IN THE UNITED STATES DISTRICT COURT
FOR THE DISTRICT OF DELAWARE

ADVANCED CLUSTER SYSTEMS, INC., Plaintiff, v. NVIDIA CORPORATION, NVIDIA SINGAPORE PTE. LTD., and NVIDIA INTERNATIONAL, INC. Defendants.	C.A. No. 19-02032-CFC **DEMAND FOR JURY TRIAL** **REDACTED:** **PUBLIC VERSION**

FIRST AMENDED COMPLAINT

Plaintiff Advanced Cluster Systems, Inc. ("ACS") hereby complains of Defendant NVIDIA Corporation, Defendant NVIDIA Singapore Pte. Ltd., and Defendant NVIDIA International, Inc. (collectively "NVIDIA"), and alleges as follows:

NATURE OF THE ACTION

1. This is a civil action for infringement of ACS's United States Patent Nos. 8,082,289; 8,140,612; 8,676,877; and 10,333,768 under the patent laws of the United States, 35 U.S.C. §§ 100, *et seq.*

THE PARTIES

2. Plaintiff ACS is a corporation incorporated under the laws of California and has its principal place of business at 220 Newport Center Drive, #11-202, Newport Beach, California 92660.

3. Defendant NVIDIA Corporation is a Delaware corporation with its principal place of business at 2788 San Tomas Expressway, Santa Clara, CA 95051. NVIDIA Corporation may

be served through its registered agent, Corporation Service Company, 251 Little Falls Drive, Wilmington, Delaware 19808.

4. On information and belief, Defendant NVIDIA Singapore Pte. Ltd. ("NVIDIA Singapore") is an entity organized under the laws of Singapore with its principal place of business located at 3/F Harbour View 1, No. 12 Science Park East Avenue, HK Science Park, Shatin, Hong Kong. On information and belief, NVIDIA Singapore may be served at its registered office, which is located at 112 Robinson Road, #05-01, Singapore 068902.

5. On information and belief, Defendant NVIDIA International, Inc. ("NVIDIA International") is an entity organized under the laws of the Cayman Islands. On information and belief, NVIDIA International may be served through its registered agent, Genesis Trust & Corporate Services Ltd., which maintains a mailing address of P.O. Box 448, Elgin Court, Elgin Avenue, George Town, Grand Cayman KY1-1106, Cayman Islands.

6. On information and belief, NVIDIA Corporation wholly owns NVIDIA International, Inc., which wholly owns NVIDIA Singapore Pte. Ltd.

JURISDICTION AND VENUE

7. ACS asserts claims for patent infringement arising under the patent laws of the United States, 35 U.S.C. §§ 100, *et seq.* This Court has subject matter jurisdiction of these claims under 28 U.S.C. §§ 1331 and 1338(a).

NVIDIA Corporation

8. This court has personal jurisdiction over NVIDIA Corporation at least because NVIDIA Corporation is organized and exists under the laws of the State of Delaware. ACS is informed and believes, and thereon alleges, that NVIDIA Corporation resides in this judicial district, has systematic and continuous contacts in this judicial district, regularly and systematically

transacts business within this district, and regularly avails itself of the benefits of this judicial district. ACS's causes of action arise, at least in part, from NVIDIA Corporation's contacts and activities in the State of Delaware.

9. Venue is proper in this judicial district under 28 U.S.C. § 1400(b) because NVIDIA Corporation is a Delaware corporation that resides in this judicial district. In addition, upon information and belief, NVIDIA Corporation has committed acts of infringement in the State of Delaware.

NVIDIA Singapore

Personal Jurisdiction Under Delaware Long-Arm Statute/Stream of Commerce

10. This Court has personal jurisdiction over NVIDIA Singapore pursuant to the Delaware long-arm statute (10 Del. C. § 3104(c)) under the "dual jurisdiction" or "stream of commerce" test, as explained below.

11. The "dual jurisdiction" or "stream of commerce" theory provides a basis for personal jurisdiction under Delaware's long-arm statute. *Graphics Properties Holdings, Inc. v. Asus Computer Int'l*, 70 F. Supp. 3d 654, 662 (D. Del. 2014). Under the stream of commerce test, personal jurisdiction is established over a foreign defendant such as NVIDIA Singapore by showing three things: (1) an intent to serve the Delaware market; (2) the introduction of products into the market as a result of this intent; and (3) that Plaintiff's cause of action arises from injuries caused by those products. *3G Licensing, S.A. v. HTC Corp.*, No. 17-cv-83-LPS-CJB, 2017 WL 6442101, at *2 (D. Del. Dec. 18, 2017).

First Prong: Intent to Serve the United States Market, Including Delaware

12. As to the first prong, a non-resident entity's "intent to serve the United States market is sufficient to establish an intent to serve the Delaware market, unless there is evidence

that the firm intended to exclude from its marketing and distribution efforts some portion of the country that includes Delaware." *Enzo Life Scis., Inc. v. Hologic Inc*., No. 16-cv-894-LPS-CJB, 2018 WL 4660355, at *3 n.5 (D. Del. Sept. 26, 2018) (quoting *Robert Bosch LLC v. Alberee Prod., Inc.*, 70 F. Supp. 3d 665, 675 (D. Del. 2014)).

13. On information and belief, during the time period relevant to this lawsuit, including from at least 2016 to 2020, NVIDIA Singapore intended to serve the United States market with products, including server products such as the DGX-1, DGX-2, DGX-A100, HGX-1, HGX-2, HGX-A100, DGX Station and GPU products for use in server products, such as the Tesla P100 GPU accelerator, Tesla V100 GPU accelerator, and NVIDIA A100 Tensor Core GPU. As explained below, Plaintiff accuses these products of infringing the Patents-in-Suit (collectively, "accused products"). *See infra* ¶¶ 94-139.

14. On information and belief, NVIDIA Singapore's intent to serve the United States market, including Delaware, is demonstrated by the fact that from 2016 through 2020 NVIDIA Singapore shipped several tons of products from Hong Kong to the United States, including the accused products. NVIDIA Singapore's U.S. parent NVIDIA Corporation received these shipments in the United States. For example, for the period 2016-2017, importation records describe these shipments as containing computer graphics cards, as summarized in the table below.

Year	Shipper	Consignee	Commodity Description	Shipments
2016	NVIDIA Singapore Pte Ltd	NVIDIA Corp	Computer Graphics Cards	3
2017	NVIDIA Singapore Pte Ltd	NVIDIA Corp	Computer Graphics Cards	2

See Exhibit I (U.S. importation records from 2016 through 2017).

15. The total reported weight of these 5 shipments is 14064 kilograms, which is about 31,000 pounds or 15.5 tons. NVIDIA Corporation has represented that [redacted]

16. On information and belief, NVIDIA Singapore's intent to serve the United States market, including Delaware, is also demonstrated by the fact that, the primary purpose of NVIDIA Singapore is to sell NVIDIA products worldwide, and U.S. consumers account for a significant portion of the worldwide sales of NVIDIA products, including the accused products, sold by NVIDIA Singapore. NVIDIA Singapore has repeatedly stated in its annual financial statements that "[t]he principal activity of the Company consists of sales of graphics processors and media and communication devices." *See, e.g.*, Exhibit J at 10 (financial statement for fiscal year ending January 25, 2015); Exhibit K at 8 (financial statement for fiscal year ending January 31, 2016). Further, as summarized in the table below, NVIDIA Singapore generates the vast majority of all revenue reported by NVIDIA Corporation on a consolidated basis:

Fiscal Year Ending In	NVIDIA Singapore Revenue	NVIDIA Corporation Revenue (Consolidated) (Ex. P at 23)
2016	$ 4,077,896,000 (Exhibit K at 4)	$ 5,010,000,000
2017	$ 5,712,400,000 (Exhibit L at 5)	$ 6,910,000,000
2018	$ 8,064,722,000 (Exhibit M at 5)	$ 9,714,000,000
2019	$ 9,727,059,000 (Exhibit N at 5)	$ 11,716,000,000
Total:	$ 27,582,077,000	$ 33,350,000,000

17. During the four fiscal years summarized above, NVIDIA Corporation reported roughly $33 billion in worldwide revenue on a consolidated basis, and NVIDIA Singapore reported roughly $27.6 billion in revenue. In other words, NVIDIA Singapore generated roughly 83 percent of NVIDIA Corporation's worldwide revenue. On information and belief, NVIDIA Singapore's role in generating revenue for NVIDIA Corporation demonstrates an intent by NVIDIA Singapore to serve the United States market because U.S. consumers provide a significant

portion of the worldwide sales for NVIDIA products, including the accused products, sold by NVIDIA Singapore.

18. NVIDIA Singapore's intent to serve the United States market, including Delaware, is further demonstrated by the fact that, as summarized in the table below, three of its four directors reside in the United States (rather than in Singapore or Hong Kong) and also hold executive level positions at NVIDIA Corporation, which is based in the United States and incorporated in Delaware.

NVIDIA Singapore Director	Country of Residence	Position at NVIDIA Corp.
Michael John Byron	United States	Vice President, Finance Operations and Systems
Karen Theresa Burns	United States	Vice President, Finance
Rebecca Peters	United States	Vice President, Corporate Affairs

Exhibit O at 5 (NVIDIA Singapore Corporate Compliance and Financial Profile).

19. On information and belief, NVIDIA Singapore has not excluded any portion of the United States that includes Delaware from its efforts to sell and ship products to the United States.

20. On information and belief, during the time period relevant to this case, NVIDIA Singapore has intended to serve the United States market, including Delaware, because (1) NVIDIA Singapore has shipped several tons of products to the United States, including the accused products, (2) NVIDIA Singapore has generated the vast majority of NVIDIA Corporation's worldwide revenue, much of which is attributable to sales of NVIDIA products to U.S. consumers, including the accused products, (3) three of NVIDIA Singapore's four directors are U.S. residents who also hold executive level positions with NVIDIA Corporation, which is incorporated in Delaware, and (4) NVIDIA Singapore has not sought to exclude any portion of the United States

that includes Delaware from its efforts to sell and ship products to the United States, including the accused products.

Second Prong: Introduction of Products Into the United States Market

21. On information and belief, NVIDIA Singapore's intent to serve the United States market resulted in the introduction of products, including the accused products, into the United States market, including in Delaware.

22. For example, the accused products are available for sale through the NVIDIA Partner Network of OEMs and service providers, many of whom service the United States. A page on the Singapore portion of NVIDIA's website notes that "NVIDIA DGX™ Systems are available through select NVIDIA Partner Network (NPN) partners." https://www.nvidia.com/en-sg/data-center/where-to-buy-dgx-systems/. The full list of partners sorted by the United States region includes 261 United States partners. https://www.nvidia.com/en-sg/about-nvidia/partners/partner-locator/. Of these NVIDIA partners, at least thirty are incorporated in Delaware. *See* https://icis.corp.delaware.gov/Ecorp/EntitySearch/ NameSearch.aspx.

Third Prong: Cause of Action Arises from Injuries Caused by Products

23. Plaintiff's cause of action for infringement of the Patents-in-Suit arises from injuries caused by the accused products introduction to and sales in the United States market, including Delaware, as a result of NVIDIA Singapore's intent to serve the United States market. As explained further below, Plaintiff alleges that the accused products infringe the Patents-in-Suit. *See infra* ¶¶ 94-139. Unlicensed usage, offers for sale, and sales of the accused products in the United States, and unlicensed importation of the accused products into the United States, injured Plaintiff.

24. In sum, this Court has personal jurisdiction over NVIDIA Singapore pursuant to the Delaware long-arm statute (10 Del. C. § 3104(c)) under the stream of commerce test.

Personal Jurisdiction Under Fed. R. Civ. P. 4(k)(2)

25. Alternatively, this Court has personal jurisdiction over NVIDIA Singapore pursuant to the federal long-arm statute, Fed. R. Civ. P. 4(k)(2).

26. Rule 4(k)(2) allows "'a court to exercise personal jurisdiction over a defendant if (1) the plaintiff's claim arises under federal law, (2) the defendant is not subject to jurisdiction in any state's courts of general jurisdiction, and (3) the exercise of jurisdiction comports with due process.'" *Bos. Sci. Corp. v. Micro-Tech Endoscopy USA Inc.*, No. 18-cv-1869-CFC-CJB, 2020 WL 229993, at *4 (D. Del. Jan. 15, 2020) (quoting *M-I Drilling Fluids UK Ltd. v. Dynamic Air Ltda*., 890 F.3d 995, 999 (Fed. Cir. 2018)), report and recommendation adopted. "'The third requirement under Rule 4(k)(2)—the due process analysis—contemplates a defendant's contacts with the entire United States, as opposed to the state in which the district court sits.'" *Id*. (same). Rule 4(k)(2) is meant to allow a district court to exercise personal jurisdiction over a foreign defendant whose contacts with the United States, but not with the forum state, satisfy due process. *Id*. (citing *M-I Drilling*, 890 F.3d at 999). Pleading personal jurisdiction alternatively under Rule 4(k)(1) and Rule 4(k)(2) is proper under Fed. R. Civ. P. 8(d). *See Touchcom, Inc. v. Bereskin & Parr*, 574 F.3d 1403, 1415 (Fed. Cir. 2009) (analyzing application of Fed. R. Civ. P. 4(k)(2) and noting that "[a]n approach that forecloses alternative arguments appears to conflict with the Federal Rules of Civil Procedure").

First Prong: Plaintiff's Claims Arise Under Federal Law

27. Plaintiff alleges that NVIDIA Singapore is liable for patent infringement pursuant to 35 U.S.C. § 271. Thus, Plaintiff's claims arise under federal law.

Second Prong: Not Subject to Jurisdiction in Any State's Courts of General Jurisdiction

28. On information and belief, NVIDIA Singapore is incorporated in Singapore, maintains its principal place of business in Hong Kong, and does not maintain any offices or subsidiaries in the United States.

29. On information and belief, NVIDIA Singapore is not subject to jurisdiction in any state's courts of general jurisdiction.

Third Prong: Due Process

30. In analyzing due process under Rule 4(k)(2), courts consider whether (1) the defendant purposefully directed its activities at residents of the United States, (2) the claim arises out of or relates to the defendant's activities with the United States, and (3) assertion of personal jurisdiction is reasonable and fair. *Bos. Sci. Corp.*, 2020 WL 229993, at *6.

31. NVIDIA Singapore purposefully directed its activities at residents of the United States. First, as explained above in paragraphs 13 and 14, which are incorporated by reference, over a period of several years NVIDIA Singapore regularly shipped tons of products, including accused products, to its parent in the United States, NVIDIA Corporation. These products were then sold throughout the United States.

32. Second, as explained above in paragraphs 15-16, which are incorporated by reference, NVIDIA Singapore generates the vast majority of revenue for NVIDIA Corporation, which is incorporated in Delaware and based in the United States. U.S. consumers provide a significant portion of worldwide sales of NVIDIA products sold by NVIDIA Singapore, including the accused products. Third, and finally, as explained above in paragraph 17, which is incorporated by reference, three out of four of NVIDIA Singapore's directors reside in the United States and

also hold executive level positions at NVIDIA Corporation, which is incorporated in Delaware and based in the United States.

35. Plaintiff's infringement claims against NVIDIA Singapore arise out of or relate to NVIDIA Singapore's activities with the United States. In particular, on information and belief, NVIDIA Singapore sold accused products and also shipped accused products to the United States. The unlicensed use, sale, and offer for sale of these accused products in the United States, and unlicensed importation of them into the United States, provide the basis for Plaintiff's infringement claims.

33. Assertion of personal jurisdiction over NVIDIA Singapore is reasonable and fair. Any burden on NVIDIA Singapore is sufficiently outweighed by the interest of the United States in adjudicating Plaintiff's claims, which are based on infringement of U.S. patents, and by the interests of Plaintiff in obtaining effective and convenient relief. *Bos. Sci. Corp*., 2020 WL 229993, at *7–*8.

34. In sum, this Court alternatively has personal jurisdiction over NVIDIA Singapore pursuant to Fed. R. Civ. P. 4(k)(2).

Venue

35. When a foreign defendant is sued in a patent infringement action, 28 U.S.C. § 1391 governs venue. *See 3G Licensing*, 2017 WL 6442101, at *2. Under § 1391, a foreign defendant may be sued in any judicial district. *See* 28 U.S.C. § 1391(c)(3).

36. On information and belief, NVIDIA Singapore is incorporated in Singapore. NVIDIA Singapore is a foreign defendant that may be sued in any judicial district. Therefore, venue is proper here in the District of Delaware.

NVIDIA International, Inc.

Personal Jurisdiction Under Delaware Long-Arm Statute/Stream of Commerce

37. This Court has personal jurisdiction over NVIDIA International, Inc. pursuant to the Delaware long-arm statute (10 Del. C. § 3104(c)) under the stream of commerce test, as explained below.

First Prong: Intent to Serve the United States Market, Including Delaware

38. On information and belief, during the time period relevant to this lawsuit, NVIDIA International intended to serve the United States market with products, including the accused products.

39. NVIDIA Corporation has represented that [redacted] NVIDIA Corporation is incorporated in Delaware and based in the United States. NVIDIA Corporation imports accused products into the United States, and markets and sells accused products throughout the United States. Additionally, on information and belief, NVIDIA International's intent to serve the United States, including Delaware, is demonstrated by the fact that NVIDIA International wholly owns NVIDIA Singapore and is wholly owned by NVIDIA Corporation, which are entities that intend to serve the United States market, including Delaware. Specifically, NVIDIA International wholly owns NVIDIA Singapore, which for the reasons explained above, intends to serve the United States market, including Delaware. *See* Exhibit N at 35 (NVIDIA Singapore 2019 financial statement) ("The Company's immediate holding corporation is NVIDIA International Inc., incorporated in the Cayman Islands."); *see supra* ¶¶ 12-20 (allegations describing activities of NVIDIA Singapore). NVIDIA International is wholly owned by NVIDIA Corporation, which is incorporated in Delaware and based in the United States. NVIDIA

Corporation imports accused products into the United States, and markets and sells accused products throughout the United States.

40. On information and belief, NVIDIA International's intent to serve the United States, including Delaware, is further demonstrated by the fact that one of NVIDIA International's directors, Karen Burns, resides in the United States (not in the Cayman Islands) and also holds an executive level position at NVIDIA Corporation as its Vice President of Finance. Ms. Burns also serves as a director of NVIDIA Singapore. As set forth above, both NVIDIA Corporation and NVIDIA Singapore intend to serve the United States, including Delaware. Ms. Burns' common involvement in all three entities demonstrates that NVIDIA International intends to serve the United States market, including Delaware, as with NVIDIA Singapore and NVIDIA Corporation.[1]

41. On information and belief, NVIDIA International has not excluded any portion of the United States that includes Delaware from its efforts to serve the United States.

42. On information and belief, during the time period relevant to this case, NVIDIA International has intended to serve the United States market, including Delaware, because (1) [redacted] in the United States; (2) NVIDIA International wholly owns NVIDIA Singapore, and is wholly owned by NVIDIA Corporation, both of which are entities that intend to serve the United States market, (3) one of NVIDIA International's directors holds the same position at NVIDIA Singapore (a frequent shipper of NVIDIA accused products to the United States) and also holds an executive level positon at NVIDIA Corporation, which is incorporated

[1] Ms. Burns' involvement as a director of NVIDIA International, Inc. was revealed in 2017 as part of the "Paradise Papers," which include leaked documents related to offshore companies in tax havens like the Cayman Islands. *See* https://offshoreleaks.icij.org/nodes/101418482

in Delaware, and (4) NVIDIA International has not sought to exclude any portion of the United States that includes Delaware from its efforts to serve the United States market.

Second Prong: Introduction of Products Into the United States Market

43. On information and belief, NVIDIA International's intent to serve the United States market resulted in the introduction of products, including the accused products, into the United States market, including in Delaware.

44. As explained [redacted] the United States. NVIDIA Corporation then imports the accused products into the United States, and markets and sells accused products nationwide.

Third Prong: Cause of Action Arises from Injuries Caused by Products

45. Plaintiff's cause of action for infringement of the Patents-in-Suit arises from injuries caused by the accused products introduced to the United States market, including Delaware, as a result of NVIDIA International's intent to serve the United States market. As explained further below, Plaintiff alleges that the accused products infringe the Patents-in-Suit. *See infra* ¶¶ 94-139. Unlicensed usage, offers for sale, and sales of the accused products in the United States, and unlicensed importation of the accused products into the United States, injured Plaintiff.

46. In sum, this Court has personal jurisdiction over NVIDIA International pursuant to the Delaware long-arm statute (10 Del. C. § 3104(c)) and the stream of commerce test.

Personal Jurisdiction Under Fed. R. Civ. P. 4(k)(2)

47. Alternatively, this Court has personal jurisdiction over NVIDIA International pursuant to the federal long-arm statute, Fed. R. Civ. P. 4(k)(2).

First Prong: Plaintiff's Claims Arise Under Federal Law

48. Plaintiff alleges that NVIDIA International is liable for patent infringement pursuant to 35 U.S.C. § 271. Thus, Plaintiff's claims arise under federal law.

Second Prong: Not Subject to Jurisdiction in Any State's Courts of General Jurisdiction

49. On information and belief, NVIDIA International is incorporated in the Cayman Islands and does not maintain any offices or subsidiaries in the United States. On information and belief, NVIDIA International is not subject to jurisdiction in any state's courts of general jurisdiction.

Third Prong: Due Process

50. NVIDIA International purposefully directed its activities at residents of the United States. First, as explained above in paragraph 38, which is incorporated by reference, NVIDIA International wholly owns NVIDIA Singapore, and is wholly owned by NVIDIA Corporation, entities that intend to serve the United States market. Second, as explained above in paragraph 39, which is incorporated by reference, NVIDIA International's director Ms. Karen Burns also serves as a director for NVIDIA Singapore, which has shipped tons of products, including the accused products, to the United States, and she resides in the United States and holds an executive level positon at NVIDIA Corporation, which is incorporated in Delaware and serves the United States.

51. Plaintiff's infringement claims against NVIDIA International arise out of or relate to NVIDIA International's activities with the United States. In particular, the unlicensed use, sale, and offer for sale of these accused products in the United States, and unlicensed importation of them into the United States, provide the basis for Plaintiff's infringement claims.

52. Assertion of personal jurisdiction over NVIDIA International is reasonable and fair. Any burden on NVIDIA International is sufficiently outweighed by the interest of the United

States in adjudicating Plaintiff's claims, which are based on infringement of U.S. patents, and by the interests of Plaintiff in obtaining effective and convenient relief.

53. In sum, this Court alternatively has personal jurisdiction over NVIDIA International pursuant to Fed. R. Civ. P. 4(k)(2).

Venue

54. On information and belief, NVIDIA International, Inc. is incorporated in the Cayman Islands. NVIDIA International is a foreign defendant that may be sued in any judicial district. *See* 28 U.S.C. § 1391(c)(3). Therefore, venue is proper here in the District of Delaware.

THE PATENTS-IN-SUIT

55. ACS realleges and reincorporates by references the allegations set forth in paragraphs 1-54 of this complaint.

56. United States Patent No. 8,082,289 titled "Cluster Computing Support for Application Programs" ("the '289 patent") is a valid, enforceable patent that the United States Patent and Trademark Office duly and lawfully issued on December 20, 2011 in full compliance with Title 35 of the United States Code. A true and correct copy of the '289 patent is attached to this complaint as Exhibit A.

57. ACS is the owner by assignment of the '289 patent with ownership of all substantial rights in the '289 patent, including the right to exclude others and to enforce, sue, and recover damages for past and future infringements.

58. The claims of the '289 patent when viewed as a whole, including as an ordered combination, are not merely the recitation of well-understood, routine, or conventional technologies or components at the time of invention, over a decade ago.

59. United States Patent No. 8,140,612 titled "Cluster Computing Support for Application Programs" ("the '612 patent") is a valid, enforceable patent that the United States Patent and Trademark Office duly and lawfully issued on March 20, 2012 in full compliance with Title 35 of the United States Code. A true and correct copy of the '612 patent is attached to this complaint as Exhibit B.

60. ACS is the owner by assignment of the '612 patent with ownership of all substantial rights in the '612 patent, including the right to exclude others and to enforce, sue and recover damages for past and future infringements.

61. The claims of the '612 patent when viewed as a whole, including as an ordered combination, are not merely the recitation of well-understood, routine, or conventional technologies or components at the time of invention, over a decade ago.

62. United States Patent No. 8,676,877 titled "Cluster Computing Using Special Purpose Microprocessors" ("the '877 patent") is a valid, enforceable patent that the United States Patent and Trademark Office duly and lawfully issued on March 18, 2014 in full compliance with Title 35 of the United States Code. A true and correct copy of the '877 patent is attached to this complaint as Exhibit C.

63. ACS is the owner by assignment of the '877 patent with ownership of all substantial rights in the '877 patent, including the right to exclude others and to enforce, sue and recover damages for past and future infringements.

64. The claims of the '877 patent when viewed as a whole, including as an ordered combination, are not merely the recitation of well-understood, routine, or conventional technologies or components at the time of invention, over a decade ago.

65. United States Patent No. 10,333,768 titled "Cluster Computing" ("the '768 patent") is a valid, enforceable patent that the United States Patent and Trademark Office duly and lawfully issued on June 25, 2019 in full compliance with Title 35 of the United States Code. A true and correct copy of the '768 patent is attached to this complaint as Exhibit D.

66. ACS is the owner by assignment of the '768 patent with ownership of all substantial rights in the '768 patent, including the right to exclude others and to enforce, sue and recover damages for past and future infringements.

67. The claims of the '768 patent when viewed as a whole, including as an ordered combination, are not merely the recitation of well-understood, routine, or conventional technologies or components at the time of invention, over a decade ago.

68. Computer clusters include a group of two or more computers, microprocessors, and/or processor cores ("nodes") that intercommunicate so that the nodes can accomplish a task as though they were a single computer. Prior to the inventions disclosed in the '289, '612, '877, and '768 patents ("Patents-in-Suit"), most software application programs were unable to benefit from the advantages offered by cluster computing, even if running on a group of computers that could act as a cluster. This is because the software application programs were not designed to benefit from the advantages offered by computer clusters. For example, many software application programs were coded to perform tasks serially, were designed to recognize and send instructions to only a single node in a cluster, or included a kernel that was designed to communication with a single node. Wolfram Research Mathematica® is an example of a software application program at the time of the invention with a kernel that was designed to communicate with a single node.

69. Prior to the invention, one method used to improve the performance of software application programs with a kernel designed to communicate with a single node was to utilize a

form of distributed computing called grid computing. Grid computing included a plurality of computer nodes that did not communicate with one another as peers. One node served as a master node that managed a plurality of slave nodes. Each slave node functioned independently, and intermediate results of one slave node did not affect other jobs in progress on other nodes. Slave nodes communicated only with the master node, and not directly with other slave nodes. An example of grid computing applied to a software application program with a kernel designed to communicate with a single node at that time was Wolfram Research gridMathematica. Distributed computing architectures like grid computing can be optimized for workloads that consist of many independent jobs of packets of work that do not need to share data between the jobs during the computational process.

70. The Patents-in-Suit are directed to an advanced, parallel mesh computing framework that makes supercomputing performance available to mainstream software. The disclosed inventions enable software application programs to benefit from the advantages computer clusters offered by, amongst other things, providing systems and methods for allowing cluster nodes to directly communicate with one another, and not just to a master node, via a communications network to perform calculations and processing on a larger scale and/or with improved speed.

FACTUAL BACKGROUND

71. ACS realleges and reincorporates by references the allegations set forth in paragraphs 1-70 of this Complaint.

72. Plaintiff ACS was founded in 2004 by Zvi Tannenbaum. Mr. Tannenbaum has a degree in electronics from the Israeli Air Force Technical School in Haifa, Israel, and devoted over 27 years to the electronic test industry prior to forming ACS. Mr. Tannenbaum's passion for

supercomputers led him to create ACS in 2004, with the mission of building innovative parallel programming tools for high-performance computing applications running on multicore, clusters, and supercomputers.

73. To accomplish his mission, Mr. Tannenbaum partnered with Dr. Dean E. Dauger, Founder and President of Dauger Research, Inc. Dr. Dauger is an accomplished scientist with extensive experience in high-performance computing. He holds a B.S. in Mathematical Physics from Harvey Mudd College, an M.S. in Physics from UCLA and a Ph.D. in Physics from UCLA. While at UCLA, Dr. Dauger co-invented the first easy-to-use, high performance cluster using a network of Macintosh computers, which garnered worldwide recognition.

74. Dr. Dauger has won multiple American Institute of Physics software contests, and he coauthored the original, award-winning Kai's Power Tools image-processing package for Adobe® Photoshop®. Dr. Dauger has also authored articles published in a variety of journals including the IEEE Journal of Computing in Science and Engineering, Journal of Computational Physics, Journal of Computer Physics Communications, and IEEE International Conference on Cluster Computing.

75. Through their partnership, Dr. Dauger and Mr. Tannenbaum developed groundbreaking systems and methods to improve parallel programming and supercomputing functionality, some of which were commercialized.

76. For example, Dr. Dauger and Mr. Tannenbaum developed a Supercomputing Engine Technology™ (SET™) product. SET is based on Message Passing Interface ("MPI") technology. Thus, SET's MPI-based library significantly simplifies parallel programming and allows a wide range of software writers to scale their code efficiently on a multitude of cores. ACS began selling SET in 2010.

77. During the development phase of SET, the SET parallelization approach was applied to Wolfram Research Mathematica, providing it with supercomputing-level parallelization. ACS commercialized the resulting product as SEM™ (Supercomputing Engine for Mathematica). SEM was a software add-on for use with Mathematica that created a standard way for each Mathematica kernel in a cluster to communicate directly with every other kernel. In contrast to typical Mathematica grid implementations that were solely master-slave or server-client implementations, SEM enabled all kernels to communicate collectively. ACS began selling SEM in 2008. Enabling Mathematica kernels to be harnessed together in this manner provided a new approach to handling supercomputing challenges at that time.

78. Dr. Dauger and Mr. Tannenbaum's innovative systems and methods at issue in this action are described and claimed in the Patents-in-Suit, including for example, U.S. Patent No. 8,082,289, which was filed on May 4, 2007.

79. In June 2007, NVIDIA released version 1.0 of its Compute Unified Device Architecture ("CUDA") language, which it purported to be "a novel hardware and programming model that . . . exposes the GPU as a truly generic data-parallel." *See* http://developer.download.nvidia.com/compute/cuda/1.0/NVIDIA_CUDA_Programming_Guide_1.0.pdf. ACS is informed and believes, and thereon alleges that Version 1.0 of CUDA did not provide for direct communication between GPUs.

80. Recognizing that the SET technology was a perfect companion to, and could vastly improve, graphics processor unit ("GPU") supercomputing, Dr. Dauger and Mr. Tannenbaum began to look for a GPU manufacturing partner or licensee. In November 2012, ACS met Stephen Jones, a senior systems software engineer at NVIDIA, at the International Conference for High-Performance Computing, Networking, Storage, and Analysis ("SC 12") in Salt Lake City, Utah.

They spoke to Mr. Jones about the patented technology and the SET product, as well as its applicability to general-purpose GPU supercomputing. Mr. Jones expressed interest in the ACS technology, and agreed to discuss ways for NVIDIA to collaborate with ACS in the weeks following SC 12.

81. Following that meeting, Mr. Tannenbaum emailed Mr. Jones a data sheet with more details about ACS's patented technology, and inquired how NVIDIA would like to proceed to discuss the applicability of ACS's patented technology to NVIDIA's GPU products. Among other information, the data sheet provided details about ACS's patented technology, including how it could be used to complement general-purpose GPUs and provide a communications infrastructure for direct communications between each GPU, as shown below.

to expedite functions. Each CPU-based process of SET could be enhanced by a GPGPU. GPGPU functions would nest inside SET-supported computations. This isolation of GPGPUs from each other therefore fits perfectly into the distributed-memory paradigm of SET, while SET would provide the communications infrastructure between GPGPU-enhanced modules.

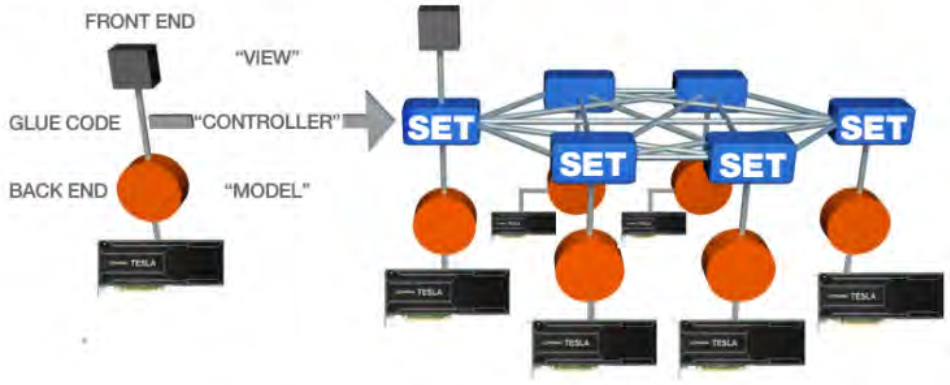


Figure 3. SET is a Perfect Companion to GPGPUs

82. The data sheet also provided actual notice of ACS's patents, citing issued U.S. Patent Nos. 8,082,289 and 8,140,612 and other "Patents Pending" as shown below.

About Advanced Cluster Systems, Inc.

Advanced Cluster Systems builds parallelization tools for high performance computing applications. The company's Supercomputing Engine Technology™ (SET™) software is positioned to advance the computer and software industries years ahead in just a few short months.

For more information, contact Zvi Tannenbaum at 949-494-0440, by email at info@adv-clustersys.com or on the web at http://www.AdvancedClusterSystems.com.

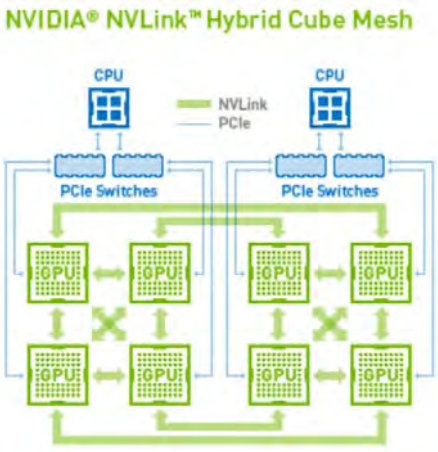
* U.S. Patents 8082289 & 8140612, Japan patent 4995902 and Patents Pending

83. ACS has continually marked its SET product and related websites with the '289 patent, the '612 patent, and the '877 patent.

84. NVIDIA did not respond further to ACS. Instead, ACS is informed and believes, and thereon alleges that NVIDIA developed its GPU interconnect architecture, called NVLink™, that combines ACS's patented technology with NVIDIA GPUs.

85. In September 2016, NVIDIA released version 8.0 of CUDA, which for the first time added support for NVIDIA's NVLink high-speed interconnect. *See* https://docs.nvidia.com/cuda/archive/8.0/cuda-toolkit-release-notes/index.html.

86. ACS is informed and believes, and thereon alleges that NVLink is an interconnect architecture that facilitates data and control transmission between multiple GPUs and Central Processing Units ("CPUs"), combined together in a hybrid cube mesh as shown below.

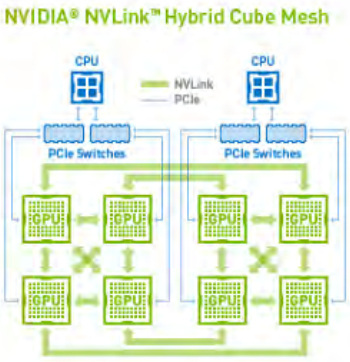


In general terms, the NVLink architecture implements the same GPU-to-GPU communication architecture claimed in the Patents-in-Suit. The details are set forth in the exemplary claim charts provided at Exhibits E-H hereto.

87. ACS is informed and believes, and thereon alleges that at least as early as May 2015, NVIDIA began developing NVLink. NVIDIA touted this new direct GPU-to-GPU interconnect architecture, as "a radical new topology."

Here's how it happened:

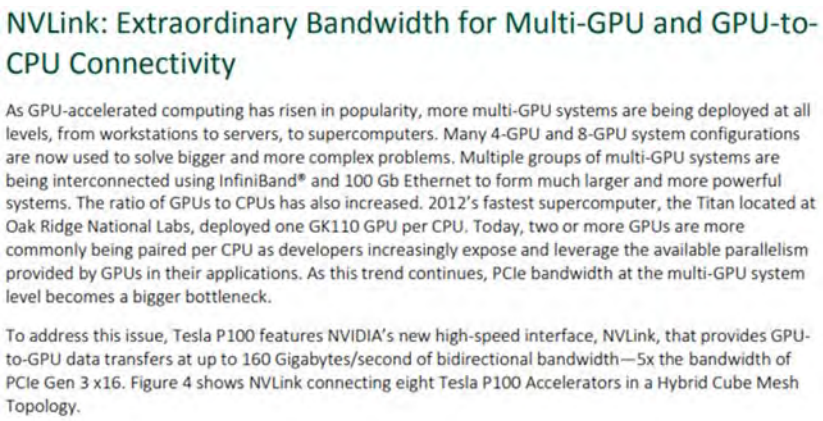
- **May 2015** — A team of engineers sketches out a radical new topology that will yoke together the eight GPUs inside DGX-1, each with 15 billion transistors. The solution: a cube mesh. The design will let users throw eight GPUs at deep learning tasks, or split the system into two separate subsystems to tackle more traditional high performance computing work. But they won't know if it will work for another seven months. The first samples of Pascal, the first GPU to use NVLink — our high-speed interconnect technology that will power the mesh — won't arrive until the last quarter of 2015 (see "What Is NVLink?")



(*See* press release titled "Blood, Software and 120 Billion Transistors: How NVIDIA Built DGX-1" (available at https://blogs.nvidia.com/blog/2016/07/11/how-nvidia-built-dgx-1/).)

88. NVIDIA also has touted the NVLink direct GPU-to-GPU interconnect as the first of its kind, and one of the key "five miracles" that made its Tesla P100 with NVIDIA NVLink product possible. (*See* GTC 2016: Tesla P100 for NVLink, World's Most Advanced Hyperscale Datacenter GPU (part 6), available at https://youtu.be/IqDKz90dNl4 at 6:35-7:00.)

89. NVIDIA also has highlighted that the NVLink direct GPU-to-GPU interconnect helped alleviate the bottleneck created by the previous interconnect topology between GPUs and CPU (i.e., the PCIe interface) as shown in the excerpt below.



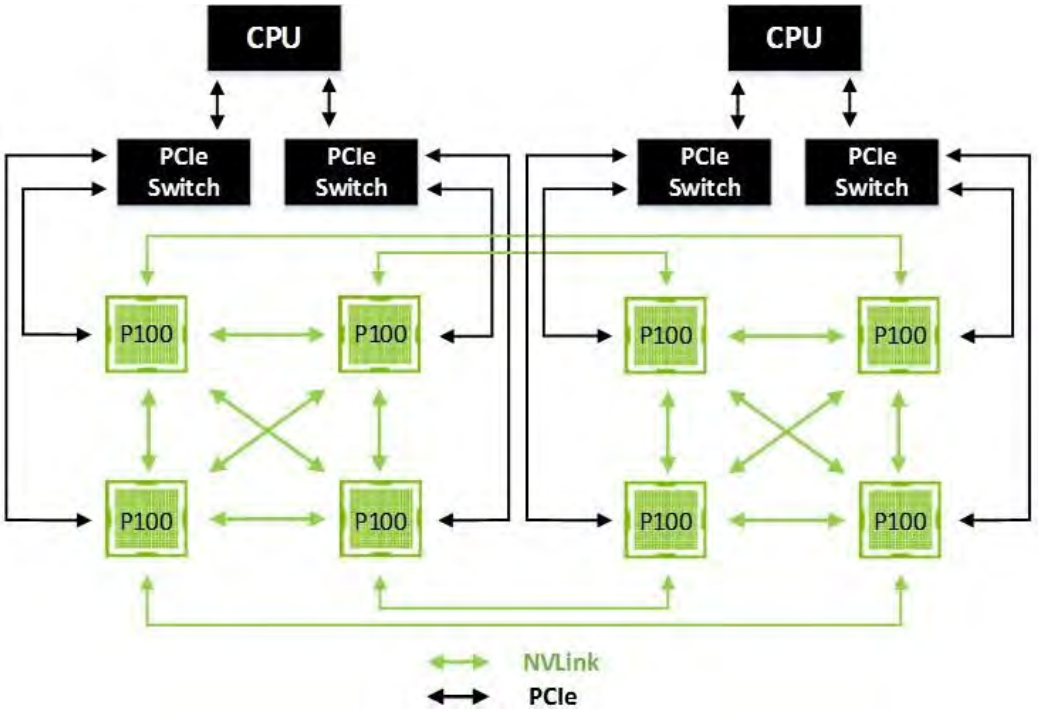
NVLink: Extraordinary Bandwidth for Multi-GPU and GPU-to-CPU Connectivity

As GPU-accelerated computing has risen in popularity, more multi-GPU systems are being deployed at all levels, from workstations to servers, to supercomputers. Many 4-GPU and 8-GPU system configurations are now used to solve bigger and more complex problems. Multiple groups of multi-GPU systems are being interconnected using InfiniBand® and 100 Gb Ethernet to form much larger and more powerful systems. The ratio of GPUs to CPUs has also increased. 2012's fastest supercomputer, the Titan located at Oak Ridge National Labs, deployed one GK110 GPU per CPU. Today, two or more GPUs are more commonly being paired per CPU as developers increasingly expose and leverage the available parallelism provided by GPUs in their applications. As this trend continues, PCIe bandwidth at the multi-GPU system level becomes a bigger bottleneck.

To address this issue, Tesla P100 features NVIDIA's new high-speed interface, NVLink, that provides GPU-to-GPU data transfers at up to 160 Gigabytes/second of bidirectional bandwidth—5x the bandwidth of PCIe Gen 3 x16. Figure 4 shows NVLink connecting eight Tesla P100 Accelerators in a Hybrid Cube Mesh Topology.

(*See* whitepaper titled "Tesla P100 for NVLink" (available at https://images.nvidia.com/content/pdf/tesla/whitepaper/pascal-architecture-whitepaper.pdf).)

90. ACS is informed and believes, and thereon alleges that NVIDIA Corporation, together with its subsidiaries NVIDIA Singapore and NVIDIA International began selling commercial products incorporating the infringing NVLink technology at least as early as 2016. For example, in April 2016 NVIDIA unveiled and subsequently began selling the Tesla P100 GPU accelerator. The Tesla P100 GPU accelerator features NVIDIA's NVLink interface to provide GPU-to-GPU data transfers. Figure 4 below depicts how NVIDIA advertises using NVLink to connect eight Tesla P100 Accelerators in a Hybrid Cube Mesh Topology.



(*See* whitepaper titled "Tesla P100 for NVLink" (available at https://images.nvidia.com/content/pdf/tesla/whitepaper/pascal-architecture-whitepaper.pdf).) NVIDIA claimed the Tesla P100 was the most advanced GPU to date due to "five miracles," including NVIDIA's NVLink high-speed GPU-to-GPU interconnect architecture.

(*see* GTC 2016: Tesla P100 for NVLink, World's Most Advanced Hyperscale Datacenter GPU (part 6), available at https://youtu.be/IqDKz90dNl4 at 6:35-7:00).

91. ACS is informed and believes, and thereon alleges that NVIDIA Corporation, together with its subsidiaries NVIDIA Singapore and NVIDIA International subsequently began

selling additional products utilizing, or especially adapted for use in, the infringing NVLink architecture, including the DGX-A100, HGX-A100, NVIDIA A100 Tensor Core GPU, DGX-2, HGX-1, HGX-2, DGX Station, and Tesla V100 GPU accelerator products.

92. ACS is informed and believes, and thereon alleges that NVIDIA Corporation, together with its subsidiaries NVIDIA Singapore and NVIDIA International have made, used, offered to sell, and/or sold within the United States, and/or has imported into the United States server and workstation products that utilize NVIDIA's NVLink architecture, including NVIDIA's DGX-1, DGX-2, DGX-A100, HGX-1, HGX-2, HGX-A100, and DGX Station (the "Accused Server Products").

93. ACS is informed and believes, and thereon alleges that NVIDIA Corporation, together with its subsidiaries NVIDIA Singapore and NVIDIA International have made, used, offered to sell, and/or sold within the United States, and/or has imported into the United States GPU products especially adapted for use in NVIDIA's NVLink architecture, including NVIDIA's Tesla P100 GPU accelerator, Tesla V100 GPU accelerator, and NVIDIA A100 Tensor Core GPU (the "Accused GPU Products").

FIRST CLAIM FOR RELIEF
INFRINGEMENT OF THE '289 PATENT

94. ACS realleges and reincorporates by references the allegations set forth in paragraphs 1-93 of this Complaint.

95. ACS is informed and believes, and thereon alleges that NVIDIA has directly infringed literally and/or through the doctrine of equivalents at least one claim of the '289 patent by making, using, selling, offering for sale, and/or importing one or more of the Accused Server Products, in violation of 35 U.S.C. § 271(a). NVIDIA continues to engage in acts of direct infringement of the '289 patent.

96. Exhibit E shows an exemplary claim chart demonstrating how the Accused Server Products use NVIDIA's NVLink technology to practice claim 29 of the '289 patent.

97. ACS is informed and believes, and thereon alleges that NVIDIA has actively, knowingly, and intentionally induced infringement of at least claim 29 of the '289 patent in violation of 35 U.S.C. § 271(b). ACS is informed and believes, and thereon alleges that NVIDIA, knowing that the Accused Server Products infringe the '289 patent as shown in Exhibit E, has actively encouraged third parties such as customers, users, system builders, NVIDIA Partner Network Deep Learning Partners, and retailers/distributors to make, use, sell, or offer for sale within the United States and/or import into the United States the Accused Server Products, knowing and intending that the actions by the third parties constitute patent infringement. ACS is informed and believes, and thereon alleges that the actions of NVIDIA actually resulted in direct infringement by the third parties.

98. For example, ACS is informed and believes, and thereon alleges that NVIDIA makes, uses, sells, offers for sale, and/or imports one or more of the Accused GPU Products for or on behalf of third parties such as customers, users, system builders, NVIDIA Partner Network Deep Learning Partners, and retailers/distributors, knowing and intending that the Accused GPU Products will be used by the third parties in a manner that practices at least claim 29 of the '289 patent as shown in Exhibit E. For example, NVIDIA published and provided marketing materials, technical specifications, whitepapers, datasheets, user manuals, and development and testing information, and other resources on its website (http://www.nvidia.com/) that instructed and encouraged third parties to integrate the Accused GPU Products into products using NVIDIA's NVLink technology that were then made, used, sold, offered for sale and/or imported into the United States. E.g., https://www.nvidia.com/en-us/data-center/tesla-p100/;

https://www.nvidia.com/en-us/data-center/tesla-v100/; https://www.nvidia.com/en-us/data-center/dgx-1/; https://www.nvidia.com/en-us/data-center/dgx-2/; https://www.nvidia.com/en-us/data-center/dgx-station/; https://www.nvidia.com/en-sg/data-center/hgx-1/; https://www.nvidia.com/en-us/data-center/hgx/; https://nvidianews.nvidia.com/news/tencent-cloud-adopts-nvidia-tesla-for-ai-cloud-computing; https://blogs.nvidia.com/blog/2017/10/03/oracle-cloud-tesla-gpus/; https://nvidianews.nvidia.com/news/china-s-top-cloud-providers-adopt-nvidia-volta-gpus-to-supercharge-next-gen-ai-services; https://blogs.nvidia.com/blog/2017/04/05/ibm-cloud-tesla-p100/. NVIDIA has also established the "NVIDIA Partner Network Deep Learning Partners" to assist customers with training, professional services, and service and support. *E.g.*, https://www.nvidia.com/en-us/data-center/where-to-buy-dgx-systems/. These activities were designed to bring infringing products that incorporate NVIDIA's Accused GPU Products to market in the United States. As a result of NVIDIA's activities, the Accused GPU Products have been used in a manner that directly infringes the '289 patent. NVIDIA continues to engage in acts of induced infringement of the '289 patent.

99. ACS is informed and believes, and thereon alleges that NVIDIA has contributed to infringement of at least claim 29 of the '289 patent under 35 U.S.C. § 271(c) by making, using, selling, offering for sale, and/or importing in the United States the Accused GPU Products and Accused Server Products, which are known by NVIDIA to be especially made or especially adapted for use in an infringement of the '289 patent, and are not staple articles or commodities of commerce suitable for substantial non-infringing use. The Accused GPU Products and Accused Server Products are specifically designed to use NVIDIA's NVLink technology in a manner that directly infringes at least claim 29 of the '289 patent. As a result, third parties such as customers,

users, system builders, NVIDIA Partner Network Deep Learning Partners, and retailers/distributors have used NVIDIA's Accused GPU Products and Accused Server Products in a manner that directly infringes the '289 patent. NVIDIA continues to engage in acts of contributory infringement of the '289 patent.

100. ACS is informed and believes, and thereon alleges that NVIDIA knew of the '289 patent at least as early as 2012, when ACS sent an email to Stephen Jones with a data sheet informing him of the patent and outlining how the technology could be used with NVIDIA's products, as well as through ACS's marking of its SET™ product. Thus since at least 2012, NVIDIA has known that its activities infringe the '289 patent. In addition, since at least 2012, NVIDIA has known or should have known that its activities induce and/or contribute to infringement of the '289 patent by third parties.

101. ACS is informed and believes, and thereon alleges that NVIDIA's infringement of the '289 patent has been, and continues to be, willful, deliberate, and intentional by continuing its acts of infringement after knowing, or after it should have known, of the '289 patent and that its conduct amounted to infringement of the '289 patent.

102. As a consequence of NVIDIA's infringement of the '289 patent, ACS has been damaged and is entitled to monetary relief in an amount to be determined at trial.

103. Unless enjoined, NVIDIA and/or others acting on behalf of NVIDIA will continue their infringing acts, thereby causing additional irreparable injury to ACS for which there is no adequate remedy at law.

104. NVIDIA's infringement of the '289 patent has been willful and deliberate, entitling ACS to enhanced damages under 35 U.S.C. § 284 and attorneys' fees and costs under 35 U.S.C. § 285.

SECOND CLAIM FOR RELIEF
INFRINGEMENT OF THE '612 PATENT

105. ACS realleges and reincorporates by references the allegations set forth in paragraphs 1-104 of this Complaint.

106. ACS is informed and believes, and thereon alleges that NVIDIA has directly infringed literally and/or through the doctrine of equivalents at least one claim of the '612 patent by making, using, selling, offering for sale, and/or importing one or more of the Accused Server Products, in violation of 35 U.S.C. § 271(a). NVIDIA continues to engage in acts of direct infringement of the '612 patent.

107. Exhibit F shows an exemplary claim chart demonstrating how the Accused Server Products use NVIDIA's NVLink technology to practice claim 1 of the '612 patent.

108. ACS is informed and believes, and thereon alleges that NVIDIA has actively, knowingly, and intentionally induced infringement of at least claim 1 of the '612 patent in violation of 35 U.S.C. § 271(b). ACS is informed and believes, and thereon alleges that NVIDIA, knowing that the Accused Server Products infringe the '612 patent as shown in Exhibit F, has actively encouraged third parties such as customers, users, system builders, NVIDIA Partner Network Deep Learning Partners, and retailers/distributors to make, use, sell, or offer for sale within the United States and/or import into the United States the Accused Server Products, knowing and intending that the actions by the third parties constitute patent infringement. ACS is informed and believes, and thereon alleges that the actions of NVIDIA actually resulted in direct infringement by the third parties.

109. For example, ACS is informed and believes, and thereon alleges that NVIDIA makes, uses, sells, offers for sale, and/or imports one or more of the Accused GPU Products for or on behalf of third parties such as customers, users, system builders, NVIDIA Partner Network

Deep Learning Partners, and retailers/distributors, knowing and intending that the Accused GPU Products will be used by the third parties in a manner that practices at least claim 1 of the '612 patent as shown in Exhibit F. For example, NVIDIA published and provided marketing materials, technical specifications, whitepapers, datasheets, user manuals, and development and testing information, and other resources on its website (http://www.nvidia.com/) that instructed and encouraged third parties to integrate the Accused GPU Products into products using NVIDIA's NVLink technology that were then made, used, sold, offered for sale and/or imported into the United States. E.g., https://www.nvidia.com/en-us/data-center/tesla-p100/; https://www.nvidia.com/en-us/data-center/tesla-v100/; https://www.nvidia.com/en-us/data-center/dgx-1/; https://www.nvidia.com/en-us/data-center/dgx-2/; https://www.nvidia.com/en-us/data-center/dgx-station/; https://www.nvidia.com/en-sg/data-center/hgx-1/; https://www.nvidia.com/en-us/data-center/hgx/; https://nvidianews.nvidia.com/news/tencent-cloud-adopts-nvidia-tesla-for-ai-cloud-computing; https://blogs.nvidia.com/blog/2017/10/03/oracle-cloud-tesla-gpus/; https://nvidianews.nvidia.com/news/china-s-top-cloud-providers-adopt-nvidia-volta-gpus-to-supercharge-next-gen-ai-services; https://blogs.nvidia.com/blog/2017/04/05/ibm-cloud-tesla-p100/. NVIDIA has also established the "NVIDIA Partner Network Deep Learning Partners" to assist customers with training, professional services, and service and support. *E.g.*, https://www.nvidia.com/en-us/data-center/where-to-buy-dgx-systems/. These activities were designed to bring infringing products that incorporate NVIDIA's Accused GPU Products to market in the United States. As a result of NVIDIA's activities, the Accused GPU Products have been used in a manner that directly infringes the '612 patent. NVIDIA continues to engage in acts of induced infringement of the '612 patent.

110. ACS is informed and believes, and thereon alleges that NVIDIA has contributed to infringement of at least claim 1 of the '612 patent under 35 U.S.C. § 271(c) by making, using, selling, offering for sale, and/or importing in the United States the Accused GPU Products and Accused Server Products, which are known by NVIDIA to be especially made or especially adapted for use in an infringement of the '612 patent, and are not staple articles or commodities of commerce suitable for substantial non-infringing use. The Accused GPU Products and Accused Server Products are specifically designed to use NVIDIA's NVLink technology in a manner that directly infringes at least claim 1 of the '612 patent. As a result, third parties such as customers, users, system builders, NVIDIA Partner Network Deep Learning Partners, and retailers/distributors have used NVIDIA's Accused GPU Products and Accused Server Products in a manner that directly infringes the '612 patent. NVIDIA continues to engage in acts of contributory infringement of the '612 patent.

111. ACS is informed and believes, and thereon alleges that NVIDIA knew of the '612 patent at least as early as 2012, when ACS sent an email to Stephen Jones with a data sheet informing him of the patent and outlining how the technology could be used with NVIDIA's products, as well as through ACS's marking of its SET™ product. Thus since at least 2012, NVIDIA has known that its activities infringe the '612 patent. In addition, since at least 2012, NVIDIA has known or should have known that its activities induce and/or contribute to infringement of the '612 patent by third parties.

112. ACS is informed and believes, and thereon alleges that NVIDIA's infringement of the '612 patent has been, and continues to be, willful, deliberate, and intentional by continuing its acts of infringement after knowing, or after it should have known, of the '612 patent and that its conduct amounted to infringement of the '612 patent.

113. As a consequence of NVIDIA's infringement of the '612 patent, ACS has been damaged and is entitled to monetary relief in an amount to be determined at trial.

114. Unless enjoined, NVIDIA and/or others acting on behalf of NVIDIA will continue their infringing acts, thereby causing additional irreparable injury to ACS for which there is no adequate remedy at law.

115. NVIDIA's infringement of the '612 patent has been willful and deliberate, entitling ACS to enhanced damages under 35 U.S.C. § 284 and attorneys' fees and costs under 35 U.S.C. § 285.

THIRD CLAIM FOR RELIEF
INFRINGEMENT OF THE '877 PATENT

116. ACS realleges and reincorporates by references the allegations set forth in paragraphs 1-115 of this Complaint.

117. ACS is informed and believes, and thereon alleges that NVIDIA has directly infringed literally and/or through the doctrine of equivalents at least one claim of the '877 patent by making, using, selling, offering for sale, and/or importing one or more of the Accused Server Products, in violation of 35 U.S.C. § 271(a). NVIDIA continues to engage in acts of direct infringement of the '877 patent.

118. Exhibit G shows an exemplary claim chart demonstrating how the Accused Server Products use NVIDIA's NVLink technology to practice claim 8 of the '877 patent.

119. ACS is informed and believes, and thereon alleges that NVIDIA has actively, knowingly, and intentionally induced infringement of at least claim 8 of the '877 patent in violation of 35 U.S.C. § 271(b). ACS is informed and believes, and thereon alleges that NVIDIA, knowing that the Accused Server Products infringe the '877 patent as shown in Exhibit G, have actively encouraged third parties such as customers, users, system builders, NVIDIA Partner Network Deep

Learning Partners, and retailers/distributors to make, use, sell, or offer for sale within the United States and/or import into the United States the Accused Server Products, knowing and intending that the actions by the third parties constitute patent infringement. ACS is informed and believes, and thereon alleges that the actions of NVIDIA actually resulted in direct infringement by the third parties.

120. For example, ACS is informed and believes, and thereon alleges that NVIDIA makes, uses, sells, offers for sale, and/or imports one or more of the Accused GPU Products for or on behalf of third parties such as customers, users, system builders, NVIDIA Partner Network Deep Learning Partners, and retailers/distributors, knowing and intending that the Accused GPU Products will be used by the third parties in a manner that practices at least claim 8 of the '877 patent as shown in Exhibit G. For example, NVIDIA published and provided marketing materials, technical specifications, whitepapers, datasheets, user manuals, and development and testing information, and other resources on its website (http://www.nvidia.com/) that instructed and encouraged third parties to integrate the Accused GPU Products into products using NVIDIA's NVLink technology that were then made, used, sold, offered for sale and/or imported into the United States. E.g., https://www.nvidia.com/en-us/data-center/tesla-p100/; https://www.nvidia.com/en-us/data-center/tesla-v100/; https://www.nvidia.com/en-us/data-center/dgx-1/; https://www.nvidia.com/en-us/data-center/dgx-2/; https://www.nvidia.com/en-us/data-center/dgx-station/; https://www.nvidia.com/en-sg/data-center/hgx-1/; https://www.nvidia.com/en-us/data-center/hgx/; https://nvidianews.nvidia.com/news/tencent-cloud-adopts-nvidia-tesla-for-ai-cloud-computing; https://blogs.nvidia.com/blog/2017/10/03/oracle-cloud-tesla-gpus/; https://nvidianews.nvidia.com/news/china-s-top-cloud-providers-adopt-nvidia-volta-gpus-to-

supercharge-next-gen-ai-services; https://blogs.nvidia.com/blog/2017/04/05/ibm-cloud-tesla-p100/. NVIDIA has also established the "NVIDIA Partner Network Deep Learning Partners" to assist customers with training, professional services, and service and support. *E.g.*, https://www.nvidia.com/en-us/data-center/where-to-buy-dgx-systems/. These activities were designed to bring infringing products that incorporate NVIDIA's Accused GPU Products to market in the United States. As a result of NVIDIA's activities, the Accused GPU Products have been used in a manner that directly infringes the '877 patent. NVIDIA continues to engage in acts of induced infringement of the '877 patent.

121. ACS is informed and believes, and thereon alleges that NVIDIA has contributed to infringement of at least claim 8 of the '877 patent under 35 U.S.C. § 271(c) by making, using, selling, offering for sale, and/or importing in the United States the Accused GPU Products and Accused Server Products, which are known by NVIDIA to be especially made or especially adapted for use in an infringement of the '877 patent, and are not staple articles or commodities of commerce suitable for substantial non-infringing use. The Accused GPU Products and Accused Server Products are specifically designed to use NVIDIA's NVLink technology in a manner that directly infringes at least claim 8 of the '877 patent. As a result, third parties such as customers, users, system builders, NVIDIA Partner Network Deep Learning Partners, and retailers/distributors have used NVIDIA's Accused GPU Products and Accused Server Products in a manner that directly infringes the '877 patent. NVIDIA continues to engage in acts of contributory infringement of the '877 patent.

122. The '877 patent is a continuation of U.S. Patent Application No. 12/040,519, now the '612 patent which is a continuation-in-part of U.S. Application No. 11/744,461, now the '289 patent.

123. ACS is informed and believes, and thereon alleges that NVIDIA knew of the '612 and '289 patents at least as early as 2012, when ACS sent an email to Stephen Jones with a data sheet informing him of the patent and outlining how the technology could be used with NVIDIA's products, as well as through ACS's marking of its SET™ product. ACS is informed and believes, and thereon alleges that NVIDIA knew or should have known of the '877 patent prior to the filing of this Complaint because it is a continuation of the '612 patent, which is a continuation-in-part of the '289 patent, and NVIDIA was explicitly informed of the '289 and '612 patents and of "Patents Pending." Thus since at least the issuance of the '877 patent in 2014, NVIDIA has known that its activities infringe the '877 patent. In addition, since at least the issuance of the '877 patent in 2014, NVIDIA knew or should have known that its activities induce and/or contribute to infringement of the '877 patent by third parties.

124. ACS is informed and believes, and thereon alleges that NVIDIA's infringement of the '877 patent has been, and continues to be, willful, deliberate, and intentional by continuing its acts of infringement after knowing, or after it should have known, of the '877 patent and that its conduct amounted to infringement of the '877 patent.

125. As a consequence of NVIDIA's infringement of the '877 patent, ACS has been damaged and is entitled to monetary relief in an amount to be determined at trial.

126. Unless enjoined, NVIDIA and/or others acting on behalf of NVIDIA will continue their infringing acts, thereby causing additional irreparable injury to ACS for which there is no adequate remedy at law.

127. NVIDIA's infringement of the '877 patent has been willful and deliberate, entitling ACS to enhanced damages under 35 U.S.C. § 284 and attorneys' fees and costs under 35 U.S.C. § 285.

FOURTH CLAIM FOR RELIEF
INFRINGEMENT OF THE '768 PATENT

128. ACS realleges and reincorporates by references the allegations set forth in paragraphs 1-127 of this Complaint.

129. ACS is informed and believes, and thereon alleges that NVIDIA has directly infringed literally and/or through the doctrine of equivalents at least one claim of the '768 patent by making, using, selling, offering for sale, and/or importing one or more of the Accused Server Products and the Accused GPU Products, in violation of 35 U.S.C. § 271(a). NVIDIA continues to engage in acts of direct infringement of the '768 patent.

130. Exhibit H shows an exemplary claim chart demonstrating how the Accused Server Products use NVIDIA's NVLink technology to practice claim 1 of the '768 patent. Exhibit H also shows an exemplary claim chart demonstrating how the Accused Server Products and the Accused GPU Products are configured to use NVIDIA's NVLink technology in a manner that practices claim 35 of the '768 patent.

131. ACS is informed and believes, and thereon alleges that NVIDIA has actively, knowingly, and intentionally induced infringement of at least claim 1 and 35 of the '768 patent in violation of 35 U.S.C. § 271(b). ACS is informed and believes, and thereon alleges that NVIDIA, knowing that the Accused Server Products and the Accused GPU Products infringe the '768 patent as shown in Exhibit H, have actively encouraged third parties such as customers, users, system builders, NVIDIA Partner Network Deep Learning Partners, and retailers/distributors to make, use, sell, or offer for sale within the United States and/or import into the United States the Accused Server Products and the Accused GPU Products, knowing and intending that the actions by the third parties constitute patent infringement. ACS is informed and believes, and thereon alleges that the actions of NVIDIA actually resulted in direct infringement by the third parties.

132. For example, ACS is informed and believes, and thereon alleges that NVIDIA makes, uses, sells, offers for sale, and/or imports one or more of the Accused GPU Products for or on behalf of third parties such as customers, users, system builders, NVIDIA Partner Network Deep Learning Partners, and retailers/distributors, knowing and intending that the Accused GPU Products will be used by the third parties in a manner that practices at least claim 1 of the '768 patent as shown in Exhibit H. For example, NVIDIA published and provided marketing materials, technical specifications, whitepapers, datasheets, user manuals, and development and testing information, and other resources on its website (http://www.nvidia.com/) that instructed and encouraged third parties to integrate the Accused GPU Products into products using NVIDIA's NVLink technology that were then made, used, sold, offered for sale and/or imported into the United States. E.g., https://www.nvidia.com/en-us/data-center/tesla-p100/; https://www.nvidia.com/en-us/data-center/tesla-v100/; https://www.nvidia.com/en-us/data-center/dgx-1/; https://www.nvidia.com/en-us/data-center/dgx-2/; https://www.nvidia.com/en-us/data-center/dgx-station/; https://www.nvidia.com/en-sg/data-center/hgx-1/; https://www.nvidia.com/en-us/data-center/hgx/; https://nvidianews.nvidia.com/news/tencent-cloud-adopts-nvidia-tesla-for-ai-cloud-computing; https://blogs.nvidia.com/blog/2017/10/03/oracle-cloud-tesla-gpus/; https://nvidianews.nvidia.com/news/china-s-top-cloud-providers-adopt-nvidia-volta-gpus-to-supercharge-next-gen-ai-services; https://blogs.nvidia.com/blog/2017/04/05/ibm-cloud-tesla-p100/. NVIDIA has also established the "NVIDIA Partner Network Deep Learning Partners" to assist customers with training, professional services, and service and support. *E.g.*, https://www.nvidia.com/en-us/data-center/where-to-buy-dgx-systems/. These activities were designed to bring infringing products that incorporate NVIDIA's Accused GPU Products to market

in the United States. As a result of NVIDIA's activities, the Accused GPU Products have been used in a manner that directly infringes the '768 patent. NVIDIA continues to engage in acts of induced infringement of the '768 patent.

133. ACS is informed and believes, and thereon alleges that NVIDIA has contributed to infringement of at least claim 1 and 35 of the '768 patent under 35 U.S.C. § 271(c) by making, using, selling, offering for sale, and/or importing in the United States the Accused GPU Products and Accused Server Products, which are known by NVIDIA to be especially made or especially adapted for use in an infringement of the '768 patent, and are not staple articles or commodities of commerce suitable for substantial non-infringing use. As a result, third parties such as customers, users, system builders, NVIDIA Partner Network Deep Learning Partners, and retailers/distributors have used NVIDIA's Accused GPU Products and Accused Server Products in a manner that directly infringes the '768 patent. NVIDIA continues to engage in acts of contributory infringement of the '768 patent.

134. The '768 patent is a continuation of U.S. Patent Application No. 13/423,063, now the '877 patent which is a continuation of U.S. Patent Application No. 12/040,519, now the '612 patent, which is a continuation-in-part of U.S. Application No. 11/744,461, now the '289 patent.

135. ACS is informed and believes, and thereon alleges that NVIDIA the knew of the '612 and '289 patents at least as early as 2012, when ACS sent an email to Stephen Jones with a data sheet informing him of the patent and outlining how the technology could be used with NVIDIA's products, as well as through ACS's marking of its SET™ product. ACS is informed and believes, and thereon alleges that NVIDIA knew or should have known of the '768 patent prior to the filing of this Complaint because it is a continuation of the '877 patent, which is a continuation of the '612 patent which is a continuation-in-part of the '289 patent, and NVIDIA

was explicitly informed of the ’289 and ’612 patents and of “Patents Pending.” Thus since at least the issuance of the ’768 patent in 2019, NVIDIA has known that its activities infringe the ’768 patent. In addition, since at least the issuance of the ’768 patent in 2019, NVIDIA knew or should have known that its activities induce and/or contribute to infringement of the ’768 patent by third parties.

136. ACS is informed and believes, and thereon alleges that NVIDIA’s infringement of the ’768 patent has been, and continues to be, willful, deliberate, and intentional by continuing its acts of infringement after knowing, or after it should have known, of the ’768 patent and that its conduct amounted to infringement of the ’768 patent.

137. As a consequence of NVIDIA’s infringement of the ’768 patent, ACS has been damaged and is entitled to monetary relief in an amount to be determined at trial.

138. Unless enjoined, NVIDIA and/or others acting on behalf of NVIDIA will continue their infringing acts, thereby causing additional irreparable injury to ACS for which there is no adequate remedy at law.

139. NVIDIA’s infringement of the ’768 patent has been willful and deliberate, entitling ACS to enhanced damages under 35 U.S.C. § 284 and attorneys’ fees and costs under 35 U.S.C. § 285.

PRAYER FOR RELIEF

WHEREFORE, ACS prays for judgment in its favor against NVIDIA for the following relief:

A. Pursuant to 35 U.S.C. § 271, a determination that NVIDIA has infringed, directly and indirectly, one or more of the ’289, ’612, ’877, and ’768 patents through the manufacture, use, offer for sale, sale and/or importation of infringing products, literally or under the doctrine of equivalents, and by way of inducement or contributory infringement;

B. Pursuant to 35 U.S.C. § 283, an injunction, including preliminary and permanent injunctive relief, enjoining NVIDIA and its officers, agents, servants, employees, attorneys and all others in active concert and/or participation with them from further infringing the ’289, ’612, ’877, and ’768 patents, and from the making, using, selling, offering for sale and/or importing infringing products;

C. Pursuant to 35 U.S.C. § 284, an award of damages sufficient to compensate ACS for NVIDIA’s infringement;

D. Pursuant to 35 U.S.C. § 284, an award increasing damages up to three times the amount found or assessed by the jury for NVIDIA’s willful infringement;

E. Pursuant to 35 U.S.C. § 285, a finding that this is an exceptional case, and an award of reasonable attorneys’ fees and costs;

F. Pursuant to 35 U.S.C. § 284, an award of prejudgment and post-judgment interest against NVIDIA;

G. Such other and further relief as this Court may deem just and proper.

DEMAND FOR A JURY TRIAL

Pursuant to Federal Rule of Civil Procedure 38(b), ACS hereby respectfully demands a trial by jury of all issues so triable raised by the Complaint.

Respectfully submitted,

October 7, 2020

/s/ Karen E. Keller

Karen E. Keller (No. 4489)
SHAW KELLER LLP
I.M. Pei Building
1105 N. Market Street, 12th Floor
Wilmington, DE 19801
(302) 298-0700
kkeller@shawkeller.com

Jon W. Gurka (Admitted *Pro Hac Vice)*
Brian C. Claassen (Admitted *Pro Hac Vice)*
Cheryl Burgess (Admitted *Pro Hac Vice)*
KNOBBE, MARTENS, OLSON & BEAR, LLP
2040 Main Street, 14th Floor
Irvine, CA 92614
(949) 760 0404
jon.gurka@knobbe.com
brian.claassen @knobbe.com
cheryl.burgess @knobbe.com

Attorneys for Plaintiff

Advanced Cluster Systems, Inc.

33231101

EXHIBIT A

US008082289B2

(12) **United States Patent**
Tannenbaum et al.

(10) **Patent No.: US 8,082,289 B2**
(45) **Date of Patent: Dec. 20, 2011**

(54) **CLUSTER COMPUTING SUPPORT FOR APPLICATION PROGRAMS**

(75) Inventors: **Zvi Tannenbaum**, Newport Beach, CA (US); **Dean E. Dauger**, Huntington Beach, CA (US)

(73) Assignee: **Advanced Cluster Systems, Inc.**, Aliso Viejo, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 1247 days.

(21) Appl. No.: **11/744,461**

(22) Filed: **May 4, 2007**

(65) **Prior Publication Data**

US 2007/0288935 A1 Dec. 13, 2007

Related U.S. Application Data

(60) Provisional application No. 60/813,738, filed on Jun. 13, 2006, provisional application No. 60/850,908, filed on Oct. 11, 2006.

(51) **Int. Cl.**
G06F 9/44 (2006.01)
G06F 15/16 (2006.01)

(52) **U.S. Cl.** **709/201**; 717/177; 719/320

(58) **Field of Classification Search** 709/201; 717/177; 719/320
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,423,046	A	6/1995	Nunnelley et al.	
5,881,315	A	3/1999	Cohen	
6,546,403	B1	4/2003	Carlson, Jr. et al.	
6,782,537	B1	8/2004	Blackmore et al.	
6,968,335	B2	11/2005	Bayliss et al.	
7,136,924	B2	11/2006	Dauger	
7,249,357	B2	7/2007	Landman et al.	
7,334,232	B2	2/2008	Jacobs et al.	
2002/0049859	A1 *	4/2002	Bruckert et al.	709/246
2003/0005266	A1	1/2003	Akkary et al.	
2003/0051062	A1	3/2003	Circenis	
2003/0135621	A1	7/2003	Romagnoli	
2003/0195931	A1	10/2003	Dauger	
2003/0195938	A1	10/2003	Howard et al.	
2004/0110209	A1	6/2004	Yokota et al.	
2004/0157203	A1	8/2004	Dunk	
2004/0252710	A1	12/2004	Jeter, Jr. et al.	
2005/0021751	A1	1/2005	Block et al.	

(Continued)

FOREIGN PATENT DOCUMENTS

JP	2002-117010	4/2002

OTHER PUBLICATIONS

International Search Report (Application No. PCT/US2007/076585) mailed Sep. 11, 2008.

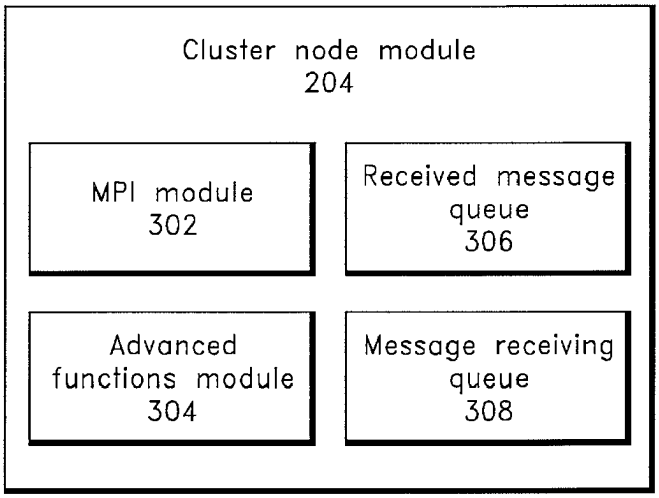
(Continued)

Primary Examiner — Larry Donaghue
(74) *Attorney, Agent, or Firm* — Knobbe Martens Olson & Bear, LLP

(57) **ABSTRACT**

A computer cluster system comprising a plurality of nodes and a software package comprising a user interface and a kernel for interpreting program code instructions is provided. In one embodiment, a cluster node module is configured to communicate with the kernel and other cluster node modules. The cluster node module accepts instructions from the user interface and interprets at least some of the instructions such that several cluster node modules in communication with one another and with a kernel can act as a computer cluster.

38 Claims, 5 Drawing Sheets



U.S. PATENT DOCUMENTS

2005/0038852	A1	2/2005	Howard	
2005/0060237	A1	3/2005	Barsness et al.	
2005/0076105	A1	4/2005	Keohane et al.	
2005/0108394	A1	5/2005	Braun et al.	
2005/0180095	A1	8/2005	Ellis	
2006/0053216	A1 *	3/2006	Deokar et al.	709/223
2006/0106931	A1	5/2006	Richoux	
2007/0073705	A1	3/2007	Gray	
2007/0288935	A1	12/2007	Tannenbaum et al.	
2008/0148244	A1	6/2008	Tannenbaum et al.	
2008/0250347	A1	10/2008	Gray et al.	
2008/0281997	A1	11/2008	Archer et al.	
2009/0222543	A1	9/2009	Tannenbaum et al.	

OTHER PUBLICATIONS

Mathematica Parallel Computing Toolkit Jan. 2005 pp. 1-90.

Dauger et al. Plug-and Play Cluster Computing: High-Performance Computing for the Mainstream Apr. 2004 pp. 22-28.

Dauger et al. "Plug-and-Play" Cluster Computing using Mac OS X 2003.

Carns et al. An Evaluation of Message Passing Implementations on Beowulf Workstations Mar. 1999 pp. 41-54.

Wolfram, Stephen: The Mathematica Book 5th Edition; Wolfram Research, Inc. 2003.

Maeder, Roman E.; Mathematica Parallel Computing Toolkit—Unleash the Power of Parallel Computing; Wolfram Research, Jan. 2005.

Tepeneu and Ida, MathGridLink—A bridge between Mathematica and the Grid, (online); Department of Computer Science, Graduate School of Systems and Information Engineering; University of Tsukuba, 2003; Retrieved from the Internet ,URL:http://www.score.is.tsukuba.ac.jp/-ida/ida2004/Recent/bib/MathGridLink.pdf.. (Abstract).

International Search Report (Application No. PCT/US06/37275) mailed Sep. 24, 2007 (2 pages).

"gridMathematica 1.1: Grid Computing Gets a Speed Boost from Mathematica 5", The Mathematica Journal, vol. 9, No. 2, 2004.

Hamscher et al., "Evaluation of Job-Scheduling Strategies for grid Computing", LNCS: Lecture Notes in Computer Science, 2000, pp. 191-202.

Jahanzeb et al., "Libra: a computational economy-based job scheduling system for clusters", Software Practice and Experience, Feb. 24, 2004, vol. 34, pp. 573-590.

Jain et al., "Data Clustering: A Review", ACM Computing Surveys, Sep. 1999, vol. 31, No. 3.

Office Action dated Jul. 11, 2011, U.S. Appl. No. 12/040,519, filed Feb. 28, 2008.

Website http://wolfram.com/products/gridmathematica, entitlled Wolfram gridMathematica, printed on Oct. 3, 2007.

* cited by examiner

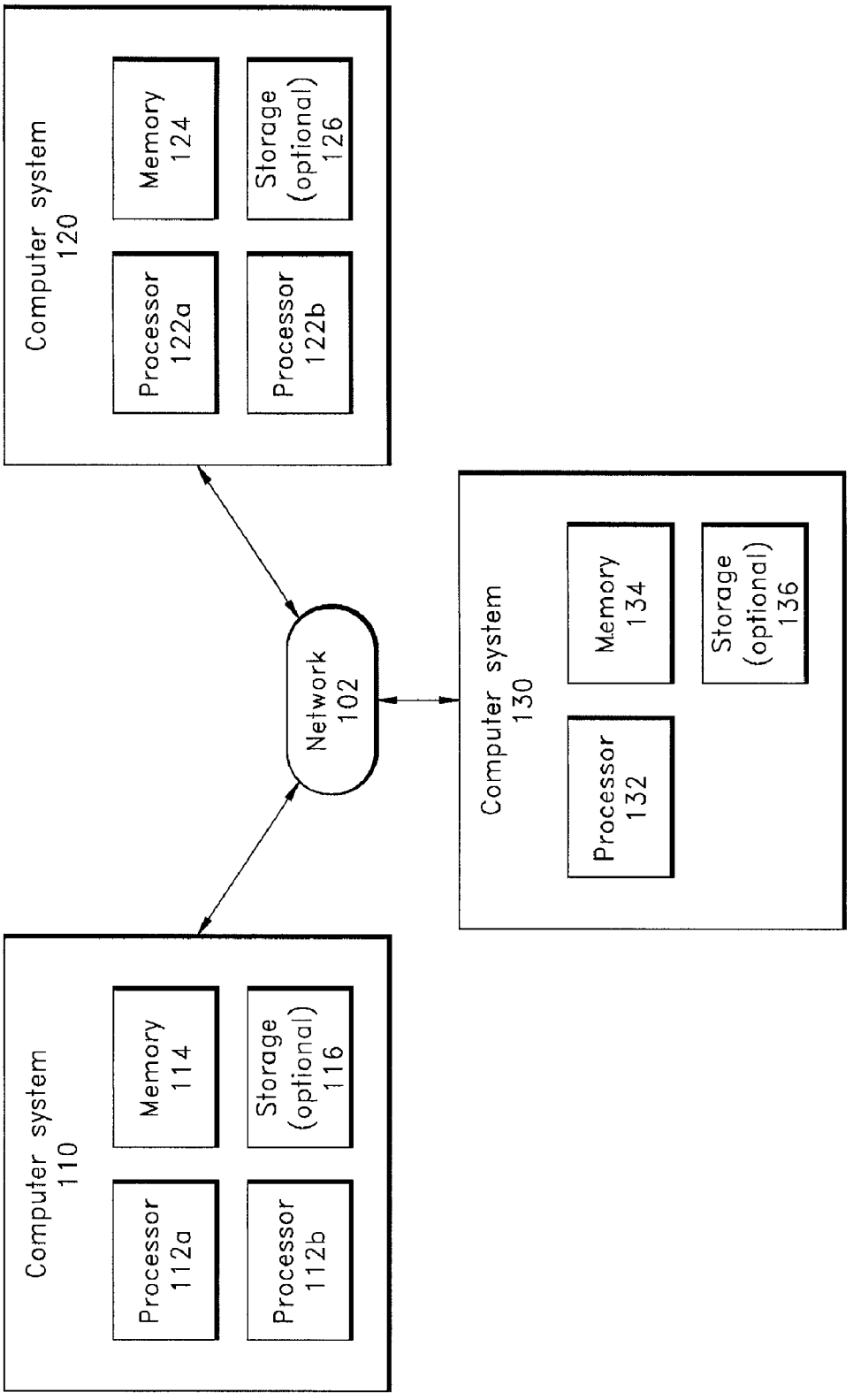

Computer system
120
Memory
124
Storage
(optional)
126
Processor
122a
Processor
122b
Network
102
Computer system
130
Memory
134
Storage
(optional)
136
Processor
132
Computer system
110
Memory
114
Storage
(optional)
116
Processor
112a
Processor
112b

FIG. 1

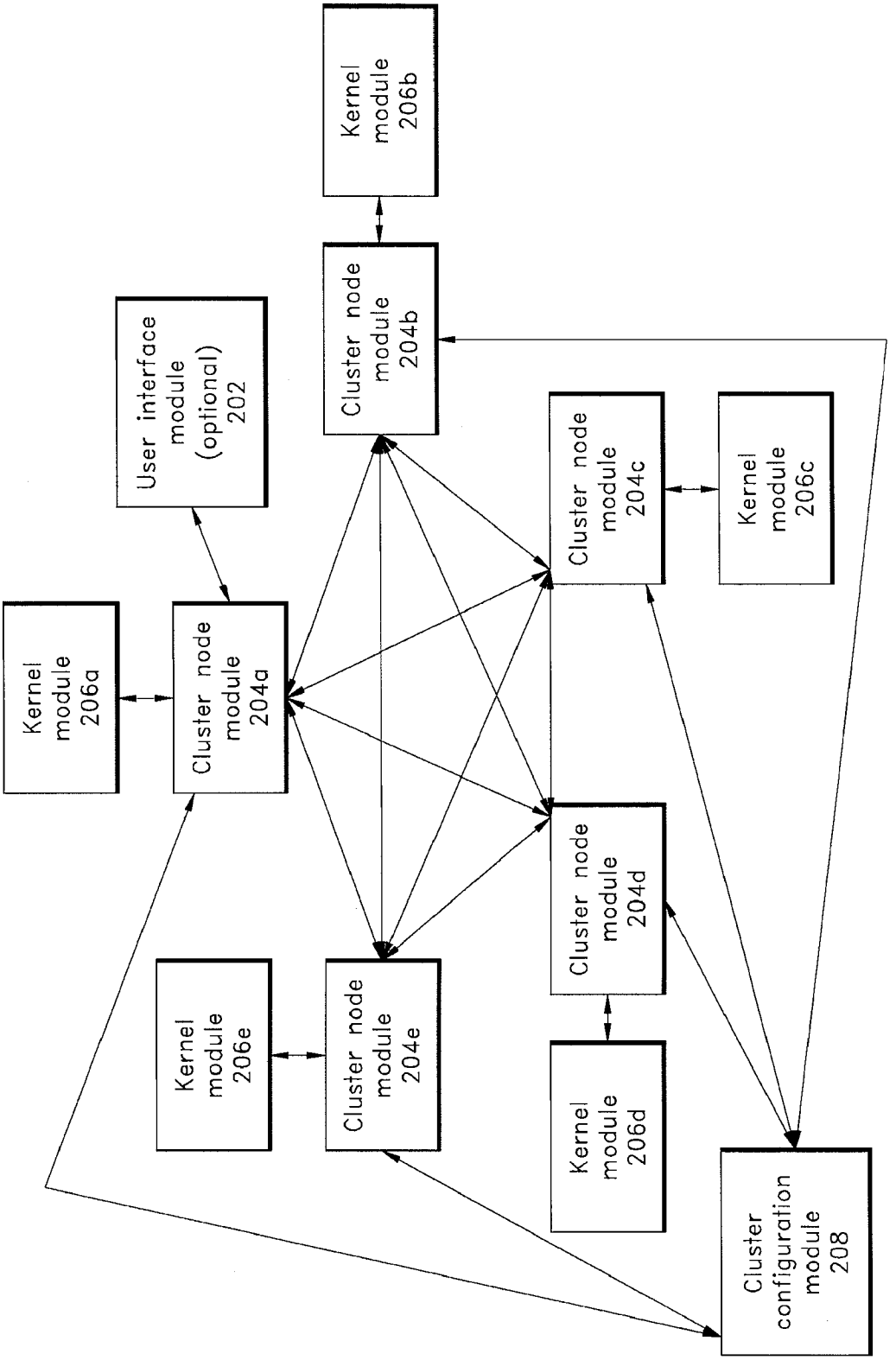

Kernel module 206b
Cluster node module 204b
User interface module (optional) 202
Cluster node module 204c
Kernel module 206c
Kernel module 206a
Cluster node module 204a
Cluster node module 204d
Kernel module 206e
Cluster node module 204e
Kernel module 206d
Cluster configuration module 208

FIG. 2

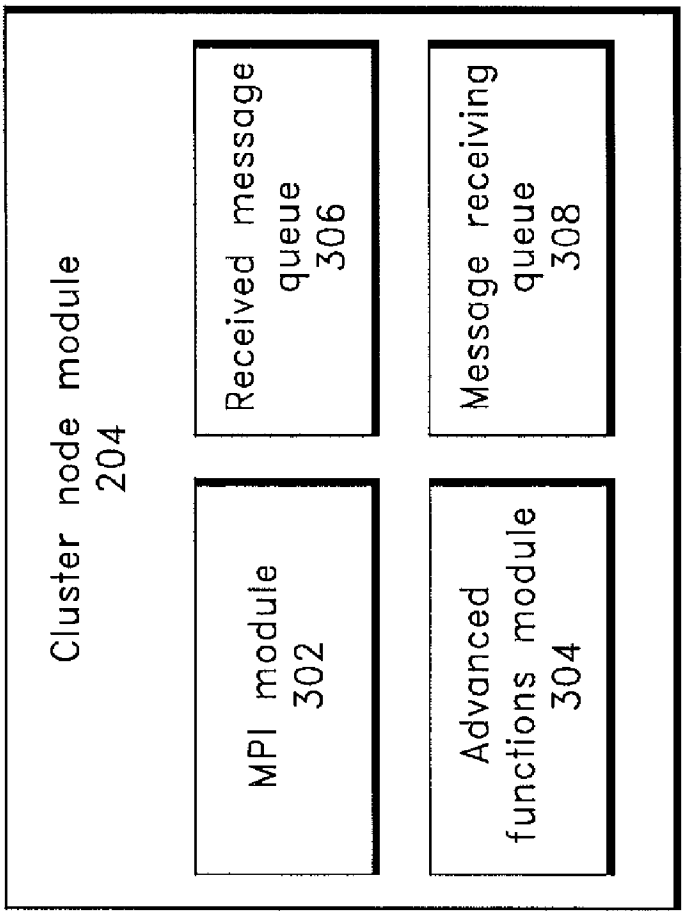

Cluster node module
204
Received message queue
306
Message receiving queue
308
MPI module
302
Advanced functions module
304

FIG. 3

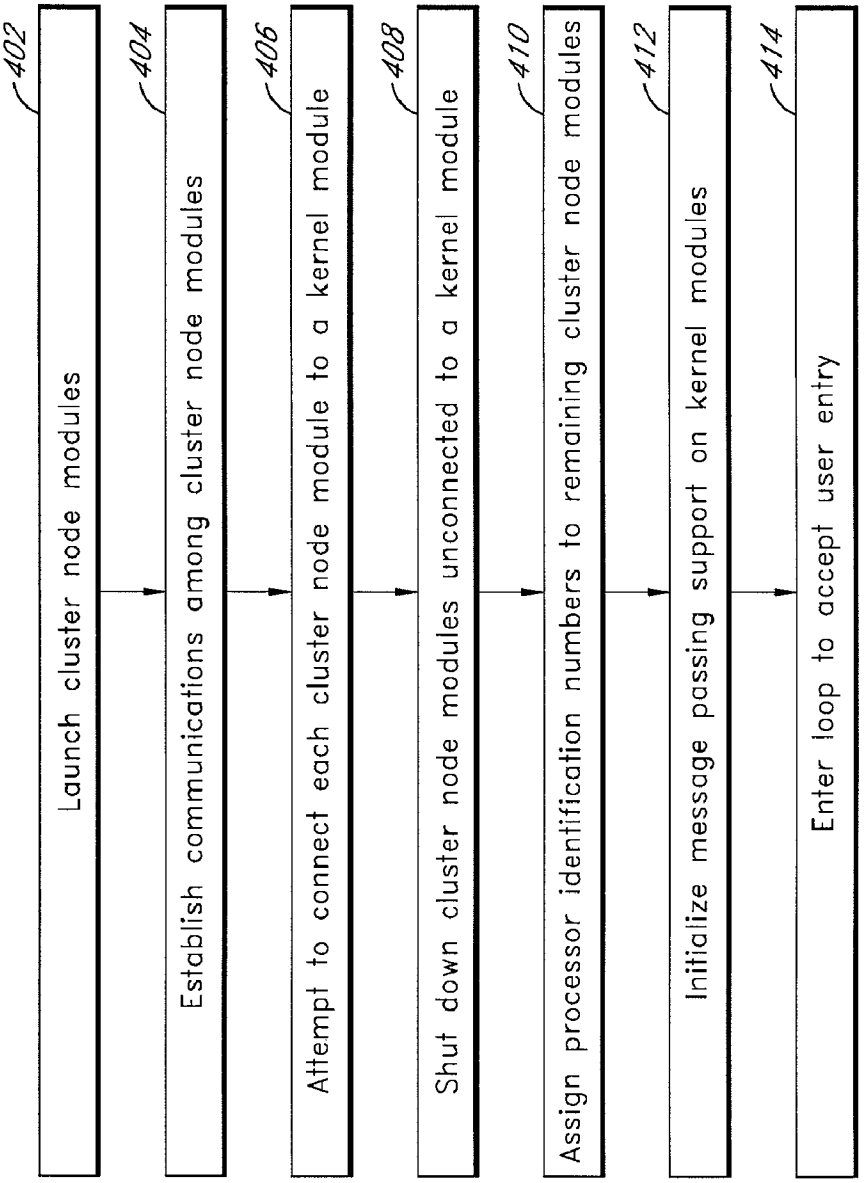

402
Launch cluster node modules
404
Establish communications among cluster node modules
406
Attempt to connect each cluster node module to a kernel module
408
Shut down cluster node modules unconnected to a kernel module
410
Assign processor identification numbers to remaining cluster node modules
412
Initialize message passing support on kernel modules
414
Enter loop to accept user entry

FIG. 4

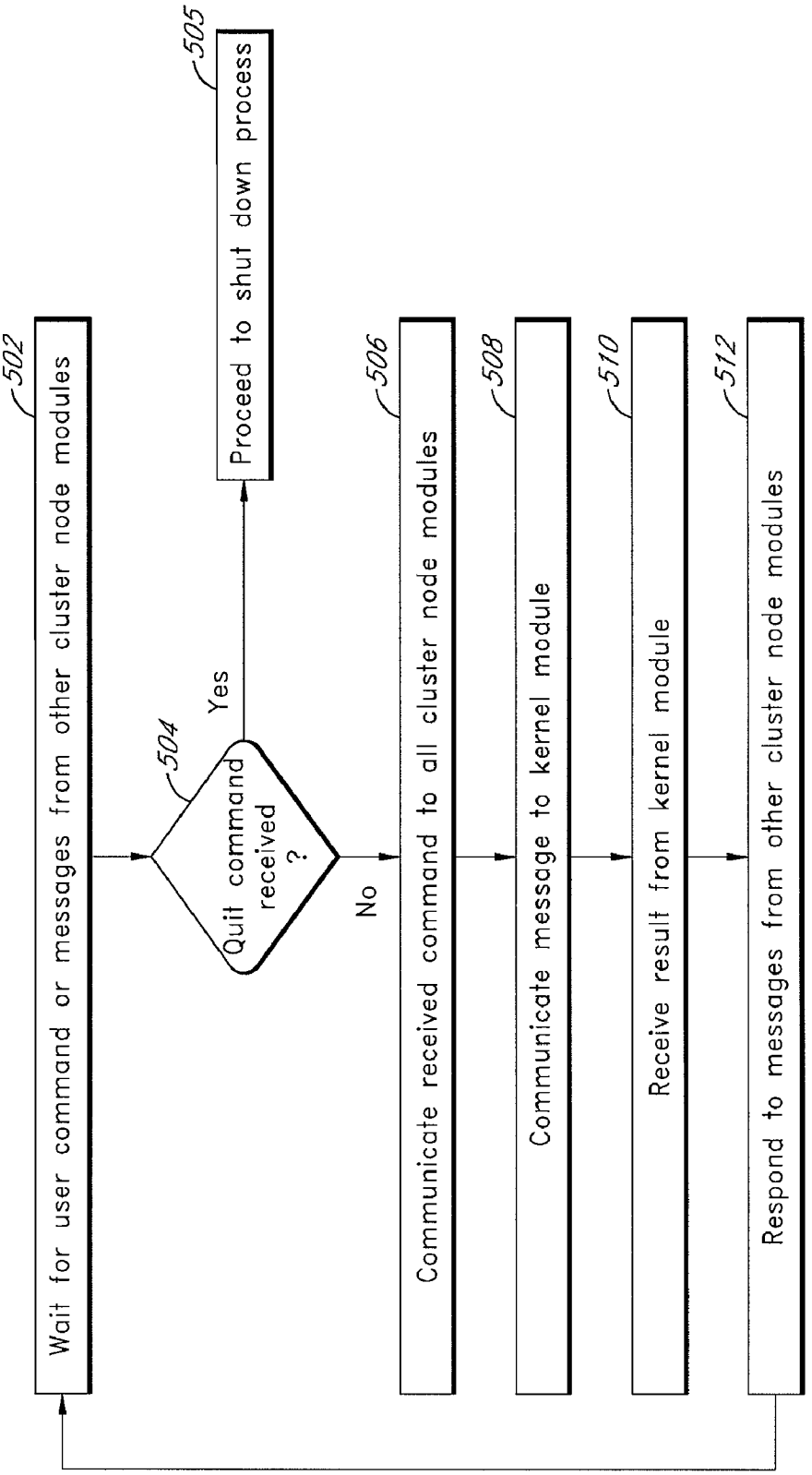

502
Wait for user command or messages from other cluster node modules
504
Quit command received ?
Yes
505
Proceed to shut down process
No
506
Communicate received command to all cluster node modules
508
Communicate message to kernel module
510
Receive result from kernel module
512
Respond to messages from other cluster node modules

FIG. 5

CLUSTER COMPUTING SUPPORT FOR APPLICATION PROGRAMS

PRIORITY INFORMATION

This application claims priority to U.S. Provisional Patent Application No. 60/813,738, filed Jun. 13, 2006 and U.S. Provisional Patent Application No. 60/850,908, filed Oct. 11, 2006. The entirety of each of the above-referenced applications is hereby incorporated by reference and made part of this specification.

BACKGROUND

1. Field of the Disclosure

The present disclosure relates to the field of cluster computing generally and to systems and methods for adding cluster computing functionality to a computer program, in particular.

2. Description of the Related Art

Computer clusters include a group of two or more computers, microprocessors, and/or processor cores ("nodes") that intercommunicate so that the nodes can accomplish a task as though they were a single computer. Many computer application programs are not currently designed to benefit from advantages that computer clusters can offer, even though they may be running on a group of nodes that could act as a cluster. Some computer programs can run on only a single node because, for example, they are coded to perform tasks serially or because they are designed to recognize or send instructions to only a single node.

Some application programs include an interpreter that executes instructions provided to the program by a user, a script, or another source. Such an interpreter is sometimes called a "kernel" because, for example, the interpreter can manage at least some hardware resources of a computer system and/or can manage communications between those resources and software (for example, the provided instructions, which can include a high-level programming language). Some software programs include a kernel that is designed to communicate with a single node. An example of a software package that includes a kernel that is designed to communicate with a single node is Mathematica® from Wolfram Research, Inc. ("Mathematica"). Mathematics software packages from other vendors and other types of software can also include such a kernel.

A product known as gridMathematica, also from Wolfram Research, Inc., gives Mathematica the capability to perform a form of grid computing known as "distributed computing." Grid computers include a plurality of nodes that generally do not communicate with one another as peers. Distributed computing can be optimized for workloads that consist of many independent jobs or packets of work, which do not need to share data between the jobs during the computational process. Grid computers include at least one node known as a master node that manages a plurality of slave nodes or computational nodes. In gridMathematica, each of a plurality of kernels runs on a single node. One kernel is designated the master kernel, which handles all input, output, and scheduling of the other kernels (the computational kernels or slave kernels). Computational kernels receive commands and data only from the node running the master kernel. Each computational kernel performs its work independently of the other computational kernels and intermediate results of one job do not affect other jobs in progress on other nodes.

SUMMARY

Embodiments described herein have several features, no single one of which is solely responsible for their desirable attributes. Without limiting the scope of the invention as expressed by the claims, some of the advantageous features will now be discussed briefly.

Some embodiments described herein provide techniques for conveniently adding cluster computing functionality to a computer application. In one embodiment, a user of a software package may be able to achieve higher performance and/or higher availability from the software package by enabling the software to benefit from a plurality of nodes in a cluster. One embodiment allows a user to create applications, using a high-level language such as Mathematica, that are able to run on a computer cluster having supercomputer-like performance. One embodiment provides access to such high-performance computing through a Mathematica Front End, a command line interface, one or more high-level commands, or a programming language such as C or FORTRAN.

One embodiment adapts a software module designed to run on a single node, such as, for example, the Mathematica kernel, to support cluster computing, even when the software module is not designed to provide such support. One embodiment provides parallelization for an application program, even if no access to the program's source code is available. One embodiment adds and supports Message Passing Interface ("MPI") calls directly from within a user interface, such as, for example, the Mathematica programming environment. In one embodiment, MPI calls are added to or made available from an interactive programming environment, such as the Mathematica Front End.

One embodiment provides a computer cluster including a first processor, a second processor, and a third processor. The cluster includes at least one computer-readable medium in communication at least one of the first processor, the second processor, or the third processor. A first kernel resides in the at least one computer-readable medium and is configured to translate commands into code for execution on the first processor. A first cluster node module resides in the at least one computer-readable medium. The first cluster node module is configured to send commands to the first kernel and receives commands from a user interface. A second kernel resides in the at least one computer-readable medium. The second kernel is configured to translate commands into code for execution on the second processor. A second cluster node module resides in the at least one computer-readable medium. The second cluster node module is configured to send commands to the second kernel and communicates with the first cluster node module. A third kernel resides in the at least one computer-readable medium. The third kernel is configured to translate commands into code for execution on the third processor. A third cluster node module resides in the at least one computer-readable medium. The third cluster node module is configured to send commands to the third kernel and configured to communicate with the first cluster node module and the second cluster node module. The first cluster node module comprises a data structure in which messages originating from the second and third cluster node modules are stored.

Another embodiment provides a computer cluster that includes a plurality of nodes and a software package including a user interface and a single-node kernel for interpreting program code instructions. A cluster node module is configured to communicate with the single-node kernel and other cluster node modules. The cluster node module accepts instructions from the user interface and interprets at least some of the instructions such that several cluster node mod-

ules in communication with one another act as a cluster. The cluster node module appears as a single-node kernel to the user interface. In one embodiment, the single-node kernel includes a Mathematical kernel. In some embodiments, the user interface can include at least one of a Mathematica front end or a command line. In some embodiments, the cluster node module includes a toolkit including library calls that implement at least a portion of MPI calls. In some embodiments, the cluster node module includes a toolkit including high-level cluster computing commands. In one embodiment, the cluster system can include a plurality of Macintosh® computers ("Macs"), Windows®-based personal computers ("PCs"), and/or Unix/Linux-based workstations.

A further embodiment provides a computer cluster including a plurality of nodes. Each node is configured to access a computer-readable medium comprising program code for a user interface and program code for a single-node kernel module configured to interpret user instructions. The cluster includes a plurality of cluster node modules. Each cluster node module is configured to communicate with a single-node kernel and with one or more other cluster node modules, to accept instructions from the user interface, and to interpret at least some of the user instructions such that the plurality of cluster node modules communicate with one another in order to act as a cluster. A communications network connects the nodes. One of the plurality of cluster node modules returns a result to the user interface.

Another embodiment provides a method of evaluating a command on a computer cluster. A command from at least one of a user interface or a script is communicated to one or more cluster node modules within the computer cluster. Each of the one or more cluster node modules communicates a message based on the command to a respective kernel module associated with the cluster node module. Each of the one or more cluster node modules receives a result from the respective kernel module associated with the cluster node module. At least one of the one or more cluster node modules responds to messages from other cluster node modules.

Another embodiment provides a computing system for executing Mathematica code on multiple nodes. The computing system includes a first node module in communication with a first Mathematica kernel executing on a first node, a second node module in communication with a second Mathematica kernel executing on a second node, and a third node module in communication with a third Mathematica kernel executing on a third node. The first node module, the second node module, and the third node module are configured to communicate with one another using a peer-to-peer architecture. In some embodiments, each of the first node module, the second node module, and third node module includes a data structure for maintaining messages originating from other node modules and a data structure for maintaining data specifying a location to which an message is expected to be received and an identifier for a node from which the message is expected to be sent.

BRIEF DESCRIPTION OF THE DRAWINGS

A general architecture that implements the various features are described with reference to the drawings. The drawings and the associated descriptions are provided to illustrate embodiments and not to limit the scope of the disclosure. Throughout the drawings, reference numbers are re-used to indicate correspondence between referenced elements.

FIG. **1** is a block diagram of one embodiment of a computer cluster.

FIG. **2** is a block diagram showing relationships between software modules running on one embodiment of a computer cluster.

FIG. **3** is a block diagram of one embodiment of a cluster node module.

FIG. **4** is a flow chart showing one embodiment of a cluster initialization process.

FIG. **5** is a flow chart showing one embodiment of the operation of a cluster node module.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

For purposes of illustration, some embodiments are described herein in the context of cluster computing with Mathematica software. The present disclosure is not limited to a single software program; the systems and methods can be used with other application software such as, for example, Maple®, MATLAB®, MathCAD®, Apple Shake®, Apple® Compressor, IDL®, other applications employing an interpreter or a kernel, Microsoft Excel®, Adobe After Effects®, Adobe Premiere®, Adobe Photoshop®, Apple Final Cut Pro®, and Apple iMovie®. Some figures and/or descriptions, however, relate to embodiments of computer clusters running Mathematica. The system can include a variety of uses, including but not limited to students, educators, scientists, engineers, mathematicians, researchers, and technicians. It is also recognized that in other embodiments, the systems and methods can be implemented as a single module and/or implemented in conjunction with a variety of other modules. Moreover, the specific implementations described herein are set forth in order to illustrate, and not to limit, the disclosure.

I. Overview

The cluster computing system described herein generally includes one or more computer systems connected to one another via a communications network or networks. The communications network can include one or more of a local area network ("LAN"), a wide area network ("WAN"), an intranet, the Internet, etc. In one embodiment, a computer system comprises one or more processors such as, for example, a microprocessor that can include one or more processing cores ("nodes"). The term "node" refers to a processing unit or subunit that is capable of single-threaded execution of code. The processors can be connected to one or more memory devices such as, for example, random access memory ("RAM"), and/or one or more optional storage devices such as, for example, a hard disk. Communications among the processors and such other devices may occur, for example, via one or more local buses of a computer system or via a LAN, a WAN, a storage area network ("SAN"), and/or any other communications network capable of carrying signals among computer system components. In one embodiment, one or more software modules such as kernels, run on nodes within the interconnected computer systems. In one embodiment, the kernels are designed to run on only a single node. In one embodiment, cluster node modules communicate with the kernels and with each other in order to implement cluster computing functionality.

FIG. **1** is a block diagram of one embodiment of a computer cluster **100** wherein computer systems **110**, **120**, **130** communicate with one another via a communications network **102**. Network **102** includes one or more of a LAN, a WAN, a wireless network, an intranet, or the Internet. In one embodiment of the computer cluster, computer system **110** includes processors **112***a*, **112***b*, memory **114**, and optional storage **116**. Other computer systems **120**, **130** can include similar devices, which generally communicate with one another

within a computer system over a local communications architecture such as a local bus (not shown). A computer system can include one or more processors, and each processor can contain one or more processor cores that are capable of single-threaded execution. Processor cores are generally independent microprocessors, but more than one can be included in a single chip package. Software code designed for single-threaded execution can generally run on one processor core at a time. For example, single-threaded software code typically does not benefit from multiple processor cores in a computer system.

FIG. **2** is a block diagram showing relationships among software modules running on one embodiment of a computer cluster **100**. In the embodiment shown in FIG. **2**, the kernel modules **206***a-e* are designed for single-threaded execution. For example, if each of the processors **112***a*, **112***b*, **122***a*, **122***b*, **132** shown in FIG. **1** includes only one processor core, two kernel modules (for example, kernel modules **206***a*, **206***b*) loaded into the memory **114** of computer system **110** could exploit at least some of the processing bandwidth of the two processors **112***a*, **112***b*. Similarly, two kernel modules **206***c*, **206***d* loaded into the memory **124** of computer system **120** could exploit at least some of the processing bandwidth of the two processors **122***a*, **122***b*. Likewise, the bandwidth of processor **132** of computer system **130** could be utilized by a single instance of a cluster node module **204***e* loaded into the computer system's memory **134**.

In the embodiment shown in FIG. **2**, each of the kernel modules **206***a-e* is in communication with a single cluster node module **204***a-e*, respectively. For example, the kernel module **206***a* is in communication with the cluster node module **204***a*, the kernel module **206***b* is in communication with the cluster node module **206***b*, and so forth. In one embodiment, one instance of a cluster node module **204***a-e* is loaded into a computer system's memory **114**, **124**, **134** for every instance of a kernel module **206***a-e* running on the system. As shown in FIG. **2**, each of the cluster node modules **204***a-e* is in communication with each of the other cluster node modules **204***a-e*. For example, one cluster node module **204***a* is in communication with all of the other cluster node modules **204***b-e*. A cluster node module **204***a* may communicate with another cluster node module **204***b* via a local bus (not shown) when, for example, both cluster node modules **204***a-b* execute on processors **112***a*, **112***b* within the same computer system **110**. A cluster node module **204***a* may also communicate with another cluster node module **204***c* over a communications network **102** when, for example, the cluster node modules **204***a, c* execute on processors **112***a*, **122***a* within different computer systems **110**, **120**.

As shown in FIG. **2**, an optional user interface module **202** such as, for example, a Mathematica front end and/or a command line interface, can connect to a cluster node module **204***a*. The user interface module can run on the same computer system **110** and/or the same microprocessor **112***a* on which the cluster node module **204***a* runs. The cluster node modules **204***a-e* provide MPI calls and/or advanced cluster functions that implement cluster computing capability for the single-threaded kernel modules. The cluster node modules **204***a-e* are configured to look and behave like a kernel module **206***a* from the perspective of the user interface module **202**. Similarly, the cluster node modules **204***a-e* are configured to look and behave like a user interface module **202** from the perspective of a kernel module **206***a*. The first cluster node module **204***a* is in communication with one or more other cluster node modules **204***b*, **204***c*, and so forth, each of which provides a set of MPI calls and/or advanced cluster commands. In one embodiment, MPI may be used to send messages between nodes in a computer cluster.

Communications can occur between any two or more cluster node modules (for example, between a cluster node module **204***a* and another cluster node module **204***c*) and not just between "adjacent" kernels. Each of the cluster node modules **204***a-e* is in communication with respective kernel modules **206***a-e*. Thus, the cluster node module **204***a* communicates with the kernel module **206***a*. MPI calls and advanced cluster commands are used to parallelize program code received from an optional user interface module **208** and distribute tasks among the kernel modules **206***a-e*. The cluster node modules **204***a-e* provide communications among kernel modules **206***a-e* while the tasks are executing. Results of evaluations performed by kernel modules **206***a-e* are communicated back to the first cluster node module **204***a* via the cluster node modules **204***a-e*, which communicates them to the user interface module **208**.

Intercommunication among kernel modules **206***a-e* during thread execution, which is made possible by cluster node modules **204***a-e*, provides advantages for addressing various types of mathematic and scientific problems, for example. Intercommunication provided by cluster computing permits exchange of information between nodes during the course of a parallel computation. Embodiments of the present disclosure provide such intercommunication for software programs such as Mathematica, while grid computing solutions can implement communication between only one master node and many slave nodes. Grid computing does not provide for communication between slave nodes during thread execution.

For purposes of providing an overview of some embodiments, certain aspects, advantages, benefits, and novel features of the invention are described herein. It is to be understood that not necessarily all such advantages or benefits can be achieved in accordance with any particular embodiment of the invention. Thus, for example, those skilled in the art will recognize that the invention can be embodied or carried out in a manner that achieves one advantage or group of advantages as taught herein without necessarily achieving other advantages or benefits as can be taught or suggested herein.

II. Computer Cluster **100**

As shown in FIG. **1**, one embodiment of a cluster system **100** includes computer systems **110**, **120**, **13.0** in communication with one another via a communications network **102**. A first computer system **110** can include one or more processors **112***a-b*, a memory device **114**, and an optional storage device **116**. Similarly, a second computer system **120** can include one or more processors **122***a-b*, a memory device **124**, and an optional storage device **126**. Likewise, a third computer system **130** can include one or more processors **132**, a memory device **134**, and an optional storage device **136**. Each of the computer systems **110**, **120**, **130** includes a network interface (not shown) for connecting to a communications network **102**, which can include one or more of a LAN, a WAN, an intranet, a wireless network, and/or the Internet.

A. Computer System **110**

In one embodiment, a first computer system **110** communicates with other computer systems **120**, **130** via a network **102** as part of a computer cluster **100**. In one embodiment, the computer system **110** is a personal computer, a workstation, a server, or a blade including one or more processors **112***a-b*, a memory device **114**, an optional storage device **116**, as well as a network interface module (not shown) for communications with the network **102**.

1. Processors **112***a-b*

In one embodiment, the computer system **110** includes one or more processors **112***a-b*. The processors **112***a-b* can be one or more general purpose single-core or multi-core microprocessors such as, for example, a Pentium® processor, a Pentium® II processor, a Pentium® Pro processor, a Pentium® III processor, Pentium® 4 processor, a Core Duo® processor, a Core 2 Duo® processor, a Xeon® processor, an Itanium® processor, a Pentium® M processor, an x86 processor, an Athlon® processor, an 8051 processor, a MIPS® processor, a PowerPC® processor, an ALPHA® processor, etc. In addition, one or more of the processors **112***a-b* can be a special purpose microprocessor such as a digital signal processor. The total number of processing cores (for example, processing units capable of single-threaded execution) within all processors **112***a-b* in the computer system **110** corresponds to the number of nodes available in the computer system **110**. For example, if the processors **112***a-b* were each Core 2 Duo® processors having two processing cores, computer system **110** would have four nodes in all. Each node can run one or more instances of a program module, such as a single-threaded kernel module.

2. Network Interface Module

The computer system **110** can also include a network interface module (not shown) that facilitates communication between the computer system **110** and other computer systems **120**, **130** via the communications network **102**.

The network interface module can use a variety of network protocols. In one embodiment, the network interface module includes TCP/IP. However, it is to be appreciated that other types of network communication protocols such as, for example, Point-to-Point Protocol ("PPP"), Server Message Block ("SMB"), Serial Line Internet Protocol ("SLIP"), tunneling PPP, AppleTalk, etc., may also be used.

3. Memory **114** and Storage **116**

The computer system **110** can include memory **114**. Memory **114** can include, for example, processor cache memory (such as processor core-specific or cache memory shared by multiple processor cores), dynamic random-access memory ("DRAM"), static random-access memory ("SRAM"), or any other type of memory device capable of storing computer data, instructions, or program code. The computer system **110** can also include optional storage **116**. Storage **116** can include, for example, one or more hard disk drives, floppy disks, flash memory, magnetic storage media, CD-ROMs, DVDs, optical storage media, or any other type of storage device capable of storing computer data, instructions, and program code.

4. Computer System **110** Information

The computer system **110** may be used in connection with various operating systems such as: Microsoft® Windows® 3.X, Windows 95®, Windows 98®, Windows NT®, Windows 2000®, Windows XP®, Windows CE®, Palm Pilot OS, OS/2, Apple® MacOS®, MacOS X®, MacOS X Server®, Disk Operating System (DOS), UNIX, Linux®, VxWorks, or IBM® OS/2®, Sun OS, Solaris OS, IRIX OS operating systems, etc.

In one embodiment, the computer system **110** is a personal computer, a laptop computer, a Blackberry® device, a portable computing device, a server, a computer workstation, a local area network of individual computers, an interactive kiosk, a personal digital assistant, an interactive wireless communications device, a handheld computer, an embedded computing device, or the like.

As can be appreciated by one of ordinary skill in the art, the computer system **110** may include various sub-routines, procedures, definitional statements, and macros. Each of the foregoing modules are typically separately compiled and linked into a single executable program. However, it is to be appreciated by one of ordinary skill in the art that the processes that are performed by selected ones of the modules may be arbitrarily redistributed to one of the other modules, combined together in a single module, made available in a shareable dynamic link library, or partitioned in any other logical way.

B. Computer System **120**

In one embodiment, a second computer system **120** communicates with other computer systems **110**, **130** via a network **102** as part of a computer cluster **100**. In one embodiment, the computer system **120** is a personal computer, a workstation, a server, or a blade including one or more processors **122***a-b*, a memory device **124**, an optional storage device **126**, as well as a network interface module (not shown) for communications with the network **102**.

1. Processors **112***a-b*

In one embodiment, the computer system **120** includes one or more processors **122***a-b*. The processors **122***a-b* can be one or more general purpose single-core or multi-core microprocessors such as a Pentium® processor, a Pentium® II processor, a Pentium® Pro processor, a Pentium® III processor, Pentium® 4 processor, a Core Duo® processor, a Core 2 Duo® processor, a Xeon® processor, an Itanium® processor, a Pentium® M processor, an x86 processor, an Athlon® processor, an 8051 processor, a MIPS® processor, a PowerPC® processor, an ALPHA® processor, etc. In addition, the processors **122***a-b* can be any special purpose microprocessors such as a digital signal processor. The total number of processing cores (for example, processing units capable of single-threaded execution) within all processors **122***a-b* in the computer system **120** corresponds to the number of nodes available in the computer system **120**. For example, if the processors **122***a-b* were each Core 2 Duo® processors having two processing cores, computer system **120** would have four nodes in all. Each node can run one or more instances of a program module, such as a single-threaded kernel module.

2. Network Interface Module

The computer system **120** can also include a network interface module (not shown) that facilitates communication between the computer system **120** and other computer systems **110**, **130** via the communications network **102**.

The network interface module can use a variety of network protocols. In one embodiment, the network interface module includes TCP/IP. However, it is to be appreciated that other types of network communication protocols such as, for example, Point-to-Point Protocol ("PPP"), Server Message Block ("SMB"), Serial Line Internet Protocol ("SLIP"), tunneling PPP, AppleTalk, etc., may also be used.

3. Memory **124** and Storage **126**

The computer system **120** can include memory **124**. Memory **124** can include, for example, processor cache memory (such as processor core-specific or cache memory shared by multiple processor cores), dynamic random-access memory ("DRAM"), static random-access memory ("SRAM"), or any other type of memory device capable of storing computer data, instructions, or program code. The computer system **120** can also include optional storage **126**. Storage **126** can include, for example, one or more hard disk drives, floppy disks, flash memory, magnetic storage media, CD-ROMs, DVDs, optical storage media, or any other type of storage device capable of storing computer data, instructions, and program code.

4. Computer System **120** Information

The computer system **120** may be used in connection with various operating systems such as: Microsoft® Windows®

3.X, Windows 95®, Windows 98®, Windows NT®, Windows 2000®, Windows XP®, Windows CE®, Palm Pilot OS, OS/2, Apple® MacOS®, MacOS X®, MacOS X Server®, Disk Operating System (DOS), UNIX, Linux®, VxWorks, or IBM® OS/2®, Sun OS, Solaris OS, IRIX OS operating systems, etc.

In one embodiment, the computer system **120** is a personal computer, a laptop computer, a Blackberry® device, a portable computing device, a server, a computer workstation, a local area network of individual computers, an interactive kiosk, a personal digital assistant, an interactive wireless communications device, a handheld computer, an embedded computing device, or the like.

As can be appreciated by one of ordinary skill in the art, the computer system **120** may include various sub-routines, procedures, definitional statements, and macros. Each of the foregoing modules are typically separately compiled and linked into a single executable program. However, it is to be appreciated by one of ordinary skill in the art that the processes that are performed by selected ones of the modules may be arbitrarily redistributed to one of the other modules, combined together in a single module, made available in a shareable dynamic link library, or partitioned in any other logical way.

C. Computer System **130**

In one embodiment, a third computer system **130** communicates with other computer systems **110**, **120** via a network **102** as part of a computer cluster **100**. In one embodiment, the computer system **130** is a personal computer, a workstation, a server, or a blade including one or more processors **132**, a memory device **134**, an optional storage device **136**, as well as a network interface module (not shown) for communications with the network **102**.

1. Processors **112***a-b*

In one embodiment, the computer system **130** includes a processor **132**. The processor **132** can be a general purpose single-core or multi-core microprocessors such as a Pentium® processor, a Pentium® II processor, a Pentium® Pro processor, a Pentium® III processor, Pentium® 4 processor, a Core Duo® processor, a Core 2 Duo® processor, a Xeon® processor, an Itanium® processor, a Pentium® M processor, an x86 processor, an Athlon® processor, an 8051 processor, a MIPS® processor, a PowerPC® processor, or an ALPHAS processor. In addition, the processor **132** can be any special purpose microprocessor such as a digital signal processor. The total number of processing cores (for example, processing units capable of single-threaded execution) within processor **132** in the computer system **130** corresponds to the number of nodes available in the computer system **130**. For example, if the processor **132** was a Core 2 Duo® processor having two processing cores, the computer system **130** would have two nodes. Each node can run one or more instances of a program module, such as a single-threaded kernel module.

2. Network Interface Module

The computer system **130** can also include a network interface module (not shown) that facilitates communication between the computer system **130** and other computer systems **110**, **120** via the communications network **102**.

The network interface module can use a variety of network protocols. In one embodiment, the network interface module includes TCP/IP. However, it is to be appreciated that other types of network communication protocols such as, for example, Point-to-Point Protocol ("PPP"), Server Message Block ("SMB"), Serial Line Internet Protocol ("SLIP"), tunneling PPP, AppleTalk, etc., may also be used.

3. Memory **134** and Storage **136**

The computer system **130** can include memory **134**. Memory **134** can include, for example, processor cache memory (such as processor core-specific or cache memory shared by multiple processor cores), dynamic random-access memory ("DRAM"), static random-access memory ("SRAM"), or any other type of memory device capable of storing computer data, instructions, or program code. The computer system **130** can also include optional storage **136**. Storage **136** can include, for example, one or more hard disk drives, floppy disks, flash memory, magnetic storage media, CD-ROMs, DVDs, optical storage media, or any other type of storage device capable of storing computer data, instructions, and program code.

4. Computer System **130** Information

The computer system **130** may be used in connection with various operating systems such as: Microsoft® Windows® 3.X, Windows 95®, Windows 98®, Windows NT®, Windows 2000®, Windows XP®, Windows CE®, Palm Pilot OS, OS/2, Apple® MacOS®, MacOS X®, MacOS X Server®, Disk Operating System (DOS), UNIX, Linux®, VxWorks, or IBM® OS/2®, Sun OS, Solaris OS, IRIX OS operating systems, etc.

In one embodiment, the computer system **130** is a personal computer, a laptop computer, a Blackberry® device, a portable computing device, a server, a computer workstation, a local area network of individual computers, an interactive kiosk, a personal digital assistant, an interactive wireless communications device, a handheld computer, an embedded computing device, or the like.

As can be appreciated by one of ordinary skill in the art, the computer system **130** may include various sub-routines, procedures, definitional statements, and macros. Each of the foregoing modules are typically separately compiled and linked into a single executable program. However, it is to be appreciated by one of ordinary skill in the art that the processes that are performed by selected ones of the modules may be arbitrarily redistributed to one of the other modules, combined together in a single module, made available in a shareable dynamic link library, or partitioned in any other logical way.

E. Communications Network **102**

In one embodiment, computer systems **110**, **120**, **130** are in communication with one another via a communications network **102**.

The communications network **102** may include one or more of any type of electronically connected group of computers including, for instance, the following networks: a virtual private network, a public Internet, a private Internet, a secure Internet, a private network, a public network, a value-added network, a wired network, a wireless network, an intranet, etc. In addition, the connectivity to the network can be, for example, a modem, Ethernet (IEEE 802.3), Gigabit Ethernet, 10-Gigabit Ethernet, Token Ring (IEEE 802.5), Fiber Distributed Datalink Interface (FDDI), Frame Relay, InfiniBand, Myrinet, Asynchronous Transfer Mode (ATM), or another interface. The communications network **102** may connect to the computer systems **110**, **120**, **130**, for example, by use of a modem or by use of a network interface card that resides in each of the systems.

In addition, the same or different communications networks **102** may be used to facilitate communication between the first computer system **110** and the second computer system **120**, between the first computer system **110** and the third computer system **130**, and between the second computer system **120** and the third computer system **130**.

III. Software Modules

As shown in FIGS. **1** and **2**, one embodiment of a cluster system **100** includes a user interface module **202** that is able to access a plurality of kernel modules **206***a-e* by communicating with a first cluster node module **204***a*. User interface module can be stored in a memory **114**, **124**, **134** while running, for example, and/or can be stored in a storage device **116**, **126**, **136**. The first cluster node module **204***a* is in communication with each of the other cluster node modules **204***b-e*. The kernel modules **206***a-e* can reside in the memory of one or more computer systems on which they run. For example, the memory **114** of the first computer system **110** can store instances of kernel modules **206***a-b*, the memory **124** of the second computer system **120** can store instances of kernel modules **206***c*-d, and the memory **134** of the third computer system **130** can store an instance of kernel module **206***e*. The kernel modules **206***a-e*, which include single-threaded program code, are each associated with one of the processors **112***a*, **112***b*, **122***a*, **122***b*, **132**. A cluster configuration module stored on one or more of the computer systems **110**, **120**, **130** or on a remote computer system, for example, can establish communication with the cluster node modules **204***a-e*. In one embodiment, communication between the cluster configuration module **208** and the cluster node modules **204***a-e* initializes the cluster node modules **204***a-e* to provide cluster computing support for the computer cluster **100**.

A. Cluster Node Module **204**

In one embodiment, the cluster node modules **204***a-e* provide a way for many kernel modules **206***a-e* such as, for example, Mathematica kernels, running on a computer cluster **100** to communicate with one another. A cluster node module **204** can include at least a portion of an application programming interface ("API") known as the Message-Passing Interface ("MPI"), which is used in several supercomputer and cluster installations. A network of connections (for example, the arrows shown in FIG. **2**) between the cluster node modules **204***a-e* can be implemented using a communications network **102**, such as, for example, TCP/IP over Ethernet, but the connections could also occur over any other type of network or local computer bus.

A cluster node module **204** can use an application-specific toolkit or interface such as, for example, Mathematica's MathLink, Add-Ons, or packets, to interact with an application. Normally used to connect a Mathematica kernel to a user interface known as the Mathematica Front End or other Mathematica kernels, MathLink is a bidirectional protocol to sends "packets" containing messages, commands, or data between any of these entities. MathLink does not allow direct cluster computing-like simultaneous communication between Mathematica kernels during execution of a command or thread. MathLink is also not designed to perform multiple simultaneous network connections. In some embodiments, a cluster node module **204** can use an application-specific toolkit such as, for example, MathLink, for connections between entities on the same computer.

When speaking about procedures or actions on a cluster or other parallel computer, not all actions happen in sequential order, nor are they required to. For example, a parallel code, as opposed to a single-processor code of the classic "Turing machine" model, has multiple copies of the parallel code running across the cluster, typically one for each processor (or "processing element" or "core"). Such parallel code is written in such a way that different instances of the same code can communicate, collaborate, and coordinate work with each other. Multiple instances of these codes can run at the same time in parallel.

If the count of the code instances is an integer N, each instance of code execution can be labeled **0** through N-**1**. For example, a computer cluster can include N connected computers, each containing a processor. The first has cluster node module **0** connected with kernel module **0** running on processor **0**. The next is cluster node module **1** and kernel module **1**, on processor **1**, and so forth for each of the N connected computers. Some steps of their procedure are collaborative, and some steps are independent. Even though these entities are not necessarily in lock-step, they do follow a pattern of initialization, main loop behavior (for example, cluster node module operation), and shut down.

In contrast, a parallel computing toolkit (PCT) that is provided as part of the gridMathematica software package does not provide a means for instances of the same code running on different nodes to communicate, collaborate, or coordinate work among the instances. The PCT provides commands that connect Mathematica kernels in a master-slave relationship rather than a peer-to-peer relationship as enabled by some embodiments disclosed herein. A computer cluster having peer-to-peer node architecture performs computations that can be more efficient, easier to design, and/or more reliable than similar computations performed on grid computers having master-slave node architecture. Moreover, the nature of some computations may not allow a programmer to harness multi-node processing power on systems that employ master-slave node architecture.

FIG. **3** shows one embodiment of a cluster node module **204** implementing MPI calls and advanced MPI functions. In the embodiment shown in FIG. **3**, cluster node module **204** includes MPI module **302**, advanced functions module **304**, received message queue **306**, and message receiving queue **308**.

1. MPI module **302**

In one embodiment, the cluster node module **204** includes an MPI module **302**. The MPI module **302** can include program code for one or more of at least five kinds of MPI instructions or calls. Selected constants, instructions, and/or calls that can be implemented by the MPI module **302** are as follows:

MPI Constants

Node identifiers are used to send messages to nodes or receive messages from them. In MPI, this is accomplished by assigning each node a unique integer ($IdProc) starting with **0**. This data, with a knowledge of the total count ($NProc), makes it possible to programmatically divide any measurable entity.

TABLE A

Constant	Description
$IdProc	The identification number of the current processor
$NProc	The number of processors in the current cluster
$mpiCommWorld	The communicator world of the entire cluster (see MPI Communicator routines, below)
mpiCommWorld	The default communicator world for the high-level routines.

Basic MPI Calls

In one embodiment, the MPI module **302** can include basic MPI calls such as, for example, relatively low-level routines that map MPI calls that are commonly used in other languages (such as C and Fortran), so that such calls can be available directly from the Mathematica user interface **204**. In some embodiments, basic MPI calls include calls that send data, equations, formulas, and/or other expressions.

Simply sending expressions from one node to another is possible with these most basic MPI calls. One node can call to send an expression while the other calls a corresponding routine to receive the sent expression. Because it is possible that the receiver has not yet called mpiRecv even if the message has left the sending node, completion of mpiSend is not a confirmation that it has been received.

TABLE B

Call	Description
mpiSend[expr, target, comm, tag]	Sends an expression expr to a node with the ID target in the communicator would comm, waiting until that expression has left this kernel
mpiRecv [expr, target, comm, tag]	Receives an expression into expr from a node with the ID target in the communicator world comm, waiting until the expression has arrived
mpiSendRecv[sendexpr, dest, recvexpr, source, comm]	Simultaneously sends the expression sendexpr to the node with the ID target and receives an expression into recvexpr from the node with the ID source in the communicator world comm, waiting until both operations have returned.

Asynchronous MPI Calls

Asynchronous calls make it possible for the kernel to do work while communications are proceeding simultaneously. It is also possible that another node may not be able to send or receive data yet, allowing one kernel to continue working while waiting.

TABLE C

Call	Description
mpiISend[expr, target, comm, tag, req]	Sends an expression expr to a processor with the ID target in the communicator world comm, returning immediately. It can be balanced with calls to mpiTest[req] until mpiTest[req] returns True.
mpiIRecv[expr, target, comm, tag, req]	Receives an expression expr from a proessor with the ID target in the communicator world comm, returning immediately. It can be balanced with calls to mpiTest[req] until mpiTest[req] returns True. The expr is not safe to access until mpiTest[req] returns True.
mpiTest[req]	Completes asynchronous behavior of mpiISend and mpiIRecv
mpWait[req]	Calls mpiTest until it returns True.
mpiWaitall[reglist]	Calls mpiWait all on every element of reqlist
mpiWaitany[reqlist]	Calls mpiTest on each element of realist until one of them returns True

The mpiISend[ ] command can be called from within a kernel module **206** (for example, a Mathematica kernel). It creates a packet containing the Mathematica expression to be sent as payload and where the expression should be sent. The packet itself is destined only for its local cluster node module. Once received by its local cluster node module, this packet is decoded and its payload is forwarded on to the cluster node module specified in the packet.

The mpiIRecv[ ] command can also be called from within a kernel module **206**. It creates a packet specifying where it expects to receive an expression and from which processor this expression is expected. Once received by its local cluster node module, this packet is decoded and its contents are stored in a message receiving queue (MRQ) **308** (FIG. **3**).

The mpiTest[ ] command can be called from within a kernel module **206**. It creates a packet specifying which message to test for completion, then waits for a reply expression to evaluate. Once received by the kernel module's associated cluster node module **204**, this packet is decoded and its message specifier is used to search for any matching expressions listed as completed in its received message queue (RMQ) **306**. If such completed expressions are found, it is sent to its local kernel module as part of the reply in mpiTest[ ]. The kernel module receives this reply expression and evaluates it, which updates the kernel module's variables as needed.

Other MPI calls are built on the fundamental calls mpiISend, mpiIRecv, and mpiTest. For example, mpiBcast, a broadcast, creates instructions to send information from the broadcast processor to all the others, while the other processors perform a Recv. Similarly, high-level calls of the toolkit can be built on top of the collection of MPI calls.

Collective MPI Calls

In one embodiment, the MPI module **302** can include program code for implementing collective MPI calls (for example, calls that provide basic multi-node data movement across nodes). Collective MPI calls can include broadcasts, gathers, transpose, and other vector and matrix operations, for example. Collective calls can also provide commonly used mechanisms to send expressions between groups of nodes.

TABLE D

Call	Description
mpiBcast[expr, root, comm]	Performs a broadcast of expr from the root processor to all the others in the communicator world comm. An expression is expected to be supplied by the root processor, while all the others expect expr to be overwritten by the incoming expression.
mpiGather[sendexpr, recvexpr, root, comm]	All processors (including root) in the communicator comm send their expression in sendexpr to the root processor, which produces a list of these expressions, in the order according to comm, in recvexpr. On the processors that are not root, recvexpr is ignored.
mpiAllgather[sendexpr, recvexpr, comm]	All processors in the communicator comm send their expression in sendexpr, which are organized into a list of these expressions, in the order according to comm, in recvexpr on all processors in comm.
mpiScatter[sendexpr, recvexpr, root, comm]	Processor root partitions the list in sendexpr into equal parts (if possible) and places each piece in recvexpr on all the processors (including root) in the communicator world comm, according the order and size of comm.
mpiAlltoall[sendexpr, recvexpr, comm]	Each processor sends equal parts of the list in sendexpr to all other processors in the communicator world comm, which each collects from all other processors are organizes into the order according to comm.

In one embodiment, the MPI module **302** includes program code for implementing parallel sums and other reduction operations on data stored across many nodes. MPI module **302** can also include program code for implementing simple parallel input/output calls (for example, calls that allow cluster system **200** to load and store objects that are located on a plurality of nodes).

TABLE E

Call	Description
mpiReduce[sendexpr, recvexpr, operation, root, comm]	Performs a collective reduction operation between expressions on all processors in the communicator world comm for every element in the list in sendexpr returning

TABLE E-continued

Call	Description
	the resulting list in recvexpr on the processor with the ID root.
mpiAllreduce[sendexpr, recvexpr, operation, comm]	Performs a collective reduction operation between expressions on all processors in the communicator world comm for every element in the list in sendexpr returning the resulting list in recvexpr on every processor.
mpiReduceScatter [sendexpr, recvexpr, operation, comm]	Performs a collective reduction operation between expressions on all processors in the communicator world comm for every element in the list in sendexpr, partitioning the resulting list into pieces for each processor's recvexpr.

These additional collective calls perform operations that reduce the data in parallel. The operation argument can be one of the constants below.

TABLE F

Constant	Description
mpiSum	Specifies that all the elements on different processors be added together in a reduction call
mpiMax	Specifies that the maximum of all the elements on different processors be chosen in a reduction call
mpiMin	Specifies that the minimum of all the elements on different processors be chosen in a reduction call

MPI Communicator Calls

In one embodiment, the MPI module **302** includes program code for implementing communicator world calls (for example, calls that would allow subsets of nodes to operate as if they were a sub-cluster). Communicators organize groups of nodes into user-defined subsets. The communicator values returned by mpiCommSplit[ ] can be used in other NPI calls instead of mpiCommWorld.

TABLE G

Call	Description
mpiCommSize[comm]	Returns the number of processors within the communicator comm
mpiCommRank[comm]	Returns the rank of this processor in the communicator comm
mpiCommDup[comm]	Returns a duplicate communicator of the communicator comm
mpiCommSplit[comm, color, key]	Creates a new communicator into several disjoint subsets each identified by color. The sort order within each subset is first by key, second according to the ordering in the previous communicator. Processors not meant to participate in any new communicator indicates this by passing the constant mpiUndefined. The corresponding communicator is returned to each calling processor.
mpiCommMap[comm] mpiCommMap[comm, target]	Returns the mapping of the communicator comm to the processor indexed according to $mpiCommWorld. Adding a second argument returns just the ID of the processor with the ID target in the communicator comm.
mpiCommFree[comm]	Frees the communicator comm

Other MPI Support Calls

Other calls that provide common functions include:

TABLE H

Call	Description
mpiWtime[ ]	Provides wall-dock time since some fixed time in the past. There is no guarantee that this time will read the same on all processors.
mpWtick[ ]	Returns the time resolution of mpiWtime[ ]
MaxByElement[in]	For every nth element of each list of the list in, chooses the maximum according to Max[ ], and returns the result as one list. Used in the mpiMax reduction operation.
MinByElement[in]	For every nth element of each list of the list in, chooses the minimum according to Min[ ], and returns the result as one list. Used in the mpiMin reduction operation.

2. Advanced Functions Module **304**

In one embodiment, the cluster node module **204** includes an advanced functions module **304**. The advanced functions module **304** can include program code that provides a toolkit of functions inconvenient or impractical to do with MPI instructions and calls implemented by the MPI module **302**. The advanced functions module **304** can rely at least partially on calls and instructions implemented by the MPI module **302** in the implementation of advanced functions. In one embodiment, the advanced functions module **304** includes a custom set of directives or functions. In an alternative embodiment, the advanced functions module **304** intercepts normal Mathematica language and converts it to one or more functions optimized for cluster execution. Such an embodiment can be easier for users familiar with Mathematica functions to use but can also complicate a program debugging process. Some functions implemented by the advanced functions module **304** can simplify operations difficult or complex to set up using parallel computing. Several examples of such functions that can be implemented by the advanced functions module **304** are shown below.

Built on the MPI calls, the calls that are described below provide commonly used communication patterns or parallel versions of Mathematica features. Unless otherwise specified, these are executed in the communicator mpiCommWorld, whose default is $mpiCommWorld, but can be changed to a valid communicator at run time.

Common Divide-and-Conquer Parallel Evaluation

In one embodiment, the advanced functions module **304** includes functions providing for basic parallelization such as, for example, routines that would perform the same operations on many data elements or inputs, stored on many nodes. These functions can be compared to parallelized for-loops and the like. The following calls address simple parallelization of common tasks. In the call descriptions, "expr" refers to an expression, and "loopspec" refers to a set of rules that determine how the expression is evaluated. In some embodiments, the advanced functions module **304** supports at least three forms of loopspec, including {var, count}, where the call iterates the variable var from 1 to the integer count; {var, start, stop}, where the call iterates the variable var every integer from start to stop; and {var, start, stop, increment}, where the call iterates the variable var from start adding increment for each iteration until var exceeds stop, allowing var to be a non-integer.

TABLE I

Call	Description
ParallelDo[expr, loopspec]	Like Do[ ] except that it evaluates expr across the cluster, rather than on just one processor. The rules for how expr is evaluated is specified in loopspec, like in Do[ ].
ParallelFunctionToList[f, count] ParallelFunctionToList[f, count, root]	Evaluates the function f[i] from 1 to count, but across the cluster, and returns these results in a list. The third argument has it gather this list into the processor whose ID is root.
ParallelTable[expr, loopspec] ParallelTable[expr, loopspec, root]	Like Table[ ] except that it evaluates expr across the cluster, rather than on just one processor, returning the locally evalulated portion. The third argument has it gather this table in to the processor whose ID is root.
ParallelFunction[f, inputs, root]	Like f[inputs] except that it evaluates f on a subset of inputs scattered across the cluster from processor root and gathered back to root.
ParallelNintegrate[expr, loopspec] ParallelNintegrate[expr, loopspec, digits]	Like Nintegrate[ ] except that it evaluates a numerical integration of expr over domains partitioned into the number of processors in the cluster, then returns the sum. The third argument has each

TABLE I-continued

Call	Description
	numerical integration execute with at least that many digits of precision.

Guard-Cell Management

In one embodiment, the advanced functions module **304** includes functions providing for guard-cell operations such as, for example, routines that perform nearest-neighbor communications to maintain edges of local arrays in any number of dimensions (optimized for 1-, 2-, and/or 3-D). Typically the space of a problem is divided into partitions. Often, however, neighboring edges of each partition can interact, so a "guard cell" is inserted on both edges as a substitute for the neighboring data. Thus the space a processor sees is two elements wider than the actual space for which the processor is responsible. EdgeCell helps maintain these guard cells.

TABLE J

Call	Description
EdgeCell[list]	Copies the second element of list to the last element of the left processor and the second-to-last element of list to the first element of the right processor while simultaneously receiving the same from its neighbors.

Matrix and Vector Manipulation

The advanced functions module **304** can also include functions providing for linear algebra operations such as, for example, parallelized versions of basic linear algebra on structures partitioned on many nodes. Such linear algebra operations can reorganize data as needed to perform matrix and vector multiplication or other operations such as determinants, trace, and the like. Matrices are partitioned and stored in processors across the cluster. These calls manipulate these matrices in common ways.

TABLE K

Call	Description
ParallelTranspose[matrix]	Like Transpose[ ] except that it transposes matrix that is in fact represented across the cluster, rather than on just one processor. It returns the portion of the transposed matrix meant for that processor.
ParallelProduct[matrix, vector]	Evaluates the product of matrix and vector, as it would on one processor, except that matrix is represented across the cluster.
ParallelDimensions[matrix]	Like Dimensions[ ] except that matrix is represented across the cluster, rather than on just one processor. It returns a list of each dimension.
ParallelTr[matrix]	Like Tr[ ] except that the matrix is represented across the cluster, rather than on just one processor. It returns the trace of this matrix.
ParallelIdentity[rank]	Like Identity[ ], it generates a new identity matrix, except that the matrix is represented across the cluster, rather than on just one processor. It returns the portion of the new matrix for this processor.
ParallelOuter[f, vector1, vector2]	Like Outer[f, vector1, vector2] except that the answer becomes a matrix represented across the cluster, rather than on just one processor. It returns the portion of the new matrix for this processor.
ParallelInverse[matrix]	Like Inverse[ ] except that the matrix is represented across the cluster, rather than on just one processor. It returns the inverse of the matrix.

Element Management

In one embodiment, the advanced functions module **304** includes element management operations. For example, a large bin of elements or particles cut up in space across the nodes may need to migrate from node to node based on rules or criteria (such as their spatial coordinate). Such operations would migrate the data from one node to another. Besides the divide-and-conquer approach, a list of elements can also be partitioned in arbitrary ways. This is useful if elements need to be organized or sorted onto multiple processors. For example, particles of a system may drift out of the space of one processor into another, so their data would need to be redistributed periodically.

TABLE L

Call	Description
ElementManage[list, switch]	Selects which elements of list will be sent to which processors according to the function switch[ ] is evaluated on each element of list. If switch is a function, switch[ ] should return the ID of the processor that element should be sent. If switch is an integer, the call assumes that each elements is itself a list, whose first element is a number ranging from 0 to the passed argument. This call returns a list of the elements, from any processor, that is switch selected for this processor.
ElementManage[list]	Each element of list can be a list of two elements, the first being the ID of the processor where the element should be sent, while the second is arbitrary data to send. This call returns those list elements, from any and all processors, whose first element is this processors ID in a list. This call is used internally by the two-argument version of ElementManage[].

Fourier Transform

In one embodiment, the advanced functions module **304** includes program code for implementing large-scale parallel fast Fourier transforms ("FFTs"). For example, such functions can perform FFTs in one, two, and/or three dimensions on large amounts of data that are not stored on one node and that are instead stored on many nodes. Fourier transforms of very large arrays can be difficult to manage, not the least of which is the memory requirements. Parallelizing the Fourier transform makes it possible to make use of all the memory available on the entire cluster, making it possible to manipulate problem sizes that no one processor could possibly do alone.

TABLE M

Call	Description
ParallelFourier[list]	Like Fourier[ ] except that list is a two- or three-dimensional list represented across the cluster, like for matrices, above. It returns the portion of the Fourier-transformed array meant for that processor.

Parallel Disk I/O

In one embodiment, the advanced functions module **304** includes parallel disk input and output calls. For example, data may need to be read in and out of the cluster in such a way that the data is distributed across the cluster evenly. The calls in the following table enable the saving data from one or more processors to storage and the retrieval data from storage.

TABLE N

Call	Description
ParallelPut[expr, filename] ParallelPut[expr, filename root] ParallelPut[expr, filename, root, comm]	Puts expr into the file with the name filename in order on processor 0. The third argument specifies that the file be written on the processor whose ID is root. The fourth uses the communicator world comm.
ParallelGet[filename] ParallelGet[filename, root] ParallelGet[filename, root, comm]	Reads and returns data from the file with the name filename on processor 0 partitioned into each processor on the cluster. The second argument specifies that the file is to be read on the processor whose ID is root. The third uses the communicator world comm.
ParallelBinaryPut[expr, type, filename] ParallelBinaryPut[expr, filename, root] ParallelBinaryPut[expr, filename, root, comm]	Puts expr into the file with the binary format type with the name filename in order on processor 0. The fourth argument specifies that the file be written on the processor whose ID is root. The fifth uses the communicator world comm.
ParallelBinaryGet[type, filename] ParallelBinaryGet[type, filename, root] ParallelBinaryGet[type, filename, root, comm]	Reads and returns data in the binary format type from the file with the name filename on processor 0 partitioned into each processor on the cluster. The third argument specifies that the file is to be read on the processor whose ID is root. The fourth uses the communicator world comm.
ParallelGetPerProcessor [expr, filename] ParallelGetPerProcessor [filename, root] ParallelGetPerProcessor [filename, root, comm]	Puts expr into the file with the name filename in order on processor 0, one line per processor. The third argument specifies that the file be written on the processor whose ID is root. The fourth uses the communicator world comm.
ParallelGetPerProcessor [filename] ParallelGetPerProcessor [filename, root]	Reads and returns data from the file with the name filename on processor 0, one line for each processor. The second argument specifies that the file is to be read on the processor whose ID is root. The third uses the communicator world comm.

TABLE N-continued

Call	Description
ParallelGetPerProcessr [filename, root, comm]	

Automatic Load Balancing

Some function calls can take an inconsistent amount of processing time to complete. For example, in Mathematica, the call f[20] could in general take much longer to evaluate than f[19]. Moreover, if one or more processors within the cluster are of different speeds (for example, if some operate at a core frequency of 2.6 GHz while other operate at less than one 1 GHz), one processor may finish a task sooner than another processor.

In some embodiments, the advanced functions module **304** includes a call that can improve the operation of the computer cluster **100** in such situations. In some embodiments, the root processor assigns a small subset of the possible calls for a function to each processor on the cluster **100**. Whichever processor returns its results first is assigned a second small subset of the possible calls. The root processor will continue to assign small subsets of the possible calls as results are received until an evaluation is complete. The order in which the processors finish can vary every time an expression is evaluated, but the root processor will continue assigning additional work to processors as they become available.

In one illustrative example, there are 4 processors and f[1] to f[100] to evaluate. One could implement this by assigning f[1], f[2], f[3], f[4] to each of processors **0** (the root can assign to oneself) through **3**. If the f[2] result came back first, then processor **1** would be assigned f[5]. If the f[4] result is returned next, f[6] would be assigned to processor **3**. The assignments continue until all results are calculated. The results are organized for output back to the user.

In alternative embodiments, the subsets of possible calls can be assigned in any order, rather than sequentially, or in batches (for example, f[1], f[5], f[9] assigned to processor **1**, etc.). Also, the subsets could be organized by delegation. For example, one processor node may not necessarily be in direct control of the other processors. Instead, a large subset could be assigned to a processor, which would in turn assign subsets of its work to other processors. The result would create a hierarchy of assignments like a vast army.

TABLE O

LoadBalanceFunctionToList[f, count] LoadBalanceFunctionToList[f, count, root]	Evaluates the function f[i] from 1 to count, but across the cluster using load-balancing techniques, and returns these results in a list. The third argument has it gather this list into the processor whose ID is root.

3. Received Message Queue **306**

In one embodiment, the cluster node module **204** includes a received message queue **306**. The received message queue **306** includes a data structure for storing messages received from other cluster node modules. Related data pertaining to the messages received, such as whether an expression has been completed, may also be stored in the received message queue **306**. The received message queue **306** may include a queue and/or another type of data structure such as, for example, a stack, a linked list, an array, a tree, etc.

4. Message Receiving Queue **308**

In one embodiment, the cluster node module **204** includes a message receiving queue **308**. The message receiving queue **308** includes a data structure for storing information about the location to which an expression is expected to be sent and the processor from which the expression is expected. The message receiving queue **308** may include a queue and/or another type of data structure such as, for example, a stack, a linked list, an array, a tree, etc.

B. Cluster Configuration Module **208**

Cluster configuration module **208** includes program code for initializing a plurality of cluster node modules to add cluster computing support to computer systems **110**, **120**, **130**. U.S. Pat. No. 7,136,924, issued to Dauger (the "'924 patent"), the entirety of which is hereby incorporated by reference and made a part of this specification, discloses a method and system for parallel operation and control of computer clusters. One method generally includes obtaining one or more personal computers having an operating system with discoverable of network services. In some embodiments, the method includes obtaining one or more processors or processor cores on which a kernel module can run. As described in the '924 patent, a cluster node control and interface (CNCI) group of software applications is copied to each node. When the CNCI applications are running on a node, the cluster configuration module **208** can permit a cluster node module **204**, in combination with a kernel module **206**, to use the node's processing resources to perform a parallel computation task as part of a computer cluster. The cluster configuration module **208** allows extensive automation of the cluster creation process in connection with the present disclosure.

C. User Interface Module **202**

In some embodiments, computer cluster **100** includes a user interface module **202**, such as, for example a Mathematica Front End or a command line interface, that includes program code for a kernel module **206** to provide graphical output, accept graphical input, and provide other methods of user communication that a graphical user interface or a command-line interface provides. To support a user interface module **202**, the behavior of a cluster node module **204***a* is altered in some embodiments. Rather than sending output to and accepting input from the user directly, the user interface module **202** activates the cluster node module **204***a* to which it is connected and specifies parameters to form a connection, such as a MathLink connection, between the cluster node module **204***a* and the user interface module **202**. The user interface module's activation of the cluster node module **204***a* can initiate the execution of instructions to activate the remaining cluster node modules **204***b-e* on the cluster and to complete the sequence to start all kernel modules **206***a-e* on the cluster. Packets from the user interface module **202**, normally intended for a kernel module **206***a*, are accepted by the cluster node module **204***a* as a user command. Output from the kernel module **206***a* associated with the cluster node module **204***a* can be forwarded back to the user interface module **202** for display to a user. Any of the cluster node modules **204***a-e* can be configured to communicate with a user interface module **202**.

D. Kernel Module **206**

A kernel module **206** typically includes program code for interpreting high-level code, commands, and/or instructions supplied by a user or a script into low-level code, such as, for example, machine language or assembly language. In one embodiment, each cluster node module **204***a-e* is connected to all other cluster node modules, while each kernel module **206***a-e* is allocated and connected only to one cluster node module **204**. In one embodiment, there is one cluster node module-kernel module pair per processor. For example, in an embodiment of a computer cluster **100** including single-processor computer systems, each cluster node module-kernel module pair could reside on a single-processor computer. If a computer contains multiple processors or processing cores, it may contain multiple cluster node module-kernel module pairs, but the pairs can still communicate over the cluster node module's network connections.

IV. Cluster Computing Methods

In one embodiment, the computer cluster **100** includes a cluster initialization process, a method of cluster node module operation, and a cluster shut down process.

A. Cluster Initialization Process

In one embodiment, a cluster configuration module **202** initializes one or more cluster node modules **204** in order to provide cluster computing support to one or more kernel modules **206**, as shown in FIG. **4**.

At **402**, cluster node modules are launched on the computer cluster **100**. In one embodiment, the cluster node module **204***a* running on a first processor **112***a* (for example, where the user is located) accesses the other processors **112***b*, **122***a-b*, **132** on the computer cluster **100** via the cluster configuration module **208** to launch cluster node modules **204***b-e* onto the entire cluster. In an alternative embodiment, the cluster configuration module **208** searches for processors **112***a-b*, **122***a-b*, **132** connected to one another via communications network **102** and launches cluster node modules **204***a-e* on each of the processors **112***a-b*, **122***a-b*, **132**.

The cluster node modules **204***a-e* establish communication with one another at **404**. In one embodiment, each of the cluster node modules **204***a-e* establish direct connections using the MPI_Init command with other cluster node modules **204***a-e* launched on the computer cluster **100** by the cluster configuration module **208**.

At **406**, each cluster node module **204** attempts to connect to a kernel module **206**. In one embodiment, each instance of the cluster node modules **204***a-e* locates, launches, and connects with a local kernel module via MathLink connections and/or similar connection tools, for example, built into the kernel module **206**.

At **408**, the cluster node modules **204** that are unconnected to a kernel module **206** are shut down. In one embodiment, each cluster node module **204** determines whether the local kernel module cannot be found or connected to. In one embodiment, each cluster node module **204** reports the failure to connect to a kernel module **206** to the other cluster node modules on computer cluster **100** and quits.

Processor identification numbers are assigned to the remaining cluster node modules **204** at **410**. In one embodiment, each remaining cluster node module **204** calculates the total number of active processors (N) and determines identification numbers describing the remaining subset of active cluster node modules **204***a-e* and kernel modules **206***a-e*. This new set of cluster node module-kernel module pairs may be numbered **0** through N-**1**, for example.

Message passing support is initialized on the kernel modules **206***a-e* at **412**. In one embodiment, each cluster node module **204** supplies initialization code (for example, Mathematica initialization code) to the local kernel module **206** to support message passing.

Finally, at **414**, the cluster node modules **204***a-e* enter a loop to accept user entry. In one embodiment, a main loop (for example, a cluster operation loop) begins execution after the cluster node module **204***a* on the first processor **112***a* returns to user control while each of the other cluster node modules **204** waits for messages from all other cluster node modules **204***a-e* connected to the network **102**.

The initialization process creates a structure enabling a way for the kernel modules **206***a-e* to send messages to one another. In some embodiments, any kernel module can send data to and receive data from any other kernel module within the cluster when initialization is complete. The cluster node module creates an illusion that a kernel module is communicating directly with the other kernel modules. The initialization process can create a relationship among kernel modules on a computer cluster **100** such as the one shown by way of example in FIG. **2**.

B. Cluster Node Module Operation

In one embodiment, a cluster node module **204** implements cluster computing support for a kernel module **206** during a main loop, as shown in FIG. **5**.

At **502**, cluster node modules **204** wait for user commands or messages from other cluster node modules. In one embodiment, the cluster node module **204***a* connected to the user interface module **202** waits for a user command, while the other cluster node modules **204***b-e* continue checking for messages.

Once a command or message is received, the method proceeds to **504**. At **504**, the cluster node module **204***a* determines whether the message received is a quit command. If a quit command is received, the cluster node module **204***a* exits the loop and proceeds to a cluster node module shut down process at **505**. If the message received is not a quit command, the process continues to **506**.

At **506**, received commands are communicated to all cluster node modules **204***a-e* on the computer cluster **100**. In one embodiment, when a user enters a command in the user interface module **202**, the cluster node module **204***a* connected to the user interface module **202** submits the user command to all other cluster node modules **204***b-e* in the computer cluster **100**. The user commands can be simple (for example, "1+1"), but can also be entire subroutines and sequences of code (such as, for example, Mathematica code), including calls to MPI from within the user interface module **202** (for example, the Mathematica Front End) to perform message passing between kernel modules **206***a-e* (for example, Mathematica kernels). These include the fundamental MPI calls, which are implemented using specially identified messages between a cluster node module **204** and its local kernel module **206**.

The message (or user command) is communicated to the kernel modules **206***a-e* at **508**. In one embodiment, the cluster node module **204***a* connected to the user interface module **202** submits the user command to the kernel module **206***a* to which it is connected. Each of the other cluster node modules **204***b-e*, after receiving the message, submits the command to the respective kernel module **206***b-e* to which it is connected.

At **510**, a cluster node module **204** receives a result from a kernel module **206**. In one embodiment, once the kernel module **206** completes its evaluation, it returns the kernel module's output to the cluster node module **204** to which it is connected. Depending on the nature of the result from the kernel module, the cluster node module **204** can report the result to a local computer system or pass the result as a

message to another cluster node module **204**. For example, the cluster node module **204***a* running on the first processor **112***a* reports the output on its local computer system **110**. For example, on the first processor **112***a*, cluster node module **204***a* only directly reports the output of kernel module **206***a*.

Messages from other cluster node modules **204** are responded to at **512**. In one embodiment, each cluster node module (for example, the cluster node module **204***a*) checks for and responds to messages from other cluster node modules **204***b-e* and from the kernel module **206***a* repeatedly until those are exhausted. In one embodiment, output messages from the kernel module **206** are forwarded to output on the local computer system. Messages from other cluster node modules **204** are forwarded to a received message queue **306** ("RMQ"). Data from each entry in the message receiving queue **308** ("MRQ") is matched with entries in the RMQ **306** (see, for example, description of the mpiIRecv[ ] call, above). If found, data from the MRQ **308** are combined into those in the RMQ **306** and marked as "completed" (see, for example, description of the mpiTest[ ] call, above). This process provides the peer-to-peer behavior of the cluster node modules **204***a-e*. Via this mechanism, code running within multiple, simultaneously running kernel modules (for example, Mathematica kernels) can interact on a pair-wise or collective basis, performing calculations, processing, or other work on a scale larger and/or faster than one kernel could have done alone. In this manner, user-entered instructions and data specifying what work will be done via user commands can be executed more quickly and/or reliably. Once responding to messages has completed, the process returns to **502**.

C. Cluster Shut Down Process

In one embodiment, a computer cluster **100** includes a procedure to shut down the system. If the operation process (or main loop) on the cluster node module **204***a* connected to the user interface module **202** detects a "Quit" or "Exit" command or otherwise receives a message from the user indicating a shut down, the sequence to shut down the cluster node modules **204***a-e* and the kernel modules **206***a-e* is activated. In one embodiment, the cluster node module **204***a* connected to the user interface module **202** sends a quit message to all other cluster node modules **204***b-e*. Each cluster node module **204** forwards the quit command to its local kernel module **206**. Once its Mathematica kernel has quit, each cluster node module **204** proceeds to tear down its communication network with other cluster node modules (for example, see description of the MPI_Finalize command, above). At the conclusion of the process, each cluster node module **204** exits execution.

V. Example Operation

For purposes of illustration, sample scenarios are discussed in which the computer cluster system is used in operation. In these sample scenarios, examples of Mathematica code are given, and descriptions of how the code would be executed by a cluster system are provided.

Basic MPI

Fundamental data available to each node includes the node's identification number and total processor count.

```
In[1]:={$IdProc, $NProc}
Out[1]:={0, 2}
```

The first element should be unique for each processor, while the second is generally the same for all. Processor **0** can see what other values are using a collective (see below) communications call such as mpiGather[ ].

```
In[2]:=mpiGather[{$IdProc, $NProc},list,0]; list
Out[2]:={{0, 2}, {1, 2}}
```

Peer-to-Peer MPI

The mpiSend and mpiRecv commands make possible basic message passing, but one needs to define which processor to target. The following defines a new variable, targetProc, so that each pair of processors will point to each other.

```
In[3]:=targetProc=If[1==Mod[$IdProc, 2],$IdProc-1,
  $IdProc+1]
Out[3]:=1
```

In this example, the even processors target its "right" processor, while the odd ones point its "left." For example, if the processors were lined up in a row and numbered in order, every even-numbered processor would pair with the processor following it in the line, and every odd-numbered processor would pair with the processor preceding it. Then a message can be sent:

```
In[4]:=If [1==Mod[ $IdProc,
2],mpiSend[N[Pi,22],targetProc, mpiCommWorld,d],
  mpiRecv[a,targetProc,mpiCommWorld,d]]
```

The If [ ] statement causes the processors to evaluate different code: the odd processor sends 22 digits of Pi, while the even receives that message. Note that these MPI calls return nothing. The received message is in the variable a:

```
In[5]:=a
Out[5]:=3.1415926535897932384626
In[6]:=Clear[a]
```

The variable a on the odd processors would have no definition. Moreover, if $NProc is 8, processor **3** sent Pi to processor **2**, processor **5** sent Pi to processor **4**, and so on. These messages were not sent through processor **0**, but they communicated on their own.

The mpiISend and mpiIRecv commands have a letter "I" to indicate asynchronous behavior, making it possible to do other work while messages are being sent and received, or if the other processor is busy. So, the above example could be done asynchronously:

```
In[7]:=If[1==Mod[$IdProc, 2],mpiISend[N[Pi,22],target-
  Proc, mpiCommWorld,d,e], mpiIRecv[a,targetProc,
  mpiCommWorld,d,e]]
```

The variable e has important data identifying the message, and mpiTest[e] can return True before the expressions are to be accessed. At this point, many other evaluations can be performed. Then, one can check using mpiTest when the data is needed:

```
In[29]:=mpiTest[e]
Out[29]:=True
In[30]:=a
Out[30]:=3.1415926535897932384626
In[31]:=Clear[a,e]
```

The mpiWait[e] command could have also have been used, which does not return until mpiTest[e] returns True. The power of using these peer-to-peer calls is that it becomes possible to construct any message-passing pattern for any problem.

Collective MPI

In some cases, such explicit control is not required and a commonly used communication pattern is sufficient. Suppose processor **0** has an expression in b that all processors are meant to have. A broadcast MPI call would do:

```
In[8]:=mpiBcast[b, 0, mpiCommWorld]
```

The second argument specifies which processor is the "root" of this broadcast; all others have their b overwritten. To collect values from all processors, use mpiGatherD:

```
In[9]:=mpiGather[b, c, 0, mpiCommWorld]
```

The variable c of processor **0** is written with a list of all the b of all the processors in mpiCommWorld. The temporal opposite is mpiScatter:

```
In[10]:=Clear[b]; a={2, 4, 5, 6}; mpiScatter[a, b, 0, mpi-
  CommWorld]; b
Out [10]:={2, 4}
```

The mpiScatter command cuts up the variable a into even pieces (when possible) and scatters them to the processors. This is the result if $NProc=2, but if $NProc=4, b would only have {2}.

MPI provides reduction operations to perform simple computations mixed with messaging. Consider the following:

```
In[11]:=a={{2+$IdProc, 45[ ],3, {1+$IdProc, $NProc[ ][ ];
  mpiReduce [a,d,mpiSum, 0,mpiCommWorld]
In[12]:=d
Out[12]:={{5, 90}, 6, {3, 4}}
```

The mpiSum constant indicates that variable a of every processor will be summed. In this case, $NProc is 2, so those elements that were not identical result in odd sums, while those that were the same are even.

Most of these calls have default values if not all are specified. For example each of the following calls will have the equivalent effect as the above mpiGather[ ] call:

```
mpiGather[b, c, 0]
mpiGather[b, c]
c=mpiGather[b]
```

High-Level Calls

High-level calls can include convenient parallel versions of commonly used application program calls (for example, Mathematica calls). For example, ParallelTable[ ] is like Table[ ], except that the evaluations are automatically performed in a distributed manner:

```
In[13]:=ParallelTable[i,{i,100},0]
Out[13]:={1,2,3,4,5, . . . ,99,100}
```

The third argument specifies that the answers are collated back to processor **0**. This is a useful, simple way to parallelize many calls to a complex function. One could define a complicated function and evaluate it over a large range of inputs:

```
In[14]:=g[x_]:=Gamma[2+0.5*(x−1)]; ParallelTable[g[i],
  {i,100},0]
Out[14]:={1, 1.32934, 2., 3.32335, 6., 11.6317, 24.,
  52.3428, 120., 287.885, 720}
```

ParallelFunctionToList[ ] also provides a simplified way to perform this form of parallelism.

Operations with Non-Trivial Communication

Matrix Operations

In some embodiments, one or more functions can help solve matrix calculations in parallel:

```
In[15]:=a=Table[i+3* $IdProc+2 j, {i, 2}, {j,4}]
Out[15]:={{3, 5, 7, 9}, {4, 6, 8, 10}}
In[16]:=t=ParallelTranspose[a]
Out [16]: {{3, 4, 6, 7}, {5, 6, 8, 9}}
```

Fourier Transforms

A Fourier transform of a large array can be solved faster in parallel, or made possible on a cluster because it can all be held in memory. A two-dimensional Fourier transform of the above example follows:

```
In[17]:=f=ParallelFourier[a]
Out[17]:={{32.+0. I, −4. −4. I, −4., −4. +4. I}, {−3. −3. I, 0.
  +0. I, 0., 0. +0. I}}
```

Edge Cell Management

Many problems require interactions between partitions, but only on the edge elements. Maintaining these edges can be performed using EdgeCell[ ].

```
In[18]:=a={2, 4, 5, 6, 7}+8*$IdProc
Out[18]:={2, 4, 5, 6, 7}
In[19]:=EdgeCell[a]; a
Out[19]:={14, 4, 5, 6, 12}
```

Element Management

In particle-based problems, items can drift through space, sometimes outside the partition of a particular processor. This can be solved with ElementManage[**1**:

```
In[20]:=list={{0,4},{1,3},{1,4},{0,5}}; fcn[x_]:=x[[1]]
In[21]:=ElementManage[list, fcn]
Out[21]:={{0, 4}, {0, 5}, {0, 4}, {0, 5}}
In[21]:=ElementManage[list, 2]
Out[21]:={{0, 4}, {0, 5}, {0, 4}, {0, 5}}
```

The second argument of ElementManage describes how to test elements of a list. The fcn identifier returns which processor is the "home" of that element. Passing an integer assumes that each element is itself a list, whose first element is a number ranging from 0 to the passed argument.

While the examples above involve Mathematica software and specific embodiments of MPI calls and cluster commands, it is recognized that these embodiments are used only to illustrate features of various embodiments of the systems and methods.

VI. Additional Embodiments

Although cluster computing techniques, modules, calls, and functions are disclosed with reference to certain embodiments, the disclosure is not intended to be limited thereby. Rather, a skilled artisan will recognize from the disclosure herein a wide number of alternatives for the exact selection of cluster calls, functions, and management systems. For example, single-node kernels can be managed using a variety of management tools and/or can be managed manually by a user, as described herein. As another example, a cluster node module can contain additional calls and procedures, including calls and procedures unrelated to cluster computing, that are not disclosed herein.

Other embodiments will be apparent to those of ordinary skill in the art from the disclosure herein. Moreover, the described embodiments have been presented by way of example only, and are not intended to limit the scope of the disclosure. Indeed, the novel methods and systems described herein can be embodied in a variety of other forms without departing from the spirit thereof. Accordingly, other combinations, omissions, substitutions and modifications will be apparent to the skilled artisan in view of the disclosure herein. Thus, the present disclosure is not intended to be limited by the disclosed embodiments, but is to be defined by reference to the appended claims. The accompanying claims and their equivalents are intended to cover forms or modifications as would fall within the scope and spirit of the inventions.

What is claimed is:

1. A computer cluster comprising:

a first processor;

a second processor;

a third processor;

at least one computer-readable medium in communication at least one of the first processor, the second processor, or the third processor;

a first kernel residing in the at least one computer-readable medium, said first kernel configured to translate commands into code for execution on the first processor;

a first cluster node module residing in the at least one computer-readable medium, said first cluster node module configured to send commands to the first kernel and receives commands from a user interface;

a second kernel residing in the at least one computer-readable medium, said second kernel configured to translate commands into code for execution on the second processor;

a second cluster node module residing in the at least one computer-readable medium, said second cluster node module configured to send commands to the second kernel and communicates with the first cluster node module;

a third kernel residing in the at least one computer-readable medium, said third kernel configured to translate commands into code for execution on the third processor; and

a third cluster node module residing in the at least one computer-readable medium, said third cluster node module configured to send commands to the third kernel and configured to communicate with the first cluster node module and the second cluster node module;

wherein the first cluster node module comprises a data structure in which messages originating from the second and third cluster node modules are stored.

2. The computer cluster of claim **1**, wherein each of the first kernel, the second kernel, and the third kernel comprises a Mathematica kernel.

3. The computer cluster of claim **1**, wherein each of the first cluster node module, the second cluster node module, and the third cluster node module comprises program code for implementing automatic load balancing on the first processor, the second processor, and the third processor.

4. The computer cluster of claim **1**, further comprising a communications network for connecting at least two of the first processor, the second processor, or the third processor to one another.

5. The computer cluster of claim **1**, wherein the at least one computer-readable medium comprises a first computer-readable medium coupled to at least the first processor and a second computer-readable medium coupled to at least the second processor, wherein the first kernel and the first cluster node module reside in the first computer-readable medium, and wherein the second kernel and the second cluster node module reside in the second computer-readable medium.

6. The computer cluster of claim **1**, further comprising a cluster configuration module that initializes at least one of the first cluster node module, the second cluster node module, or the third cluster node module.

7. The computer cluster of claim **1**, wherein the second cluster node module communicates with the first cluster node module using commands that implement at least a portion of a Message-Passing Interface.

8. The computer cluster of claim **1**, wherein the data structure of the first cluster node module comprises a received message queue.

9. The computer cluster of claim **8**, wherein the first cluster node module further comprises a message receiving queue, in which data specifying the location to which the first cluster node module can expect to receive an expression and an identifier of a processor from which the expression is expected to be sent are stored.

10. The computer cluster of claim **1**, wherein the second cluster node module further comprises a received message queue in which messages originating from the first and third cluster node modules are stored.

11. The computer cluster of claim **1**, wherein the first processor and the second processor reside in the same computer system.

12. The computer cluster of claim **1**, wherein the first processor and the second processor reside on the same die.

13. The computer cluster of claim **1**, wherein the first cluster node module comprises an advanced functions module, and wherein the advanced functions module comprises a call that evaluates an expression across the computer cluster in parallel.

14. The computer cluster of claim **13**, wherein the advanced functions module comprises a call that calculates a Fourier transform across the computer cluster in parallel.

15. The computer cluster of claim **13**, wherein the advanced functions module comprises one or more calls that implement guard-cell operations.

16. The computer cluster of claim **13**, wherein the advanced functions module comprises one or more calls that evaluate matrix operations across the computer cluster in parallel.

17. A computer cluster comprising:

a plurality of nodes, wherein each node is configured to access a computer-readable medium comprising program code for a user interface and program code for a single-node kernel module configured to interpret user instructions;

a plurality of cluster node modules, wherein each cluster node module is configured to communicate with a single-node kernel and with one or more other cluster node modules, to accept instructions from the user interface, and to interpret at least some of the user instructions such that the plurality of cluster node modules communicate with one another in order to act as a cluster; and

a communications network to connect the nodes;

wherein one of the plurality of cluster node modules returns a result to the user interface.

18. The computer cluster of claim **17**, wherein at least one of the plurality of cluster node modules comprises program code for storing data to one or more storage disks.

19. The computer cluster of claim **18**, wherein the at least one of the plurality of cluster node modules further comprises program code for retrieving stored data from one or more storage disks.

20. The computer cluster of claim **17**, wherein at least one of the plurality of cluster node modules comprises program code for collecting data from the plurality of cluster node modules and writing the data to a storage device associated with a single computer system.

21. The computer cluster of claim **17**, wherein at least one of the plurality of cluster node modules comprises program code for reading data from a storage device associated with a single computer system and distributing the data to the plurality of cluster node modules within the computer cluster.

22. The computer cluster of claim **17**, wherein at least one of the plurality of cluster node modules comprises program code for writing temporary data to a storage device associated with the same computer system on which the respective cluster node module is running.

23. The computer cluster of claim **17**, wherein the communications network comprises at least one of a local bus, a local-area network, a wide-area network, an Ethernet network, a Token Ring network, a Frame Relay network, a Fiber Distributed Datalink Interface network, or an Asynchronous Transfer Mode network.

24. The computer cluster of claim **17**, wherein the user interface comprises a Mathematica front end.

25. The computer cluster of claim **17**, wherein the user interface comprises a command line.

26. The computer cluster of claim **17**, wherein each of the plurality of cluster node modules comprises a toolkit comprising library calls that implement at least a portion of a Message-Passing Interface.

27. The computer cluster of claim **17**, wherein each of the plurality of cluster node modules comprises a toolkit comprising one or more high-level cluster computing commands.

28. The computer cluster of claim **17**, wherein the cluster computer system comprises at least one of a plurality of Macintosh® computers, a plurality of Windows®-based computers, or a plurality of computers running a Unix or Linux operating system.

29. A method of evaluating a command on a computer cluster comprising:

communicating a command from at least one of a user interface or a script to one or more cluster node modules within the computer cluster;

for each of the one or more cluster node modules, communicating a message based on the command to a respective kernel module associated with the cluster node module;

for each of the one or more cluster node modules, receiving a result from the respective kernel module associated with the cluster node module; and

for at least one of the one or more cluster node modules, responding to messages from other cluster node modules.

30. The method of claim **29**, wherein communicating a message based on the command to a respective kernel module associated with the cluster node module comprises communicating a specially identified message to the respective kernel module.

31. The method of claim **29**, further comprising, for at least one of the cluster node modules, forwarding the result to at least one of a user interface or a script running on a same computer system as the cluster node module.

32. The method of claim **29**, further comprising, for at least one of the cluster node modules, communicating the result in a message to one or more other cluster node modules.

33. The method of claim **29**, wherein responding to messages from other cluster node modules comprises:

forwarding messages from the other cluster node modules to a received message queue;

matching data from each entry in a message receiving queue with entries in the received message queue;

combining data from the message receiving queue with matching data in the received message queue; and

marking the matching data as completed.

34. The method of claim **29**, wherein communicating a command from at least one of a user interface or a script to one or more cluster node modules within the computer cluster comprises communicating an instruction from a Mathematica Front End to a first cluster node module, wherein the first cluster node module forwards the instruction to other cluster node modules running on the computer cluster.

35. The method of claim **34**, wherein the first cluster node module forwards the instruction to other cluster node modules running on the computer cluster using a command from a Message-Passing Interface.

36. The method of claim **29**, wherein each of the one or more cluster node modules communicates with a respective kernel module using MathLink.

37. A computing system for executing Mathematica code on multiple nodes comprising:

a first node module in communication with a first Mathematica kernel executing on a first node;

a second node module in communication with a second Mathematica kernel executing on a second node; and

a third node module in communication with a third Mathematica kernel executing on a third node;

wherein the first node module, the second node module, and the third node module are configured to communicate with one another using a peer-to-peer architecture.

38. The computing system of claim **37**, wherein each of the first node module, the second node module, and the third node module comprises:

a data structure for maintaining messages originating from other node modules; and

a data structure for maintaining data specifying:

a location to which a message is expected to be received; and

an identifier for a node from which the message is expected to be sent.

* * * * *

EXHIBIT B

US008140612B2

(12) **United States Patent**
Tannenbaum et al.

(10) **Patent No.: US 8,140,612 B2**
(45) **Date of Patent: *Mar. 20, 2012**

(54) **CLUSTER COMPUTING SUPPORT FOR APPLICATION PROGRAMS**

(75) Inventors: **Zvi Tannenbaum**, Newport Beach, CA (US); **Dean E. Dauger**, Huntington Beach, CA (US)

(73) Assignee: **Advanced Cluster Systems, Inc.**, Aliso Viejo, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 1039 days.

This patent is subject to a terminal disclaimer.

(21) Appl. No.: **12/040,519**

(22) Filed: **Feb. 29, 2008**

(65) **Prior Publication Data**

US 2008/0148244 A1 Jun. 19, 2008

Related U.S. Application Data

(63) Continuation-in-part of application No. 11/744,461, filed on May 4, 2007, now Pat. No. 8,082,289.

(60) Provisional application No. 60/813,738, filed on Jun. 13, 2006, provisional application No. 60/850,908, filed on Oct. 11, 2006.

(51) **Int. Cl.**
G06F 9/44 (2006.01)
G06F 15/16 (2006.01)

(52) **U.S. Cl.** **709/201**; 717/177; 719/320

(58) **Field of Classification Search** 709/201; 717/177; 719/320

See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,423,046	A	6/1995	Nunnelley et al.
5,881,315	A	3/1999	Cohen
6,546,403	B1	4/2003	Carlson, Jr. et al.
6,782,537	B1	8/2004	Blackmore et al.
6,968,335	B2	11/2005	Bayliss et al.
7,136,924	B2	11/2006	Dauger
7,249,357	B2	7/2007	Landman et al.
7,334,232	B2	2/2008	Jacobs et al.
2002/0049859	A1	4/2002	Bruckert et al.
2003/0005266	A1	1/2003	Akkary et al.
2003/0051062	A1	3/2003	Circenis
2003/0135621	A1	7/2003	Romagnoli

(Continued)

FOREIGN PATENT DOCUMENTS

JP 2002117010 4/2002

OTHER PUBLICATIONS

International Search Report (Application No. PCT/US2007/070585) mailed Sep. 11, 2008.

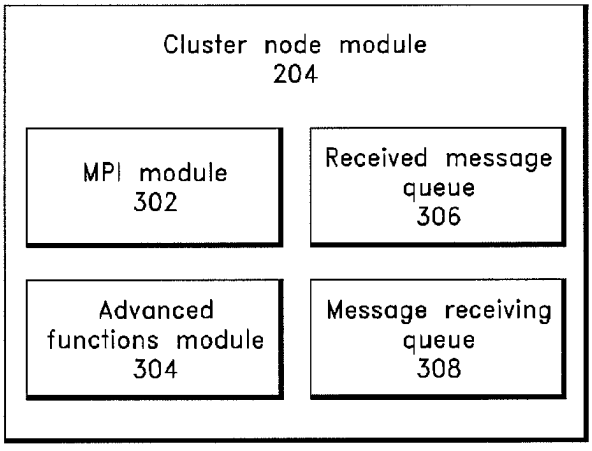
(Continued)

Primary Examiner — Larry Donaghue
(74) *Attorney, Agent, or Firm* — Knobbe, Martens, Olson & Bear, LLP

(57) **ABSTRACT**

A computer cluster system comprising a plurality of nodes and a software package comprising a user interface and a kernel for interpreting program code instructions is provided. In one embodiment, a cluster node module is configured to communicate with the kernel and other cluster node modules. The cluster node module accepts instructions from the user interface and interprets at least some of the instructions such that several cluster node modules in communication with one another and with a kernel can act as a computer cluster.

29 Claims, 5 Drawing Sheets



U.S. PATENT DOCUMENTS

2003/0195931	A1	10/2003	Dauger	
2003/0195938	A1	10/2003	Howard et al.	
2004/0110209	A1	6/2004	Yokota et al.	
2004/0157203	A1	8/2004	Dunk	
2004/0252710	A1	12/2004	Jeter, Jr. et al.	
2005/0021751	A1	1/2005	Block et al.	
2005/0038852	A1	2/2005	Howard	
2005/0060237	A1	3/2005	Barsness et al.	
2005/0076105	A1	4/2005	Keohane et al.	
2005/0108394	A1	5/2005	Braun et al.	
2005/0180095	A1	8/2005	Ellis	
2006/0053216	A1	3/2006	Deokar et al.	
2006/0106931	A1	5/2006	Richoux	
2007/0073705	A1	3/2007	Gray	
2007/0288935	A1 *	12/2007	Tannenbaum et al.	719/319
2008/0148244	A1	6/2008	Tannenbaum et al.	
2008/0250347	A1	10/2008	Gray et al.	
2008/0281997	A1	11/2008	Archer et al.	
2009/0222543	A1	9/2009	Tannenbaum et al.	

OTHER PUBLICATIONS

Mathematica Parallel Computing Toolkit Jan. 2005 pp. 1-90.

Dauger et al. Plug-and Play Cluster Computing: High-Performance Computing For the Mainstream Apr. 2004 pp. 22-28.

Dauger et al. "Plug-and-Play" Cluster Computing using Mac OS X 2003.

Carns et al. An Evaluation of Message Passing Implementations on Beowulf Workstations Mar. 1999 pp. 41-54.

Wolfram, Stephen: The Mathematica Book 5TH Edition; Wolfram Research, Inc. 2003.

Maeder, Roman E.; Mathematica Parallel Computing Toolkit—Unleash the Power of Parallel Computing; Wolfram Research, Jan. 2005.

Tepeneu and Ida, MathGridLink—A bridge between Mathematica and the Grid, (online); Department of Computer Science, Graduate School of Systems and Information Engineering; University of Tsukuba, 2003; Retrieved from the Internet ,URL:http://www.score.is.tsukuba.ac.jp/-ida/ida2004/Recent/bib/MathGridLink.pdf.. (Abstract).

International Search Report (Application No. PCT/US06/37275) mailed Sep. 24, 2007 (2 pages).

"GridMathematica 1.1: Grid Computing Gets a Speed Boost from Mathematica 5", The Mathematica Journal, vol. 9, No. 2, 2004.

"Wolfram gridMathematica", Wolfram Research, Inc., printed on Oct. 3, 2007.

Hamscher et al., "Evaluation of Job-Scheduling Strategies for grid Computing", LNCS: Lecture Notes in Computer Science, 2000, pp. 191-202.

Jahanzeb et al., "Libra: a computational economy-based job scheduling system for clusters", Software Practice and Experience, Feb. 24, 2004, vol. 34, pp. 573-590.

Jain et al., "Data Clustering: A Review", ACM Computing Surveys, Sep. 1999, vol. 31, No. 3.

Notice of Allowance dated Oct. 18, 2011, U.S. Appl. No. 11/744,461.

Office Action dated Apr. 1, 2011, U.S. Appl. No. 11/744,461, filed May 4, 2007.

Response to Office Action dated Jul. 1, 2011, U.S. Appl. No. 11/744,461, filed May 4, 2007.

* cited by examiner

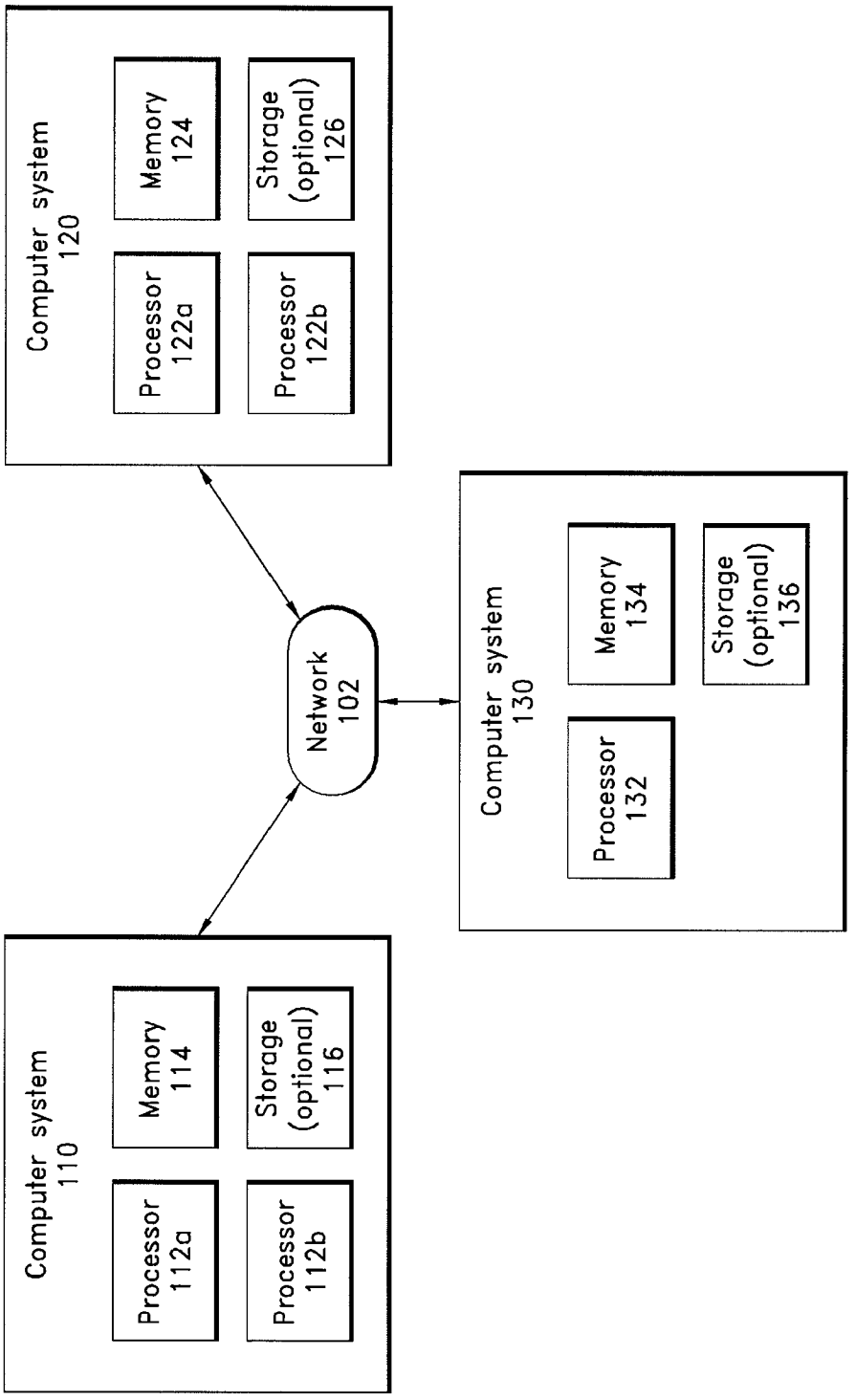

Computer system
120
Memory
124
Storage
(optional)
126
Processor
122a
Processor
122b
Network
102
Computer system
130
Memory
134
Storage
(optional)
136
Processor
132
Computer system
110
Memory
114
Storage
(optional)
116
Processor
112a
Processor
112b

FIG. 1

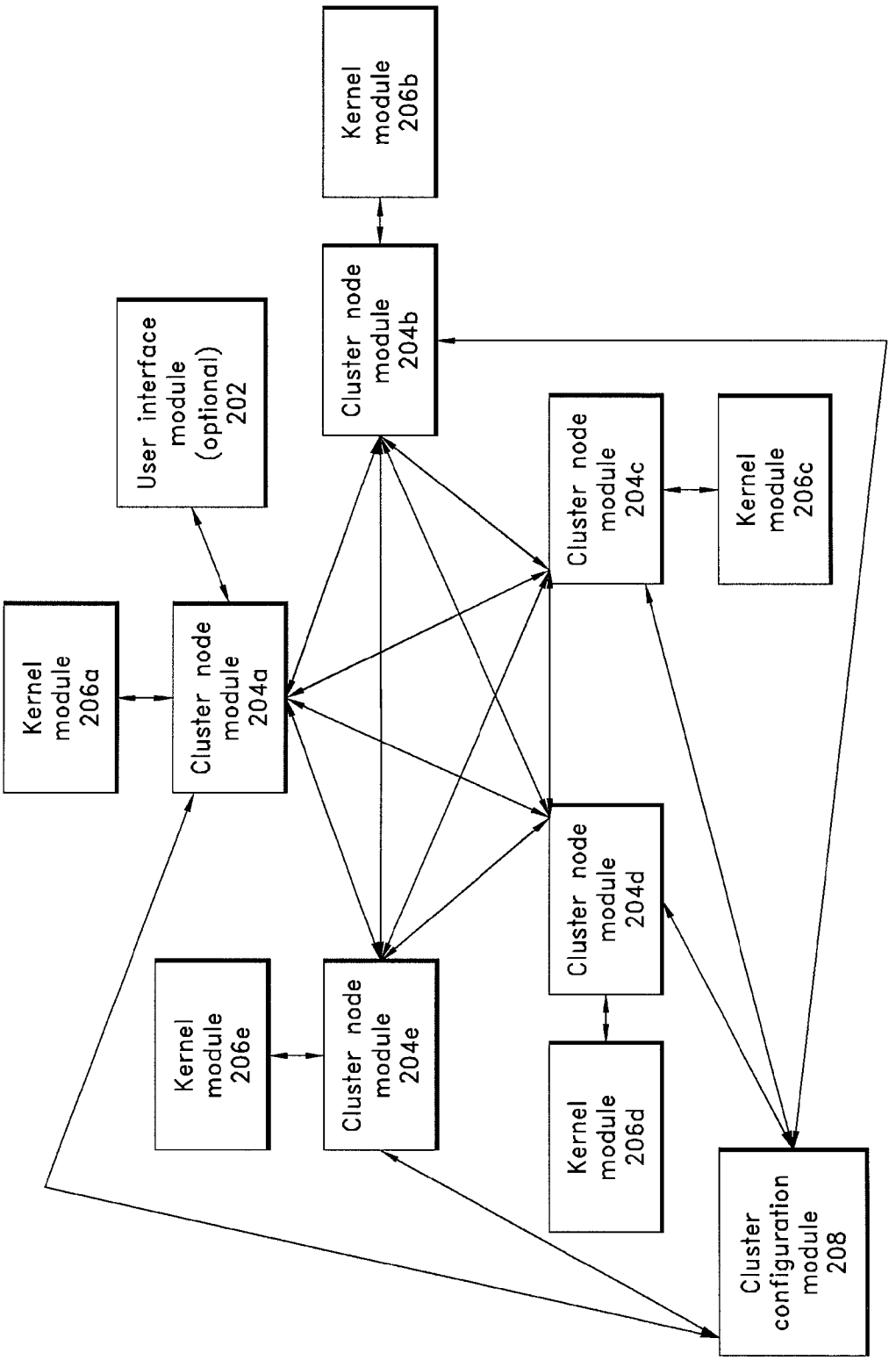

Kernel module 206b
Cluster node module 204b
User interface module (optional) 202
Cluster node module 204c
Kernel module 206c
Kernel module 206a
Cluster node module 204a
Cluster node module 204d
Kernel module 206e
Cluster node module 204e
Kernel module 206d
Cluster configuration module 208

FIG. 2

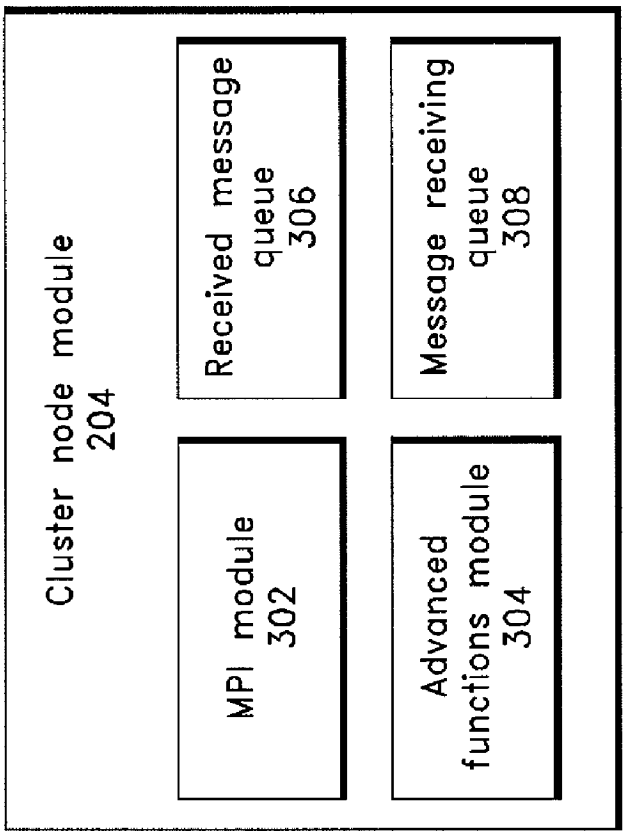

Cluster node module
204
Received message queue
306
Message receiving queue
308
MPI module
302
Advanced functions module
304

FIG. 3

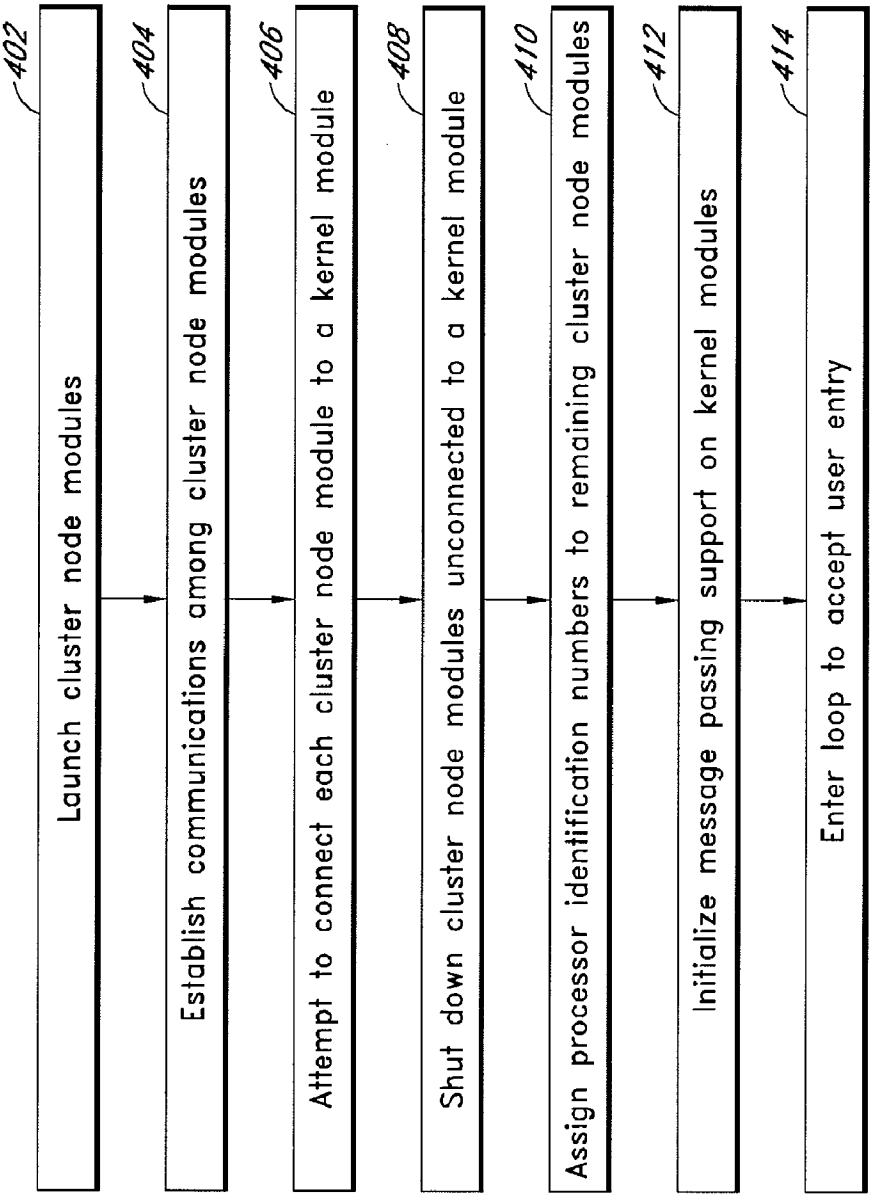

402
Launch cluster node modules
404
Establish communications among cluster node modules
406
Attempt to connect each cluster node module to a kernel module
408
Shut down cluster node modules unconnected to a kernel module
410
Assign processor identification numbers to remaining cluster node modules
412
Initialize message passing support on kernel modules
414
Enter loop to accept user entry

FIG. 4

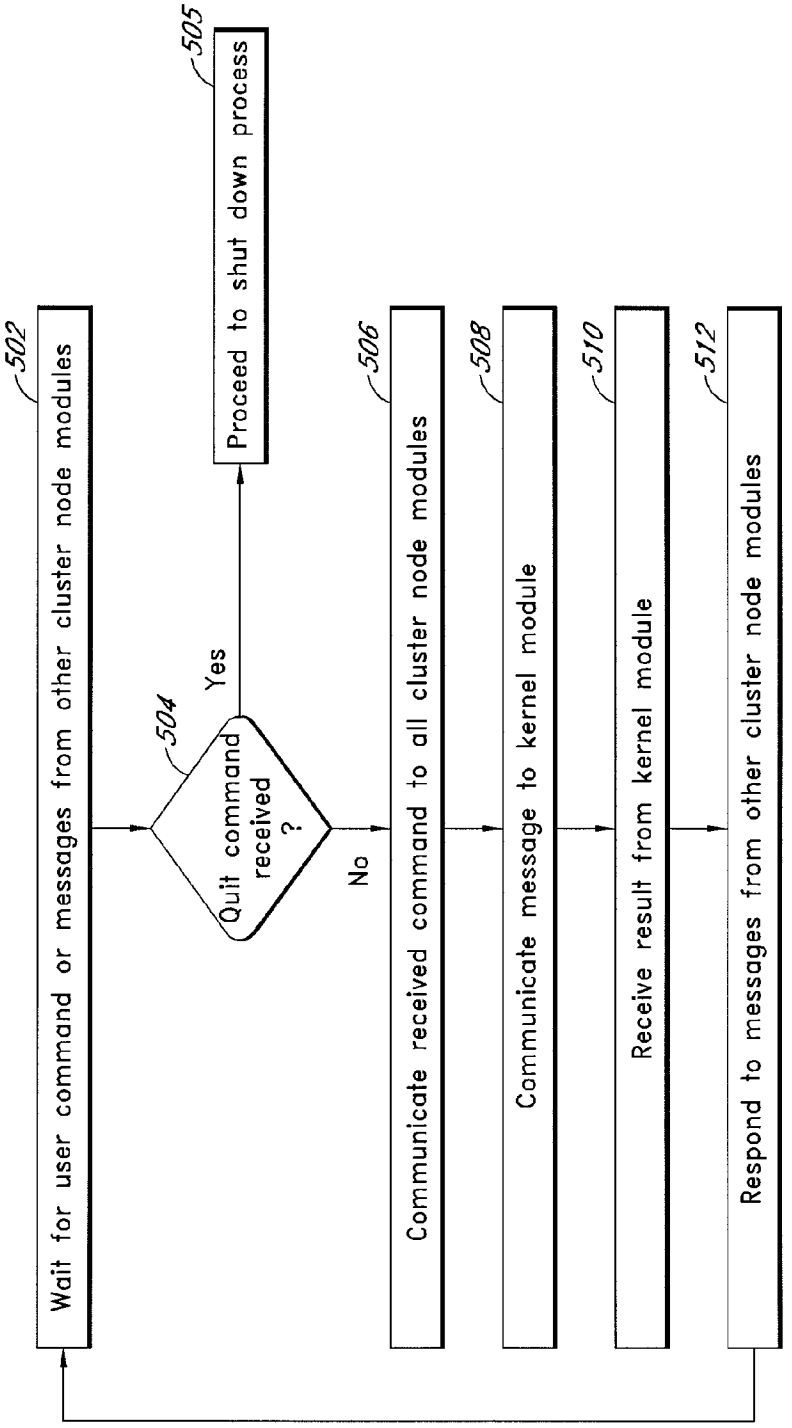

502
Wait for user command or messages from other cluster node modules
504
Quit command received ?
Yes
505
Proceed to shut down process
No
506
Communicate received command to all cluster node modules
508
Communicate message to kernel module
510
Receive result from kernel module
512
Respond to messages from other cluster node modules

FIG. 5

CLUSTER COMPUTING SUPPORT FOR APPLICATION PROGRAMS

PRIORITY INFORMATION

This application is a continuation-in-part of U.S. patent application Ser. No. 11/744,461, filed May 4, 2007 now U.S. Pat. No. 8,082,289, which claims priority to U.S. Provisional Patent Application No. 60/813,738, filed Jun. 13, 2006, and U.S. Provisional Patent Application No. 60/850,908, filed Oct. 11, 2006. The entirety of each of the above-referenced applications is hereby incorporated by reference and made part of this specification.

BACKGROUND

1. Field of the Disclosure

The present disclosure relates to the field of cluster computing generally and to systems and methods for adding cluster computing functionality to a computer program, in particular.

2. Description of the Related Art

Computer clusters include a group of two or more computers, microprocessors, and/or processor cores ("nodes") that intercommunicate so that the nodes can accomplish a task as though they were a single computer. Many computer application programs are not currently designed to benefit from advantages that computer clusters can offer, even though they may be running on a group of nodes that could act as a cluster. Some computer programs can run on only a single node because, for example, they are coded to perform tasks serially or because they are designed to recognize or send instructions to only a single node.

Some application programs include an interpreter that executes instructions provided to the program by a user, a script, or another source. Such an interpreter is sometimes called a "kernel" because, for example, the interpreter can manage at least some hardware resources of a computer system and/or can manage communications between those resources and software (for example, the provided instructions, which can include a high-level programming language). Some software programs include a kernel that is designed to communicate with a single node. An example of a software package that includes a kernel that is designed to communicate with a single node is Mathematica® from Wolfram Research, Inc. ("Mathematica"). Mathematics software packages from other vendors and other types of software can also include such a kernel.

A product known as gridMathematica, also from Wolfram Research, Inc., gives Mathematica the capability to perform a form of grid computing known as "distributed computing." Grid computers include a plurality of nodes that generally do not communicate with one another as peers. Distributed computing can be optimized for workloads that consist of many independent jobs or packets of work, which do not need to share data between the jobs during the computational process. Grid computers include at least one node known as a master node that manages a plurality of slave nodes or computational nodes. In gridMathematica, each of a plurality of kernels runs on a single node. One kernel is designated the master kernel, which handles all input, output, and scheduling of the other kernels (the computational kernels or slave kernels). Computational kernels receive commands and data only from the node running the master kernel. Each computational kernel performs its work independently of the other computational kernels and intermediate results of one job do not affect other jobs in progress on other nodes.

SUMMARY

Embodiments described herein have several features, no single one of which is solely responsible for their desirable attributes. Without limiting the scope of the invention as expressed by the claims, some of the advantageous features will now be discussed briefly.

Some embodiments described herein provide techniques for conveniently adding cluster computing functionality to a computer application. In one embodiment, a user of a software package may be able to achieve higher performance and/or higher availability from the software package by enabling the software to benefit from a plurality of nodes in a cluster. One embodiment allows a user to create applications, using a high-level language such as Mathematica, that are able to run on a computer cluster having supercomputer-like performance. One embodiment provides access to such high-performance computing through a Mathematica Front End, a command line interface, one or more high-level commands, or a programming language such as C or FORTRAN.

One embodiment adapts a software module designed to run on a single node, such as, for example, the Mathematica kernel, to support cluster computing, even when the software module is not designed to provide such support. One embodiment provides parallelization for an application program, even if no access to the program's source code is available. One embodiment adds and supports Message Passing Interface ("MPI") calls directly from within a user interface, such as, for example, the Mathematica programming environment. In one embodiment, MPI calls are added to or made available from an interactive programming environment, such as the Mathematica Front End.

One embodiment provides a computer cluster including a first processor, a second processor, and a third processor. The cluster includes at least one computer-readable medium in communication at least one of the first processor, the second processor, or the third processor. A first kernel resides in the at least one computer-readable medium and is configured to translate commands into code for execution on the first processor. A first cluster node module resides in the at least one computer-readable medium. The first cluster node module is configured to send commands to the first kernel and receives commands from a user interface. A second kernel resides in the at least one computer-readable medium. The second kernel is configured to translate commands into code for execution on the second processor. A second cluster node module resides in the at least one computer-readable medium. The second cluster node module is configured to send commands to the second kernel and communicates with the first cluster node module. A third kernel resides in the at least one computer-readable medium. The third kernel is configured to translate commands into code for execution on the third processor. A third cluster node module resides in the at least one computer-readable medium. The third cluster node module is configured to send commands to the third kernel and configured to communicate with the first cluster node module and the second cluster node module. The first cluster node module comprises a data structure in which messages originating from the second and third cluster node modules are stored.

Another embodiment provides a computer cluster that includes a plurality of nodes and a software package including a user interface and a single-node kernel for interpreting program code instructions. A cluster node module is configured to communicate with the single-node kernel and other

cluster node modules. The cluster node module accepts instructions from the user interface and interprets at least some of the instructions such that several cluster node modules in communication with one another act as a cluster. The cluster node module appears as a single-node kernel to the user interface. In one embodiment, the single-node kernel includes a Mathematical kernel. In some embodiments, the user interface can include at least one of a Mathematica front end or a command line. In some embodiments, the cluster node module includes a toolkit including library calls that implement at least a portion of MPI calls. In some embodiments, the cluster node module includes a toolkit including high-level cluster computing commands. In one embodiment, the cluster system can include a plurality of Macintosh® computers ("Macs"), Windows®-based personal computers ("PCs"), and/or Unix/Linux-based workstations.

A further embodiment provides a computer cluster including a plurality of nodes. Each node is configured to access a computer-readable medium comprising program code for a user interface and program code for a single-node kernel module configured to interpret user instructions. The cluster includes a plurality of cluster node modules. Each cluster node module is configured to communicate with a single-node kernel and with one or more other cluster node modules, to accept instructions from the user interface, and to interpret at least some of the user instructions such that the plurality of cluster node modules communicate with one another in order to act as a cluster. A communications network connects the nodes. One of the plurality of cluster node modules returns a result to the user interface.

Another embodiment provides a method of evaluating a command on a computer cluster. A command from at least one of a user interface or a script is communicated to one or more cluster node modules within the computer cluster. Each of the one or more cluster node modules communicates a message based on the command to a respective kernel module associated with the cluster node module. Each of the one or more cluster node modules receives a result from the respective kernel module associated with the cluster node module. At least one of the one or more cluster node modules responds to messages from other cluster node modules.

Another embodiment provides a computing system for executing Mathematica code on multiple nodes. The computing system includes a first node module in communication with a first Mathematica kernel executing on a first node, a second node module in communication with a second Mathematica kernel executing on a second node, and a third node module in communication with a third Mathematica kernel executing on a third node. The first node module, the second node module, and the third node module are configured to communicate with one another using a peer-to-peer architecture. In some embodiments, each of the first node module, the second node module, and third node module includes a data structure for maintaining messages originating from other node modules and a data structure for maintaining data specifying a location to which an message is expected to be received and an identifier for a node from which the message is expected to be sent.

BRIEF DESCRIPTION OF THE DRAWINGS

A general architecture that implements the various features are described with reference to the drawings. The drawings and the associated descriptions are provided to illustrate embodiments and not to limit the scope of the disclosure. Throughout the drawings, reference numbers are re-used to indicate correspondence between referenced elements.

FIG. **1** is a block diagram of one embodiment of a computer cluster.

FIG. **2** is a block diagram showing relationships between software modules running on one embodiment of a computer cluster.

FIG. **3** is a block diagram of one embodiment of a cluster node module.

FIG. **4** is a flow chart showing one embodiment of a cluster initialization process.

FIG. **5** is a flow chart showing one embodiment of the operation of a cluster node module.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

For purposes of illustration, some embodiments are described herein in the context of cluster computing with Mathematica software. The present disclosure is not limited to a single software program; the systems and methods can be used with other application software such as, for example, Maple®, MATLAB®, MathCAD®, Apple Shake®, Apple® Compressor, IDL®, other applications employing an interpreter or a kernel, Microsoft Excel®, Adobe After Effects®, Adobe Premiere®, Adobe Photoshop®, Apple Final Cut Pro®, and Apple iMovie®. Some figures and/or descriptions, however, relate to embodiments of computer clusters running Mathematica. The system can include a variety of uses, including but not limited to students, educators, scientists, engineers, mathematicians, researchers, and technicians. It is also recognized that in other embodiments, the systems and methods can be implemented as a single module and/or implemented in conjunction with a variety of other modules. Moreover, the specific implementations described herein are set forth in order to illustrate, and not to limit, the disclosure.

I. Overview

The cluster computing system described herein generally includes one or more computer systems connected to one another via a communications network or networks. The communications network can include one or more of a local area network ("LAN"), a wide area network ("WAN"), an intranet, the Internet, etc. In one embodiment, a computer system comprises one or more processors such as, for example, a microprocessor that can include one or more processing cores ("nodes"). The term "node" refers to a processing unit or subunit that is capable of single-threaded execution of code. The processors can be connected to one or more memory devices such as, for example, random access memory ("RAM"), and/or one or more optional storage devices such as, for example, a hard disk. Communications among the processors and such other devices may occur, for example, via one or more local buses of a computer system or via a LAN, a WAN, a storage area network ("SAN"), and/or any other communications network capable of carrying signals among computer system components. In one embodiment, one or more software modules such as kernels, run on nodes within the interconnected computer systems. In one embodiment, the kernels are designed to run on only a single node. In one embodiment, cluster node modules communicate with the kernels and with each other in order to implement cluster computing functionality.

FIG. **1** is a block diagram of one embodiment of a computer cluster **100** wherein computer systems **110**, **120**, **130** communicate with one another via a communications network **102**. Network **102** includes one or more of a LAN, a WAN, a wireless network, an intranet, or the Internet. In one embodi-

ment of the computer cluster, computer system **110** includes processors **112***a*, **112***b*, memory **114**, and optional storage **116**. Other computer systems **120**, **130** can include similar devices, which generally communicate with one another within a computer system over a local communications architecture such as a local bus (not shown). A computer system can include one or more processors, and each processor can contain one or more processor cores that are capable of single-threaded execution. Processor cores are generally independent microprocessors, but more than one can be included in a single chip package. Software code designed for single-threaded execution can generally run on one processor core at a time. For example, single-threaded software code typically does not benefit from multiple processor cores in a computer system.

FIG. **2** is a block diagram showing relationships among software modules running on one embodiment of a computer cluster **100**. In the embodiment shown in FIG. **2**, the kernel modules **206***a-e* are designed for single-threaded execution. For example, if each of the processors **112***a*, **112***b*, **122***a*, **122***b*, **132** shown in FIG. **1** includes only one processor core, two kernel modules (for example, kernel modules **206***a*, **206***b*) loaded into the memory **114** of computer system **110** could exploit at least some of the processing bandwidth of the two processors **112***a*, **112***b*. Similarly, two kernel modules **206***c*, **206***d* loaded into the memory **124** of computer system **120** could exploit at least some of the processing bandwidth of the two processors **122***a*, **122***b*. Likewise, the bandwidth of processor **132** of computer system **130** could be utilized by a single instance of a cluster node module **204***e* loaded into the computer system's memory **134**.

In the embodiment shown in FIG. **2**, each of the kernel modules **206***a-e* is in communication with a single cluster node module **204***a-e*, respectively. For example, the kernel module **206***a* is in communication with the cluster node module **204***a*, the kernel module **206***b* is in communication with the cluster node module **206***b*, and so forth. In one embodiment, one instance of a cluster node module **204***a-e* is loaded into a computer system's memory **114**, **124**, **134** for every instance of a kernel module **206***a-e* running on the system. As shown in FIG. **2**, each of the cluster node modules **204***a-e* is in communication with each of the other cluster node modules **204***a-e*. For example, one cluster node module **204***a* is in communication with all of the other cluster node modules **204***b-e*. A cluster node module **204***a* may communicate with another cluster node module **204***b* via a local bus (not shown) when, for example, both cluster node modules **204***a-b* execute on processors **112***a*, **112***b* within the same computer system **110**. A cluster node module **204***a* may also communicate with another cluster node module **204***c* over a communications network **102** when, for example, the cluster node modules **204***a, c* execute on processors **112***a*, **122***a* within different computer systems **110**, **120**.

As shown in FIG. **2**, an optional user interface module **202** such as, for example, a Mathematica front end and/or a command line interface, can connect to a cluster node module **204***a*. The user interface module can run on the same computer system **110** and/or the same microprocessor **112***a* on which the cluster node module **204***a* runs. The cluster node modules **204***a-e* provide MPI calls and/or advanced cluster functions that implement cluster computing capability for the single-threaded kernel modules. The cluster node modules **204***a-e* are configured to look and behave like a kernel module **206***a* from the perspective of the user interface module **202**. Similarly, the cluster node modules **204***a-e* are configured to look and behave like a user interface module **202** from the perspective of a kernel module **206***a*. The first cluster node module **204***a* is in communication with one or more other cluster node modules **204***b*, **204***c*, and so forth, each of which provides a set of MPI calls and/or advanced cluster commands. In one embodiment, MPI may be used to send messages between nodes in a computer cluster.

Communications can occur between any two or more cluster node modules (for example, between a cluster node module **204***a* and another cluster node module **204***c*) and not just between "adjacent" kernels. Each of the cluster node modules **204***a-e* is in communication with respective kernel modules **206***a-e*. Thus, the cluster node module **204***a* communicates with the kernel module **206***a*. MPI calls and advanced cluster commands are used to parallelize program code received from an optional user interface module **208** and distribute tasks among the kernel modules **206***a-e*. The cluster node modules **204***a-e* provide communications among kernel modules **206***a-e* while the tasks are executing. Results of evaluations performed by kernel modules **206***a-e* are communicated back to the first cluster node module **204***a* via the cluster node modules **204***a-e*, which communicates them to the user interface module **208**.

Intercommunication among kernel modules **206***a-e* during thread execution, which is made possible by cluster node modules **204***a-e*, provides advantages for addressing various types of mathematic and scientific problems, for example. Intercommunication provided by cluster computing permits exchange of information between nodes during the course of a parallel computation. Embodiments of the present disclosure provide such intercommunication for software programs such as Mathematica, while grid computing solutions can implement communication between only one master node and many slave nodes. Grid computing does not provide for communication between slave nodes during thread execution.

For purposes of providing an overview of some embodiments, certain aspects, advantages, benefits, and novel features of the invention are described herein. It is to be understood that not necessarily all such advantages or benefits can be achieved in accordance with any particular embodiment of the invention. Thus, for example, those skilled in the art will recognize that the invention can be embodied or carried out in a manner that achieves one advantage or group of advantages as taught herein without necessarily achieving other advantages or benefits as can be taught or suggested herein.

II. Computer Cluster **100**

As shown in FIG. **1**, one embodiment of a cluster system **100** includes computer systems **110**, **120**, **130** in communication with one another via a communications network **102**. A first computer system **110** can include one or more processors **112***a-b*, a memory device **114**, and an optional storage device **116**. Similarly, a second computer system **120** can include one or more processors **122***a-b*, a memory device **124**, and an optional storage device **126**. Likewise, a third computer system **130** can include one or more processors **132**, a memory device **134**, and an optional storage device **136**. Each of the computer systems **110**, **120**, **130** includes a network interface (not shown) for connecting to a communications network **102**, which can include one or more of a LAN, a WAN, an intranet, a wireless network, and/or the Internet.

A. Computer System **110**

In one embodiment, a first computer system **110** communicates with other computer systems **120**, **130** via a network **102** as part of a computer cluster **100**. In one embodiment, the computer system **110** is a personal computer, a workstation, a server, or a blade including one or more processors **112***a-b*, a

memory device **114**, an optional storage device **116**, as well as a network interface module (not shown) for communications with the network **102**.

1. Processors **112***a-b*

In one embodiment, the computer system **110** includes one or more processors **112***a-b*. The processors **112***a-b* can be one or more general purpose single-core or multi-core microprocessors such as, for example, a Pentium® processor, a Pentium® II processor, a Pentium® Pro processor, a Pentium® III processor, Pentium® 4 processor, a Core Duo® processor, a Core 2 Duo® processor, a Xeon® processor, an Itanium® processor, a Pentium®M processor, an x86 processor, an Athlon® processor, an 8051 processor, a MIPS® processor, a PowerPC® processor, an ALPHA® processor, etc. In addition, one or more of the processors **112***a-b* can be a special purpose microprocessor such as a digital signal processor. The total number of processing cores (for example, processing units capable of single-threaded execution) within all processors **112***a-b* in the computer system **110** corresponds to the number of nodes available in the computer system **110**. For example, if the processors **112***a-b* were each Core 2 Duo® processors having two processing cores, computer system **110** would have four nodes in all. Each node can run one or more instances of a program module, such as a single-threaded kernel module.

2. Network Interface Module

The computer system **110** can also include a network interface module (not shown) that facilitates communication between the computer system **110** and other computer systems **120**, **130** via the communications network **102**.

The network interface module can use a variety of network protocols. In one embodiment, the network interface module includes TCP/IP. However, it is to be appreciated that other types of network communication protocols such as, for example, Point-to-Point Protocol ("PPP"), Server Message Block ("SMB"), Serial Line Internet Protocol ("SLIP"), tunneling PPP, AppleTalk, etc., may also be used.

3. Memory **114** and Storage **116**

The computer system **110** can include memory **114**. Memory **114** can include, for example, processor cache memory (such as processor core-specific or cache memory shared by multiple processor cores), dynamic random-access memory ("DRAM"), static random-access memory ("SRAM"), or any other type of memory device capable of storing computer data, instructions, or program code. The computer system **110** can also include optional storage **116**. Storage **116** can include, for example, one or more hard disk drives, floppy disks, flash memory, magnetic storage media, CD-ROMs, DVDs, optical storage media, or any other type of storage device capable of storing computer data, instructions, and program code.

4. Computer System **110** Information

The computer system **110** may be used in connection with various operating systems such as: Microsoft® Windows® 3.X, Windows 95®, Windows 98®, Windows NT®, Windows 2000®, Windows XP®, Windows CE®, Palm Pilot OS, OS/2, Apple® MacOS®, MacOS X®, MacOS X Server®, Disk Operating System (DOS), UNIX, Linux®, VxWorks, or IBM® OS/2®, Sun OS, Solaris OS, IRIX OS operating systems, etc.

In one embodiment, the computer system **110** is a personal computer, a laptop computer, a Blackberry® device, a portable computing device, a server, a computer workstation, a local area network of individual computers, an interactive kiosk, a personal digital assistant, an interactive wireless communications device, a handheld computer, an embedded computing device, or the like.

As can be appreciated by one of ordinary skill in the art, the computer system **110** may include various sub-routines, procedures, definitional statements, and macros. Each of the foregoing modules are typically separately compiled and linked into a single executable program. However, it is to be appreciated by one of ordinary skill in the art that the processes that are performed by selected ones of the modules may be arbitrarily redistributed to one of the other modules, combined together in a single module, made available in a shareable dynamic link library, or partitioned in any other logical way.

B. Computer System **120**

In one embodiment, a second computer system **120** communicates with other computer systems **110**, **130** via a network **102** as part of a computer cluster **100**. In one embodiment, the computer system **120** is a personal computer, a workstation, a server, or a blade including one or more processors **122***a-b*, a memory device **124**, an optional storage device **126**, as well as a network interface module (not shown) for communications with the network **102**.

1. Processors **112***a-b*

In one embodiment, the computer system **120** includes one or more processors **122***a-b*. The processors **122***a-b* can be one or more general purpose single-core or multi-core microprocessors such as a Pentium® processor, a Pentium® II processor, a Pentium® Pro processor, a Pentium® III processor, Pentium® 4 processor, a Core Duo® processor, a Core 2 Duo® processor, a Xeon® processor, an Itanium® processor, a Pentium® M processor, an x86 processor, an Athlon® processor, an 8051 processor, a MIPS® processor, a PowerPC® processor, an ALPHA® processor, etc. In addition, the processors **122***a-b* can be any special purpose microprocessors such as a digital signal processor. The total number of processing cores (for example, processing units capable of single-threaded execution) within all processors **122***a-b* in the computer system **120** corresponds to the number of nodes available in the computer system **120**. For example, if the processors **122***a-b* were each Core 2 Duo® processors having two processing cores, computer system **120** would have four nodes in all. Each node can run one or more instances of a program module, such as a single-threaded kernel module.

2. Network Interface Module

The computer system **120** can also include a network interface module (not shown) that facilitates communication between the computer system **120** and other computer systems **110**, **130** via the communications network **102**.

The network interface module can use a variety of network protocols. In one embodiment, the network interface module includes TCP/IP. However, it is to be appreciated that other types of network communication protocols such as, for example, Point-to-Point Protocol ("PPP"), Server Message Block ("SMB"), Serial Line Internet Protocol ("SLIP"), tunneling PPP, AppleTalk, etc., may also be used.

3. Memory **124** and Storage **126**

The computer system **120** can include memory **124**. Memory **124** can include, for example, processor cache memory (such as processor core-specific or cache memory shared by multiple processor cores), dynamic random-access memory ("DRAM"), static random-access memory ("SRAM"), or any other type of memory device capable of storing computer data, instructions, or program code. The computer system **120** can also include optional storage **126**. Storage **126** can include, for example, one or more hard disk drives, floppy disks, flash memory, magnetic storage media, CD-ROMs, DVDs, optical storage media, or any other type of storage device capable of storing computer data, instructions, and program code.

4. Computer System **120** Information

The computer system **120** may be used in connection with various operating systems such as: Microsoft® Windows® 3.X, Windows 95®, Windows 98®, Windows NT®, Windows 2000®, Windows XP®, Windows CE®, Palm Pilot OS, OS/2, Apple® MacOS®, MacOS X®, MacOS X Server®, Disk Operating System (DOS), UNIX, Linux®, VxWorks, or IBM® OS/2®, Sun OS, Solaris OS, IRIX OS operating systems, etc.

In one embodiment, the computer system **120** is a personal computer, a laptop computer, a Blackberry® device, a portable computing device, a server, a computer workstation, a local area network of individual computers, an interactive kiosk, a personal digital assistant, an interactive wireless communications device, a handheld computer, an embedded computing device, or the like.

As can be appreciated by one of ordinary skill in the art, the computer system **120** may include various sub-routines, procedures, definitional statements, and macros. Each of the foregoing modules are typically separately compiled and linked into a single executable program. However, it is to be appreciated by one of ordinary skill in the art that the processes that are performed by selected ones of the modules may be arbitrarily redistributed to one of the other modules, combined together in a single module, made available in a shareable dynamic link library, or partitioned in any other logical way.

C. Computer System **130**

In one embodiment, a third computer system **130** communicates with other computer systems **110**, **120** via a network **102** as part of a computer cluster **100**. In one embodiment, the computer system **130** is a personal computer, a workstation, a server, or a blade including one or more processors **132**, a memory device **134**, an optional storage device **136**, as well as a network interface module (not shown) for communications with the network **102**.

1. Processors **112***a-b*

In one embodiment, the computer system **130** includes a processor **132**. The processor **132** can be a general purpose single-core or multi-core microprocessors such as a Pentium® processor, a Pentium® II processor, a Pentium® Pro processor, a Pentium® III processor, Pentium® 4 processor, a Core Duo® processor, a Core 2 Duo® processor, a Xeon® processor, an Itanium® processor, a Pentium® M processor, an x86 processor, an Athlon® processor, an 8051 processor, a MIPS® processor, a PowerPC® processor, or an ALPHA® processor. In addition, the processor **132** can be any special purpose microprocessor such as a digital signal processor. The total number of processing cores (for example, processing units capable of single-threaded execution) within processor **132** in the computer system **130** corresponds to the number of nodes available in the computer system **130**. For example, if the processor **132** was a Core 2 Duo® processor having two processing cores, the computer system **130** would have two nodes. Each node can run one or more instances of a program module, such as a single-threaded kernel module.

2. Network Interface Module

The computer system **130** can also include a network interface module (not shown) that facilitates communication between the computer system **130** and other computer systems **110**, **120** via the communications network **102**.

The network interface module can use a variety of network protocols. In one embodiment, the network interface module includes TCP/IP. However, it is to be appreciated that other types of network communication protocols such as, for example, Point-to-Point Protocol ("PPP"), Server Message Block ("SMB"), Serial Line Internet Protocol ("SLIP"), tunneling PPP, AppleTalk, etc., may also be used.

3. Memory **134** and Storage **136**

The computer system **130** can include memory **134**. Memory **134** can include, for example, processor cache memory (such as processor core-specific or cache memory shared by multiple processor cores), dynamic random-access memory ("DRAM"), static random-access memory ("SRAM"), or any other type of memory device capable of storing computer data, instructions, or program code. The computer system **130** can also include optional storage **136**. Storage **136** can include, for example, one or more hard disk drives, floppy disks, flash memory, magnetic storage media, CD-ROMs, DVDs, optical storage media, or any other type of storage device capable of storing computer data, instructions, and program code.

4. Computer System **130** Information

The computer system **130** may be used in connection with various operating systems such as: Microsoft® Windows® 3.X, Windows 95®, Windows 98®, Windows NT®, Windows 2000®, Windows XP®, Windows CE®, Palm Pilot OS, OS/2, Apple® MacOS®, MacOS X®, MacOS X Server®, Disk Operating System (DOS), UNIX, Linux®, VxWorks, or IBM® OS/2®, Sun OS, Solaris OS, IRIX OS operating systems, etc.

In one embodiment, the computer system **130** is a personal computer, a laptop computer, a Blackberry® device, a portable computing device, a server, a computer workstation, a local area network of individual computers, an interactive kiosk, a personal digital assistant, an interactive wireless communications device, a handheld computer, an embedded computing device, or the like.

As can be appreciated by one of ordinary skill in the art, the computer system **130** may include various sub-routines, procedures, definitional statements, and macros. Each of the foregoing modules are typically separately compiled and linked into a single executable program. However, it is to be appreciated by one of ordinary skill in the art that the processes that are performed by selected ones of the modules may be arbitrarily redistributed to one of the other modules, combined together in a single module, made available in a shareable dynamic link library, or partitioned in any other logical way.

E. Communications Network **102**

In one embodiment, computer systems **110**, **120**, **130** are in communication with one another via a communications network **102**.

The communications network **102** may include one or more of any type of electronically connected group of computers including, for instance, the following networks: a virtual private network, a public Internet, a private Internet, a secure Internet, a private network, a public network, a value-added network, a wired network, a wireless network, an intranet, etc. In addition, the connectivity to the network can be, for example, a modem, Ethernet (IEEE 802.3), Gigabit Ethernet, 10-Gigabit Ethernet, Token Ring (IEEE 802.5), Fiber Distributed Datalink Interface (FDDI), Frame Relay, InfiniBand, Myrinet, Asynchronous Transfer Mode (ATM), or another interface. The communications network **102** may connect to the computer systems **110**, **120**, **130**, for example, by use of a modem or by use of a network interface card that resides in each of the systems.

In addition, the same or different communications networks **102** may be used to facilitate communication between the first computer system **110** and the second computer system **120**, between the first computer system **110** and the third

computer system **130**, and between the second computer system **120** and the third computer system **130**.

III. Software Modules

As shown in FIGS. **1** and **2**, one embodiment of a cluster system **100** includes a user interface module **202** that is able to access a plurality of kernel modules **206***a-e* by communicating with a first cluster node module **204***a*. User interface module can be stored in a memory **114**, **124**, **134** while running, for example, and/or can be stored in a storage device **116**, **126**, **136**. The first cluster node module **204***a* is in communication with each of the other cluster node modules **204***b-e*. The kernel modules **206***a-e* can reside in the memory of one or more computer systems on which they run. For example, the memory **114** of the first computer system **110** can store instances of kernel modules **206***a-b*, the memory **124** of the second computer system **120** can store instances of kernel modules **206***c-d*, and the memory **134** of the third computer system **130** can store an instance of kernel module **206***e*. The kernel modules **206***a-e*, which include single-threaded program code, are each associated with one of the processors **112***a*, **112***b*, **122***a*, **122***b*, **132**. A cluster configuration module stored on one or more of the computer systems **110**, **120**, **130** or on a remote computer system, for example, can establish communication with the cluster node modules **204***a-e*. In one embodiment, communication between the cluster configuration module **208** and the cluster node modules **204***a-e* initializes the cluster node modules **204***a-e* to provide cluster computing support for the computer cluster **100**.

A. Cluster Node Module **204**

In one embodiment, the cluster node modules **204***a-e* provide a way for many kernel modules **206***a-e* such as, for example, Mathematica kernels, running on a computer cluster **100** to communicate with one another. A cluster node module **204** can include at least a portion of an application programming interface ("API") known as the Message-Passing Interface ("MPI"), which is used in several supercomputer and cluster installations. A network of connections (for example, the arrows shown in FIG. **2**) between the cluster node modules **204***a-e* can be implemented using a communications network **102**, such as, for example, TCP/IP over Ethernet, but the connections could also occur over any other type of network or local computer bus.

A cluster node module **204** can use an application-specific toolkit or interface such as, for example, Mathematica's MathLink, Add-Ons, or packets, to interact with an application. Normally used to connect a Mathematica kernel to a user interface known as the Mathematica Front End or other Mathematica kernels, MathLink is a bidirectional protocol to sends "packets" containing messages, commands, or data between any of these entities. MathLink does not allow direct cluster computing-like simultaneous communication between Mathematica kernels during execution of a command or thread. MathLink is also not designed to perform multiple simultaneous network connections. In some embodiments, a cluster node module **204** can use an application-specific toolkit such as, for example, MathLink, for connections between entities on the same computer.

When speaking about procedures or actions on a cluster or other parallel computer, not all actions happen in sequential order, nor are they required to. For example, a parallel code, as opposed to a single-processor code of the classic "Turing machine" model, has multiple copies of the parallel code running across the cluster, typically one for each processor (or "processing element" or "core"). Such parallel code is written in such a way that different instances of the same code can communicate, collaborate, and coordinate work with each other. Multiple instances of these codes can run at the same time in parallel.

If the count of the code instances is an integer N, each instance of code execution can be labeled 0 through N−1. For example, a computer cluster can include N connected computers, each containing a processor. The first has cluster node module 0 connected with kernel module 0 running on processor 0. The next is cluster node module 1 and kernel module 1, on processor 1, and so forth for each of the N connected computers. Some steps of their procedure are collaborative, and some steps are independent. Even though these entities are not necessarily in lock-step, they do follow a pattern of initialization, main loop behavior (for example, cluster node module operation), and shut down.

In contrast, a parallel computing toolkit (PCT) that is provided as part of the gridMathematica software package does not provide a means for instances of the same code running on different nodes to communicate, collaborate, or coordinate work among the instances. The PCT provides commands that connect Mathematica kernels in a master-slave relationship rather than a peer-to-peer relationship as enabled by some embodiments disclosed herein. A computer cluster having peer-to-peer node architecture performs computations that can be more efficient, easier to design, and/or more reliable than similar computations performed on grid computers having master-slave node architecture. Moreover, the nature of some computations may not allow a programmer to harness multi-node processing power on systems that employ master-slave node architecture.

FIG. **3** shows one embodiment of a cluster node module **204** implementing MPI calls and advanced MPI functions. In the embodiment shown in FIG. **3**, cluster node module **204** includes MPI module **302**, advanced functions module **304**, received message queue **306**, and message receiving queue **308**.

1. MPI Module **302**

In one embodiment, the cluster node module **204** includes an MPI module **302**. The MPI module **302** can include program code for one or more of at least five kinds of MPI instructions or calls. Selected constants, instructions, and/or calls that can be implemented by the MPI module **302** are as follows:

MPI Constants

Node identifiers are used to send messages to nodes or receive messages from them. In MPI, this is accomplished by assigning each node a unique integer ($IdProc) starting with 0. This data, with a knowledge of the total count ($NProc), makes it possible to programmatically divide any measurable entity.

TABLE A

Constant	Description
$IdProc	The identification number of the current processor
$NProc	The number of processors in the current cluster
$mpiCommWorld	The communicator world of the entire cluster (see MPI Communicator routines, below)
mpiCommWorld	The default communicator world for the high-level routines.

Basic MPI Calls

In one embodiment, the MPI module **302** can include basic MPI calls such as, for example, relatively low-level routines that map MPI calls that are commonly used in other languages (such as C and Fortran), so that such calls can be available

directly from the Mathematica user interface **204**. In some embodiments, basic MPI calls include calls that send data, equations, formulas, and/or other expressions.

Simply sending expressions from one node to another is possible with these most basic MPI calls. One node can call to send an expression while the other calls a corresponding routine to receive the sent expression. Because it is possible that the receiver has not yet called mpiRecv even if the message has left the sending node, completion of mpiSend is not a confirmation that it has been received.

TABLE B

Call	Description
mpiSend[expr, target, comm, tag]	Sends an expression expr to a node with the ID target in the communicator would comm, waiting until that expression has left this kernel
mpiRecv[expr, target, comm, tag]	Receives an expression into expr from a node with the ID target in the communicator world comm, waiting until the expression has arrived
mpiSendRecv [sendexpr, dest, recvexpr, source, comm]	Simultaneously sends the expression sendexpr to the node with the ID target and receives an expression into recvexpr from the node with the ID source in the communicator world comm, waiting until both operations have returned.

Asynchronous MPI Calls

Asynchronous calls make it possible for the kernel to do work while communications are proceeding simultaneously. It is also possible that another node may not be able to send or receive data yet, allowing one kernel to continue working while waiting.

TABLE C

Call	Description
mpiISend[expr, target, comm, tag, req]	Sends an expression expr to a processor with the ID target in the communicator world comm, returning immediately. It can be balanced with calls to mpiTest[req] until mpiTest[req] returns True.
mpiIRecv[expr, target, comm, tag, req]	Receives an expression expr from a proessor with the ID target in the communicator world comm, returning immediately. It can be balanced with calls to mpiTest[req] until mpiTest[req] returns True. The expr is not safe to access until mpiTest[req] returns True.
mpiTest[req]	Completes asynchronous behavior of mpiISend and mpiIRecv
mpWait[req]	Calls mpiTest until it returns True.
mpiWaitall[reglist]	Calls mpiWait all on every element of reqlist
mpiWaitany[reqlist]	Calls mpiTest on each element of realist until one of them returns True

The mpilSend[ ] command can be called from within a kernel module **206** (for example, a Mathematica kernel). It creates a packet containing the Mathematica expression to be sent as payload and where the expression should be sent. The packet itself is destined only for its local cluster node module. Once received by its local cluster node module, this packet is decoded and its payload is forwarded on to the cluster node module specified in the packet.

The mpiIRecv[ ] command can also be called from within a kernel module **206**. It creates a packet specifying where it expects to receive an expression and from which processor this expression is expected. Once received by its local cluster node module, this packet is decoded and its contents are stored in a message receiving queue (MRQ) **308** (FIG. **3**).

The mpiTest[ ] command can be called from within a kernel module **206**. It creates a packet specifying which message to test for completion, then waits for a reply expression to evaluate. Once received by the kernel module's associated cluster node module **204**, this packet is decoded and its message specifier is used to search for any matching expressions listed as completed in its received message queue (RMQ) **306**. If such completed expressions are found, it is sent to its local kernel module as part of the reply in mpiTest[ ]. The kernel module receives this reply expression and evaluates it, which updates the kernel module's variables as needed.

Other MPI calls are built on the fundamental calls mpiISend, mpiIRecv, and mpiTest. For example, mpiBcast, a broadcast, creates instructions to send information from the broadcast processor to all the others, while the other processors perform a Recv. Similarly, high-level calls of the toolkit can be built on top of the collection of MPI calls.

Collective MPI Calls

In one embodiment, the MPI module **302** can include program code for implementing collective MPI calls (for example, calls that provide basic multi-node data movement across nodes). Collective MPI calls can include broadcasts, gathers, transpose, and other vector and matrix operations, for example. Collective calls can also provide commonly used mechanisms to send expressions between groups of nodes.

TABLE D

Call	Description
mpiBcast[expr, root, comm]	Performs a broadcast of expr from the root processor to all the others in the communicator world comm. An expression is expected to be supplied by the root processor, while all the others expect expr to be overwritten by the incoming expression.
mpiGather[sendexpr, recvexpr, root, comm]	All processors (including root) in the communicator comm send their expression in sendexpr to the root processor, which produces a list of these expressions, in the order according to comm, in recvexpr. On the processors that are not root, recvexpr is ignored.
mpiAllgather[sendexpr, recvexpr, comm]	All processors in the communicator comm send their expression in sendexpr, which are organized into a list of these expressions, in the order according to comm, in recvexpr on all processors in comm.
mpiScatter[sendexpr, recvexpr, root, comm]	Processor root partitions the list in sendexpr into equal parts (if possible) and places each piece in recvexpr on all the processors (including root) in the communicator world comm, according the order and size of comm.
mpiAlltoall[sendexpr, recvexpr, comm]	Each processor sends equal parts of the list in sendexpr to all other processors in the communicator world comm, which each collects from all other processors are organizes into the order according to comm.

In one embodiment, the MPI module **302** includes program code for implementing parallel sums and other reduction operations on data stored across many nodes. MPI module **302** can also include program code for implementing simple parallel input/output calls (for example, calls that allow cluster system **200** to load and store objects that are located on a plurality of nodes).

TABLE E

Call	Description
mpiReduce[sendexpr, recvexpr, operation, root, comm]	Performs a collective reduction operation between expressions on all processors in the communicator world comm for every element in the list in sendexpr returning the resulting list in recvexpr on the processor with the ID root.

TABLE E-continued

Call	Description
mpiAllreduce [sendexpr, recvexpr, operation, comm]	Performs a collective reduction operation between expressions on all processors in the communicator world comm for every element in the list in sendexpr returning the resulting list in recvexpr on every processor.
mpiReduceScatter [sendexpr, recvexpr, operation, comm]	Performs a collective reduction operation between expressions on all processors in the communicator world comm for every element in the list in sendexpr, partitioning the resulting list into pieces for each processor's recvexpr.

These additional collective calls perform operations that reduce the data in parallel. The operation argument can be one of the constants below.

TABLE F

Constant	Description
mpiSum	Specifies that all the elements on different processors be added together in a reduction call
mpiMax	Specifies that the maximum of all the elements on different processors be chosen in a reduction call
mpiMin	Specifies that the minimum of all the elements on different processors be chosen in a reduction call

MPI Communicator Calls

In one embodiment, the MPI module **302** includes program code for implementing communicator world calls (for example, calls that would allow subsets of nodes to operate as if they were a sub-cluster). Communicators organize groups of nodes into user-defined subsets. The communicator values returned by mpiCommSplit[ ] can be used in other MPI calls instead of mpiCommWorld.

TABLE G

Call	Description
mpiCommSize[comm]	Returns the number of processors within the communicator comm
mpiCommRank[comm]	Returns the rank of this processor in the communicator comm
mpiCommDup[comm]	Returns a duplicate communicator of the communicator comm
mpiCommSplit[comm, color, key]	Creates a new communicator into several disjoint subsets each identified by color. The sort order within each subset is first by key, second according to the ordering in the previous communicator. Processors not meant to participate in any new communicator indicates this by passing the constant mpiUndefined. The corresponding communicator is returned to each calling processor.
mpiCommMap[comm] mpiCommMap[comm, target]	Returns the mapping of the communicator comm to the processor indexed according to $mpiCommWorld. Adding a second argument returns just the ID of the processor with the ID target in the communicator comm.
mpiCommFree[comm]	Frees the communicator comm

Other MPI Support Calls

Other calls that provide common functions include:

TABLE H

Call	Description
mpiWtime[ ]	Provides wall-dock time since some fixed time in the past. There is no guarantee that this time will read the same on all processors.
mpWtick[ ]	Returns the time resolution of mpiWtime[ ]
MaxByElement[in]	For every nth element of each list of the list in, chooses the maximum according to Max[ ], and returns the result as one list. Used in the mpiMax reduction operation.
MinByElement[in]	For every nth element of each list of the list in, chooses the minimum according to Min[ ], and returns the result as one list. Used in the mpiMin reduction operation.

2. Advanced Functions Module **304**

In one embodiment, the cluster node module **204** includes an advanced functions module **304**. The advanced functions module **304** can include program code that provides a toolkit of functions inconvenient or impractical to do with MPI instructions and calls implemented by the MPI module **302**. The advanced functions module **304** can rely at least partially on calls and instructions implemented by the MPI module **302** in the implementation of advanced functions. In one embodiment, the advanced functions module **304** includes a custom set of directives or functions. In an alternative embodiment, the advanced functions module **304** intercepts normal Mathematica language and converts it to one or more functions optimized for cluster execution. Such an embodiment can be easier for users familiar with Mathematica functions to use but can also complicate a program debugging process. Some functions implemented by the advanced functions module **304** can simplify operations difficult or complex to set up using parallel computing. Several examples of such functions that can be implemented by the advanced functions module **304** are shown below.

Built on the MPI calls, the calls that are described below provide commonly used communication patterns or parallel versions of Mathematica features. Unless otherwise specified, these are executed in the communicator mpiCommWorld, whose default is $mpiCommWorld, but can be changed to a valid communicator at run time.

Common Divide-and-Conquer Parallel Evaluation

In one embodiment, the advanced functions module **304** includes functions providing for basic parallelization such as, for example, routines that would perform the same operations on many data elements or inputs, stored on many nodes. These functions can be compared to parallelized for-loops and the like. The following calls address simple parallelization of common tasks. In the call descriptions, "expr" refers to an expression, and "loopspec" refers to a set of rules that determine how the expression is evaluated. In some embodiments, the advanced functions module **304** supports at least three forms of loopspec, including {var, count}, where the call iterates the variable var from 1 to the integer count; {var, start, stop}, where the call iterates the variable var every integer from start to stop; and {var, start, stop, increment}, where the call iterates the variable var from start adding increment for each iteration until var exceeds stop, allowing var to be a non-integer.

TABLE I

Call	Description
ParallelDo[expr, loopspec]	Like Do[ ] except that it evaluates expr across the cluster, rather than on just one processor. The rules for how expr is evaluated is specified in loopspec, like in Do[ ].

TABLE I-continued

Call	Description
ParallelFunctionToList [f, count]	Evaluates the function f[i] from 1 to count, but across the cluster, and returns these results in a list. The third argument has it gather this list into the processor whose ID is root.
ParallelFunctionToList [f, count, root]	
ParallelTable[expr, loopspec]	Like Table[ ] except that it evaluates expr across the cluster, rather than on just one processor, returning the locally evalulated portion. The third argument has it gather this table in to the processor whose ID is root.
ParallelTable[expr, loopspec, root]	
ParallelFunction[f, inputs, root]	Like f[inputs] except that it evaluates f on a subset of inputs scattered across the cluster from processor root and gathered back to root.
ParallelNintegrate [expr, loopspec]	Like Nintegrate[ ] except that it evaluates a numerical integration of expr over domains partitioned into the number of processors in the cluster, then returns the sum. The third argument has each numerical integration execute with at least that many digits of precision.
ParallelNintegrate [expr, loopspec, digits]	

Guard-Cell Management

In one embodiment, the advanced functions module **304** includes functions providing for guard-cell operations such as, for example, routines that perform nearest-neighbor communications to maintain edges of local arrays in any number of dimensions (optimized for 1-, 2-, and/or 3-D). Typically the space of a problem is divided into partitions. Often, however, neighboring edges of each partition can interact, so a "guard cell" is inserted on both edges as a substitute for the neighboring data. Thus the space a processor sees is two elements wider than the actual space for which the processor is responsible. EdgeCell helps maintain these guard cells.

TABLE J

Call	Description
EdgeCell[list]	Copies the second element of list to the last element of the left processor and the second-to-last element of list to the first element of the right processor while simultaneously receiving the same from its neighbors.

Matrix and Vector Manipulation

The advanced functions module **304** can also include functions providing for linear algebra operations such as, for example, parallelized versions of basic linear algebra on structures partitioned on many nodes. Such linear algebra operations can reorganize data as needed to perform matrix and vector multiplication or other operations such as determinants, trace, and the like. Matrices are partitioned and stored in processors across the cluster. These calls manipulate these matrices in common ways.

TABLE K

Call	Description
ParallelTranspose [matrix]	Like Transpose[ ] except that it transposes matrix that is in fact represented across the cluster, rather than on just one processor. It returns the portion of the transposed matrix meant for that processor.
ParallelProduct [matrix, vector]	Evaluates the product of matrix and vector, as it would on one processor, except that matrix is represented across the cluster.
ParallelDimensions [matrix]	Like Dimensions[ ] except that matrix is represented across the cluster, rather than on just one processor. It returns a list of each dimension.
ParallelTr[matrix]	Like Tr[ ] except that the matrix is represented across the cluster, rather than on just one processor. It returns the trace of this matrix.

TABLE K-continued

Call	Description
ParallelIdentity [rank]	Like Identity[ ], it generates a new identity matrix, except that the matrix is represented across the cluster, rather than on just one processor. It returns the portion of the new matrix for this processor.
ParallelOuter[f, vector1, vector2]	Like Outer[f, vector1, vector2] except that the answer becomes a matrix represented across the cluster, rather than on just one processor. It returns the portion of the new matrix for this processor.
ParallelInverse [matrix]	Like Inverse[ ] except that the matrix is represented across the cluster, rather than on just one processor. It returns the inverse of the matrix.

Element Management

In one embodiment, the advanced functions module **304** includes element management operations. For example, a large bin of elements or particles cut up in space across the nodes may need to migrate from node to node based on rules or criteria (such as their spatial coordinate). Such operations would migrate the data from one node to another. Besides the divide-and-conquer approach, a list of elements can also be partitioned in arbitrary ways. This is useful if elements need to be organized or sorted onto multiple processors. For example, particles of a system may drift out of the space of one processor into another, so their data would need to be redistributed periodically.

TABLE L

Call	Description
ElementManage [list, switch]	Selects which elements of list will be sent to which processors according to the function switch[ ] is evaluated on each element of list. If switch is a function, switch[ ] should return the ID of the processor that element should be sent. If switch is an integer, the call assumes that each elements is itself a list, whose first element is a number ranging from 0 to the passed argument. This call returns a list of the elements, from any processor, that is switch selected for this processor.
ElementManage [list]	Each element of list can be a list of two elements, the first being the ID of the processor where the element should be sent, while the second is arbitrary data to send. This call returns those list elements, from any and all processors, whose first element is this processors ID in a list. This call is used internally by the two-argument version of ElementManage[ ].

Fourier Transform

In one embodiment, the advanced functions module **304** includes program code for implementing large-scale parallel fast Fourier transforms ("FFTs"). For example, such functions can perform FFTs in one, two, and/or three dimensions on large amounts of data that are not stored on one node and that are instead stored on many nodes. Fourier transforms of very large arrays can be difficult to manage, not the least of which is the memory requirements. Parallelizing the Fourier transform makes it possible to make use of all the memory available on the entire cluster, making it possible to manipulate problem sizes that no one processor could possibly do alone.

TABLE M

Call	Description
ParallelFourier[list]	Like Fourier[ ] except that list is a two- or three-dimensional list represented across the cluster, like for matrices, above. It returns the portion of the Fourier-transformed array meant for that processor.

Parallel Disk I/O

In one embodiment, the advanced functions module **304** includes parallel disk input and output calls. For example, data may need to be read in and out of the cluster in such a way that the data is distributed across the cluster evenly. The calls in the following table enable the saving data from one or more processors to storage and the retrieval data from storage.

TABLE N

Call	Description
ParallelPut[expr, filename] ParallelPut[expr, filename root] ParallelPut[expr, filename, root, comm]	Puts expr into the file with the name filename in order on processor 0. The third argument specifies that the file be written on the processor whose ID is root. The fourth uses the communicator world comm.
ParallelGet[filename] ParallelGet[filename, root] ParallelGet[filename, root, comm]	Reads and returns data from the file with the name filename on processor 0 partitioned into each processor on the cluster. The second argument specifies that the file is to be read on the processor whose ID is root. The third uses the communicator world comm.
ParallelBinaryPut[expr, type, filename] ParallelBinaryPut[expr, filename, root] ParallelBinaryPut[expr, filename, root, comm]	Puts expr into the file with the binary format type with the name filename in order on processor 0. The fourth argument specifiest hat the file be written on the processor whose ID is root. The fifth uses the communicator world comm.
ParallelBinaryGet[type, filename] ParallelBinaryGet[type, filename, root] ParallelBinaryGet[type, filename, root, comm]	Reads and returns data in the binary format type from the file with the name filename on processor 0 partitioned into each processor on the cluster. The third argument specifies that the file is to be read on the processor whose ID is root. The fourth uses the communicator world comm.
ParallelGetPerProcessor [expr, filename] ParallelGetPerProcessor [filename, root] ParallelGetPerProcessor [filename, root, comm]	Puts expr into the file with the name filename in order on processor 0, one line per processor. The third argument specifies that the file be written on the processor whose ID is root. The fourth uses the communicator world comm.
ParallelGetPerProcessor [filename] ParallelGetPerProcessor [filename, root] ParallelGetPerProcessr [filename, root, comm]	Reads and returns data from the file with the name filename on processor 0, one line for each processor. The second argument specifies that the file is to be read on the processor whose ID is root. The third uses the communicator world comm.

Automatic Load Balancing

Some function calls can take an inconsistent amount of processing time to complete. For example, in Mathematica, the call f[20] could in general take much longer to evaluate than f[19]. Moreover, if one or more processors within the cluster are of different speeds (for example, if some operate at a core frequency of 2.6 GHz while other operate at less than one 1 GHz), one processor may finish a task sooner than another processor.

In some embodiments, the advanced functions module **304** includes a call that can improve the operation of the computer cluster **100** in such situations. In some embodiments, the root processor assigns a small subset of the possible calls for a function to each processor on the cluster **100**. Whichever processor returns its results first is assigned a second small subset of the possible calls. The root processor will continue to assign small subsets of the possible calls as results are received until an evaluation is complete. The order in which the processors finish can vary every time an expression is evaluated, but the root processor will continue assigning additional work to processors as they become available.

In one illustrative example, there are 4 processors and f[1] to f[100] to evaluate. One could implement this by assigning f[1], f[2], f[3], f[4] to each of processors 0 (the root can assign to oneself) through 3. If the f[2] result came back first, then processor **1** would be assigned f[5]. If the f[4] result is returned next, f[6] would be assigned to processor **3**. The assignments continue until all results are calculated. The results are organized for output back to the user.

In alternative embodiments, the subsets of possible calls can be assigned in any order, rather than sequentially, or in batches (for example, f[1], f[5], f[9] assigned to processor **1**, etc.). Also, the subsets could be organized by delegation. For example, one processor node may not necessarily be in direct control of the other processors. Instead, a large subset could be assigned to a processor, which would in turn assign subsets of its work to other processors. The result would create a hierarchy of assignments like a vast army.

TABLE O

LoadBalanceFunctionToList[f, count] LoadBalanceFunctionToList[f, count, root]	Evaluates the function f[i] from 1 to count, but across the cluster using load-balancing techniques, and returns these results in a list. The third argument has it gather this list into the processor whose ID is root.

3. Received Message Queue **306**

In one embodiment, the cluster node module **204** includes a received message queue **306**. The received message queue **306** includes a data structure for storing messages received from other cluster node modules. Related data pertaining to the messages received, such as whether an expression has been completed, may also be stored in the received message queue **306**. The received message queue **306** may include a queue and/or another type of data structure such as, for example, a stack, a linked list, an array, a tree, etc.

4. Message Receiving Queue **308**

In one embodiment, the cluster node module **204** includes a message receiving queue **308**. The message receiving queue **308** includes a data structure for storing information about the location to which an expression is expected to be sent and the processor from which the expression is expected. The message receiving queue **308** may include a queue and/or another type of data structure such as, for example, a stack, a linked list, an array, a tree, etc.

B. Cluster Configuration Module **208**

Cluster configuration module **208** includes program code for initializing a plurality of cluster node modules to add cluster computing support to computer systems **110**, **120**, **130**. U.S. Pat. No. 7,136,924, issued to Dauger (the "'924 patent"), the entirety of which is hereby incorporated by reference and made a part of this specification, discloses a method and system for parallel operation and control of computer clusters. One method generally includes obtaining one or more personal computers having an operating system with discoverable of network services. In some embodiments, the method includes obtaining one or more processors or processor cores on which a kernel module can run. As described in the '924 patent, a cluster node control and interface (CNCI) group of software applications is copied to each node. When the CNCI applications are running on a node, the cluster configuration module **208** can permit a cluster node module **204**, in combination with a kernel module **206**, to use the

node's processing resources to perform a parallel computation task as part of a computer cluster. The cluster configuration module **208** allows extensive automation of the cluster creation process in connection with the present disclosure.

C. User Interface Module **202**

In some embodiments, computer cluster **100** includes a user interface module **202**, such as, for example a Mathematica Front End or a command line interface, that includes program code for a kernel module **206** to provide graphical output, accept graphical input, and provide other methods of user communication that a graphical user interface or a command-line interface provides. To support a user interface module **202**, the behavior of a cluster node module **204***a* is altered in some embodiments. Rather than sending output to and accepting input from the user directly, the user interface module **202** activates the cluster node module **204***a* to which it is connected and specifies parameters to form a connection, such as a MathLink connection, between the cluster node module **204***a* and the user interface module **202**. The user interface module's activation of the cluster node module **204***a* can initiate the execution of instructions to activate the remaining cluster node modules **204***b-e* on the cluster and to complete the sequence to start all kernel modules **206***a-e* on the cluster. Packets from the user interface module **202**, normally intended for a kernel module **206***a*, are accepted by the cluster node module **204***a* as a user command. Output from the kernel module **206***a* associated with the cluster node module **204***a* can be forwarded back to the user interface module **202** for display to a user. Any of the cluster node modules **204***a-e* can be configured to communicate with a user interface module **202**.

D. Kernel Module **206**

A kernel module **206** typically includes program code for interpreting high-level code, commands, and/or instructions supplied by a user or a script into low-level code, such as, for example, machine language or assembly language. In one embodiment, each cluster node module **204***a-e* is connected to all other cluster node modules, while each kernel module **206***a-e* is allocated and connected only to one cluster node module **204**. In one embodiment, there is one cluster node module-kernel module pair per processor. For example, in an embodiment of a computer cluster **100** including single-processor computer systems, each cluster node module-kernel module pair could reside on a single-processor computer. If a computer contains multiple processors or processing cores, it may contain multiple cluster node module-kernel module pairs, but the pairs can still communicate over the cluster node module's network connections.

IV. Cluster Computing Methods

In one embodiment, the computer cluster **100** includes a cluster initialization process, a method of cluster node module operation, and a cluster shut down process.

A. Cluster Initialization Process

In one embodiment, a cluster configuration module **202** initializes one or more cluster node modules **204** in order to provide cluster computing support to one or more kernel modules **206**, as shown in FIG. **4**.

At **402**, cluster node modules are launched on the computer cluster **100**. In one embodiment, the cluster node module **204***a* running on a first processor **112***a* (for example, where the user is located) accesses the other processors **112***b*, **122***a-b*, **132** on the computer cluster **100** via the cluster configuration module **208** to launch cluster node modules **204***b-e* onto the entire cluster. In an alternative embodiment, the cluster configuration module **208** searches for processors **112***a-b*, **122***a-b*, **132** connected to one another via communications network **102** and launches cluster node modules **204***a-e* on each of the processors **112***a-b*, **122***a-b*, **132**.

The cluster node modules **204***a-e* establish communication with one another at **404**. In one embodiment, each of the cluster node modules **204***a-e* establish direct connections using the MPI_Init command with other cluster node modules **204***a-e* launched on the computer cluster **100** by the cluster configuration module **208**.

At **406**, each cluster node module **204** attempts to connect to a kernel module **206**. In one embodiment, each instance of the cluster node modules **204***a-e* locates, launches, and connects with a local kernel module via MathLink connections and/or similar connection tools, for example, built into the kernel module **206**.

At **408**, the cluster node modules **204** that are unconnected to a kernel module **206** are shut down. In one embodiment, each cluster node module **204** determines whether the local kernel module cannot be found or connected to. In one embodiment, each cluster node module **204** reports the failure to connect to a kernel module **206** to the other cluster node modules on computer cluster **100** and quits.

Processor identification numbers are assigned to the remaining cluster node modules **204** at **410**. In one embodiment, each remaining cluster node module **204** calculates the total number of active processors (N) and determines identification numbers describing the remaining subset of active cluster node modules **204***a-e* and kernel modules **206***a-e*. This new set of cluster node module-kernel module pairs may be numbered 0 through N−1, for example.

Message passing support is initialized on the kernel modules **206***a-e* at **412**. In one embodiment, each cluster node module **204** supplies initialization code (for example, Mathematica initialization code) to the local kernel module **206** to support message passing.

Finally, at **414**, the cluster node modules **204***a-e* enter a loop to accept user entry. In one embodiment, a main loop (for example, a cluster operation loop) begins execution after the cluster node module **204***a* on the first processor **112***a* returns to user control while each of the other cluster node modules **204** waits for messages from all other cluster node modules **204***a-e* connected to the network **102**.

The initialization process creates a structure enabling a way for the kernel modules **206***a-e* to send messages to one another. In some embodiments, any kernel module can send data to and receive data from any other kernel module within the cluster when initialization is complete. The cluster node module creates an illusion that a kernel module is communicating directly with the other kernel modules. The initialization process can create a relationship among kernel modules on a computer cluster **100** such as the one shown by way of example in FIG. **2**.

B. Cluster Node Module Operation

In one embodiment, a cluster node module **204** implements cluster computing support for a kernel module **206** during a main loop, as shown in FIG. **5**.

At **502**, cluster node modules **204** wait for user commands or messages from other cluster node modules. In one embodiment, the cluster node module **204***a* connected to the user interface module **202** waits for a user command, while the other cluster node modules **204***b-e* continue checking for messages.

Once a command or message is received, the method proceeds to **504**. At **504**, the cluster node module **204***a* determines whether the message received is a quit command. If a quit command is received, the cluster node module **204***a* exits

the loop and proceeds to a cluster node module shut down process at **505**. If the message received is not a quit command, the process continues to **506**.

At **506**, received commands are communicated to all cluster node modules **204***a-e* on the computer cluster **100**. In one embodiment, when a user enters a command in the user interface module **202**, the cluster node module **204***a* connected to the user interface module **202** submits the user command to all other cluster node modules **204***b-e* in the computer cluster **100**. The user commands can be simple (for example, "1+1"), but can also be entire subroutines and sequences of code (such as, for example, Mathematica code), including calls to MPI from within the user interface module **202** (for example, the Mathematica Front End) to perform message passing between kernel modules **206***a-e* (for example, Mathematica kernels). These include the fundamental MPI calls, which are implemented using specially identified messages between a cluster node module **204** and its local kernel module **206**.

The message (or user command) is communicated to the kernel modules **206***a-e* at **508**. In one embodiment, the cluster node module **204***a* connected to the user interface module **202** submits the user command to the kernel module **206***a* to which it is connected. Each of the other cluster node modules **204***b-e*, after receiving the message, submits the command to the respective kernel module **206***b-e* to which it is connected.

At **510**, a cluster node module **204** receives a result from a kernel module **206**. In one embodiment, once the kernel module **206** completes its evaluation, it returns the kernel module's output to the cluster node module **204** to which it is connected. Depending on the nature of the result from the kernel module, the cluster node module **204** can report the result to a local computer system or pass the result as a message to another cluster node module **204**. For example, the cluster node module **204***a* running on the first processor **112***a* reports the output on its local computer system **110**. For example, on the first processor **112***a*, cluster node module **204***a* only directly reports the output of kernel module **206***a*.

Messages from other cluster node modules **204** are responded to at **512**. In one embodiment, each cluster node module (for example, the cluster node module **204***a*) checks for and responds to messages from other cluster node modules **204***b-e* and from the kernel module **206***a* repeatedly until those are exhausted. In one embodiment, output messages from the kernel module **206** are forwarded to output on the local computer system. Messages from other cluster node modules **204** are forwarded to a received message queue **306** ("RMQ"). Data from each entry in the message receiving queue **308** ("MRQ") is matched with entries in the RMQ **306** (see, for example, description of the mpilRecv[ ] call, above). If found, data from the MRQ **308** are combined into those in the RMQ **306** and marked as "completed" (see, for example, description of the mpiTest[ ] call, above). This process provides the peer-to-peer behavior of the cluster node modules **204***a-e*. Via this mechanism, code running within multiple, simultaneously running kernel modules (for example, Mathematica kernels) can interact on a pair-wise or collective basis, performing calculations, processing, or other work on a scale larger and/or faster than one kernel could have done alone. In this manner, user-entered instructions and data specifying what work will be done via user commands can be executed more quickly and/or reliably. Once responding to messages has completed, the process returns to **502**.

In some embodiments, a computer system includes software, such as an operating system, that divides memory and/or other system resources into a user space, a kernel space, an application space (for example, a portion of the user space allocated to an application program), and/or an operating system space (for example, a portion of the user space allocated to an operating system). In some embodiments, some or all of the cluster node modules **204***a-e* are implemented in the application space of a computer system. In further embodiments, at least some of the cluster node modules **204***a-e* are implemented in the operating system space of a computer system. For example, some cluster node modules in a computer cluster may operate in the application space while others operate in the operating system space.

In some embodiments, some or all of the functionality of the cluster node modules **204***a-e* is incorporated into or integrated with the operating system. The operating system can add cluster computing functionality to application programs, for example, by implementing at least some of the methods, modules, data structures, commands, functions, and processes discussed herein. Other suitable variations of the techniques described herein can be employed, as would be recognized by one skilled in the art.

In some embodiments, the operating system or components of the operating system can identify and launch the front end **202** and the kernels **206**. The operating system or its components can connect the front end **202** and kernels **206** to one another in the same manner as a cluster node module **204** would or by a variation of one of the techniques described previously. The operating system can also be responsible for maintaining the communications network **102** that connects the modules to one another. In some embodiments, the operating system implements at least some MPI-style calls, such as, for example, collective MPI-style calls. In some embodiments, the operating system includes an application programming interface (API) library of cluster subroutine calls that is exposed to application programs. Applications programs can use the API library to assist with launching and operating the computer cluster.

C. Cluster Shut Down Process

In one embodiment, a computer cluster **100** includes a procedure to shut down the system. If the operation process (or main loop) on the cluster node module **204***a* connected to the user interface module **202** detects a "Quit" or "Exit" command or otherwise receives a message from the user indicating a shut down, the sequence to shut down the cluster node modules **204***a-e* and the kernel modules **206***a-e* is activated. In one embodiment, the cluster node module **204***a* connected to the user interface module **202** sends a quit message to all other cluster node modules **204***b-e*. Each cluster node module **204** forwards the quit command to its local kernel module **206**. Once its Mathematica kernel has quit, each cluster node module **204** proceeds to tear down its communication network with other cluster node modules (for example, see description of the MPI_Finalize command, above). At the conclusion of the process, each cluster node module **204** exits execution.

V. Example Operation

For purposes of illustration, sample scenarios are discussed in which the computer cluster system is used in operation. In these sample scenarios, examples of Mathematica code are given, and descriptions of how the code would be executed by a cluster system are provided.

Basic MPI

Fundamental data available to each node includes the node's identification number and total processor count.

```
In[1]:={$IdProc,$NProc}

Out[1]:={0,2}
```

The first element should be unique for each processor, while the second is generally the same for all. Processor 0 can see what other values are using a collective (see below) communications call such as mpiGather[ ].

```
In[2]:=mpiGather[{$IdProc,$NProc},list,0];list

Out[2]:={{0,2},{1,2}}
```

Peer-to-Peer MPI

The mpiSend and mpiRecv commands make possible basic message passing, but one needs to define which processor to target. The following defines a new variable, targetProc, so that each pair of processors will point to each other.

```
In[3]:=targetProc=If[1==Mod [$IdProc,2],$IdProc-1,
    $IdProc+1]

Out[3]:=1
```

In this example, the even processors target its "right" processor, while the odd ones point its "left." For example, if the processors were lined up in a row and numbered in order, every even-numbered processor would pair with the processor following it in the line, and every odd-numbered processor would pair with the processor preceding it. Then a message can be sent:

```
In[4]:=If [1==Mod [$IdProc,2],mpiSend[N[Pi,22],
    targetProc, mpiCommWorld,d], mpiRecv[a,tar-
    getProc,mpiCommWorld,d]]
```

The If [ ] statement causes the processors to evaluate different code: the odd processor sends 22 digits of Pi, while the even receives that message. Note that these MPI calls return nothing. The received message is in the variable a:

```
In[5]:=a

Out[5]:=3.1415926535897932384626

In[6]:=Clear[a]
```

The variable a on the odd processors would have no definition. Moreover, if $NProc is 8, processor 3 sent Pi to processor 2, processor 5 sent Pi to processor 4, and so on. These messages were not sent through processor 0, but they communicated on their own.

The mpiISend and mpiIRecv commands have a letter "I" to indicate asynchronous behavior, making it possible to do other work while messages are being sent and received, or if the other processor is busy. So, the above example could be done asynchronously:

```
In[7]:=If[1==Mod [$IdProc,2],mpiISend[N[Pi,22],
    targetProc, mpiCommWorld,d,e], mpiIRecv[a,
    targetProc,mpiCommWorld,d,e]]
```

The variable e has important data identifying the message, and mpiTest[e] can return True before the expressions are to be accessed. At this point, many other evaluations can be performed. Then, one can check using mpiTest when the data is needed:

```
In[29]:=mpiTest[e]

Out[29]:=True

In[30]:=a

Out[30]:=3.1415926535897932384626

In[31]:=Clear[a,e]
```

The mpiWait[e] command could have also have been used, which does not return until mpiTest[e] returns True. The power of using these peer-to-peer calls is that it becomes possible to construct any message-passing pattern for any problem.

Collective MPI

In some cases, such explicit control is not required and a commonly used communication pattern is sufficient. Suppose processor 0 has an expression in b that all processors are meant to have. A broadcast MPI call would do:

```
In[8]:=mpiBcast[b,0,mpiCommWorld]
```

The second argument specifies which processor is the "root" of this broadcast; all others have their b overwritten. To collect values from all processors, use mpiGatherD:

```
In[9]:=mpiGather[b,c,0,mpiCommWorld]
```

The variable c of processor 0 is written with a list of all the b of all the processors in mpiCommWorld. The temporal opposite is mpiScatter:

```
In[10]:=Clear[b];a={2,4,5,6};mpiScatter[a,b,0, mpi-
    CommWorld];b

Out[10]:={2,4}
```

The mpiScafter command cuts up the variable a into even pieces (when possible) and scatters them to the processors. This is the result if $NProc=2, but if $NProc=4, b would only have {2}.

MPI provides reduction operations to perform simple computations mixed with messaging. Consider the following:

```
In[11]:=a={{2+$IdProc,45[ ],3,{1+$IdProc,
    $NProc[ ] [ ]; mpiReduce [a,d,mpiSum, 0,mpi-
    CommWorld]

In[12]:=d

Out[12]:={{5,90},6,{3,4}}
```

The mpiSum constant indicates that variable a of every processor will be summed. In this case, $NProc is 2, so those elements that were not identical result in odd sums, while those that were the same are even.

Most of these calls have default values if not all are specified. For example each of the following calls will have the equivalent effect as the above mpiGather[ ] call:

```
mpiGather[b,c,0]

mpiGather[b,c]

c=mpiGather[b]
```

High-Level Calls

High-level calls can include convenient parallel versions of commonly used application program calls (for example, Mathematica calls). For example, ParallelTable[ ] is like Table[ ], except that the evaluations are automatically performed in a distributed manner:

```
In[13]:=ParallelTable[i,{i,100},0]

Out[13]:={1,2,3,4,5, . . . , 99,100}
```

The third argument specifies that the answers are collated back to processor 0. This is a useful, simple way to parallelize many calls to a complex function. One could define a complicated function and evaluate it over a large range of inputs:

```
In[14]:=g[x_]:=Gamma[2+0.5*(x-1)]; ParallelTable
    [g[i],{i,100},0]

Out[14]:={1,1.32934,2.,3.32335,6.,11.6317,24.,
    52.3428, 120.,287.885,720}
```

ParallelFunctionToList[ ] also provides a simplified way to perform this form of parallelism.

Operations with Non-Trivial Communication

Matrix Operations

In some embodiments, one or more functions can help solve matrix calculations in parallel:

```
In[15]:=a=Table [i+3*$IdProc+2 j,{i,2},{j,4}]

Out[15]:={{3,5,7,9},{4,6,8,10}}

In[16]:=t=ParallelTranspose[a]

Out[16]:={{3,4,6,7},{5,6,8,9}}
```

Fourier Transforms

A Fourier transform of a large array can be solved faster in parallel, or made possible on a cluster because it can all be held in memory. A two-dimensional Fourier transform of the above example follows:

```
In[17]:=f=ParallelFourier[a]

Out[17]:={{32.+0.I,-4.-4.I,-4.,-4.+4.I},{-3.-3.I,0.+
    0.I,0.,0.+0.I}}
```

Edge Cell Management

Many problems require interactions between partitions, but only on the edge elements. Maintaining these edges can be performed using EdgeCell[ ].

```
In[18]:=a={2,4,5,6,7}+8*$IdProc

Out[18]:={2,4,5,6,7}

In[19]:=EdgeCell [a];a

Out[19]:={14,4,5,6,12}
```

Element Management

In particle-based problems, items can drift through space, sometimes outside the partition of a particular processor. This can be solved with ElementManage[ ]:

```
In[20]:=list={{0,4},{1,3},{1,4},{0,5}};fcn[x_]:=
    x[[1]]

In[21]:=ElementManage[list,fcn]

Out[21]:={{0,4},{0,5},{0,4},{0,5}}

In[21]:=ElementManage[list,2]

Out[21]:={{0,4},{0,5},{0,4},{0,5}}
```

The second argument of ElementManage describes how to test elements of a list. The fcn identifier returns which processor is the "home" of that element. Passing an integer assumes that each element is itself a list, whose first element is a number ranging from 0 to the passed argument.

While the examples above involve Mathematica software and specific embodiments of MPI calls and cluster commands, it is recognized that these embodiments are used only to illustrate features of various embodiments of the systems and methods.

VI. Additional Embodiments

Although cluster computing techniques, modules, calls, and functions are disclosed with reference to certain embodiments, the disclosure is not intended to be limited thereby. Rather, a skilled artisan will recognize from the disclosure herein a wide number of alternatives for the exact selection of cluster calls, functions, and management systems. For example, single-node kernels can be managed using a variety of management tools and/or can be managed manually by a user, as described herein. As another example, a cluster node module can contain additional calls and procedures, including calls and procedures unrelated to cluster computing, that are not disclosed herein.

Other embodiments will be apparent to those of ordinary skill in the art from the disclosure herein. Moreover, the described embodiments have been presented by way of example only, and are not intended to limit the scope of the disclosure. Indeed, the novel methods and systems described herein can be embodied in a variety of other forms without departing from the spirit thereof. Accordingly, other combinations, omissions, substitutions and modifications will be apparent to the skilled artisan in view of the disclosure herein. Thus, the present disclosure is not intended to be limited by the disclosed embodiments, but is to be defined by reference to the appended claims. The accompanying claims and their equivalents are intended to cover forms or modifications as would fall within the scope and spirit of the inventions.

What is claimed is:

1. A computer cluster comprising:

a plurality of nodes, wherein each node is configured to access a computer-readable medium comprising program code for a single-node kernel module configured to interpret user instructions;

a plurality of cluster node modules, wherein each cluster node module is stored in a computer-readable medium and configured to communicate with a single-node kernel and with one or more other cluster node modules, to accept instructions from a user interface or a script, and to interpret at least some of the user instructions such that the plurality of cluster node modules communicate with one another in order to act as a cluster; and

a communications network to connect the nodes;

wherein the plurality of cluster node modules are configured to cooperate to interpret, translate, or interpret and translate commands for execution by a plurality of single-node kernel modules, and wherein at least one of the plurality of cluster node modules returns a result to the user interface.

2. The computer cluster of claim **1**, wherein the plurality of cluster node modules reside in an operating system space.

3. The computer cluster of claim **1**, wherein the plurality of cluster node modules are integrated into an operating system.

4. The computer cluster of claim **1**, wherein an operating system incorporates at least one of the plurality of cluster node modules.

5. The computer cluster of claim **4**, wherein the operating system comprises a programming interface exposed to an application program, the programming interface being configured to allow the application program to access functionality supplied by the at least one cluster node module.

6. The computer cluster of claim **1**, wherein each of the plurality of cluster node modules is configured to use an application-specific toolkit comprising commands that can be interpreted by at least one of the plurality of single-node kernels into code executable by a hardware processor, wherein each of the plurality of cluster node modules is paired with a single-node kernel that is able to interpret the commands of the application-specific toolkit, and wherein each cluster node module and single-node kernel pairing is associated with a unique identifier.

7. The computer cluster of claim **1**, wherein each of the plurality of single-node kernels is configured to access a partitioned space to perform processing of data.

8. The computer cluster of claim 1, wherein each of the plurality of cluster node modules is configured to access a partitioned space to perform processing of data.

9. The computer cluster of claim 1, wherein one of the plurality of cluster node modules is configured to receive a command from a user interface or from a script, activate at least some other cluster node modules, and initialize operation of a computer cluster in which each of the plurality of cluster node modules is configured to access a partitioned space to perform processing of data.

10. The computer cluster of claim 1, wherein at least a portion of the computer cluster runs on a computer system that is remote from one or more other computer systems running the computer cluster.

11. The computer cluster of claim 1, wherein a cluster node module is stored in a computer-readable medium that is separate from a computer-readable medium in which at least some other cluster node modules are stored.

12. The computer cluster of claim 1, wherein a single-node kernel is stored in a computer-readable medium that is separate from a computer-readable medium in which at least some other single-node kernels are stored.

13. The computer cluster of claim 1, wherein each of at least two single-node kernels are configured to perform processing of data in memory shared by multiple nodes.

14. The computer cluster of claim 1, wherein each of at least two cluster node modules are configured to perform processing of data in memory shared by multiple nodes.

15. The computer cluster of claim 1, wherein a user interface module is stored on a computer system that has one or more local storage devices.

16. A method of evaluating a command on a computer cluster comprising:

communicating a command from at least one of a user interface or a script to one or more cluster node modules within the computer cluster;

for each of the one or more cluster node modules, communicating a message based on the command to a respective kernel module associated with the cluster node module;

for each of the one or more cluster node modules, receiving a result from the respective kernel module associated with the cluster node module; and

for at least one of the one or more cluster node modules, responding to messages from other cluster node modules;

wherein the one or more cluster node modules reside in an operating system space.

17. The method of claim 16, wherein responding to messages from other cluster node modules comprises:

forwarding messages from the other cluster node modules to a received message queue;

matching data from each entry in a message receiving queue with entries in the received message queue;

combining data from the message receiving queue with matching data in the received message queue; and

marking the matching data as completed.

18. The method of claim 16, wherein the one or more cluster node modules are integrated into an operating system.

19. The method of claim 16, wherein an operating system incorporates at least one of the one or more cluster node modules.

20. The method of claim 19, wherein the operating system comprises a programming interface exposed to an application program, the programming interface being configured to allow the application program to access functionality supplied by the at least one of the one or more cluster node modules.

21. A computing system for executing Mathematica code on multiple nodes comprising:

a first node module in communication with a first Mathematica kernel executing on a first node;

a second node module in communication with a second Mathematica kernel executing on a second node; and

a third node module in communication with a third Mathematica kernel executing on a third node;

wherein the first node module, the second node module, and the third node module are configured to communicate with one another using a peer-to-peer architecture; and

wherein the first node module resides in an operating system space.

22. The computing system of claim 21, wherein each of the first node module, the second node module, and the third node module comprises:

a data structure for maintaining messages originating from other node modules; and

a data structure for maintaining data specifying:

a location to which a message is expected to be received; and

an identifier for a node from which the message is expected to be sent.

23. The computing system of claim 21, wherein the first node module is integrated into an operating system.

24. The computing system of claim 21, wherein an operating system incorporates at least one of the first node module, the second node module, or the third node module.

25. The computing system of claim 24, wherein the operating system comprises a programming interface exposed to an application program, the programming interface being configured to allow the application program to access functionality supplied by the at least one node module.

26. The computer cluster of claim 9, wherein the processing of data involves communications among at least three cluster node modules.

27. The computer cluster of claim 9, wherein the processing of data involves exchanging data between two or more partitioned spaces.

28. The computer cluster of claim 9, wherein the processing of data occurs on fewer than all cluster node modules connected to the communications network.

29. The computer cluster of claim 1, wherein a user interface module is stored in a computer-readable medium that is separate from a computer-readable medium in which at least some of the plurality of cluster node modules are stored.

* * * * *

EXHIBIT C

US008676877B2

(12) **United States Patent**
Tannenbaum et al.

(10) **Patent No.: US 8,676,877 B2**
(45) **Date of Patent: Mar. 18, 2014**

(54) **CLUSTER COMPUTING USING SPECIAL PURPOSE MICROPROCESSORS**

(75) Inventors: **Zvi Tannenbaum**, Newport Beach, CA (US); **Dean E. Dauger**, Huntington Beach, CA (US)

(73) Assignee: **Advanced Cluster Systems, Inc.**, Aliso Viejo, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 174 days.

(21) Appl. No.: **13/423,063**

(22) Filed: **Mar. 16, 2012**

(65) **Prior Publication Data**

US 2013/0097406 A1 Apr. 18, 2013

Related U.S. Application Data

(63) Continuation of application No. 12/040,519, filed on Feb. 29, 2008, now Pat. No. 8,140,612, which is a continuation-in-part of application No. 11/744,461, filed on May 4, 2007, now Pat. No. 8,082,289.

(60) Provisional application No. 60/813,738, filed on Jun. 13, 2006, provisional application No. 60/850,908, filed on Oct. 11, 2006.

(51) **Int. Cl.**
G06F 9/44 (2006.01)
G06F 15/16 (2006.01)

(52) **U.S. Cl.**
USPC **709/201**; 717/177; 719/320

(58) **Field of Classification Search**
USPC 709/201; 719/320; 717/177
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,423,046	A	6/1995	Nunnelley et al.
5,881,315	A	3/1999	Cohen
6,108,699	A	8/2000	Moiin
6,202,080	B1	3/2001	Lu et al.
6,546,403	B1	4/2003	Carlson, Jr. et al.
6,578,068	B1	6/2003	Bowman-Amuah
6,751,698	B1	6/2004	Deneroff et al.
6,782,537	B1	8/2004	Blackmore et al.
6,968,335	B2	11/2005	Bayliss et al.
7,093,004	B2	8/2006	Bernardin et al.

(Continued)

FOREIGN PATENT DOCUMENTS

JP	08-87473 A	2/1996
JP	2002117010	4/2002

OTHER PUBLICATIONS

"gridMathematica 1.1: Grid Computing Gets a Speed Boost from Mathematica 5", The Mathematica Journal, vol. 9, No. 2, 2004.
"Wolfram gridMathematica", Wolfram Research, Inc., 2007.

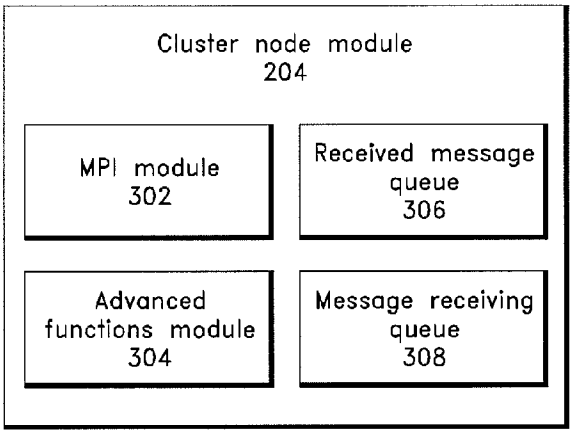
(Continued)

Primary Examiner — Larry Donaghue
(74) *Attorney, Agent, or Firm* — Knobbe, Martens, Olson & Bear, LLP

(57) **ABSTRACT**

In some embodiments, a computer cluster system comprises a plurality of nodes and a software package comprising a user interface and a kernel for interpreting program code instructions. In certain embodiments, a cluster node module is configured to communicate with the kernel and other cluster node modules. The cluster node module can accept instructions from the user interface and can interpret at least some of the instructions such that several cluster node modules in communication with one another and with a kernel can act as a computer cluster.

14 Claims, 5 Drawing Sheets



(56) **References Cited**

U.S. PATENT DOCUMENTS

7,136,924	B2	11/2006	Dauger
7,174,381	B2	2/2007	Gulko et al.
7,249,357	B2	7/2007	Landman et al.
7,334,232	B2	2/2008	Jacobs et al.
7,437,460	B2	10/2008	Chidambaran et al.
7,472,193	B2	12/2008	Dauger
7,554,949	B2	6/2009	Chen
7,937,455	B2	5/2011	Saha et al.
8,082,289	B2	12/2011	Tannenbaum et al.
8,140,612	B2	3/2012	Tannenbaum et al.
2002/0049859	A1	4/2002	Bruckert et al.
2003/0005266	A1	1/2003	Akkary et al.
2003/0051062	A1	3/2003	Circenis
2003/0135621	A1	7/2003	Romagnoli
2003/0195931	A1	10/2003	Dauger
2003/0195938	A1	10/2003	Howard et al.
2004/0110209	A1	6/2004	Yokota et al.
2004/0157203	A1	8/2004	Dunk
2005/0021751	A1	1/2005	Block et al.
2005/0038852	A1	2/2005	Howard
2005/0060237	A1	3/2005	Barsness et al.
2005/0076105	A1	4/2005	Keohane et al.
2005/0108394	A1	5/2005	Braun et al.
2005/0180095	A1	8/2005	Ellis
2006/0053216	A1	3/2006	Deokar et al.
2006/0106931	A1	5/2006	Richoux
2007/0073705	A1	3/2007	Gray
2008/0250347	A1	10/2008	Gray et al.
2008/0281997	A1	11/2008	Archer et al.
2009/0222543	A1	9/2009	Tannenbaum et al.

OTHER PUBLICATIONS

Carns et al., "An Evaluation of Message Passing Implementations on Beowulf Workstations", Aerospace Conference, Mar. 6-13, 1999, IEEE 1999, vol. 5, pp. 41-54.

Dauger et al., "Plug-and-Play Cluster Computing using Mac OS X", IEEE International Conference, Dec. 1-4, 2003, Paper appears in Cluster Computing Jan. 8, 2004, pp. 430-435.

Dauger et al., Plug-and Play Cluster Computing: High-Performance Computing for the Mainstream, IEEE Computing in Science and Engineering, Mar./Apr. 2005, pp. 27-33.

Hamscher et al., "Evaluation of Job-Scheduling Strategies for grid Computing", LNCS: Lecture Notes in Computer Science, 2000, pp. 191-202.

International Search Report dated Sep. 11, 2008, International Application No. PCT/US07/70585.

Jahanzeb et al., "Libra: a computational economy-based job scheduling system for clusters", Software Practice and Experience, Feb. 24, 2004, vol. 34, pp. 573-590.

Jain et al., "Data Clustering: A Review", ACM Computing Surveys, Sep. 1999, vol. 31, No. 3.

Maeder, R., "Mathematica Parallel Computing Toolkit: Unleash the Power of Parallel Computing", Wolfram Research, Jan. 2005.

Tepeneu and Ida, "MathGridLink—A bridge between Mathematica and the Grid", Nippon Sofutowea Kagakkai Taikai Ronbunshu, 2003, vol. 20, pp. 74-77.

Wolfram, Stephen: The Mathematica Book 5th Edition; Wolfram Research, Inc. 2003.

Kepner, Jeremy et al., "Parallel Matlab: The Next Generation", Aug. 20, 2004, MIT Lincoln Laboratory, Lexington, MA.

Kim, Hahn et al., "Introduction to Parallel Programming and pMatlab v2.0", 2011, MIT Lincoln Laboratory, Lexington, MA.

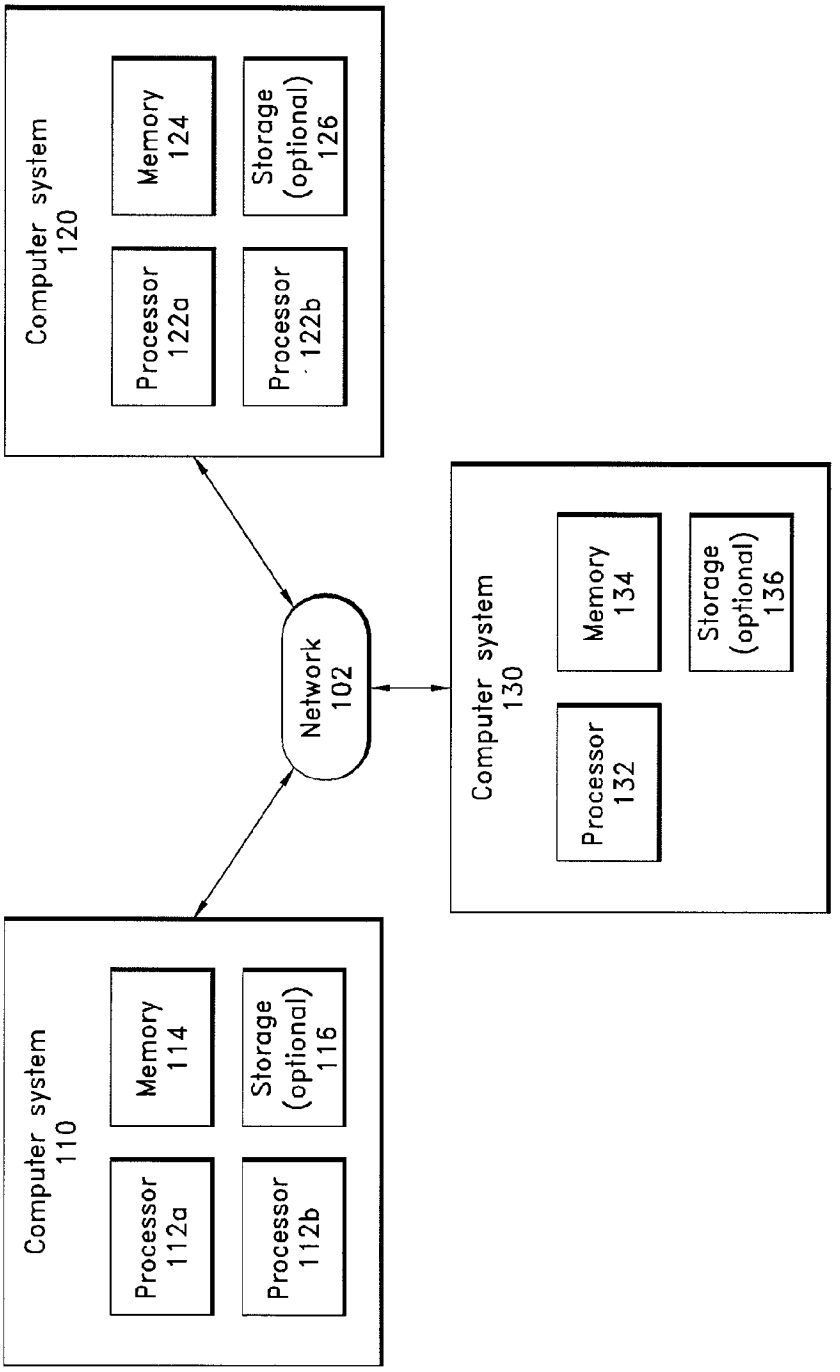

Computer system
120
Memory
124
Storage
(optional)
126
Processor
122a
Processor
122b
Network
102
Computer system
130
Memory
134
Storage
(optional)
136
Processor
132
Computer system
110
Memory
114
Storage
(optional)
116
Processor
112a
Processor
112b

FIG. 1

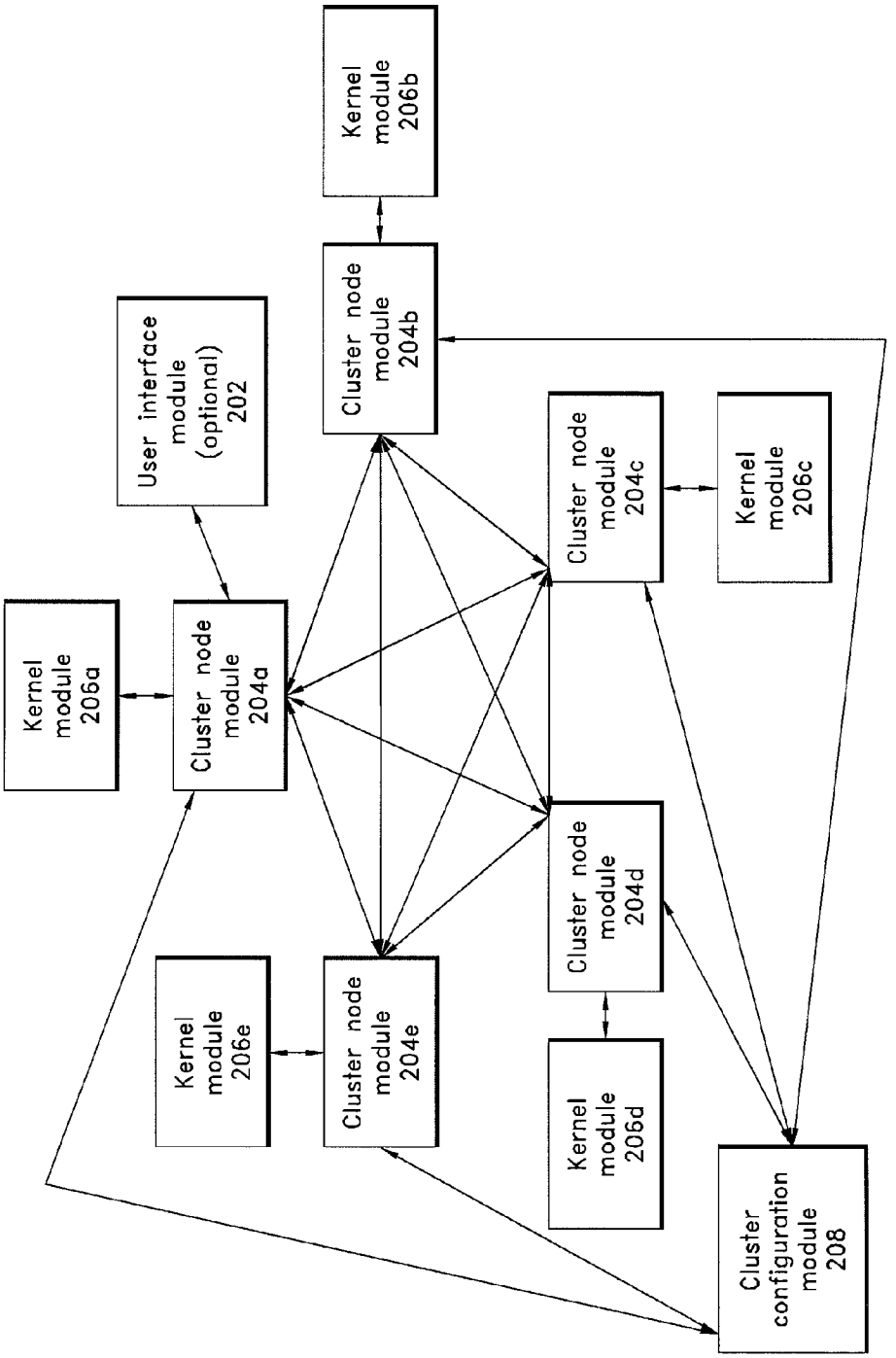

Kernel module 206b
Cluster node module 204b
User interface module (optional) 202
Cluster node module 204c
Kernel module 206c
Kernel module 206a
Cluster node module 204a
Cluster node module 204d
Kernel module 206e
Cluster node module 204e
Kernel module 206d
Cluster configuration module 208

FIG. 2

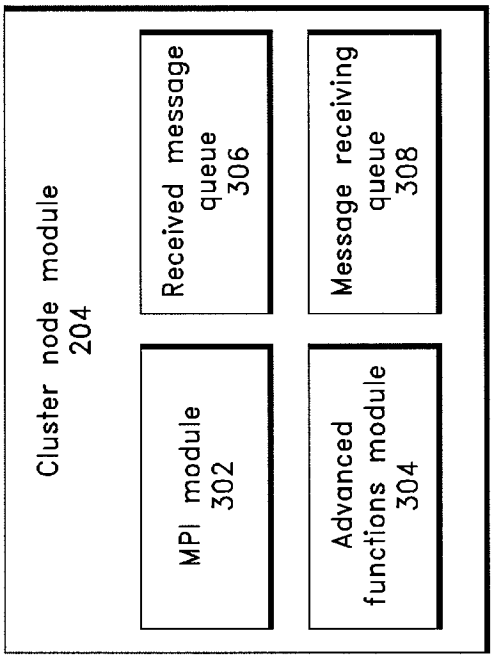

Cluster node module
204
Received message queue
306
Message receiving queue
308
MPI module
302
Advanced functions module
304

FIG. 3

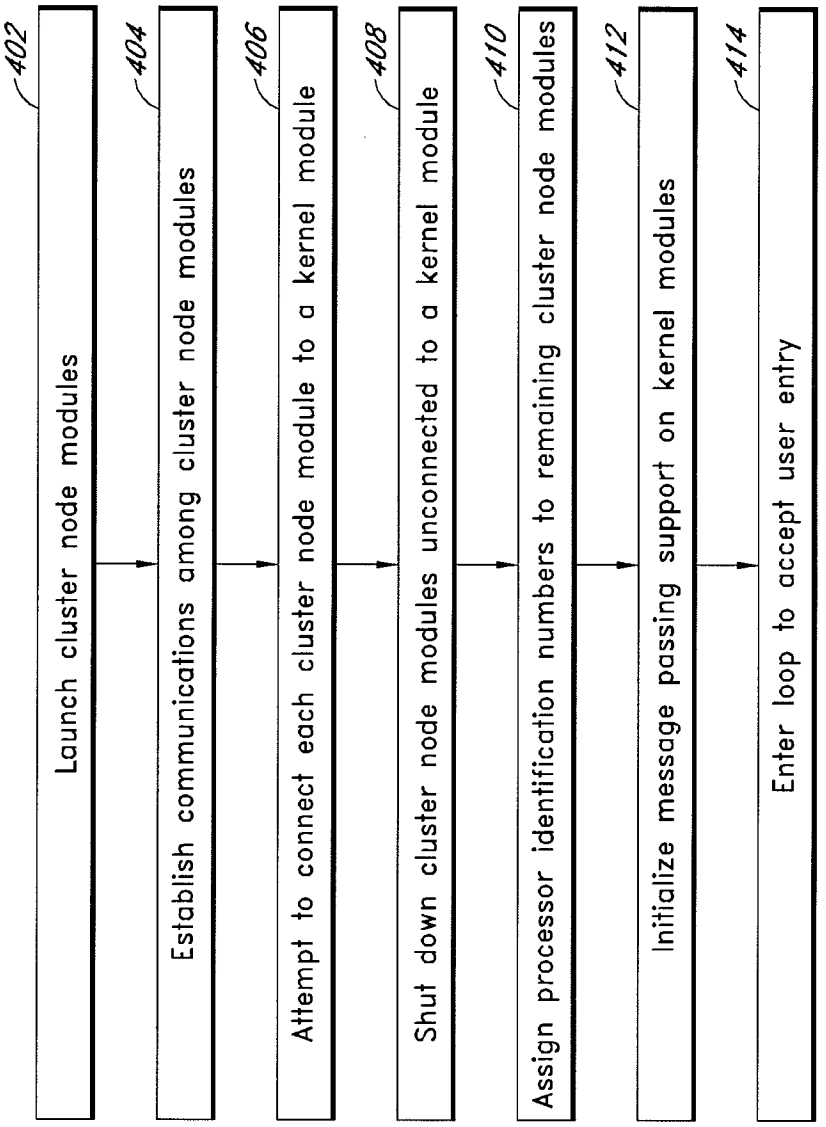

402
Launch cluster node modules
404
Establish communications among cluster node modules
406
Attempt to connect each cluster node module to a kernel module
408
Shut down cluster node modules unconnected to a kernel module
410
Assign processor identification numbers to remaining cluster node modules
412
Initialize message passing support on kernel modules
414
Enter loop to accept user entry

FIG. 4

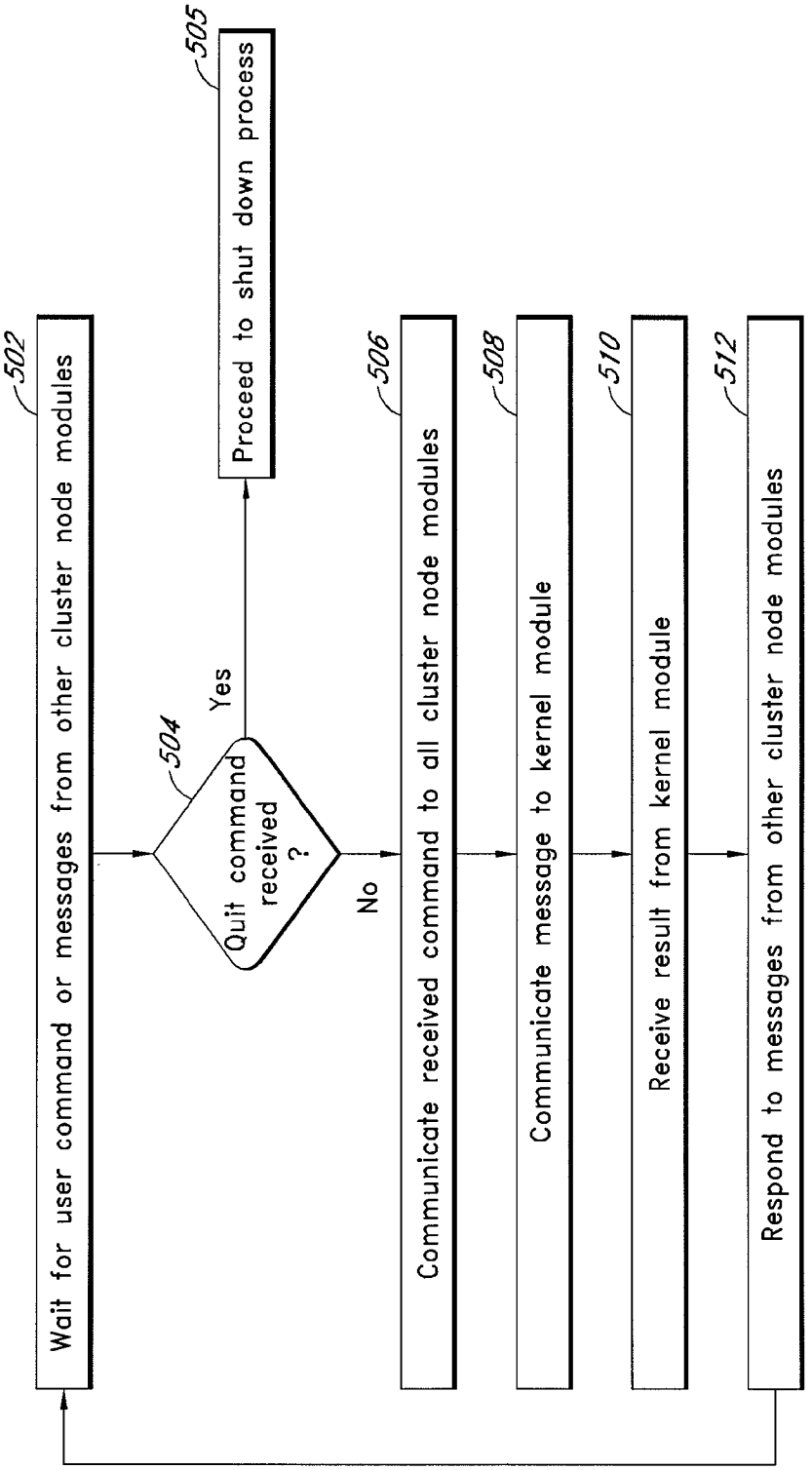

502
Wait for user command or messages from other cluster node modules
504
Quit command received ?
Yes
505
Proceed to shut down process
No
506
Communicate received command to all cluster node modules
508
Communicate message to kernel module
510
Receive result from kernel module
512
Respond to messages from other cluster node modules

FIG. 5

CLUSTER COMPUTING USING SPECIAL PURPOSE MICROPROCESSORS

RELATED APPLICATIONS

This application is a continuation of U.S. patent application Ser. No. 12/040,519, filed Feb. 29, 2008 now U.S. Pat. No. 8,140,612, which is a continuation-in-part of U.S. patent application Ser. No. 11/744,461, filed May 4, 2007, now U.S. Pat. No. 8,082,289, which claims the benefit under 35 U.S.C. §119(e) of U.S. Provisional Patent Application No. 60/813,738, filed Jun. 13, 2006, and U.S. Provisional Patent Application No. 60/850,908, filed Oct. 11, 2006. The entire contents of each of the above-referenced applications are incorporated by reference herein and made a part of this specification.

BACKGROUND

1. Field

The present disclosure relates to the field of cluster computing generally and to systems and methods for adding cluster computing functionality to a computer program, in particular.

2. Description of Related Art

Computer clusters include a group of two or more computers, microprocessors, and/or processor cores ("nodes") that intercommunicate so that the nodes can accomplish a task as though they were a single computer. Many computer application programs are not currently designed to benefit from advantages that computer clusters can offer, even though they may be running on a group of nodes that could act as a cluster. Some computer programs can run on only a single node because, for example, they are coded to perform tasks serially or because they are designed to recognize or send instructions to only a single node.

Some application programs include an interpreter that executes instructions provided to the program by a user, a script, or another source. Such an interpreter is sometimes called a "kernel" because, for example, the interpreter can manage at least some hardware resources of a computer system and/or can manage communications between those resources and software (for example, the provided instructions, which can include a high-level programming language). Some software programs include a kernel that is designed to communicate with a single node. An example of a software package that includes a kernel that is designed to communicate with a single node is Mathematica® from Wolfram Research, Inc. ("Mathematica"). Mathematics software packages from other vendors and other types of software can also include such a kernel.

A product known as gridMathematica, also from Wolfram Research, Inc., gives Mathematica the capability to perform a form of grid computing known as "distributed computing." Grid computers include a plurality of nodes that generally do not communicate with one another as peers. Distributed computing can be optimized for workloads that consist of many independent jobs or packets of work, which do not need to share data between the jobs during the computational process. Grid computers include at least one node known as a master node that manages a plurality of slave nodes or computational nodes. In gridMathematica, each of a plurality of kernels runs on a single node. One kernel is designated the master kernel, which handles all input, output, and scheduling of the other kernels (the computational kernels or slave kernels). Computational kernels receive commands and data only from the node running the master kernel. Each computational kernel performs its work independently of the other computational kernels and intermediate results of one job do not affect other jobs in progress on other nodes.

SUMMARY

Embodiments described herein have several features, no single one of which is solely responsible for their desirable attributes. Without limiting the scope of the invention as expressed by the claims, some of the advantageous features will now be discussed briefly.

Some embodiments described herein provide techniques for conveniently adding cluster computing functionality to a computer application. In one embodiment, a user of a software package may be able to achieve higher performance and/or higher availability from the software package by enabling the software to benefit from a plurality of nodes in a cluster. One embodiment allows a user to create applications, using a high-level language such as Mathematica, that are able to run on a computer cluster having supercomputer-like performance. One embodiment provides access to such high-performance computing through a Mathematica Front End, a command line interface, one or more high-level commands, or a programming language such as C or FORTRAN.

One embodiment adapts a software module designed to run on a single node, such as, for example, the Mathematica kernel, to support cluster computing, even when the software module is not designed to provide such support. One embodiment provides parallelization for an application program, even if no access to the program's source code is available. One embodiment adds and supports Message Passing Interface ("MPI") calls directly from within a user interface, such as, for example, the Mathematica programming environment. In one embodiment, MPI calls are added to or made available from an interactive programming environment, such as the Mathematica Front End.

One embodiment provides a computer cluster including a first processor, a second processor, and a third processor. The cluster includes at least one computer-readable medium in communication at least one of the first processor, the second processor, or the third processor. A first kernel resides in the at least one computer-readable medium and is configured to translate commands into code for execution on the first processor. A first cluster node module resides in the at least one computer-readable medium. The first cluster node module is configured to send commands to the first kernel and receives commands from a user interface. A second kernel resides in the at least one computer-readable medium. The second kernel is configured to translate commands into code for execution on the second processor. A second cluster node module resides in the at least one computer-readable medium. The second cluster node module is configured to send commands to the second kernel and communicates with the first cluster node module. A third kernel resides in the at least one computer-readable medium. The third kernel is configured to translate commands into code for execution on the third processor. A third cluster node module resides in the at least one computer-readable medium. The third cluster node module is configured to send commands to the third kernel and configured to communicate with the first cluster node module and the second cluster node module. The first cluster node module comprises a data structure in which messages originating from the second and third cluster node modules are stored.

Another embodiment provides a computer cluster that includes a plurality of nodes and a software package including a user interface and a single-node kernel for interpreting program code instructions. A cluster node module is config-

ured to communicate with the single-node kernel and other cluster node modules. The cluster node module accepts instructions from the user interface and interprets at least some of the instructions such that several cluster node modules in communication with one another act as a cluster. The cluster node module appears as a single-node kernel to the user interface. In one embodiment, the single-node kernel includes a Mathematical kernel. In some embodiments, the user interface can include at least one of a Mathematica front end or a command line. In some embodiments, the cluster node module includes a toolkit including library calls that implement at least a portion of MPI calls. In some embodiments, the cluster node module includes a toolkit including high-level cluster computing commands. In one embodiment, the cluster system can include a plurality of Macintosh® computers ("Macs"), Windows®-based personal computers ("PCs"), and/or Unix/Linux-based workstations.

A further embodiment provides a computer cluster including a plurality of nodes. Each node is configured to access a computer-readable medium comprising program code for a user interface and program code for a single-node kernel module configured to interpret user instructions. The cluster includes a plurality of cluster node modules. Each cluster node module is configured to communicate with a single-node kernel and with one or more other cluster node modules, to accept instructions from the user interface, and to interpret at least some of the user instructions such that the plurality of cluster node modules communicate with one another in order to act as a cluster. A communications network connects the nodes. One of the plurality of cluster node modules returns a result to the user interface.

Another embodiment provides a method of evaluating a command on a computer cluster. A command from at least one of a user interface or a script is communicated to one or more cluster node modules within the computer cluster. Each of the one or more cluster node modules communicates a message based on the command to a respective kernel module associated with the cluster node module. Each of the one or more cluster node modules receives a result from the respective kernel module associated with the cluster node module. At least one of the one or more cluster node modules responds to messages from other cluster node modules.

Another embodiment provides a computing system for executing Mathematica code on multiple nodes. The computing system includes a first node module in communication with a first Mathematica kernel executing on a first node, a second node module in communication with a second Mathematica kernel executing on a second node, and a third node module in communication with a third Mathematica kernel executing on a third node. The first node module, the second node module, and the third node module are configured to communicate with one another using a peer-to-peer architecture. In some embodiments, each of the first node module, the second node module, and third node module includes a data structure for maintaining messages originating from other node modules and a data structure for maintaining data specifying a location to which an message is expected to be received and an identifier for a node from which the message is expected to be sent.

BRIEF DESCRIPTION OF THE DRAWINGS

A general architecture that implements the various features are described with reference to the drawings. The drawings and the associated descriptions are provided to illustrate embodiments and not to limit the scope of the disclosure. Throughout the drawings, reference numbers are re-used to indicate correspondence between referenced elements.

FIG. **1** is a block diagram of one embodiment of a computer cluster.

FIG. **2** is a block diagram showing relationships between software modules running on one embodiment of a computer cluster.

FIG. **3** is a block diagram of one embodiment of a cluster node module.

FIG. **4** is a flow chart showing one embodiment of a cluster initialization process.

FIG. **5** is a flow chart showing one embodiment of the operation of a cluster node module.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

For purposes of illustration, some embodiments are described herein in the context of cluster computing with Mathematica software. The present disclosure is not limited to a single software program; the systems and methods can be used with other application software such as, for example, Maple®, MATLAB®, MathCAD®, Apple Shake®, Apple® Compressor, IDL®, other applications employing an interpreter or a kernel, Microsoft Excel®, Adobe After Effects®, Adobe Premiere®, Adobe Photoshop®, Apple Final Cut Pro®, and Apple iMovie®. Some figures and/or descriptions, however, relate to embodiments of computer clusters running Mathematica. The system can include a variety of uses, including but not limited to students, educators, scientists, engineers, mathematicians, researchers, and technicians. It is also recognized that in other embodiments, the systems and methods can be implemented as a single module and/or implemented in conjunction with a variety of other modules. Moreover, the specific implementations described herein are set forth in order to illustrate, and not to limit, the disclosure.

I. Overview

The cluster computing system described herein generally includes one or more computer systems connected to one another via a communications network or networks. The communications network can include one or more of a local area network ("LAN"), a wide area network ("WAN"), an intranet, the Internet, etc. In one embodiment, a computer system comprises one or more processors such as, for example, a microprocessor that can include one or more processing cores ("nodes"). The term "node" refers to a processing unit or subunit that is capable of single-threaded execution of code. The processors can be connected to one or more memory devices such as, for example, random access memory ("RAM"), and/or one or more optional storage devices such as, for example, a hard disk. Communications among the processors and such other devices may occur, for example, via one or more local buses of a computer system or via a LAN, a WAN, a storage area network ("SAN"), and/or any other communications network capable of carrying signals among computer system components. In one embodiment, one or more software modules such as kernels, run on nodes within the interconnected computer systems. In one embodiment, the kernels are designed to run on only a single node. In one embodiment, cluster node modules communicate with the kernels and with each other in order to implement cluster computing functionality.

FIG. **1** is a block diagram of one embodiment of a computer cluster **100** wherein computer systems **110**, **120**, **130** communicate with one another via a communications network

102. Network 102 includes one or more of a LAN, a WAN, a wireless network, an intranet, or the Internet. In one embodiment of the computer cluster, computer system 110 includes processors 112*a*, 112*b*, memory 114, and optional storage 116. Other computer systems 120, 130 can include similar devices, which generally communicate with one another within a computer system over a local communications architecture such as a local bus (not shown). A computer system can include one or more processors, and each processor can contain one or more processor cores that are capable of single-threaded execution. Processor cores are generally independent microprocessors, but more than one can be included in a single chip package. Software code designed for single-threaded execution can generally run on one processor core at a time. For example, single-threaded software code typically does not benefit from multiple processor cores in a computer system.

FIG. 2 is a block diagram showing relationships among software modules running on one embodiment of a computer cluster 100. In the embodiment shown in FIG. 2, the kernel modules 206*a-e* are designed for single-threaded execution. For example, if each of the processors 112*a*, 112*b*, 122*a*, 122*b*, 132 shown in FIG. 1 includes only one processor core, two kernel modules (for example, kernel modules 206*a*, 206*b*) loaded into the memory 114 of computer system 110 could exploit at least some of the processing bandwidth of the two processors 112*a*, 112*b*. Similarly, two kernel modules 206*c*, 206*d* loaded into the memory 124 of computer system 120 could exploit at least some of the processing bandwidth of the two processors 122*a*, 122*b*. Likewise, the bandwidth of processor 132 of computer system 130 could be utilized by a single instance of a cluster node module 204*e* loaded into the computer system's memory 134.

In the embodiment shown in FIG. 2, each of the kernel modules 206*a-e* is in communication with a single cluster node module 204*a-e*, respectively. For example, the kernel module 206*a* is in communication with the cluster node module 204*a*, the kernel module 206*b* is in communication with the cluster node module 206*b*, and so forth. In one embodiment, one instance of a cluster node module 204*a-e* is loaded into a computer system's memory 114, 124, 134 for every instance of a kernel module 206*a-e* running on the system. As shown in FIG. 2, each of the cluster node modules 204*a-e* is in communication with each of the other cluster node modules 204*a-e*. For example, one cluster node module 204*a* is in communication with all of the other cluster node modules 204*b-e*. A cluster node module 204*a* may communicate with another cluster node module 204*b* via a local bus (not shown) when, for example, both cluster node modules 204*a-b* execute on processors 112*a*, 112*b* within the same computer system 110. A cluster node module 204*a* may also communicate with another cluster node module 204*c* over a communications network 102 when, for example, the cluster node modules 204*a, c* execute on processors 112*a*, 122*a* within different computer systems 110, 120.

As shown in FIG. 2, an optional user interface module 202 such as, for example, a Mathematica front end and/or a command line interface, can connect to a cluster node module 204*a*. The user interface module can run on the same computer system 110 and/or the same microprocessor 112*a* on which the cluster node module 204*a* runs. The cluster node modules 204*a-e* provide MPI calls and/or advanced cluster functions that implement cluster computing capability for the single-threaded kernel modules. The cluster node modules 204*a-e* are configured to look and behave like a kernel module 206*a* from the perspective of the user interface module 202. Similarly, the cluster node modules 204*a-e* are configured to look and behave like a user interface module 202 from the perspective of a kernel module 206*a*. The first cluster node module 204*a* is in communication with one or more other cluster node modules 204*b*, 204*c*, and so forth, each of which provides a set of MPI calls and/or advanced cluster commands. In one embodiment, MPI may be used to send messages between nodes in a computer cluster.

Communications can occur between any two or more cluster node modules (for example, between a cluster node module 204*a* and another cluster node module 204*c*) and not just between "adjacent" kernels. Each of the cluster node modules 204*a-e* is in communication with respective kernel modules 206*a-e*. Thus, the cluster node module 204*a* communicates with the kernel module 206*a*. MPI calls and advanced cluster commands are used to parallelize program code received from an optional user interface module 208 and distribute tasks among the kernel modules 206*a-e*. The cluster node modules 204*a-e* provide communications among kernel modules 206*a-e* while the tasks are executing. Results of evaluations performed by kernel modules 206*a-e* are communicated back to the first cluster node module 204*a* via the cluster node modules 204*a-e*, which communicates them to the user interface module 208.

Intercommunication among kernel modules 206*a-e* during thread execution, which is made possible by cluster node modules 204*a-e*, provides advantages for addressing various types of mathematic and scientific problems, for example. Intercommunication provided by cluster computing permits exchange of information between nodes during the course of a parallel computation. Embodiments of the present disclosure provide such intercommunication for software programs such as Mathematica, while grid computing solutions can implement communication between only one master node and many slave nodes. Grid computing does not provide for communication between slave nodes during thread execution.

For purposes of providing an overview of some embodiments, certain aspects, advantages, benefits, and novel features of the invention are described herein. It is to be understood that not necessarily all such advantages or benefits can be achieved in accordance with any particular embodiment of the invention. Thus, for example, those skilled in the art will recognize that the invention can be embodied or carried out in a manner that achieves one advantage or group of advantages as taught herein without necessarily achieving other advantages or benefits as can be taught or suggested herein.

II. Computer Cluster 100

As shown in FIG. 1, one embodiment of a cluster system 100 includes computer systems 110, 120, 130 in communication with one another via a communications network 102. A first computer system 110 can include one or more processors 112*a-b*, a memory device 114, and an optional storage device 116. Similarly, a second computer system 120 can include one or more processors 122*a-b*, a memory device 124, and an optional storage device 126. Likewise, a third computer system 130 can include one or more processors 132, a memory device 134, and an optional storage device 136. Each of the computer systems 110, 120, 130 includes a network interface (not shown) for connecting to a communications network 102, which can include one or more of a LAN, a WAN, an intranet, a wireless network, and/or the Internet.

A. Computer System 110

In one embodiment, a first computer system 110 communicates with other computer systems 120, 130 via a network 102 as part of a computer cluster 100. In one embodiment, the

computer system **110** is a personal computer, a workstation, a server, or a blade including one or more processors **112***a-b*, a memory device **114**, an optional storage device **116**, as well as a network interface module (not shown) for communications with the network **102**.

1. Processors **112***a-b*

In one embodiment, the computer system **110** includes one or more processors **112***a-b*. The processors **112***a-b* can be one or more general purpose single-core or multi-core microprocessors such as, for example, a Pentium® processor, a Pentium® II processor, a Pentium® Pro processor, a Pentium® III processor, Pentium® 4 processor, a Core Duo® processor, a Core 2 Duo® processor, a Xeon® processor, an Itanium® processor, a Pentium® M processor, an x86 processor, an Athlon® processor, an 8051 processor, a MIPS® processor, a PowerPC® processor, an ALPHA® processor, etc. In addition, one or more of the processors **112***a-b* can be a special purpose microprocessor such as a digital signal processor. The total number of processing cores (for example, processing units capable of single-threaded execution) within all processors **112***a-b* in the computer system **110** corresponds to the number of nodes available in the computer system **110**. For example, if the processors **112***a-b* were each Core 2 Duo® processors having two processing cores, computer system **110** would have four nodes in all. Each node can run one or more instances of a program module, such as a single-threaded kernel module.

2. Network Interface Module

The computer system **110** can also include a network interface module (not shown) that facilitates communication between the computer system **110** and other computer systems **120**, **130** via the communications network **102**.

The network interface module can use a variety of network protocols. In one embodiment, the network interface module includes TCP/IP. However, it is to be appreciated that other types of network communication protocols such as, for example, Point-to-Point Protocol ("PPP"), Server Message Block ("SMB"), Serial Line Internet Protocol ("SLIP"), tunneling PPP, AppleTalk, etc., may also be used.

3. Memory **114** and Storage **116**

The computer system **110** can include memory **114**. Memory **114** can include, for example, processor cache memory (such as processor core-specific or cache memory shared by multiple processor cores), dynamic random-access memory ("DRAM"), static random-access memory ("SRAM"), or any other type of memory device capable of storing computer data, instructions, or program code. The computer system **110** can also include optional storage **116**. Storage **116** can include, for example, one or more hard disk drives, floppy disks, flash memory, magnetic storage media, CD-ROMs, DVDs, optical storage media, or any other type of storage device capable of storing computer data, instructions, and program code.

4. Computer System **110** Information

The computer system **110** may be used in connection with various operating systems such as: Microsoft® Windows® 3.X, Windows 95®, Windows 98®, Windows NT®, Windows 2000®, Windows XP®, Windows CE®, Palm Pilot OS, OS/2, Apple® MacOS®, MacOS X®, MacOS X Server®, Disk Operating System (DOS), UNIX, Linux®, VxWorks, or IBM® OS/2®, Sun OS, Solaris OS, IRIX OS operating systems, etc.

In one embodiment, the computer system **110** is a personal computer, a laptop computer, a Blackberry® device, a portable computing device, a server, a computer workstation, a local area network of individual computers, an interactive kiosk, a personal digital assistant, an interactive wireless communications device, a handheld computer, an embedded computing device, or the like.

As can be appreciated by one of ordinary skill in the art, the computer system **110** may include various sub-routines, procedures, definitional statements, and macros. Each of the foregoing modules are typically separately compiled and linked into a single executable program. However, it is to be appreciated by one of ordinary skill in the art that the processes that are performed by selected ones of the modules may be arbitrarily redistributed to one of the other modules, combined together in a single module, made available in a shareable dynamic link library, or partitioned in any other logical way.

B. Computer System **120**

In one embodiment, a second computer system **120** communicates with other computer systems **110**, **130** via a network **102** as part of a computer cluster **100**. In one embodiment, the computer system **120** is a personal computer, a workstation, a server, or a blade including one or more processors **122***a-b*, a memory device **124**, an optional storage device **126**, as well as a network interface module (not shown) for communications with the network **102**.

1. Processors **112***a-b*

In one embodiment, the computer system **120** includes one or more processors **122***a-b*. The processors **122***a-b* can be one or more general purpose single-core or multi-core microprocessors such as a Pentium® processor, a Pentium® II processor, a Pentium® Pro processor, a Pentium® III processor, Pentium® 4 processor, a Core Duo® processor, a Core 2 Duo® processor, a Xeon® processor, an Itanium® processor, a Pentium® M processor, an x86 processor, an Athlon® processor, an 8051 processor, a MIPS® processor, a PowerPC® processor, an ALPHA® processor, etc. In addition, the processors **122***a-b* can be any special purpose microprocessors such as a digital signal processor. The total number of processing cores (for example, processing units capable of single-threaded execution) within all processors **122***a-b* in the computer system **120** corresponds to the number of nodes available in the computer system **120**. For example, if the processors **122***a-b* were each Core 2 Duo® processors having two processing cores, computer system **120** would have four nodes in all. Each node can run one or more instances of a program module, such as a single-threaded kernel module.

2. Network Interface Module

The computer system **120** can also include a network interface module (not shown) that facilitates communication between the computer system **120** and other computer systems **110**, **130** via the communications network **102**.

The network interface module can use a variety of network protocols. In one embodiment, the network interface module includes TCP/IP. However, it is to be appreciated that other types of network communication protocols such as, for example, Point-to-Point Protocol ("PPP"), Server Message Block ("SMB"), Serial Line Internet Protocol ("SLIP"), tunneling PPP, AppleTalk, etc., may also be used.

3. Memory **124** and Storage **126**

The computer system **120** can include memory **124**. Memory **124** can include, for example, processor cache memory (such as processor core-specific or cache memory shared by multiple processor cores), dynamic random-access memory ("DRAM"), static random-access memory ("SRAM"), or any other type of memory device capable of storing computer data, instructions, or program code. The computer system **120** can also include optional storage **126**. Storage **126** can include, for example, one or more hard disk drives, floppy disks, flash memory, magnetic storage media,

CD-ROMs, DVDs, optical storage media, or any other type of storage device capable of storing computer data, instructions, and program code.

4. Computer System **120** Information

The computer system **120** may be used in connection with various operating systems such as: Microsoft® Windows® 3.X, Windows 95®, Windows 98®, Windows NT®, Windows 2000®, Windows XP®, Windows CE®, Palm Pilot OS, OS/2, Apple® MacOS®, MacOS X®, MacOS X Server®, Disk Operating System (DOS), UNIX, Linux®, VxWorks, or IBM® OS/2®, Sun OS, Solaris OS, IRIX OS operating systems, etc.

In one embodiment, the computer system **120** is a personal computer, a laptop computer, a Blackberry® device, a portable computing device, a server, a computer workstation, a local area network of individual computers, an interactive kiosk, a personal digital assistant, an interactive wireless communications device, a handheld computer, an embedded computing device, or the like.

As can be appreciated by one of ordinary skill in the art, the computer system **120** may include various sub-routines, procedures, definitional statements, and macros. Each of the foregoing modules are typically separately compiled and linked into a single executable program. However, it is to be appreciated by one of ordinary skill in the art that the processes that are performed by selected ones of the modules may be arbitrarily redistributed to one of the other modules, combined together in a single module, made available in a shareable dynamic link library, or partitioned in any other logical way.

C. Computer System **130**

In one embodiment, a third computer system **130** communicates with other computer systems **110**, **120** via a network **102** as part of a computer cluster **100**. In one embodiment, the computer system **130** is a personal computer, a workstation, a server, or a blade including one or more processors **132**, a memory device **134**, an optional storage device **136**, as well as a network interface module (not shown) for communications with the network **102**.

1. Processors **112***a-b*

In one embodiment, the computer system **130** includes a processor **132**. The processor **132** can be a general purpose single-core or multi-core microprocessors such as a Pentium® processor, a Pentium® II processor, a Pentium® Pro processor, a Pentium® III processor, Pentium® 4 processor, a Core Duo® processor, a Core 2 Duo® processor, a Xeon® processor, an Itanium® processor, a Pentium® M processor, an x86 processor, an Athlon® processor, an 8051 processor, a MIPS® processor, a PowerPC® processor, or an ALPHA® processor. In addition, the processor **132** can be any special purpose microprocessor such as a digital signal processor. The total number of processing cores (for example, processing units capable of single-threaded execution) within processor **132** in the computer system **130** corresponds to the number of nodes available in the computer system **130**. For example, if the processor **132** was a Core 2 Duo® processor having two processing cores, the computer system **130** would have two nodes. Each node can run one or more instances of a program module, such as a single-threaded kernel module.

2. Network Interface Module

The computer system **130** can also include a network interface module (not shown) that facilitates communication between the computer system **130** and other computer systems **110**, **120** via the communications network **102**.

The network interface module can use a variety of network protocols. In one embodiment, the network interface module includes TCP/IP. However, it is to be appreciated that other types of network communication protocols such as, for example, Point-to-Point Protocol ("PPP"), Server Message Block ("SMB"), Serial Line Internet Protocol ("SLIP"), tunneling PPP, AppleTalk, etc., may also be used.

3. Memory **134** and Storage **136**

The computer system **130** can include memory **134**. Memory **134** can include, for example, processor cache memory (such as processor core-specific or cache memory shared by multiple processor cores), dynamic random-access memory ("DRAM"), static random-access memory ("SRAM"), or any other type of memory device capable of storing computer data, instructions, or program code. The computer system **130** can also include optional storage **136**. Storage **136** can include, for example, one or more hard disk drives, floppy disks, flash memory, magnetic storage media, CD-ROMs, DVDs, optical storage media, or any other type of storage device capable of storing computer data, instructions, and program code.

4. Computer System **130** Information

The computer system **130** may be used in connection with various operating systems such as: Microsoft® Windows® 3.X, Windows 95®, Windows 98®, Windows NT®, Windows 2000®, Windows XP®, Windows CE®, Palm Pilot OS, OS/2, Apple® MacOS®, MacOS X®, MacOS X Server®, Disk Operating System (DOS), UNIX, Linux®, VxWorks, or IBM® OS/2®, Sun OS, Solaris OS, IRIX OS operating systems, etc.

In one embodiment, the computer system **130** is a personal computer, a laptop computer, a Blackberry® device, a portable computing device, a server, a computer workstation, a local area network of individual computers, an interactive kiosk, a personal digital assistant, an interactive wireless communications device, a handheld computer, an embedded computing device, or the like.

As can be appreciated by one of ordinary skill in the art, the computer system **130** may include various sub-routines, procedures, definitional statements, and macros. Each of the foregoing modules are typically separately compiled and linked into a single executable program. However, it is to be appreciated by one of ordinary skill in the art that the processes that are performed by selected ones of the modules may be arbitrarily redistributed to one of the other modules, combined together in a single module, made available in a shareable dynamic link library, or partitioned in any other logical way.

E. Communications Network **102**

In one embodiment, computer systems **110**, **120**, **130** are in communication with one another via a communications network **102**.

The communications network **102** may include one or more of any type of electronically connected group of computers including, for instance, the following networks: a virtual private network, a public Internet, a private Internet, a secure Internet, a private network, a public network, a value-added network, a wired network, a wireless network, an intranet, etc. In addition, the connectivity to the network can be, for example, a modem, Ethernet (IEEE 802.3), Gigabit Ethernet, 10-Gigabit Ethernet, Token Ring (IEEE 802.5), Fiber Distributed Datalink Interface (FDDI), Frame Relay, InfiniBand, Myrinet, Asynchronous Transfer Mode (ATM), or another interface. The communications network **102** may connect to the computer systems **110**, **120**, **130**, for example, by use of a modem or by use of a network interface card that resides in each of the systems.

In addition, the same or different communications networks **102** may be used to facilitate communication between the first computer system **110** and the second computer sys-

tem **120**, between the first computer system **110** and the third computer system **130**, and between the second computer system **120** and the third computer system **130**.

III. Software Modules

As shown in FIGS. **1** and **2**, one embodiment of a cluster system **100** includes a user interface module **202** that is able to access a plurality of kernel modules **206***a-e* by communicating with a first cluster node module **204***a*. User interface module can be stored in a memory **114**, **124**, **134** while running, for example, and/or can be stored in a storage device **116**, **126**, **136**. The first cluster node module **204***a* is in communication with each of the other cluster node modules **204***b-e*. The kernel modules **206***a-e* can reside in the memory of one or more computer systems on which they run. For example, the memory **114** of the first computer system **110** can store instances of kernel modules **206***a-b*, the memory **124** of the second computer system **120** can store instances of kernel modules **206***c-d*, and the memory **134** of the third computer system **130** can store an instance of kernel module **206***e*. The kernel modules **206***a-e*, which include single-threaded program code, are each associated with one of the processors **112***a*, **112***b*, **122***a*, **122***b*, **132**. A cluster configuration module stored on one or more of the computer systems **110**, **120**, **130** or on a remote computer system, for example, can establish communication with the cluster node modules **204***a-e*. In one embodiment, communication between the cluster configuration module **208** and the cluster node modules **204***a-e* initializes the cluster node modules **204***a-e* to provide cluster computing support for the computer cluster **100**.

A. Cluster Node Module **204**

In one embodiment, the cluster node modules **204***a-e* provide a way for many kernel modules **206***a-e* such as, for example, Mathematica kernels, running on a computer cluster **100** to communicate with one another. A cluster node module **204** can include at least a portion of an application programming interface ("API") known as the Message-Passing Interface ("MPI"), which is used in several supercomputer and cluster installations. A network of connections (for example, the arrows shown in FIG. **2**) between the cluster node modules **204***a-e* can be implemented using a communications network **102**, such as, for example, TCP/IP over Ethernet, but the connections could also occur over any other type of network or local computer bus.

A cluster node module **204** can use an application-specific toolkit or interface such as, for example, Mathematica's MathLink, Add-Ons, or packets, to interact with an application. Normally used to connect a Mathematica kernel to a user interface known as the Mathematica Front End or other Mathematica kernels, MathLink is a bidirectional protocol to sends "packets" containing messages, commands, or data between any of these entities. MathLink does not allow direct cluster computing-like simultaneous communication between Mathematica kernels during execution of a command or thread. MathLink is also not designed to perform multiple simultaneous network connections. In some embodiments, a cluster node module **204** can use an application-specific toolkit such as, for example, MathLink, for connections between entities on the same computer.

When speaking about procedures or actions on a cluster or other parallel computer, not all actions happen in sequential order, nor are they required to. For example, a parallel code, as opposed to a single-processor code of the classic "Turing machine" model, has multiple copies of the parallel code running across the cluster, typically one for each processor (or "processing element" or "core"). Such parallel code is written in such a way that different instances of the same code can communicate, collaborate, and coordinate work with each other. Multiple instances of these codes can run at the same time in parallel.

If the count of the code instances is an integer N, each instance of code execution can be labeled 0 through N−1. For example, a computer cluster can include N connected computers, each containing a processor. The first has cluster node module 0 connected with kernel module 0 running on processor 0. The next is cluster node module 1 and kernel module 1, on processor 1, and so forth for each of the N connected computers. Some steps of their procedure are collaborative, and some steps are independent. Even though these entities are not necessarily in lock-step, they do follow a pattern of initialization, main loop behavior (for example, cluster node module operation), and shut down.

In contrast, a parallel computing toolkit (PCT) that is provided as part of the gridMathematica software package does not provide a means for instances of the same code running on different nodes to communicate, collaborate, or coordinate work among the instances. The PCT provides commands that connect Mathematica kernels in a master-slave relationship rather than a peer-to-peer relationship as enabled by some embodiments disclosed herein. A computer cluster having peer-to-peer node architecture performs computations that can be more efficient, easier to design, and/or more reliable than similar computations performed on grid computers having master-slave node architecture. Moreover, the nature of some computations may not allow a programmer to harness multi-node processing power on systems that employ master-slave node architecture.

FIG. **3** shows one embodiment of a cluster node module **204** implementing MPI calls and advanced MPI functions. In the embodiment shown in FIG. **3**, cluster node module **204** includes MPI module **302**, advanced functions module **304**, received message queue **306**, and message receiving queue **308**.

1. MPI Module **302**

In one embodiment, the cluster node module **204** includes an MPI module **302**. The MPI module **302** can include program code for one or more of at least five kinds of MPI instructions or calls. Selected constants, instructions, and/or calls that can be implemented by the MPI module **302** are as follows:

MPI Constants

Node identifiers are used to send messages to nodes or receive messages from them. In MPI, this is accomplished by assigning each node a unique integer ($IdProc) starting with 0. This data, with a knowledge of the total count ($NProc), makes it possible to programmatically divide any measurable entity.

TABLE A

Constant	Description
$IdProc	The identification number of the current processor
$NProc	The number of processors in the current cluster
$mpiCommWorld	The communicator world of the entire cluster (see MPI Communicator routines, below)
mpiCommWorld	The default communicator world for the high-level routines.

Basic MPI Calls

In one embodiment, the MPI module **302** can include basic MPI calls such as, for example, relatively low-level routines that map MPI calls that are commonly used in other languages

(such as C and Fortran), so that such calls can be available directly from the Mathematica user interface **204**. In some embodiments, basic MPI calls include calls that send data, equations, formulas, and/or other expressions.

Simply sending expressions from one node to another is possible with these most basic MPI calls. One node can call to send an expression while the other calls a corresponding routine to receive the sent expression. Because it is possible that the receiver has not yet called mpiRecv even if the message has left the sending node, completion of mpiSend is not a confirmation that it has been received.

TABLE B

Call	Description
mpiSend[expr, target, comm, tag]	Sends an expression expr to a node with the ID target in the communicator would comm, waiting until that expression has left this kernel
mpiRecv [expr, target, comm, tag]	Receives an expression into expr from a node with the ID target in the communicator world comm, waiting until the expression has arrived
mpiSendRecv[sendexpr, dest, recvexpr, source, comm]	Simultaneously sends the expression sendexpr to the node with the ID target and receives an expression into recvexpr from the node with the ID source in the communicator world comm, waiting until both operations have returned.

Asynchronous MPI Calls

Asynchronous calls make it possible for the kernel to do work while communications are proceeding simultaneously. It is also possible that another node may not be able to send or receive data yet, allowing one kernel to continue working while waiting.

TABLE C

Call	Description
mpiISend[expr, target, comm, tag, req]	Sends an expression expr to a processor with the ID target in the communicator world comm, returning immediately. It can be balanced with calls to mpiTest[req] until mpiTest[req] returns True.
mpiIRecv[expr, target, comm, tag, req]	Receives an expression expr from a proessor with the ID target in the communicator world comm, returning immediately. It can be balanced with calls to mpiTest[req] until mpiTest[req] returns True. The expr is not safe to access until mpiTest[req] returns True.
mpiTest[req]	Completes asynchronous behavior of mpiISend and mpiIRecv
mpWait[req]	Calls mpiTest until it returns True.
mpiWaitall[reglist]	Calls mpiWait all on every element of reqlist
mpiWaitany[reqlist]	Calls mpiTest on each element of realist until one of them returns True

The mpiISend[ ] command can be called from within a kernel module **206** (for example, a Mathematica kernel). It creates a packet containing the Mathematica expression to be sent as payload and where the expression should be sent. The packet itself is destined only for its local cluster node module. Once received by its local cluster node module, this packet is decoded and its payload is forwarded on to the cluster node module specified in the packet.

The mpiIRecv[ ] command can also be called from within a kernel module **206**. It creates a packet specifying where it expects to receive an expression and from which processor this expression is expected. Once received by its local cluster node module, this packet is decoded and its contents are stored in a message receiving queue (MRQ) **308** (FIG. **3**).

The mpiTest[ ] command can be called from within a kernel module **206**. It creates a packet specifying which message to test for completion, then waits for a reply expression to evaluate. Once received by the kernel module's associated cluster node module **204**, this packet is decoded and its message specifier is used to search for any matching expressions listed as completed in its received message queue (RMQ) **306**. If such completed expressions are found, it is sent to its local kernel module as part of the reply in mpiTest[ ]. The kernel module receives this reply expression and evaluates it, which updates the kernel module's variables as needed.

Other MPI calls are built on the fundamental calls mpiISend, mpiIRecv, and mpiTest. For example, mpiBcast, a broadcast, creates instructions to send information from the broadcast processor to all the others, while the other processors perform a Recv. Similarly, high-level calls of the toolkit can be built on top of the collection of MPI calls.

Collective MPI Calls

In one embodiment, the MPI module **302** can include program code for implementing collective MPI calls (for example, calls that provide basic multi-node data movement across nodes). Collective MPI calls can include broadcasts, gathers, transpose, and other vector and matrix operations, for example. Collective calls can also provide commonly used mechanisms to send expressions between groups of nodes.

TABLE D

Call	Description
mpiBcast[expr, root, comm]	Performs a broadcast of expr from the root processor to all the others in the communicator world comm. An expression is expected to be supplied by the root processor, while all the others expect expr to be overwritten by the incoming expression.
mpiGather[sendexpr, recvexpr, root, comm]	All processors (including root) in the communicator comm send their expression in sendexpr to the root processor, which produces a list of these expressions, in the order according to comm, in recvexpr. On the processors that are not root, recvexpr is ignored.
mpiAllgather[sendexpr, recvexpr, comm]	All processors in the communicator comm send their expression in sendexpr, which are organized into a list of these expressions, in the order according to comm, in recvexpr on all processors in comm.
mpiScatter[sendexpr, recvexpr, root, comm]	Processor root partitions the list in sendexpr into equal parts (if possible) and places each piece in recvexpr on all the processors (including root) in the communicator world comm, according the order and size of comm.
mpiAlltoall[sendexpr, recvexpr, comm]	Each processor sends equal parts of the list in sendexpr to all other processors in the communicator world comm, which each collects from all other processors are organizes into the order according to comm.

In one embodiment, the MPI module **302** includes program code for implementing parallel sums and other reduction operations on data stored across many nodes. MPI module **302** can also include program code for implementing simple parallel input/output calls (for example, calls that allow cluster system **200** to load and store objects that are located on a plurality of nodes).

TABLE E

Call	Description
mpiReduce[sendexpr, recvexpr, operation, root, comm]	Performs a collective reduction operation between expressions on all processors in the communicator world comm for every element in the list in sendexpr returning the resulting list in recvexpr on the processor with the ID root.
mpiAllreduce[sendexpr, recvexpr, operation, comm]	Performs a collective reduction operation between expressions on all processors in the communicator world comm for every element in the list in sendexpr returning the resulting list in recvexpr on every processor.
mpiReduceScatter [sendexpr, recvexpr, operation, comm]	Performs a collective reduction operation between expressions on all processors in the communicator world comm for every element in the list in sendexpr, partitioning the resulting list into pieces for each processor's recvexpr.

These additional collective calls perform operations that reduce the data in parallel. The operation argument can be one of the constants below.

TABLE F

Constant	Description
mpiSum	Specifies that all the elements on different processors be added together in a reduction call
mpiMax	Specifies that the maximum of all the elements on different processors be chosen in a reduction call
mpiMin	Specifies that the minimum of all the elements on different processors be chosen in a reduction call

MPI Communicator Calls

In one embodiment, the MPI module **302** includes program code for implementing communicator world calls (for example, calls that would allow subsets of nodes to operate as if they were a sub-cluster). Communicators organize groups of nodes into user-defined subsets. The communicator values returned by mpiCommSplit[ ] can be used in other MPI calls instead of mpiCommWorld.

TABLE G

Call	Description
mpiCommSize[comm]	Returns the number of processors within the communicator comm
mpiCommRank[comm]	Returns the rank of this processor in the communicator comm
mpiCommDup[comm]	Returns a duplicate communicator of the communicator comm
mpiCommSplit[comm, color, key]	Creates a new communicator into several disjoint subsets each identified by color. The sort order within each subset is first by key, second according to the ordering in the previous communicator. Processors not meant to participate in any new communicator indicates this by passing the constant mpiUndefined. The corresponding communicator is returned to each calling processor.
mpiCommMap[comm] mpiCommMap[comm, target]	Returns the mapping of the communicator comm to the processor indexed according to $mpiCommWorld. Adding a second argument returns just the ID of the processor with the ID target in the communicator comm.
mpiCommFree[comm]	Frees the communicator comm

Other MPI Support Calls

Other calls that provide common functions include:

TABLE H

Call	Description
mpiWtime[ ]	Provides wall-dock time since some fixed time in the past. There is no guarantee that this time will read the same on all processors.
mpWtick[ ]	Returns the time resolution of mpiWtime[ ]
MaxByElement[in]	For every nth element of each list of the list in, chooses the maximum according to Max[ ], and returns the result as one list. Used in the mpiMax reduction operation.
MinByElement[in]	For every nth element of each list of the list in, chooses the minimum according to Min[ ], and returns the result as one list. Used in the mpiMin reduction operation.

2. Advanced Functions Module **304**

In one embodiment, the cluster node module **204** includes an advanced functions module **304**. The advanced functions module **304** can include program code that provides a toolkit of functions inconvenient or impractical to do with MPI instructions and calls implemented by the MPI module **302**. The advanced functions module **304** can rely at least partially on calls and instructions implemented by the MPI module **302** in the implementation of advanced functions. In one embodiment, the advanced functions module **304** includes a custom set of directives or functions. In an alternative embodiment, the advanced functions module **304** intercepts normal Mathematica language and converts it to one or more functions optimized for cluster execution. Such an embodiment can be easier for users familiar with Mathematica functions to use but can also complicate a program debugging process. Some functions implemented by the advanced functions module **304** can simplify operations difficult or complex to set up using parallel computing. Several examples of such functions that can be implemented by the advanced functions module **304** are shown below.

Built on the MPI calls, the calls that are described below provide commonly used communication patterns or parallel versions of Mathematica features. Unless otherwise specified, these are executed in the communicator mpiCommWorld, whose default is $mpiCommWorld, but can be changed to a valid communicator at run time.

Common Divide-and-Conquer Parallel Evaluation

In one embodiment, the advanced functions module **304** includes functions providing for basic parallelization such as, for example, routines that would perform the same operations on many data elements or inputs, stored on many nodes. These functions can be compared to parallelized for-loops and the like. The following calls address simple parallelization of common tasks. In the call descriptions, "expr" refers to an expression, and "loopspec" refers to a set of rules that determine how the expression is evaluated. In some embodiments, the advanced functions module **304** supports at least three forms of loopspec, including {var, count}, where the call iterates the variable var from 1 to the integer count; {var, start, stop}, where the call iterates the variable var every integer from start to stop; and {var, start, stop, increment}, where the call iterates the variable var from start adding increment for each iteration until var exceeds stop, allowing var to be a non-integer.

TABLE I

Call	Description
ParallelDo[expr, loopspec]	Like Do[ ] except that it evaluates expr across the cluster, rather than on just one processor. The rules for how expr is evaluated is specified in loopspec, like in Do[ ].

TABLE I-continued

Call	Description
ParallelFunctionToList[f, count]	Evaluates the function f[i] from 1 to count, but across the cluster, and returns these results in a list. The third argument has it gather this list into the processor whose ID is root.
ParallelFunctionToList[f, count, root]	
ParallelTable[expr, loopspec]	Like Table[ ] except that it evaluates expr across the cluster, rather than on just one processor, returning the locally evalulated portion. The third argument has it gather this table in to the processor whose ID is root.
ParallelTable[expr, loopspec, root]	
ParallelFunction[f, inputs, root]	Like f[inputs] except that it evaluates f on a subset of inputs scattered across the cluster from processor root and gathered back to root.
ParallelNintegrate[expr, loopspec]	Like Nintegrate[ ] except that it evaluates a numerical integration of expr over domains partitioned into the number of processors in the cluster, then returns the sum. The third argument has each numerical integration execute with at least that many digits of precision.
ParallelNintegrate[expr, loopspec, digits]	

TABLE I-continued

Call	Description

Guard-Cell Management

In one embodiment, the advanced functions module **304** includes functions providing for guard-cell operations such as, for example, routines that perform nearest-neighbor communications to maintain edges of local arrays in any number of dimensions (optimized for 1-, 2-, and/or 3-D). Typically the space of a problem is divided into partitions. Often, however, neighboring edges of each partition can interact, so a "guard cell" is inserted on both edges as a substitute for the neighboring data. Thus the space a processor sees is two elements wider than the actual space for which the processor is responsible. EdgeCell helps maintain these guard cells.

TABLE J

Call	Description
EdgeCell[list]	Copies the second element of list to the last element of the left processor and the second-to-last element of list to the first element of the right processor while simultaneously receiving the same from its neighbors.

Matrix and Vector Manipulation

The advanced functions module **304** can also include functions providing for linear algebra operations such as, for example, parallelized versions of basic linear algebra on structures partitioned on many nodes. Such linear algebra operations can reorganize data as needed to perform matrix and vector multiplication or other operations such as determinants, trace, and the like. Matrices are partitioned and stored in processors across the cluster. These calls manipulate these matrices in common ways.

TABLE K

Call	Description
ParallelTranspose[matrix]	Like Transpose[ ] except that it transposes matrix that is in fact represented across the cluster, rather than on just one processor. It returns the portion of the transposed matrix meant for that processor.
ParallelProduct[matrix, vector]	Evaluates the product of matrix and vector, as it would on one processor, except that matrix is represented across the cluster.
ParallelDimensions[matrix]	Like Dimensions[ ] except that matrix is represented across the cluster, rather than on just one processor. It returns a list of each dimension.
ParallelTr[matrix]	Like Tr[ ] except that the matrix is represented across the cluster, rather than on just one processor. It returns the trace of this matrix.
ParallelIdentity[rank]	Like Identity[ ], it generates a new identity matrix, except that the matrix is represented across the cluster, rather than on just one processor. It returns the portion of the new matrix for this processor.
ParallelOuter[f, vector1, vector2]	Like Outer[f, vector1, vector2] except that the answer becomes a matrix represented across the cluster, rather than on just one processor. It returns the portion of the new matrix for this processor.
ParallelInverse[matrix]	Like Inverse[ ] except that the matrix is represented across the cluster, rather than on just one processor. It returns the inverse of the matrix.

Element Management

In one embodiment, the advanced functions module **304** includes element management operations. For example, a large bin of elements or particles cut up in space across the nodes may need to migrate from node to node based on rules or criteria (such as their spatial coordinate). Such operations would migrate the data from one node to another. Besides the divide-and-conquer approach, a list of elements can also be partitioned in arbitrary ways. This is useful if elements need to be organized or sorted onto multiple processors. For example, particles of a system may drift out of the space of one processor into another, so their data would need to be redistributed periodically.

TABLE L

Call	Description
ElementManage[list, switch]	Selects which elements of list will be sent to which processors according to the function switch[ ] is evaluated on each element of list.

TABLE L-continued

Call	Description
ElementManage[list]	If switch is a function, switch[ ] should return the ID of the processor that element should be sent. If switch is an integer, the call assumes that each elements is itself a list, whose first element is a number ranging from 0 to the passed argument. This call returns a list of the elements, from any processor, that is switch selected for this processor. Each element of list can be a list of two elements, the first being the ID of the processor where the element should be sent, while the second is arbitrary data to send. This call returns those list elements, from any and all processors, whose first element is this processors ID in a list. This call is used internally by the two-argument version of ElementManage[ ].

Fourier Transform

In one embodiment, the advanced functions module **304** includes program code for implementing large-scale parallel fast Fourier transforms ("FFTs"). For example, such functions can perform FFTs in one, two, and/or three dimensions on large amounts of data that are not stored on one node and that are instead stored on many nodes. Fourier transforms of very large arrays can be difficult to manage, not the least of which is the memory requirements. Parallelizing the Fourier transform makes it possible to make use of all the memory available on the entire cluster, making it possible to manipulate problem sizes that no one processor could possibly do alone.

TABLE M

Call	Description
ParallelFourier[list]	Like Fourier[ ] except that list is a two- or three-dimensional list represented across the cluster, like for matrices, above. It returns the portion of the Fourier-transformed array meant for that processor.

Parallel Disk I/O

In one embodiment, the advanced functions module **304** includes parallel disk input and output calls. For example, data may need to be read in and out of the cluster in such a way that the data is distributed across the cluster evenly. The calls in the following table enable the saving data from one or more processors to storage and the retrieval data from storage.

TABLE N

Call	Description
ParallelPut[expr, filename] ParallelPut[expr, filename root] ParallelPut[expr, filename, root, comm]	Puts expr into the file with the name filename in order on processor 0. The third argument specifies that the file be written on the processor whose ID is root. The fourth uses the communicator world comm.
ParallelGet[filename] ParallelGet[filename, root] ParallelGet[filename, root, comm]	Reads and returns data from the file with the name filename on processor 0 partitioned into each processor on the cluster. The second argument specifies that the file is to be read on the processor whose ID is root. The third uses the communicator world comm.
ParallelBinaryPut[expr, type, filename] ParallelBinaryPut[expr, filename, root] ParallelBinaryPut[expr, filename, root, comm]	Puts expr into the file with the binary format type with the name filename in order on processor 0. The fourth argument specifies that the file be written on the processor whose ID is root. The fifth uses the communicator world comm.
ParallelBinaryGet[type, filename] ParallelBinaryGet[type, filename, root] ParallelBinaryGet[type, filename, root, comm]	Reads and returns data in the binary format type from the file with the name filename on processor 0 partitioned into each processor on the cluster. The third argument specifies that the file is to be read on the processor whose ID is root. The fourth uses the communicator world comm.
ParallelGetPerProcessor [expr, filename] ParallelGetPerProcessor [filename, root] ParallelGetPerProcessor [filename, root, comm]	Puts expr into the file with the name filename in order on processor 0, one line per processor. The third argument specifies that the file be written on the processor whose ID is root. The fourth uses the communicator world comm.
ParallelGetPerProcessor [filename] ParallelGetPerProcessor [filename, root] ParallelGetPerProcessr [filename, root, comm]	Reads and returns data from the file with the name filename on processor 0, one line for each processor. The second argument specifies that the file is to be read on the processor whose ID is root. The third uses the communicator world comm.

Automatic Load Balancing

Some function calls can take an inconsistent amount of processing time to complete. For example, in Mathematica, the call f[20] could in general take much longer to evaluate than f[19]. Moreover, if one or more processors within the cluster are of different speeds (for example, if some operate at a core frequency of 2.6 GHz while other operate at less than one 1 GHz), one processor may finish a task sooner than another processor.

In some embodiments, the advanced functions module **304** includes a call that can improve the operation of the computer cluster **100** in such situations. In some embodiments, the root processor assigns a small subset of the possible calls for a function to each processor on the cluster **100**. Whichever processor returns its results first is assigned a second small subset of the possible calls. The root processor will continue to assign small subsets of the possible calls as results are received until an evaluation is complete. The order in which the processors finish can vary every time an expression is evaluated, but the root processor will continue assigning additional work to processors as they become available.

In one illustrative example, there are 4 processors and f[1] to f[100] to evaluate. One could implement this by assigning f[1], f[2], f[3], f[4] to each of processors 0 (the root can assign to oneself) through 3. If the f[2] result came back first, then processor 1 would be assigned f[5]. If the f[4] result is returned next, f[6] would be assigned to processor 3. The assignments continue until all results are calculated. The results are organized for output back to the user.

In alternative embodiments, the subsets of possible calls can be assigned in any order, rather than sequentially, or in batches (for example, f[1], f[5], f[9] assigned to processor **1**, etc.). Also, the subsets could be organized by delegation. For example, one processor node may not necessarily be in direct control of the other processors. Instead, a large subset could be assigned to a processor, which would in turn assign subsets of its work to other processors. The result would create a hierarchy of assignments like a vast army.

TABLE O

LoadBalanceFunctionToList[f, count] LoadBalanceFunctionToList[f, count, root]	Evaluates the function f[i] from 1 to count, but across the cluster using load-balancing techniques, and returns these results in a list. The third argument has it gather this list into the processor whose ID is root.

3. Received Message Queue **306**

In one embodiment, the cluster node module **204** includes a received message queue **306**. The received message queue **306** includes a data structure for storing messages received from other cluster node modules. Related data pertaining to the messages received, such as whether an expression has been completed, may also be stored in the received message queue **306**. The received message queue **306** may include a queue and/or another type of data structure such as, for example, a stack, a linked list, an array, a tree, etc.

4. Message Receiving Queue **308**

In one embodiment, the cluster node module **204** includes a message receiving queue **308**. The message receiving queue **308** includes a data structure for storing information about the location to which an expression is expected to be sent and the processor from which the expression is expected. The message receiving queue **308** may include a queue and/or another type of data structure such as, for example, a stack, a linked list, an array, a tree, etc.

B. Cluster Configuration Module **208**

Cluster configuration module **208** includes program code for initializing a plurality of cluster node modules to add cluster computing support to computer systems **110**, **120**, **130**. U.S. Pat. No. 7,136,924, issued to Dauger (the "'924 patent"), the entirety of which is hereby incorporated by reference and made a part of this specification, discloses a method and system for parallel operation and control of computer clusters. One method generally includes obtaining one or more personal computers having an operating system with discoverable network services. In some embodiments, the method includes obtaining one or more processors or processor cores on which a kernel module can run. As described in the '924 patent, a cluster node control and interface (CNCI) group of software applications is copied to each node. When the CNCI applications are running on a node, the cluster configuration module **208** can permit a cluster node module **204**, in combination with a kernel module **206**, to use the node's processing resources to perform a parallel computation task as part of a computer cluster. The cluster configuration module **208** allows extensive automation of the cluster creation process in connection with the present disclosure.

C. User Interface Module **202**

In some embodiments, computer cluster **100** includes a user interface module **202**, such as, for example a Mathematica Front End or a command line interface, that includes program code for a kernel module **206** to provide graphical output, accept graphical input, and provide other methods of user communication that a graphical user interface or a command-line interface provides. To support a user interface module **202**, the behavior of a cluster node module **204***a* is altered in some embodiments. Rather than sending output to and accepting input from the user directly, the user interface module **202** activates the cluster node module **204***a* to which it is connected and specifies parameters to form a connection, such as a MathLink connection, between the cluster node module **204***a* and the user interface module **202**. The user interface module's activation of the cluster node module **204***a* can initiate the execution of instructions to activate the remaining cluster node modules **204***b-e* on the cluster and to complete the sequence to start all kernel modules **206***a-e* on the cluster. Packets from the user interface module **202**, normally intended for a kernel module **206***a*, are accepted by the cluster node module **204***a* as a user command. Output from the kernel module **206***a* associated with the cluster node module **204***a* can be forwarded back to the user interface module **202** for display to a user. Any of the cluster node modules **204***a-e* can be configured to communicate with a user interface module **202**.

D. Kernel Module **206**

A kernel module **206** typically includes program code for interpreting high-level code, commands, and/or instructions supplied by a user or a script into low-level code, such as, for example, machine language or assembly language. In one embodiment, each cluster node module **204***a-e* is connected to all other cluster node modules, while each kernel module **206***a-e* is allocated and connected only to one cluster node module **204**. In one embodiment, there is one cluster node module-kernel module pair per processor. For example, in an embodiment of a computer cluster **100** including single-processor computer systems, each cluster node module-kernel module pair could reside on a single-processor computer. If a computer contains multiple processors or processing cores, it may contain multiple cluster node module-kernel module pairs, but the pairs can still communicate over the cluster node module's network connections.

IV. Cluster Computing Methods

In one embodiment, the computer cluster **100** includes a cluster initialization process, a method of cluster node module operation, and a cluster shut down process.

A. Cluster Initialization Process

In one embodiment, a cluster configuration module **202** initializes one or more cluster node modules **204** in order to provide cluster computing support to one or more kernel modules **206**, as shown in FIG. **4**.

At **402**, cluster node modules are launched on the computer cluster **100**. In one embodiment, the cluster node module **204***a* running on a first processor **112***a* (for example, where the user is located) accesses the other processors **112***b*, **122***a-b*, **132** on the computer cluster **100** via the cluster configuration module **208** to launch cluster node modules **204***b-e* onto the entire cluster. In an alternative embodiment, the cluster configuration module **208** searches for processors **112***a-b*, **122***a-b*, **132** connected to one another via communications network **102** and launches cluster node modules **204***a-e* on each of the processors **112***a-b*, **122***a-b*, **132**.

The cluster node modules **204***a-e* establish communication with one another at **404**. In one embodiment, each of the cluster node modules **204***a-e* establish direct connections using the MPI_Init command with other cluster node modules **204***a-e* launched on the computer cluster **100** by the cluster configuration module **208**.

At **406**, each cluster node module **204** attempts to connect to a kernel module **206**. In one embodiment, each instance of the cluster node modules **204***a-e* locates, launches, and connects with a local kernel module via MathLink connections and/or similar connection tools, for example, built into the kernel module **206**.

At **408**, the cluster node modules **204** that are unconnected to a kernel module **206** are shut down. In one embodiment, each cluster node module **204** determines whether the local kernel module cannot be found or connected to. In one embodiment, each cluster node module **204** reports the failure to connect to a kernel module **206** to the other cluster node modules on computer cluster **100** and quits.

Processor identification numbers are assigned to the remaining cluster node modules **204** at **410**. In one embodiment, each remaining cluster node module **204** calculates the total number of active processors (N) and determines identification numbers describing the remaining subset of active cluster node modules **204***a-e* and kernel modules **206***a-e*. This new set of cluster node module-kernel module pairs may be numbered 0 through N−1, for example.

Message passing support is initialized on the kernel modules **206***a-e* at **412**. In one embodiment, each cluster node module **204** supplies initialization code (for example, Mathematica initialization code) to the local kernel module **206** to support message passing.

Finally, at **414**, the cluster node modules **204***a-e* enter a loop to accept user entry. In one embodiment, a main loop (for example, a cluster operation loop) begins execution after the cluster node module **204***a* on the first processor **112***a* returns to user control while each of the other cluster node modules **204** waits for messages from all other cluster node modules **204***a-e* connected to the network **102**.

The initialization process creates a structure enabling a way for the kernel modules **206***a-e* to send messages to one another. In some embodiments, any kernel module can send data to and receive data from any other kernel module within the cluster when initialization is complete. The cluster node module creates an illusion that a kernel module is communicating directly with the other kernel modules. The initialization process can create a relationship among kernel modules on a computer cluster **100** such as the one shown by way of example in FIG. **2**.

B. Cluster Node Module Operation

In one embodiment, a cluster node module **204** implements cluster computing support for a kernel module **206** during a main loop, as shown in FIG. **5**.

At **502**, cluster node modules **204** wait for user commands or messages from other cluster node modules. In one embodiment, the cluster node module **204***a* connected to the user interface module **202** waits for a user command, while the other cluster node modules **204***b-e* continue checking for messages.

Once a command or message is received, the method proceeds to **504**. At **504**, the cluster node module **204***a* determines whether the message received is a quit command. If a quit command is received, the cluster node module **204***a* exits the loop and proceeds to a cluster node module shut down process at **505**. If the message received is not a quit command, the process continues to **506**.

At **506**, received commands are communicated to all cluster node modules **204***a-e* on the computer cluster **100**. In one embodiment, when a user enters a command in the user interface module **202**, the cluster node module **204***a* connected to the user interface module **202** submits the user command to all other cluster node modules **204***b-e* in the computer cluster **100**. The user commands can be simple (for example, "1+1"), but can also be entire subroutines and sequences of code (such as, for example, Mathematica code), including calls to MPI from within the user interface module **202** (for example, the Mathematica Front End) to perform message passing between kernel modules **206***a-e* (for example, Mathematica kernels). These include the fundamental MPI calls, which are implemented using specially identified messages between a cluster node module **204** and its local kernel module **206**.

The message (or user command) is communicated to the kernel modules **206***a-e* at **508**. In one embodiment, the cluster node module **204***a* connected to the user interface module **202** submits the user command to the kernel module **206***a* to which it is connected. Each of the other cluster node modules **204***b-e*, after receiving the message, submits the command to the respective kernel module **206***b-e* to which it is connected.

At **510**, a cluster node module **204** receives a result from a kernel module **206**. In one embodiment, once the kernel module **206** completes its evaluation, it returns the kernel module's output to the cluster node module **204** to which it is connected. Depending on the nature of the result from the kernel module, the cluster node module **204** can report the result to a local computer system or pass the result as a message to another cluster node module **204**. For example, the cluster node module **204***a* running on the first processor **112***a* reports the output on its local computer system **110**. For example, on the first processor **112***a*, cluster node module **204***a* only directly reports the output of kernel module **206***a*.

Messages from other cluster node modules **204** are responded to at **512**. In one embodiment, each cluster node module (for example, the cluster node module **204***a*) checks for and responds to messages from other cluster node modules **204***b-e* and from the kernel module **206***a* repeatedly until those are exhausted. In one embodiment, output messages from the kernel module **206** are forwarded to output on the local computer system. Messages from other cluster node modules **204** are forwarded to a received message queue **306** ("RMQ"). Data from each entry in the message receiving queue **308** ("MRQ") is matched with entries in the RMQ **306** (see, for example, description of the mpiIRecv[ ] call, above). If found, data from the MRQ **308** are combined into those in the RMQ **306** and marked as "completed" (see, for example, description of the mpiTest[ ] call, above). This process provides the peer-to-peer behavior of the cluster node modules **204***a-e*. Via this mechanism, code running within multiple, simultaneously running kernel modules (for example, Mathematica kernels) can interact on a pair-wise or collective basis, performing calculations, processing, or other work on a scale larger and/or faster than one kernel could have done alone. In this manner, user-entered instructions and data

specifying what work will be done via user commands can be executed more quickly and/or reliably. Once responding to messages has completed, the process returns to **502**.

In some embodiments, a computer system includes software, such as an operating system, that divides memory and/or other system resources into a user space, a kernel space, an application space (for example, a portion of the user space allocated to an application program), and/or an operating system space (for example, a portion of the user space allocated to an operating system). In some embodiments, some or all of the cluster node modules **204***a-e* are implemented in the application space of a computer system. In further embodiments, at least some of the cluster node modules **204***a-e* are implemented in the operating system space of a computer system. For example, some cluster node modules in a computer cluster may operate in the application space while others operate in the operating system space.

In some embodiments, some or all of the functionality of the cluster node modules **204***a-e* is incorporated into or integrated with the operating system. The operating system can add cluster computing functionality to application programs, for example, by implementing at least some of the methods, modules, data structures, commands, functions, and processes discussed herein. Other suitable variations of the techniques described herein can be employed, as would be recognized by one skilled in the art.

In some embodiments, the operating system or components of the operating system can identify and launch the front end **202** and the kernels **206**. The operating system or its components can connect the front end **202** and kernels **206** to one another in the same manner as a cluster node module **204** would or by a variation of one of the techniques described previously. The operating system can also be responsible for maintaining the communications network **102** that connects the modules to one another. In some embodiments, the operating system implements at least some MPI-style calls, such as, for example, collective MPI-style calls. In some embodiments, the operating system includes an application programming interface (API) library of cluster subroutine calls that is exposed to application programs. Applications programs can use the API library to assist with launching and operating the computer cluster.

C. Cluster Shut Down Process

In one embodiment, a computer cluster **100** includes a procedure to shut down the system. If the operation process (or main loop) on the cluster node module **204***a* connected to the user interface module **202** detects a "Quit" or "Exit" command or otherwise receives a message from the user indicating a shut down, the sequence to shut down the cluster node modules **204***a-e* and the kernel modules **206***a-e* is activated. In one embodiment, the cluster node module **204***a* connected to the user interface module **202** sends a quit message to all other cluster node modules **204***b-e*. Each cluster node module **204** forwards the quit command to its local kernel module **206**. Once its Mathematica kernel has quit, each cluster node module **204** proceeds to tear down its communication network with other cluster node modules (for example, see description of the MPI_Finalize command, above). At the conclusion of the process, each cluster node module **204** exits execution.

V. Example Operation

For purposes of illustration, sample scenarios are discussed in which the computer cluster system is used in operation. In these sample scenarios, examples of Mathematica code are given, and descriptions of how the code would be executed by a cluster system are provided.

Basic MPI

Fundamental data available to each node includes the node's identification number and total processor count.

```
In[1]:= {$IdProc, $NProc}
Out[1]:= {0, 2}
```

The first element should be unique for each processor, while the second is generally the same for all. Processor 0 can see what other values are using a collective (see below) communications call such as mpiGather[ ].

```
In[2]:= mpiGather[{$IdProc, $NProc},list,0]; list
Out[2]:= {{0, 2}, {1, 2}}
```

Peer-to-Peer MPI

The mpiSend and mpiRecv commands make possible basic message passing, but one needs to define which processor to target. The following defines a new variable, targetProc, so that each pair of processors will point to each other.

```
In[3]:= targetProc=If[1==Mod[$IdProc, 2 ],$IdProc-1,$IdProc+1]
Out[3]:= 1
```

In this example, the even processors target its "right" processor, while the odd ones point its "left." For example, if the processors were lined up in a row and numbered in order, every even-numbered processor would pair with the processor following it in the line, and every odd-numbered processor would pair with the processor preceding it. Then a message can be sent:

```
In[4]:= If [ 1==Mod[ $IdProc , 2],mpiSend[N[Pi,22],targetProc,
    mpiCommWorld,d], mpiRecv[a,targetProc,mpiCommWorld,d]]
```

The If [ ] statement causes the processors to evaluate different code: the odd processor sends 22 digits of Pi, while the even receives that message. Note that these MPI calls return nothing. The received message is in the variable a:

```
In[5]:= a
Out[5]:= 3.141592653589793238 4626
In[6]:= Clear[a]
```

The variable a on the odd processors would have no definition. Moreover, if $NProc is 8, processor 3 sent Pi to processor 2, processor 5 sent Pi to processor 4, and so on. These messages were not sent through processor 0, but they communicated on their own.

The mpiISend and mpiIRecv commands have a letter "I" to indicate asynchronous behavior, making it possible to do other work while messages are being sent and received, or if the other processor is busy. So, the above example could be done asynchronously:

```
In[7]:= If[1==Mod[$IdProc, 2],mpiISend[N[Pi,22],targetProc,
        mpiCommWorld,d,e],
        mpiIRecv[a,targetProc,mpiCommWorld,d,e]]
```

The variable e has important data identifying the message, and mpiTest[e] can return True before the expressions are to be accessed. At this point, many other evaluations can be performed. Then, one can check using mpiTest when the data is needed:

```
In[29]:= mpiTest[e]
Out[29]:= True
In[30]:= a
Out[30]:= 3.1415926535897932384626
In[31]:= Clear[a,e]
```

The mpiWait[e] command could have also have been used, which does not return until mpiTest[e] returns True. The power of using these peer-to-peer calls is that it becomes possible to construct any message-passing pattern for any problem.

Collective MPI

In some cases, such explicit control is not required and a commonly used communication pattern is sufficient. Suppose processor 0 has an expression in b that all processors are meant to have. A broadcast MPI call would do:

In[8]:=mpiBcast[b, 0, mpiCommWorld]

The second argument specifies which processor is the "root" of this broadcast; all others have their b overwritten. To collect values from all processors, use mpiGatherD:

In[9]:=mpiGather[b, c, 0, mpiCommWorld]

The variable c of processor 0 is written with a list of all the b of all the processors in mpiCommWorld. The temporal opposite is mpiScatter:

```
In[10]:= Clear[b]; a = {2, 4, 5, 6}; mpiScatter[a, b, 0,
        mpiCommWorld]; b
Out[10]:= {2, 4}
```

The mpiScatter command cuts up the variable a into even pieces (when possible) and scatters them to the processors. This is the result if $NProc=2, but if $NProc=4, b would only have {2}.

MPI provides reduction operations to perform simple computations mixed with messaging. Consider the following:

```
In[11]:= a = {{2 + $IdProc, 45[ ],3,{1 + $IdProc,$NProc[ ][ ];
        mpiReduce [a, d, mpiSum, 0, mpiCommWorld ]
In[12]:= d
Out[12]:= {{5, 90}, 6, {3, 4}}
```

The mpiSum constant indicates that variable a of every processor will be summed. In this case, $NProc is 2, so those elements that were not identical result in odd sums, while those that were the same are even.

Most of these calls have default values if not all are specified. For example each of the following calls will have the equivalent effect as the above mpiGather[ ] call:

```
mpiGather[b, c, 0]
mpiGather[b, c]
c = mpiGather[b]
```

High-Level Calls

High-level calls can include convenient parallel versions of commonly used application program calls (for example, Mathematica calls). For example, ParallelTable[ ] is like Table[ ], except that the evaluations are automatically performed in a distributed manner:

```
In[13]:= ParallelTable[i,{i,100},0]
Out[13]:= {1,2,3,4,5, . . . ,99,100}
```

The third argument specifies that the answers are collated back to processor 0. This is a useful, simple way to parallelize many calls to a complex function. One could define a complicated function and evaluate it over a large range of inputs:

```
In[14]:= g[x_] := Gamma[2 + 0.5*(x-1)];
        ParallelTable[g[i],{i,100},0]
Out[14]:= {1, 1.32934, 2., 3.32335, 6., 11.6317, 24., 52.3428,
        120., 287.885, 720}
```

ParallelFunctionToList[ ] also provides a simplified way to perform this form of parallelism.

Operations with Non-Trivial Communication

Matrix Operations

In some embodiments, one or more functions can help solve matrix calculations in parallel:

```
In[15]:= a = Table[i+ 3* $IdProc + 2 j, {i, 2}, {j,4}]
Out[15] := {{3, 5, 7, 9}, {4, 6, 8, 10}}
In[16]:= t = ParallelTranspose[a]
Out[16]:= {{3, 4, 6, 7}, {5, 6, 8, 9}}
```

Fourier Transforms

A Fourier transform of a large array can be solved faster in parallel, or made possible on a cluster because it can all be held in memory. A two-dimensional Fourier transform of the above example follows:

```
In[17]:= f = ParallelFourier[a]
Out[17]:= {{32. + 0. I, -4. - 4. I, -4., -4. + 4. I}, {-3. -
        3. I, 0. + 0. I, 0., 0. + 0. I}}
```

Edge Cell Management

Many problems require interactions between partitions, but only on the edge elements. Maintaining these edges can be performed using EdgeCell[ ].

```
In[18]:= a = {2, 4, 5, 6, 7 }+8*$IdProc
Out[18]:= {2, 4, 5, 6, 7}
In[19]:= EdgeCell[a]; a
Out[19]:= {14, 4, 5, 6, 12}
```

Element Management

In particle-based problems, items can drift through space, sometimes outside the partition of a particular processor. This can be solved with ElementManage[1:

```
In[20]:= list={{0,4},{1,3},{1,4},{0,5}}; fcn[x_]:=x[[1]]
In[21]:= ElementManage[list, fcn]
Out[21]:= {{0, 4}, {0, 5}, {0, 4}, {0, 5}}
In[21]:= ElementManage[list, 2]
Out[21]:= {{0, 4}, {0, 5}, {0, 4}, {0, 5}}
```

The second argument of ElementManage describes how to test elements of a list. The fcn identifier returns which processor is the "home" of that element. Passing an integer assumes that each element is itself a list, whose first element is a number ranging from 0 to the passed argument.

While the examples above involve Mathematica software and specific embodiments of MPI calls and cluster commands, it is recognized that these embodiments are used only to illustrate features of various embodiments of the systems and methods.

VI. Additional Embodiments

Although cluster computing techniques, modules, calls, and functions are disclosed with reference to certain embodiments, the disclosure is not intended to be limited thereby. Rather, a skilled artisan will recognize from the disclosure herein a wide number of alternatives for the exact selection of cluster calls, functions, and management systems. For example, single-node kernels can be managed using a variety of management tools and/or can be managed manually by a user, as described herein. As another example, a cluster node module can contain additional calls and procedures, including calls and procedures unrelated to cluster computing, that are not disclosed herein.

Other embodiments will be apparent to those of ordinary skill in the art from the disclosure herein. Moreover, the described embodiments have been presented by way of example only, and are not intended to limit the scope of the disclosure. Indeed, the novel methods and systems described herein can be embodied in a variety of other forms without departing from the spirit thereof. Accordingly, other combinations, omissions, substitutions and modifications will be apparent to the skilled artisan in view of the disclosure herein. Thus, the present disclosure is not intended to be limited by the disclosed embodiments, but is to be defined by reference to the appended claims. The accompanying claims and their equivalents are intended to cover forms or modifications as would fall within the scope and spirit of the inventions.

What is claimed is:

1. A system for performing an instruction received from a front end by executing commands on one or more special purpose microprocessors, the system comprising:

a plurality of nodes, wherein each node is configured to access a computer-readable memory system comprising program code for a single-node kernel module, and wherein each single-node kernel module is configured to interpret instructions received by the single-node kernel module into commands that are executable by a special purpose microprocessor;

a plurality of cluster node modules, wherein each cluster node module is stored in a computer-readable memory system and configured to communicate with a single-node kernel and with one or more other cluster node modules, to accept instructions, and to interpret at least some of the instructions such that the plurality of cluster node modules communicate with one another in order to act as a cluster in executing commands using one or more hardware processors; and

a communications system configured to connect the plurality of nodes;

wherein the plurality of cluster node modules cooperate to interpret and translate, as needed, the instruction for execution by a plurality of single-node kernel modules, and wherein at least one of the plurality of cluster node modules returns a result to the front end.

2. The system of claim **1**, wherein the special purpose microprocessor comprises a digital signal processor.

3. The system of claim **1**, wherein the plurality of nodes are organized into two or more groups of node subsets.

4. The system of claim **3**, wherein at least one of the two or more groups of node subsets exchange data with the special purpose microprocessor.

5. The system of claim **1**, wherein the special purpose microprocessor comprises multiple processor cores.

6. The system of claim **1**, wherein at least one of the plurality of cluster node modules resides in processor cache memory.

7. The system of claim **1**, wherein each single-node kernel modules resides in processor cache memory.

8. A system for performing an instruction received from a front end by executing commands on one or more hardware processors, the system comprising:

a plurality of nodes, wherein each node is configured to access a computer-readable memory system comprising program code for a single-node kernel module, and wherein each single-node kernel module is configured to interpret instructions received by the single-node kernel module into commands that are executable by a hardware processor, wherein the commands are configured to perform computations one or more elements from a list of elements;

a plurality of cluster node modules, wherein each cluster node module is stored in a computer-readable memory system and configured to communicate with a single-node kernel and with one or more other cluster node modules, to accept instructions, and to interpret at least some of the instructions such that the plurality of cluster node modules communicate with one another in order to act as a cluster in performing computations on the list of elements; and

a communications system configured to connect the plurality of nodes;

wherein the plurality of cluster node modules cooperate to interpret and translate, as needed, the instruction for execution by a plurality of single-node kernel modules, wherein the list of elements is partitioned for execution on the plurality of nodes, and wherein at least one of the plurality of cluster node modules returns a result to the front end.

9. The system of claim **8**, wherein one or more elements of the list of elements migrates to a different node.

10. A method of performing an instruction received from a front end by executing commands on one or more special purpose microprocessors, the method comprising:

communicating an instruction from a front end to one or more cluster node modules connected to one another by a communications system;

for each of the one or more cluster node modules, communicating a message based on the instruction to a respective kernel module associated with the cluster node mod-

ule, wherein the respective kernel module is configured to interpret the message into commands that are executable by a special purpose microprocessor;

for each of the one or more cluster node modules, receiving a result from the respective kernel module associated with the cluster node module; and

for at least one of the one or more cluster node modules, responding to messages from other cluster node modules.

11. The method of claim **10**, wherein communicating a message based on the command to a respective kernel module associated with the cluster node module comprises communicating a specially identified message to the respective kernel module.

12. The method of claim **10**, wherein responding to messages from other cluster node modules comprises:

forwarding messages from the other cluster node modules to a received message queue;

matching data from each entry in a message receiving queue with entries in the received message queue;

combining data from the message receiving queue with matching data in the received message queue; and

marking the matching data as completed.

13. The method of claim **10**, wherein communicating a command from at least one of a user interface or a script to one or more cluster node modules within the computer cluster comprises communicating an instruction from a Mathematica Front End to a first cluster node module, wherein the first cluster node module forwards the instruction to other cluster node modules running on the computer cluster.

14. The method of claim **10**, wherein each of the one or more cluster node modules communicates with a respective kernel module using MathLink.

* * * * *

EXHIBIT D

US010333768B2

(12) **United States Patent**
Tannenbaum et al.

(10) **Patent No.: US 10,333,768 B2**
(45) **Date of Patent: *Jun. 25, 2019**

(54) **CLUSTER COMPUTING**

(71) Applicant: **Advanced Cluster Systems, Inc.**, Aliso Viejo, CA (US)

(72) Inventors: **Zvi Tannenbaum**, Newport Beach, CA (US); **Dean E. Dauger**, Huntington Beach, CA (US)

(73) Assignee: **Advanced Cluster Systems, Inc.**, Aliso Viejo, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 524 days.

This patent is subject to a terminal disclaimer.

(21) Appl. No.: **14/181,112**

(22) Filed: **Feb. 14, 2014**

(65) **Prior Publication Data**

US 2014/0372586 A1 Dec. 18, 2014

Related U.S. Application Data

(63) Continuation of application No. 13/423,063, filed on Mar. 16, 2012, now Pat. No. 8,676,877, which is a (Continued)

(51) **Int. Cl.**
G06F 9/50 (2006.01)
G06F 9/54 (2006.01)
(Continued)

(52) **U.S. Cl.**
CPC ***H04L 41/04*** (2013.01); ***G06F 9/5072*** (2013.01); ***G06F 9/54*** (2013.01); ***G06F 15/76*** (2013.01)

(58) **Field of Classification Search**
USPC 709/223
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,423,046 A 6/1995 Nunnelley et al.
5,881,315 A 3/1999 Cohen
(Continued)

FOREIGN PATENT DOCUMENTS

JP 08-87473 A 2/1996
JP H11-126196 5/1999
(Continued)

OTHER PUBLICATIONS

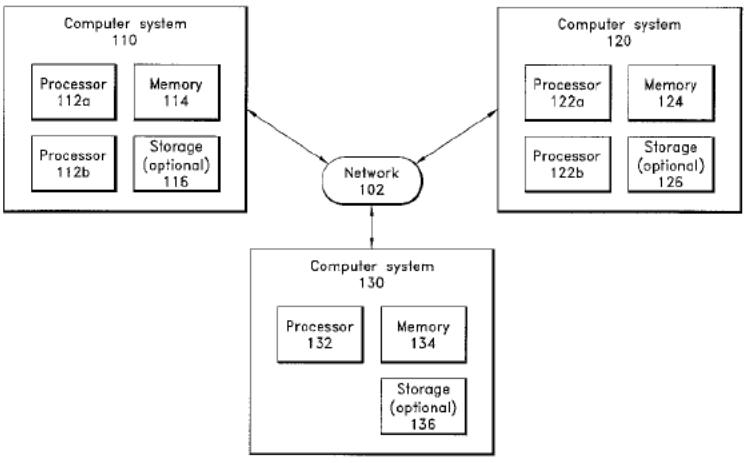
"GridMathematica 1.1: Grid Computing Gets a Speed Boost from Mathematica 5", The Mathematica Journal, vol. 9, No. 2, 2004.
(Continued)

Primary Examiner — Hermon Asres
(74) *Attorney, Agent, or Firm* — Knobbe Martens Olson & Bear LLP

(57) **ABSTRACT**

In some embodiments, a computer cluster system comprises a plurality of nodes and a software package comprising a user interface and a kernel for interpreting program code instructions. In certain embodiments, a cluster node module is configured to communicate with the kernel and other cluster node modules. The cluster node module can accept instructions from the user interface and can interpret at least some of the instructions such that several cluster node modules in communication with one another and with a kernel can act as a computer cluster.

39 Claims, 5 Drawing Sheets



Related U.S. Application Data

continuation of application No. 12/040,519, filed on Feb. 29, 2008, now Pat. No. 8,140,612, which is a continuation-in-part of application No. 11/744,461, filed on May 4, 2007, now Pat. No. 8,082,289.

(60) Provisional application No. 60/813,738, filed on Jun. 13, 2006, provisional application No. 60/850,908, filed on Oct. 11, 2006.

(51) **Int. Cl.**

G06F 15/76	(2006.01)
H04L 12/24	(2006.01)

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,930,768	A	7/1999	Hooban	
6,006,259	A	12/1999	Adelman et al.	
6,067,609	A	5/2000	Meeker et al.	
6,108,699	A	8/2000	Moiin	
6,195,680	B1	2/2001	Goldszmidt et al.	
6,202,080	B1	3/2001	Lu et al.	
6,546,403	B1	4/2003	Carlson, Jr. et al.	
6,578,068	B1	6/2003	Bowman-Amuah	
6,604,134	B2	8/2003	Haury	
6,751,698	B1	6/2004	Deneroff et al.	
6,782,537	B1	8/2004	Blackmore et al.	
6,968,335	B2	11/2005	Bayliss et al.	
6,968,359	B1 *	11/2005	Miller	G06F 9/5061 709/205
7,031,944	B2	4/2006	Tanioka	
7,093,004	B2	8/2006	Bernardin et al.	
7,136,924	B2	11/2006	Dauger	
7,139,882	B2	11/2006	Suzuoki et al.	
7,174,381	B2	2/2007	Gulko et al.	
7,177,874	B2	2/2007	Jardin	
7,249,357	B2	7/2007	Landman et al.	
7,315,877	B2	1/2008	Bhanot et al.	
7,334,232	B2	2/2008	Jacobs et al.	
7,437,460	B2	10/2008	Chidambaran et al.	
7,472,193	B2	12/2008	Dauger	
7,502,915	B2	3/2009	Jacob et al.	
7,519,652	B2	4/2009	Page et al.	
7,533,141	B2	5/2009	Nadgir et al.	
7,533,389	B2	5/2009	Verbeke et al.	
7,554,949	B2	6/2009	Chen	
7,568,131	B2	7/2009	Vertes	
7,644,130	B2	1/2010	Magro et al.	
7,698,390	B1	4/2010	Harkness et al.	
7,716,323	B2	5/2010	Gole et al.	
7,757,236	B1	7/2010	Singh	
7,788,314	B2	8/2010	Holt	
7,835,022	B2	11/2010	Matsumoto	
7,937,455	B2	5/2011	Saha et al.	
8,082,289	B2	12/2011	Tannenbaum et al.	
8,117,288	B2	2/2012	Bhanot et al.	
8,140,612	B2	3/2012	Tannenbaum et al.	
8,402,080	B2	3/2013	Tannenbaum et al.	
8,402,083	B2	3/2013	Tannenbaum et al.	
8,601,101	B1 *	12/2013	Singh	H04L 69/40 370/254
8,676,877	B2	3/2014	Tannenbaum et al.	
8,849,889	B1	9/2014	Tannenbaum et al.	
9,405,564	B2	8/2016	Muellers et al.	
2002/0049859	A1	4/2002	Bruckert et al.	
2003/0005068	A1	1/2003	Nickel et al.	
2003/0005266	A1	1/2003	Dauger	
2003/0051062	A1	3/2003	Circenis	
2003/0135621	A1	7/2003	Romagnoli	
2003/0195931	A1	10/2003	Dauger	
2003/0195938	A1 *	10/2003	Howard	G06F 8/45 709/208
2004/0015968	A1	1/2004	Neiman	
2004/0098359	A1	5/2004	Bayliss	
2004/0098447	A1	5/2004	Verbeke	
2004/0110209	A1	6/2004	Yokota et al.	
2004/0157203	A1	8/2004	Dunk	
2004/0195572	A1	10/2004	Kato et al.	
2004/0252710	A1	12/2004	Jeter, Jr. et al.	
2004/0254984	A1	12/2004	Dinker	
2005/0015460	A1	1/2005	Goyle et al.	
2005/0021751	A1 *	1/2005	Block	G06F 9/54 709/225
2005/0038852	A1	2/2005	Howard	
2005/0060237	A1	3/2005	Barsness et al.	
2005/0076105	A1	4/2005	Keohane et al.	
2005/0097300	A1	5/2005	Gildea et al.	
2005/0108394	A1	5/2005	Braun et al.	
2005/0154789	A1	7/2005	Fellenstein et al.	
2005/0180095	A1	8/2005	Ellis	
2005/0188088	A1	8/2005	Fellenstein et al.	
2006/0026601	A1	2/2006	Solt	
2006/0053216	A1	3/2006	Deokar et al.	
2006/0059473	A1	3/2006	Moler	
2006/0106931	A1	5/2006	Richoux	
2007/0044099	A1	2/2007	Rajput	
2007/0073705	A1 *	3/2007	Gray	G06F 9/451
2007/0094532	A1 *	4/2007	Sengupta	G06F 11/3632 714/5.1
2007/0124363	A1	5/2007	Lurie et al.	
2008/0250347	A1	10/2008	Gray et al.	
2008/0281997	A1	11/2008	Archer et al.	

FOREIGN PATENT DOCUMENTS

JP	11-328134	11/1999
JP	2002117010	4/2002
JP	2004-061359	2/2004
JP	2004-247405	9/2004
JP	2005-063033	10/2005
WF	1368948	12/2003
WO	WO 2004/086270	10/2004

OTHER PUBLICATIONS

"Wolfram gridMathematica", Wolfram Research, Inc., 2007.

Carns et al., "An Evaluation of Message Passing Implementations on Beowulf Workstations", Aerospace Conference, Mar. 6-13, 1999, IEEE 1999, vol. 5, pp. 41-54.

Dauger et al., "Plug-and-Play Cluster Computing using Mac OS X", IEEE International Conference, Dec. 1-4, 2003, Paper appears in Cluster Computing Jan. 8, 2004, pp. 430-435.

Dauger et al., Plug-and Play Cluster Computing: High-Performance Computing for the Mainstream, IEEE Computing in Science and Engineering, Mar./Apr. 2005, pp. 27-33.

Hamscher et al., "Evaluation of Job-Scheduling Strategies for grid Computing", LNCS: Lecture Notes in Computer Science, 2000, pp. 191-202.

International Search Report dated Sep. 11, 2008, International Application No. PCT/US07/70585.

Jahanzeb et al., "Libra: a computational economy-based job scheduling system for clusters", Software Practice and Experience, Feb. 24, 2004, vol. 34, pp. 573-590.

Jain et al., "Data Clustering: A Review", ACM Computing Surveys, Sep. 1999, vol. 31, No. 3.

Kepner, Jeremy et al., "Parallel Matlab: The Next Generation", Aug. 20, 2004, MIT Lincoln Laboratory, Lexington, MA.

Kim, Hahn et al., "Introduction to Parallel Programming and pMatlab v2.0", 2011, MIT Lincoln Laboratory, Lexington, MA.

Maeder, R., "Mathematica Parallel Computing Toolkit: Unleash the Power of Parallel Computing", Wolfram Research, Jan. 2005.

Tepeneu and Ida, "MathGridLink—A bridge between Mathematica and the Grid", Nippon Sofutowea Kagakkai Taikai Ronbunshu, 2003, vol. 20, pp. 74-77.

Wolfram, Stephen: The Mathematica Book 5th Edition; Wolfram Research, Inc. 2003.

Haynos, Matt, "Perspectives on grid: Grid computing—next-generation distributed computing", Jan. 27, 2004, IBM Developer Works, http://www-106.ibm.com/developerworks/library/gr-heritage.

(56) **References Cited**

OTHER PUBLICATIONS

"Pingpong MPI Benchmark—SEM vs 'grid'," Jan. 2009, Dauger Research, Inc., http://daugerresearch.com/pooch/mathmatica/.
Wolfram Research, "Mathematics Parallel Computing Toolkit", Jan. 2005, pp. 1-95.
English translation of Matsumura, et al., "Construction of distributed computing system for large-scale matrices intended or reduction of communication blocks," The Special Interest Group Technical Reports of IPSJ, Information Processing Society of Japan, Mar. 5, 1998, vol. 98, No. 18, pp. 19-24.
Konishi, et al., "Performance Evaluation of Parallel Computing Tool for MATLAB on PC Grid Environment," IPSJ SIG Technical Reports, Aug. 5, 2005, vol. 2005, No. 84. 9 pages.
Matsumura, et al. "Construction of distributed computing system for large-scale matrices intended or reduction of communication blocks," Aug. 5, 1998, vol. 98, No. 18. 9 pages.
Tatebe, et al., "Efficient Implementation of MPI Using Remote Memory Write," Transactions of Information Processing Society of Japan, May 1999, vol. 40, No. 5. 14 pages.
U.S. Appl. No. 60/799,474, filed May 10, 2006, titled "Graphical Interface for Monitoring Status of a Concurrent Computing Process".
Roman E. Maeder "Mathematica Parallel Computing Toolkit—Unleash the Power of Parallel Computing", Jan. 1, 2005, pp. i-x, 1-90. URL:http//media.wolfam.com/documents/ParallelComputingToolkitDoumentation.pdf; retrieved on Aug. 21, 2015.
European Search Report received in Application No. 17207443.7, dated Sep. 20, 2018, in 9 pages.

* cited by examiner

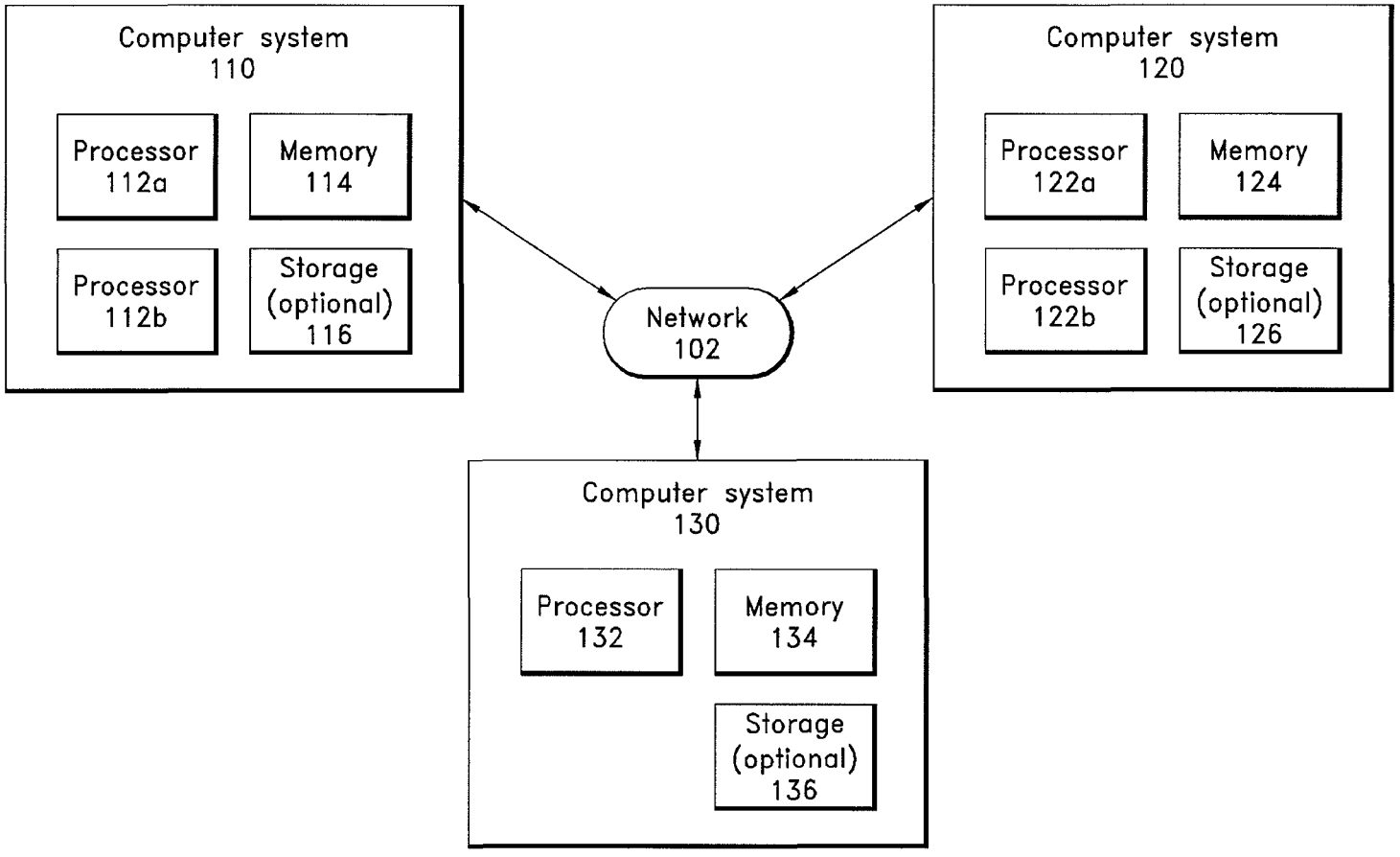

Computer system
110
Processor
112a
Memory
114
Processor
112b
Storage
(optional)
116
Network
102
Computer system
120
Processor
122a
Memory
124
Processor
122b
Storage
(optional)
126
Computer system
130
Processor
132
Memory
134
Storage
(optional)
136

FIG. 1

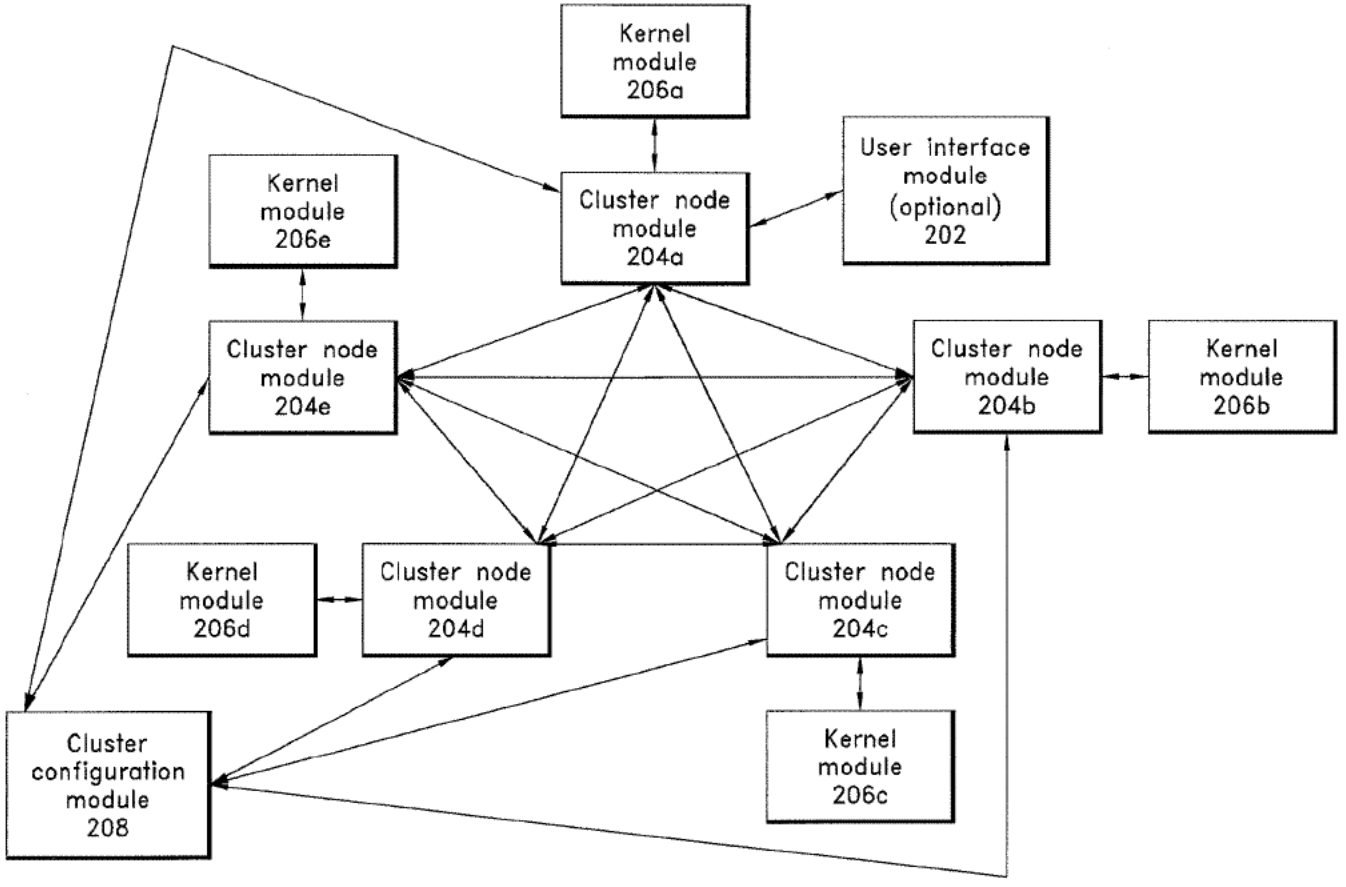

Kernel module 206a
User interface module (optional) 202
Kernel module 206e
Cluster node module 204a
Cluster node module 204e
Cluster node module 204b
Kernel module 206b
Kernel module 206d
Cluster node module 204d
Cluster node module 204c
Cluster configuration module 208
Kernel module 206c

FIG. 2

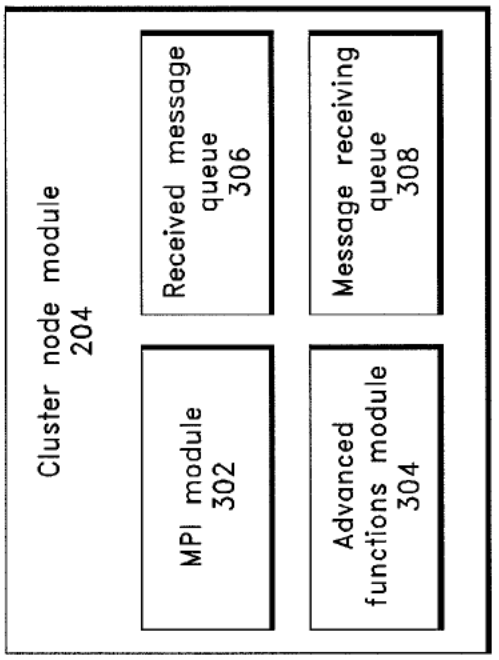

Cluster node module
204
MPI module
302
Advanced functions module
304
Received message queue
306
Message receiving queue
308

FIG. 3

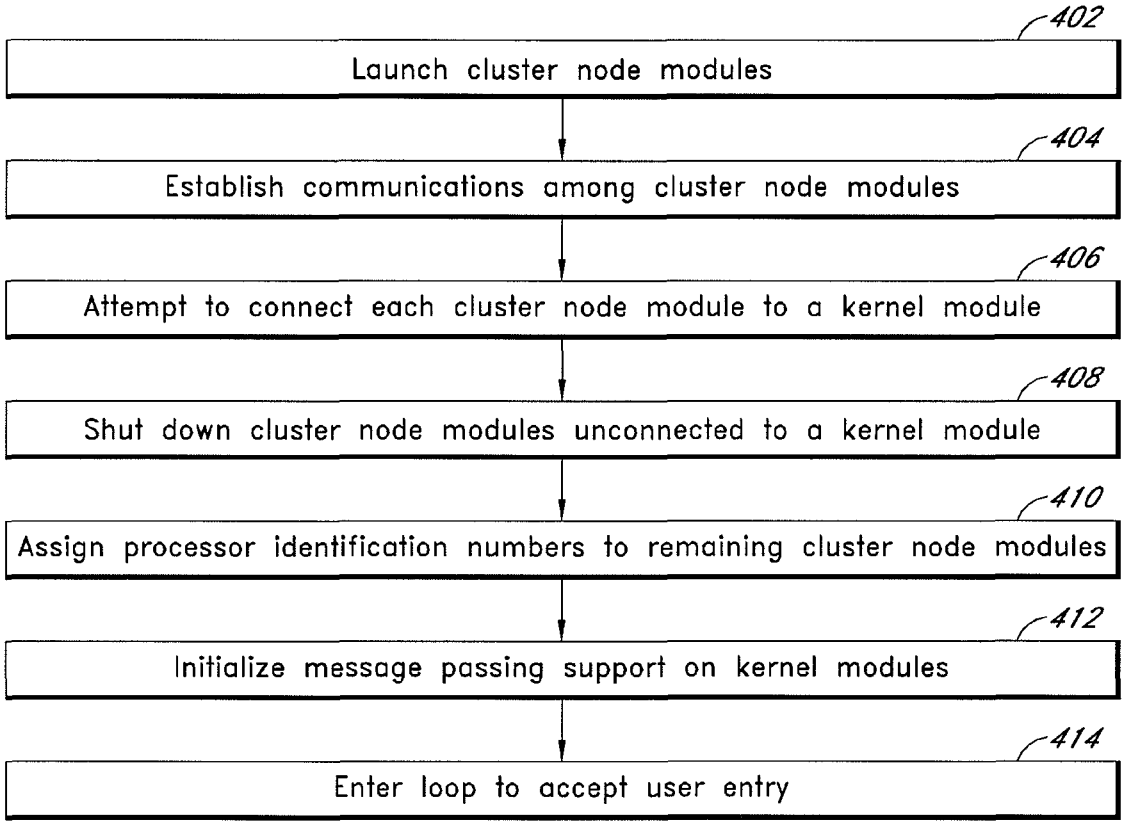

402
Launch cluster node modules
404
Establish communications among cluster node modules
406
Attempt to connect each cluster node module to a kernel module
408
Shut down cluster node modules unconnected to a kernel module
410
Assign processor identification numbers to remaining cluster node modules
412
Initialize message passing support on kernel modules
414
Enter loop to accept user entry

FIG. 4

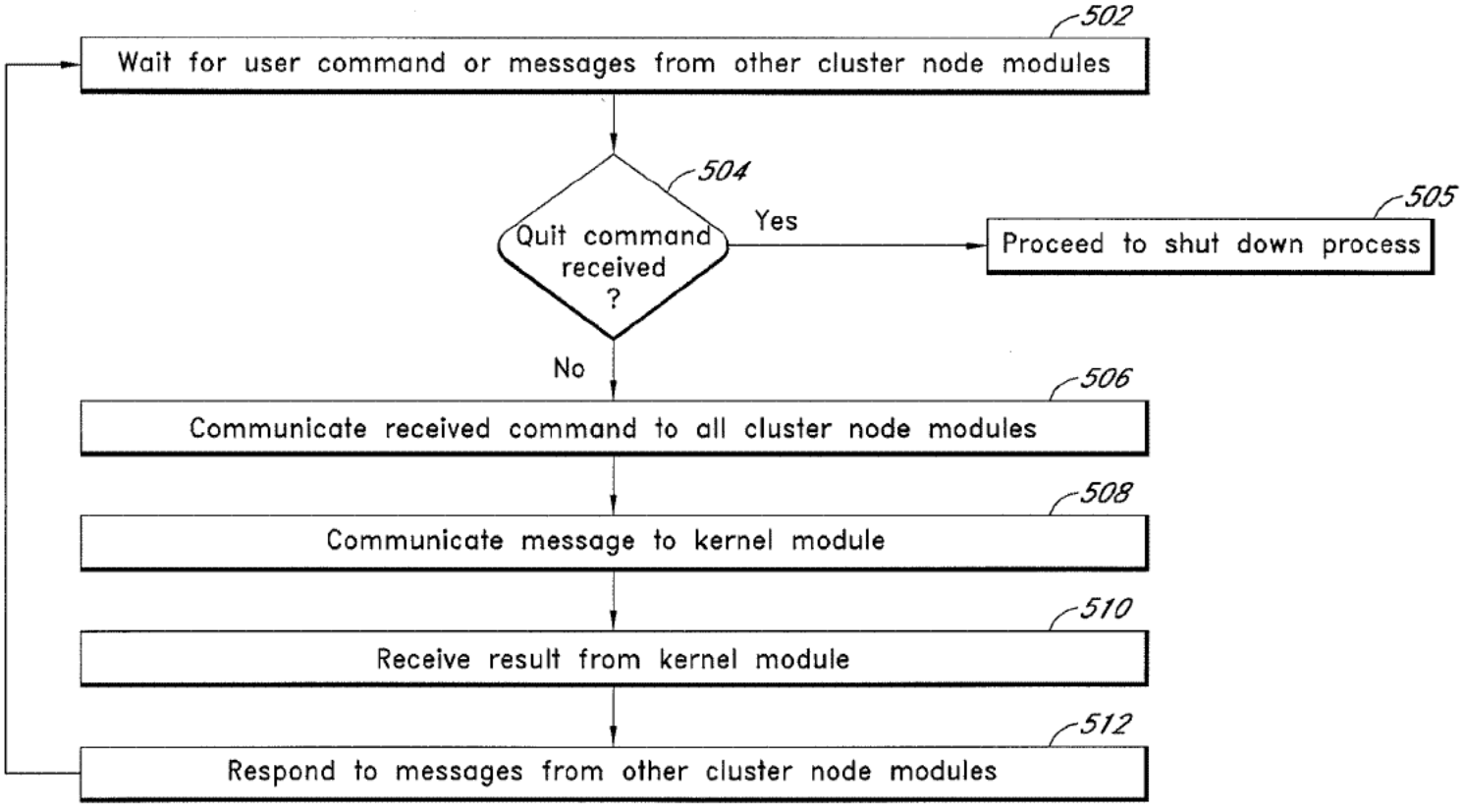

502
Wait for user command or messages from other cluster node modules
504
Quit command received ?
Yes
505
Proceed to shut down process
No
506
Communicate received command to all cluster node modules
508
Communicate message to kernel module
510
Receive result from kernel module
512
Respond to messages from other cluster node modules

FIG. 5

CLUSTER COMPUTING

INCORPORATION BY REFERENCE TO ANY PRIORITY APPLICATIONS

Any and all applications for which a foreign or domestic priority claim is identified in the Application Data Sheet as filed with the present application are incorporated by reference under 37 CFR 1.57 and made a part of this specification.

BACKGROUND

Field

The present disclosure relates to the field of cluster computing generally and to systems and methods for adding cluster computing functionality to a computer program, in particular.

Description of Related Art

Computer clusters include a group of two or more computers, microprocessors, and/or processor cores ("nodes") that intercommunicate so that the nodes can accomplish a task as though they were a single computer. Many computer application programs are not currently designed to benefit from advantages that computer clusters can offer, even though they may be running on a group of nodes that could act as a cluster. Some computer programs can run on only a single node because, for example, they are coded to perform tasks serially or because they are designed to recognize or send instructions to only a single node.

Some application programs include an interpreter that executes instructions provided to the program by a user, a script, or another source. Such an interpreter is sometimes called a "kernel" because, for example, the interpreter can manage at least some hardware resources of a computer system and/or can manage communications between those resources and software (for example, the provided instructions, which can include a high-level programming language). Some software programs include a kernel that is designed to communicate with a single node. An example of a software package that includes a kernel that is designed to communicate with a single node is Mathematica® from Wolfram Research, Inc. ("Mathematica"). Mathematics software packages from other vendors and other types of software can also include such a kernel.

A product known as gridMathematica, also from Wolfram Research, Inc., gives Mathematica the capability to perform a form of grid computing known as "distributed computing." Grid computers include a plurality of nodes that generally do not communicate with one another as peers. Distributed computing can be optimized for workloads that consist of many independent jobs or packets of work, which do not need to share data between the jobs during the computational process. Grid computers include at least one node known as a master node that manages a plurality of slave nodes or computational nodes. In gridMathematica, each of a plurality of kernels runs on a single node. One kernel is designated the master kernel, which handles all input, output, and scheduling of the other kernels (the computational kernels or slave kernels). Computational kernels receive commands and data only from the node running the master kernel. Each computational kernel performs its work independently of the other computational kernels and intermediate results of one job do not affect other jobs in progress on other nodes.

SUMMARY

Embodiments described herein have several features, no single one of which is solely responsible for their desirable attributes. Without limiting the scope of the invention as expressed by the claims, some of the advantageous features will now be discussed briefly.

Some embodiments described herein provide techniques for conveniently adding cluster computing functionality to a computer application. In one embodiment, a user of a software package may be able to achieve higher performance and/or higher availability from the software package by enabling the software to benefit from a plurality of nodes in a cluster. One embodiment allows a user to create applications, using a high-level language such as Mathematica, that are able to run on a computer cluster having supercomputer-like performance. One embodiment provides access to such high-performance computing through a Mathematica Front End, a command line interface, one or more high-level commands, or a programming language such as C or FORTRAN.

One embodiment adapts a software module designed to run on a single node, such as, for example, the Mathematica kernel, to support cluster computing, even when the software module is not designed to provide such support. One embodiment provides parallelization for an application program, even if no access to the program's source code is available. One embodiment adds and supports Message Passing Interface ("MPI") calls directly from within a user interface, such as, for example, the Mathematica programming environment. In one embodiment, MPI calls are added to or made available from an interactive programming environment, such as the Mathematica Front End.

One embodiment provides a computer cluster including a first processor, a second processor, and a third processor. The cluster includes at least one computer-readable medium in communication at least one of the first processor, the second processor, or the third processor. A first kernel resides in the at least one computer-readable medium and is configured to translate commands into code for execution on the first processor. A first cluster node module resides in the at least one computer-readable medium. The first cluster node module is configured to send commands to the first kernel and receives commands from a user interface. A second kernel resides in the at least one computer-readable medium. The second kernel is configured to translate commands into code for execution on the second processor. A second cluster node module resides in the at least one computer-readable medium. The second cluster node module is configured to send commands to the second kernel and communicates with the first cluster node module. A third kernel resides in the at least one computer-readable medium. The third kernel is configured to translate commands into code for execution on the third processor. A third cluster node module resides in the at least one computer-readable medium. The third cluster node module is configured to send commands to the third kernel and configured to communicate with the first cluster node module and the second cluster node module. The first cluster node module comprises a data structure in which messages originating from the second and third cluster node modules are stored.

Another embodiment provides a computer cluster that includes a plurality of nodes and a software package including a user interface and a single-node kernel for interpreting program code instructions. A cluster node module is configured to communicate with the single-node kernel and other cluster node modules. The cluster node module accepts instructions from the user interface and interprets at least some of the instructions such that several cluster node modules in communication with one another act as a cluster. The cluster node module appears as a single-node kernel to

the user interface. In one embodiment, the single-node kernel includes a Mathematical kernel. In some embodiments, the user interface can include at least one of a Mathematica front end or a command line. In some embodiments, the cluster node module includes a toolkit including library calls that implement at least a portion of MPI calls. In some embodiments, the cluster node module includes a toolkit including high-level cluster computing commands. In one embodiment, the cluster system can include a plurality of Macintosh® computers ("Macs"), Windows®-based personal computers ("PCs"), and/or Unix/Linux-based workstations.

A further embodiment provides a computer cluster including a plurality of nodes. Each node is configured to access a computer-readable medium comprising program code for a user interface and program code for a single-node kernel module configured to interpret user instructions. The cluster includes a plurality of cluster node modules. Each cluster node module is configured to communicate with a single-node kernel and with one or more other cluster node modules, to accept instructions from the user interface, and to interpret at least some of the user instructions such that the plurality of cluster node modules communicate with one another in order to act as a cluster. A communications network connects the nodes. One of the plurality of cluster node modules returns a result to the user interface.

Another embodiment provides a method of evaluating a command on a computer cluster. A command from at least one of a user interface or a script is communicated to one or more cluster node modules within the computer cluster. Each of the one or more cluster node modules communicates a message based on the command to a respective kernel module associated with the cluster node module. Each of the one or more cluster node modules receives a result from the respective kernel module associated with the cluster node module. At least one of the one or more cluster node modules responds to messages from other cluster node modules.

Another embodiment provides a computing system for executing Mathematica code on multiple nodes. The computing system includes a first node module in communication with a first Mathematica kernel executing on a first node, a second node module in communication with a second Mathematica kernel executing on a second node, and a third node module in communication with a third Mathematica kernel executing on a third node. The first node module, the second node module, and the third node module are configured to communicate with one another using a peer-to-peer architecture. In some embodiments, each of the first node module, the second node module, and third node module includes a data structure for maintaining messages originating from other node modules and a data structure for maintaining data specifying a location to which an message is expected to be received and an identifier for a node from which the message is expected to be sent.

BRIEF DESCRIPTION OF THE DRAWINGS

A general architecture that implements the various features are described with reference to the drawings. The drawings and the associated descriptions are provided to illustrate embodiments and not to limit the scope of the disclosure. Throughout the drawings, reference numbers are re-used to indicate correspondence between referenced elements.

FIG. **1** is a block diagram of one embodiment of a computer cluster.

FIG. **2** is a block diagram showing relationships between software modules running on one embodiment of a computer cluster.

FIG. **3** is a block diagram of one embodiment of a cluster node module.

FIG. **4** is a flow chart showing one embodiment of a cluster initialization process.

FIG. **5** is a flow chart showing one embodiment of the operation of a cluster node module.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

For purposes of illustration, some embodiments are described herein in the context of cluster computing with Mathematica software. The present disclosure is not limited to a single software program; the systems and methods can be used with other application software such as, for example, Maple®, MATLAB®, MathCAD®, Apple Shake®, Apple® Compressor, IDL®, other applications employing an interpreter or a kernel, Microsoft Excel®, Adobe After Effects®, Adobe Premiere®, Adobe Photoshop®, Apple Final Cut Pro®, and Apple iMovie®. Some figures and/or descriptions, however, relate to embodiments of computer clusters running Mathematica. The system can include a variety of uses, including but not limited to students, educators, scientists, engineers, mathematicians, researchers, and technicians. It is also recognized that in other embodiments, the systems and methods can be implemented as a single module and/or implemented in conjunction with a variety of other modules. Moreover, the specific implementations described herein are set forth in order to illustrate, and not to limit, the disclosure.

I. Overview

The cluster computing system described herein generally includes one or more computer systems connected to one another via a communications network or networks. The communications network can include one or more of a local area network ("LAN"), a wide area network ("WAN"), an intranet, the Internet, etc. In one embodiment, a computer system comprises one or more processors such as, for example, a microprocessor that can include one or more processing cores ("nodes"). The term "node" refers to a processing unit or subunit that is capable of single-threaded execution of code. The processors can be connected to one or more memory devices such as, for example, random access memory ("RAM"), and/or one or more optional storage devices such as, for example, a hard disk. Communications among the processors and such other devices may occur, for example, via one or more local buses of a computer system or via a LAN, a WAN, a storage area network ("SAN"), and/or any other communications network capable of carrying signals among computer system components. In one embodiment, one or more software modules such as kernels, run on nodes within the interconnected computer systems. In one embodiment, the kernels are designed to run on only a single node. In one embodiment, cluster node modules communicate with the kernels and with each other in order to implement cluster computing functionality.

FIG. **1** is a block diagram of one embodiment of a computer cluster **100** wherein computer systems **110**, **120**, **130** communicate with one another via a communications network **102**. Network **102** includes one or more of a LAN, a WAN, a wireless network, an intranet, or the Internet. In

one embodiment of the computer cluster, computer system **110** includes processors **112***a*, **112***b*, memory **114**, and optional storage **116**. Other computer systems **120**, **130** can include similar devices, which generally communicate with one another within a computer system over a local communications architecture such as a local bus (not shown). A computer system can include one or more processors, and each processor can contain one or more processor cores that are capable of single-threaded execution. Processor cores are generally independent microprocessors, but more than one can be included in a single chip package. Software code designed for single-threaded execution can generally run on one processor core at a time. For example, single-threaded software code typically does not benefit from multiple processor cores in a computer system.

FIG. **2** is a block diagram showing relationships among software modules running on one embodiment of a computer cluster **100**. In the embodiment shown in FIG. **2**, the kernel modules **206***a-e* are designed for single-threaded execution. For example, if each of the processors **112***a*, **112***b*, **122***a*, **122***b*, **132** shown in FIG. **1** includes only one processor core, two kernel modules (for example, kernel modules **206***a*, **206***b*) loaded into the memory **114** of computer system **110** could exploit at least some of the processing bandwidth of the two processors **112***a*, **112***b*. Similarly, two kernel modules **206***c*, **206***d* loaded into the memory **124** of computer system **120** could exploit at least some of the processing bandwidth of the two processors **122***a*, **122***b*. Likewise, the bandwidth of processor **132** of computer system **130** could be utilized by a single instance of a cluster node module **204***e* loaded into the computer system's memory **134**.

In the embodiment shown in FIG. **2**, each of the kernel modules **206***a-e* is in communication with a single cluster node module **204***a-e*, respectively. For example, the kernel module **206***a* is in communication with the cluster node module **204***a*, the kernel module **206***b* is in communication with the cluster node module **206***b*, and so forth. In one embodiment, one instance of a cluster node module **204***a-e* is loaded into a computer system's memory **114**, **124**, **134** for every instance of a kernel module **206***a-e* running on the system. As shown in FIG. **2**, each of the cluster node modules **204***a-e* is in communication with each of the other cluster node modules **204***a-e*. For example, one cluster node module **204***a* is in communication with all of the other cluster node modules **204***b-e*. A cluster node module **204***a* may communicate with another cluster node module **204***b* via a local bus (not shown) when, for example, both cluster node modules **204***a-b* execute on processors **112***a*, **112***b* within the same computer system **110**. A cluster node module **204***a* may also communicate with another cluster node module **204***c* over a communications network **102** when, for example, the cluster node modules **204***a, c* execute on processors **112***a*, **122***a* within different computer systems **110**, **120**.

As shown in FIG. **2**, an optional user interface module **202** such as, for example, a Mathematica front end and/or a command line interface, can connect to a cluster node module **204***a*. The user interface module can run on the same computer system **110** and/or the same microprocessor **112***a* on which the cluster node module **204***a* runs. The cluster node modules **204***a-e* provide MPI calls and/or advanced cluster functions that implement cluster computing capability for the single-threaded kernel modules. The cluster node modules **204***a-e* are configured to look and behave like a kernel module **206***a* from the perspective of the user interface module **202**. Similarly, the cluster node modules **204***a-e* are configured to look and behave like a user interface module **202** from the perspective of a kernel module **206***a*. The first cluster node module **204***a* is in communication with one or more other cluster node modules **204***b*, **204***c*, and so forth, each of which provides a set of MPI calls and/or advanced cluster commands. In one embodiment, MPI may be used to send messages between nodes in a computer cluster.

Communications can occur between any two or more cluster node modules (for example, between a cluster node module **204***a* and another cluster node module **204***c*) and not just between "adjacent" kernels. Each of the cluster node modules **204***a-e* is in communication with respective kernel modules **206***a-e*. Thus, the cluster node module **204***a* communicates with the kernel module **206***a*. MPI calls and advanced cluster commands are used to parallelize program code received from an optional user interface module **208** and distribute tasks among the kernel modules **206***a-e*. The cluster node modules **204***a-e* provide communications among kernel modules **206***a-e* while the tasks are executing. Results of evaluations performed by kernel modules **206***a-e* are communicated back to the first cluster node module **204***a* via the cluster node modules **204***a-e*, which communicates them to the user interface module **208**.

Intercommunication among kernel modules **206***a-e* during thread execution, which is made possible by cluster node modules **204***a-e*, provides advantages for addressing various types of mathematic and scientific problems, for example. Intercommunication provided by cluster computing permits exchange of information between nodes during the course of a parallel computation. Embodiments of the present disclosure provide such intercommunication for software programs such as Mathematica, while grid computing solutions can implement communication between only one master node and many slave nodes. Grid computing does not provide for communication between slave nodes during thread execution.

For purposes of providing an overview of some embodiments, certain aspects, advantages, benefits, and novel features of the invention are described herein. It is to be understood that not necessarily all such advantages or benefits can be achieved in accordance with any particular embodiment of the invention. Thus, for example, those skilled in the art will recognize that the invention can be embodied or carried out in a manner that achieves one advantage or group of advantages as taught herein without necessarily achieving other advantages or benefits as can be taught or suggested herein.

II. Computer Cluster **100**

As shown in FIG. **1**, one embodiment of a cluster system **100** includes computer systems **110**, **120**, **130** in communication with one another via a communications network **102**. A first computer system **110** can include one or more processors **112***a-b*, a memory device **114**, and an optional storage device **116**. Similarly, a second computer system **120** can include one or more processors **122***a-b*, a memory device **124**, and an optional storage device **126**. Likewise, a third computer system **130** can include one or more processors **132**, a memory device **134**, and an optional storage device **136**. Each of the computer systems **110**, **120**, **130** includes a network interface (not shown) for connecting to a communications network **102**, which can include one or more of a LAN, a WAN, an intranet, a wireless network, and/or the Internet.

A. Computer System **110**

In one embodiment, a first computer system **110** communicates with other computer systems **120**, **130** via a network **102** as part of a computer cluster **100**. In one embodiment, the computer system **110** is a personal computer, a workstation, a server, or a blade including one or more processors **112***a-b*, a memory device **114**, an optional storage device **116**, as well as a network interface module (not shown) for communications with the network **102**.

1. Processors **112***a-b*

In one embodiment, the computer system **110** includes one or more processors **112***a-b*. The processors **112***a-b* can be one or more general purpose single-core or multi-core microprocessors such as, for example, a Pentium® processor, a Pentium® II processor, a Pentium® Pro processor, a Pentium® III processor, Pentium® 4 processor, a Core Duo® processor, a Core 2 Duo® processor, a Xeon® processor, an Itanium® processor, a Pentium® M processor, an x86 processor, an Athlon® processor, an 8051 processor, a MIPS® processor, a PowerPC® processor, an ALPHA® processor, etc. In addition, one or more of the processors **112***a-b* can be a special purpose microprocessor such as a digital signal processor. The total number of processing cores (for example, processing units capable of single-threaded execution) within all processors **112***a-b* in the computer system **110** corresponds to the number of nodes available in the computer system **110**. For example, if the processors **112***a-b* were each Core 2 Duo® processors having two processing cores, computer system **110** would have four nodes in all. Each node can run one or more instances of a program module, such as a single-threaded kernel module.

2. Network Interface Module

The computer system **110** can also include a network interface module (not shown) that facilitates communication between the computer system **110** and other computer systems **120**, **130** via the communications network **102**.

The network interface module can use a variety of network protocols. In one embodiment, the network interface module includes TCP/IP. However, it is to be appreciated that other types of network communication protocols such as, for example, Point-to-Point Protocol ("PPP"), Server Message Block ("SMB"), Serial Line Internet Protocol ("SLIP"), tunneling PPP, AppleTalk, etc., may also be used.

3. Memory **114** and Storage **116**

The computer system **110** can include memory **114**. Memory **114** can include, for example, processor cache memory (such as processor core-specific or cache memory shared by multiple processor cores), dynamic random-access memory ("DRAM"), static random-access memory ("SRAM"), or any other type of memory device capable of storing computer data, instructions, or program code. The computer system **110** can also include optional storage **116**. Storage **116** can include, for example, one or more hard disk drives, floppy disks, flash memory, magnetic storage media, CD-ROMs, DVDs, optical storage media, or any other type of storage device capable of storing computer data, instructions, and program code.

4. Computer System **110** Information

The computer system **110** may be used in connection with various operating systems such as: Microsoft® Windows® 3.X, Windows 95®, Windows 98®, Windows NT®, Windows 2000®, Windows XP®, Windows CE®, Palm Pilot OS, OS/2, Apple® MacOS®, MacOS X®, MacOS X Server®, Disk Operating System (DOS), UNIX, Linux®, VxWorks, or IBM® OS/2®, Sun OS, Solaris OS, IRIX OS operating systems, etc.

In one embodiment, the computer system **110** is a personal computer, a laptop computer, a Blackberry® device, a portable computing device, a server, a computer workstation, a local area network of individual computers, an interactive kiosk, a personal digital assistant, an interactive wireless communications device, a handheld computer, an embedded computing device, or the like.

As can be appreciated by one of ordinary skill in the art, the computer system **110** may include various sub-routines, procedures, definitional statements, and macros. Each of the foregoing modules are typically separately compiled and linked into a single executable program. However, it is to be appreciated by one of ordinary skill in the art that the processes that are performed by selected ones of the modules may be arbitrarily redistributed to one of the other modules, combined together in a single module, made available in a shareable dynamic link library, or partitioned in any other logical way.

B. Computer System **120**

In one embodiment, a second computer system **120** communicates with other computer systems **110**, **130** via a network **102** as part of a computer cluster **100**. In one embodiment, the computer system **120** is a personal computer, a workstation, a server, or a blade including one or more processors **122***a-b*, a memory device **124**, an optional storage device **126**, as well as a network interface module (not shown) for communications with the network **102**.

1. Processors **112***a-b*

In one embodiment, the computer system **120** includes one or more processors **122***a-b*. The processors **122***a-b* can be one or more general purpose single-core or multi-core microprocessors such as a Pentium® processor, a Pentium® II processor, a Pentium® Pro processor, a Pentium® III processor, Pentium® 4 processor, a Core Duo® processor, a Core 2 Duo® processor, a Xeon® processor, an Itanium® processor, a Pentium® M processor, an x86 processor, an Athlon® processor, an 8051 processor, a MIPS® processor, a PowerPC® processor, an ALPHA® processor, etc. In addition, the processors **122***a-b* can be any special purpose microprocessors such as a digital signal processor. The total number of processing cores (for example, processing units capable of single-threaded execution) within all processors **122***a-b* in the computer system **120** corresponds to the number of nodes available in the computer system **120**. For example, if the processors **122***a-b* were each Core 2 Duo® processors having two processing cores, computer system **120** would have four nodes in all. Each node can run one or more instances of a program module, such as a single-threaded kernel module.

2. Network Interface Module

The computer system **120** can also include a network interface module (not shown) that facilitates communication between the computer system **120** and other computer systems **110**, **130** via the communications network **102**.

The network interface module can use a variety of network protocols. In one embodiment, the network interface module includes TCP/IP. However, it is to be appreciated that other types of network communication protocols such as, for example, Point-to-Point Protocol ("PPP"), Server Message Block ("SMB"), Serial Line Internet Protocol ("SLIP"), tunneling PPP, AppleTalk, etc., may also be used.

3. Memory **124** and Storage **126**

The computer system **120** can include memory **124**. Memory **124** can include, for example, processor cache memory (such as processor core-specific or cache memory shared by multiple processor cores), dynamic random-access memory ("DRAM"), static random-access memory

("SRAM"), or any other type of memory device capable of storing computer data, instructions, or program code. The computer system **120** can also include optional storage **126**. Storage **126** can include, for example, one or more hard disk drives, floppy disks, flash memory, magnetic storage media, CD-ROMs, DVDs, optical storage media, or any other type of storage device capable of storing computer data, instructions, and program code.

4. Computer System **120** Information

The computer system **120** may be used in connection with various operating systems such as: Microsoft® Windows® 3.X, Windows 95®, Windows 98®, Windows NT®, Windows 2000®, Windows XP®, Windows CE®, Palm Pilot OS, OS/2, Apple® MacOS®, MacOS X®, MacOS X Server®, Disk Operating System (DOS), UNIX, Linux®, VxWorks, or IBM® OS/2®, Sun OS, Solaris OS, IRIX OS operating systems, etc.

In one embodiment, the computer system **120** is a personal computer, a laptop computer, a Blackberry® device, a portable computing device, a server, a computer workstation, a local area network of individual computers, an interactive kiosk, a personal digital assistant, an interactive wireless communications device, a handheld computer, an embedded computing device, or the like.

As can be appreciated by one of ordinary skill in the art, the computer system **120** may include various sub-routines, procedures, definitional statements, and macros. Each of the foregoing modules are typically separately compiled and linked into a single executable program. However, it is to be appreciated by one of ordinary skill in the art that the processes that are performed by selected ones of the modules may be arbitrarily redistributed to one of the other modules, combined together in a single module, made available in a shareable dynamic link library, or partitioned in any other logical way.

C. Computer System **130**

In one embodiment, a third computer system **130** communicates with other computer systems **110**, **120** via a network **102** as part of a computer cluster **100**. In one embodiment, the computer system **130** is a personal computer, a workstation, a server, or a blade including one or more processors **132**, a memory device **134**, an optional storage device **136**, as well as a network interface module (not shown) for communications with the network **102**.

1. Processors **112***a-b*

In one embodiment, the computer system **130** includes a processor **132**. The processor **132** can be a general purpose single-core or multi-core microprocessors such as a Pentium® processor, a Pentium® II processor, a Pentium® Pro processor, a Pentium® III processor, Pentium® 4 processor, a Core Duo® processor, a Core 2 Duo® processor, a Xeon® processor, an Itanium® processor, a Pentium® M processor, an x86 processor, an Athlon® processor, an 8051 processor, a MIPS® processor, a PowerPC® processor, or an ALPHA® processor. In addition, the processor **132** can be any special purpose microprocessor such as a digital signal processor. The total number of processing cores (for example, processing units capable of single-threaded execution) within processor **132** in the computer system **130** corresponds to the number of nodes available in the computer system **130**. For example, if the processor **132** was a Core 2 Duo® processor having two processing cores, the computer system **130** would have two nodes. Each node can run one or more instances of a program module, such as a single-threaded kernel module.

2. Network Interface Module

The computer system **130** can also include a network interface module (not shown) that facilitates communication between the computer system **130** and other computer systems **110**, **120** via the communications network **102**.

The network interface module can use a variety of network protocols. In one embodiment, the network interface module includes TCP/IP. However, it is to be appreciated that other types of network communication protocols such as, for example, Point-to-Point Protocol ("PPP"), Server Message Block ("SMB"), Serial Line Internet Protocol ("SLIP"), tunneling PPP, AppleTalk, etc., may also be used.

3. Memory **134** and Storage **136**

The computer system **130** can include memory **134**. Memory **134** can include, for example, processor cache memory (such as processor core-specific or cache memory shared by multiple processor cores), dynamic random-access memory ("DRAM"), static random-access memory ("SRAM"), or any other type of memory device capable of storing computer data, instructions, or program code. The computer system **130** can also include optional storage **136**. Storage **136** can include, for example, one or more hard disk drives, floppy disks, flash memory, magnetic storage media, CD-ROMs, DVDs, optical storage media, or any other type of storage device capable of storing computer data, instructions, and program code.

4. Computer System **130** Information

The computer system **130** may be used in connection with various operating systems such as: Microsoft® Windows® 3.X, Windows 95®, Windows 98®, Windows NT®, Windows 2000®, Windows XP®, Windows CE®, Palm Pilot OS, OS/2, Apple® MacOS®, MacOS X®, MacOS X Server®, Disk Operating System (DOS), UNIX, Linux®, VxWorks, or IBM® OS/2®, Sun OS, Solaris OS, IRIX OS operating systems, etc.

In one embodiment, the computer system **130** is a personal computer, a laptop computer, a Blackberry® device, a portable computing device, a server, a computer workstation, a local area network of individual computers, an interactive kiosk, a personal digital assistant, an interactive wireless communications device, a handheld computer, an embedded computing device, or the like.

As can be appreciated by one of ordinary skill in the art, the computer system **130** may include various sub-routines, procedures, definitional statements, and macros. Each of the foregoing modules are typically separately compiled and linked into a single executable program. However, it is to be appreciated by one of ordinary skill in the art that the processes that are performed by selected ones of the modules may be arbitrarily redistributed to one of the other modules, combined together in a single module, made available in a shareable dynamic link library, or partitioned in any other logical way.

E. Communications Network **102**

In one embodiment, computer systems **110**, **120**, **130** are in communication with one another via a communications network **102**.

The communications network **102** may include one or more of any type of electronically connected group of computers including, for instance, the following networks: a virtual private network, a public Internet, a private Internet, a secure Internet, a private network, a public network, a value-added network, a wired network, a wireless network, an intranet, etc. In addition, the connectivity to the network can be, for example, a modem, Ethernet (IEEE 802.3), Gigabit Ethernet, 10-Gigabit Ethernet, Token Ring (IEEE 802.5), Fiber Distributed Datalink Interface (FDDI), Frame

Relay, InfiniBand, Myrinet, Asynchronous Transfer Mode (ATM), or another interface. The communications network **102** may connect to the computer systems **110**, **120**, **130**, for example, by use of a modem or by use of a network interface card that resides in each of the systems.

In addition, the same or different communications networks **102** may be used to facilitate communication between the first computer system **110** and the second computer system **120**, between the first computer system **110** and the third computer system **130**, and between the second computer system **120** and the third computer system **130**.

III. Software Modules

As shown in FIGS. **1** and **2**, one embodiment of a cluster system **100** includes a user interface module **202** that is able to access a plurality of kernel modules **206***a-e* by communicating with a first cluster node module **204***a*. User interface module can be stored in a memory **114**, **124**, **134** while running, for example, and/or can be stored in a storage device **116**, **126**, **136**. The first cluster node module **204***a* is in communication with each of the other cluster node modules **204***b-e*. The kernel modules **206***a-e* can reside in the memory of one or more computer systems on which they run. For example, the memory **114** of the first computer system **110** can store instances of kernel modules **206***a-b*, the memory **124** of the second computer system **120** can store instances of kernel modules **206***c-d*, and the memory **134** of the third computer system **130** can store an instance of kernel module **206***e*. The kernel modules **206***a-e*, which include single-threaded program code, are each associated with one of the processors **112***a*, **112***b*, **122***a*, **122***b*, **132**. A cluster configuration module stored on one or more of the computer systems **110**, **120**, **130** or on a remote computer system, for example, can establish communication with the cluster node modules **204***a-e*. In one embodiment, communication between the cluster configuration module **208** and the cluster node modules **204***a-e* initializes the cluster node modules **204***a-e* to provide cluster computing support for the computer cluster **100**.

A. Cluster Node Module **204**

In one embodiment, the cluster node modules **204***a-e* provide a way for many kernel modules **206***a-e* such as, for example, Mathematica kernels, running on a computer cluster **100** to communicate with one another. A cluster node module **204** can include at least a portion of an application programming interface ("API") known as the Message-Passing Interface ("MPI"), which is used in several supercomputer and cluster installations. A network of connections (for example, the arrows shown in FIG. **2**) between the cluster node modules **204***a-e* can be implemented using a communications network **102**, such as, for example, TCP/IP over Ethernet, but the connections could also occur over any other type of network or local computer bus.

A cluster node module **204** can use an application-specific toolkit or interface such as, for example, Mathematica's MathLink, Add-Ons, or packets, to interact with an application. Normally used to connect a Mathematica kernel to a user interface known as the Mathematica Front End or other Mathematica kernels, MathLink is a bidirectional protocol to sends "packets" containing messages, commands, or data between any of these entities. MathLink does not allow direct cluster computing-like simultaneous communication between Mathematica kernels during execution of a command or thread. MathLink is also not designed to perform multiple simultaneous network connections. In some embodiments, a cluster node module **204** can use an application-specific toolkit such as, for example, MathLink, for connections between entities on the same computer.

When speaking about procedures or actions on a cluster or other parallel computer, not all actions happen in sequential order, nor are they required to. For example, a parallel code, as opposed to a single-processor code of the classic "Turing machine" model, has multiple copies of the parallel code running across the cluster, typically one for each processor (or "processing element" or "core"). Such parallel code is written in such a way that different instances of the same code can communicate, collaborate, and coordinate work with each other. Multiple instances of these codes can run at the same time in parallel.

If the count of the code instances is an integer N, each instance of code execution can be labeled 0 through N−1. For example, a computer cluster can include N connected computers, each containing a processor. The first has cluster node module 0 connected with kernel module 0 running on processor 0. The next is cluster node module 1 and kernel module 1, on processor 1, and so forth for each of the N connected computers. Some steps of their procedure are collaborative, and some steps are independent. Even though these entities are not necessarily in lock-step, they do follow a pattern of initialization, main loop behavior (for example, cluster node module operation), and shut down.

In contrast, a parallel computing toolkit (PCT) that is provided as part of the gridMathematica software package does not provide a means for instances of the same code running on different nodes to communicate, collaborate, or coordinate work among the instances. The PCT provides commands that connect Mathematica kernels in a master-slave relationship rather than a peer-to-peer relationship as enabled by some embodiments disclosed herein. A computer cluster having peer-to-peer node architecture performs computations that can be more efficient, easier to design, and/or more reliable than similar computations performed on grid computers having master-slave node architecture. Moreover, the nature of some computations may not allow a programmer to harness multi-node processing power on systems that employ master-slave node architecture.

FIG. **3** shows one embodiment of a cluster node module **204** implementing MPI calls and advanced MPI functions. In the embodiment shown in FIG. **3**, cluster node module **204** includes MPI module **302**, advanced functions module **304**, received message queue **306**, and message receiving queue **308**.

1. MPI module **302**

In one embodiment, the cluster node module **204** includes an MPI module **302**. The MPI module **302** can include program code for one or more of at least five kinds of MPI instructions or calls. Selected constants, instructions, and/or calls that can be implemented by the MPI module **302** are as follows:

MPI Constants

Node identifiers are used to send messages to nodes or receive messages from them. In MPI, this is accomplished by assigning each node a unique integer ($IdProc) starting with 0. This data, with a knowledge of the total count ($NProc), makes it possible to programmatically divide any measurable entity.

TABLE A

Constant	Description
$IdProc	The identification number of the current processor
$NProc	The number of processors in the current cluster

TABLE A-continued

Constant	Description
$mpiCommWorld	The communicator world of the entire cluster (see MPI Communicator routines, below)
mpiCommWorld	The default communicator world for the high-level routines.

Basic MPI Calls

In one embodiment, the MPI module **302** can include basic MPI calls such as, for example, relatively low-level routines that map MPI calls that are commonly used in other languages (such as C and Fortran), so that such calls can be available directly from the Mathematica user interface **204**. In some embodiments, basic MPI calls include calls that send data, equations, formulas, and/or other expressions.

Simply sending expressions from one node to another is possible with these most basic MPI calls. One node can call to send an expression while the other calls a corresponding routine to receive the sent expression. Because it is possible that the receiver has not yet called mpiRecv even if the message has left the sending node, completion of mpiSend is not a confirmation that it has been received.

TABLE B

Call	Description
mpiSend[expr, target, comm, tag]	Sends an expression expr to a node with the ID target in the communicator would comm, waiting until that expression has left this kernel
mpiRecv [expr, target, comm, tag]	Receives an expression into expr from a node with the ID target in the communicator world comm, waiting until the expression has arrived
mpiSendRecv[sendexpr, dest, recvexpr, source, comm]	Simultaneously sends the expression sendexpr to the node with the ID target and receives an expression into recvexpr from the node with the ID source in the communicator world comm, waiting until both operations have returned.

Asynchronous MPI Calls

Asynchronous calls make it possible for the kernel to do work while communications are proceeding simultaneously. It is also possible that another node may not be able to send or receive data yet, allowing one kernel to continue working while waiting.

TABLE C

Call	Description
mpiISend[expr, target, comm, tag, req]	Sends an expression expr to a processor with the ID target in the communicator world comm, returning immediately. It can be balanced with calls to mpiTest[req] until mpiTest[req] returns True.
mpiIRecv[expr, target, comm, tag, req]	Receives an expression expr from a proessor with the ID target in the communicator world comm, returning immediately. It can be balanced with calls to mpiTest[req] until mpiTest[req] returns True. The expr is not safe to access until mpiTest[req] returns True.
mpiTest[req]	Completes asynchronous behavior of mpiISend and mpiIRecv
mpWait[req]	Calls mpiTest until it returns True.
mpiWaitall[reglist]	Calls mpiWait all on every element of reqlist
mpiWaitany[reqlist]	Calls mpiTest on each element of realist until one of them returns True

The mpilSend[ ] command can be called from within a kernel module **206** (for example, a Mathematica kernel). It creates a packet containing the Mathematica expression to be sent as payload and where the expression should be sent. The packet itself is destined only for its local cluster node module. Once received by its local cluster node module, this packet is decoded and its payload is forwarded on to the cluster node module specified in the packet.

The mpilRecv[ ] command can also be called from within a kernel module **206**. It creates a packet specifying where it expects to receive an expression and from which processor this expression is expected. Once received by its local cluster node module, this packet is decoded and its contents are stored in a message receiving queue (MRQ) **308** (FIG. **3**).

The mpiTest[ ] command can be called from within a kernel module **206**. It creates a packet specifying which message to test for completion, then waits for a reply expression to evaluate. Once received by the kernel module's associated cluster node module **204**, this packet is decoded and its message specifier is used to search for any matching expressions listed as completed in its received message queue (RMQ) **306**. If such completed expressions are found, it is sent to its local kernel module as part of the reply in mpiTest[ ]. The kernel module receives this reply expression and evaluates it, which updates the kernel module's variables as needed.

Other MPI calls are built on the fundamental calls mpilSend, mpilRecv, and mpiTest. For example, mpiBcast, a broadcast, creates instructions to send information from the broadcast processor to all the others, while the other processors perform a Recv. Similarly, high-level calls of the toolkit can be built on top of the collection of MPI calls.

Collective MPI Calls

In one embodiment, the MPI module **302** can include program code for implementing collective MPI calls (for example, calls that provide basic multi-node data movement across nodes). Collective MPI calls can include broadcasts, gathers, transpose, and other vector and matrix operations, for example. Collective calls can also provide commonly used mechanisms to send expressions between groups of nodes.

TABLE D

Call	Description
mpiBcast[expr, root, comm]	Performs a broadcast of expr from the root processor to all the others in the communicator world comm. An expression is expected to be supplied by the root processor, while all the others expect expr to be overwritten by the incoming expression.
mpiGather[sendexpr, recvexpr, root, comm]	All processors (including root) in the communicator comm send their expression in sendexpr to the root processor, which produces a list of these expressions, in the order according to comm, in recvexpr. On the processors that are not root, recvexpr is ignored.
mpiAllgather[sendexpr, recvexpr, comm]	All processors in the communicator comm send their expression in sendexpr, which are organized into a list of these expressions, in the order according to comm, in recvexpr on all processors in comm.
mpiScatter[sendexpr, recvexpr, root, comm]	Processor root partitions the list in sendexpr into equal parts (if possible) and places each piece in recvexpr on all the processors (including root) in the communicator world comm, according the order and size of comm.

TABLE D-continued

Call	Description
mpiAlltoall[sendexpr, recvexpr, comm]	Each processor sends equal parts of the list in sendexpr to all other processors in the communicator world comm, which each collects from all other processors are organizes into the order according to comm.

In one embodiment, the MPI module **302** includes program code for implementing parallel sums and other reduction operations on data stored across many nodes. MPI module **302** can also include program code for implementing simple parallel input/output calls (for example, calls that allow cluster system **200** to load and store objects that are located on a plurality of nodes).

TABLE E

Call	Description
mpiReduce[sendexpr, recvexpr, operation, root, comm]	Performs a collective reduction operation between expressions on all processors in the communicator world comm for every element in the list in sendexpr returning the resulting list in recvexpr on the processor with the ID root.
mpiAllreduce[sendexpr, recvexpr, operation, comm]	Performs a collective reduction operation between expressions on all processors in the communicator world comm for every element in the list in sendexpr returning the resulting list in recvexpr on every processor.
mpiReduceScatter [sendexpr, recvexpr, operation, comm]	Performs a collective reduction operation between expressions on all processors in the communicator world comm for every element in the list in sendexpr, partitioning the resulting list into pieces for each processor's recvexpr.

These additional collective calls perform operations that reduce the data in parallel. The operation argument can be one of the constants below.

TABLE F

Constant	Description
mpiSum	Specifies that all the elements on different processors be added together in a reduction call
mpiMax	Specifies that the maximum of all the elements on different processors be chosen in a reduction call
mpiMin	Specifies that the minimum of all the elements on different processors be chosen in a reduction call

MPI Communicator Calls

In one embodiment, the MPI module **302** includes program code for implementing communicator world calls (for example, calls that would allow subsets of nodes to operate as if they were a sub-cluster). Communicators organize groups of nodes into user-defined subsets. The communicator values returned by mpiCommSplit[ ] can be used in other MPI calls instead of mpiCommWorld.

TABLE G

Call	Description
mpiCommSize[comm]	Returns the number of processors within the communicator comm
mpiCommRank[comm]	Returns the rank of this processor in the communicator comm

TABLE G-continued

Call	Description
mpiCommDup[comm]	Returns a duplicate communicator of the communicator comm
mpiCommSplit[comm, color, key]	Creates a new communicator into several disjoint subsets each identified by color. The sort order within each subset is first by key, second according to the ordering in the previous communicator. Processors not meant to participate in any new communicator indicates this by passing the constant mpiUndefined. The corresponding communicator is returned to each calling processor.
mpiCommMap[comm]	Returns the mapping of the communicator comm to the processor indexed according to $mpiCommWorld. Adding a second argument returns just the ID of the processor with the ID target in the communicator comm.
mpiCommMap[comm, target]	
mpiCommFree[comm]	Frees the communicator comm

Other MPI Support Calls

Other calls that provide common functions include:

TABLE H

Call	Description
mpiWtime[]	Provides wall-dock time since some fixed time in the past. There is no guarantee that this time will read the same on all processors.
mpWtick[]	Returns the time resolution of mpiWtime[]
MaxByElement[in]	For every nth element of each list of the list in, chooses the maximum according to Max[], and returns the result as one list. Used in the mpiMax reduction operation.
MinByElement[in]	For every nth element of each list of the list in, chooses the minimum according to Min[], and returns the result as one list. Used in the mpiMin reduction operation.

2. Advanced Functions Module **304**

In one embodiment, the cluster node module **204** includes an advanced functions module **304**. The advanced functions module **304** can include program code that provides a toolkit of functions inconvenient or impractical to do with MPI instructions and calls implemented by the MPI module **302**. The advanced functions module **304** can rely at least partially on calls and instructions implemented by the MPI module **302** in the implementation of advanced functions. In one embodiment, the advanced functions module **304** includes a custom set of directives or functions. In an alternative embodiment, the advanced functions module **304** intercepts normal Mathematica language and converts it to one or more functions optimized for cluster execution. Such an embodiment can be easier for users familiar with Mathematica functions to use but can also complicate a program debugging process. Some functions implemented by the advanced functions module **304** can simplify operations difficult or complex to set up using parallel computing. Several examples of such functions that can be implemented by the advanced functions module **304** are shown below.

Built on the MPI calls, the calls that are described below provide commonly used communication patterns or parallel versions of Mathematica features. Unless otherwise specified, these are executed in the communicator mpiCommWorld, whose default is $mpiCommWorld, but can be changed to a valid communicator at run time.

Common Divide-and-Conquer Parallel Evaluation

In one embodiment, the advanced functions module **304** includes functions providing for basic parallelization such as, for example, routines that would perform the same operations on many data elements or inputs, stored on many nodes. These functions can be compared to parallelized for-loops and the like. The following calls address simple parallelization of common tasks. In the call descriptions, "expr" refers to an expression, and "loopspec" refers to a set of rules that determine how the expression is evaluated. In some embodiments, the advanced functions module **304** supports at least three forms of loopspec, including {var, count}, where the call iterates the variable var from 1 to the integer count; {var, start, stop}, where the call iterates the variable var every integer from start to stop; and {var, start, stop, increment}, where the call iterates the variable var from start adding increment for each iteration until var exceeds stop, allowing var to be a non-integer.

TABLE I

Call	Description
ParallelDo[expr, loopspec]	Like Do[] except that it evaluates expr across the cluster, rather than on just one processor. The rules for how expr is evaluated is specified in loopspec, like in Do[].
ParallelFunctionToList[f, count]	Evaluates the function f[i] from 1 to count, but across the cluster, and returns these results in a list. The third argument has it gather this list into the processor whose ID is root.
ParallelFunctionToList[f, count, root]	
ParallelTable[expr, loopspec]	Like Table[] except that it evaluates expr across the cluster, rather than on just one processor, returning the locally evalulated portion. The third argument has it gather this table in to the processor whose ID is root.
ParallelTable[expr, loopspec, root]	
ParallelFunction[f, inputs, root]	Like f[inputs] except that it evaluates f on a subset of inputs scattered across the cluster from processor root and gathered back to root.
ParallelNintegrate[expr, loopspec]	Like Nintegrate[] except that it evaluates a numerical integration of expr over domains partitioned into the number of processors in the cluster, then returns the sum. The third argument has each numerical integration execute with at least that many digits of precision.
ParallelNintegrate[expr, loopspec, digits]	

Guard-Cell Management

In one embodiment, the advanced functions module **304** includes functions providing for guard-cell operations such as, for example, routines that perform nearest-neighbor communications to maintain edges of local arrays in any number of dimensions (optimized for 1-, 2-, and/or 3-D). Typically the space of a problem is divided into partitions. Often, however, neighboring edges of each partition can interact, so a "guard cell" is inserted on both edges as a substitute for the neighboring data. Thus the space a processor sees is two elements wider than the actual space for which the processor is responsible. EdgeCell helps maintain these guard cells.

TABLE J

Call	Description
EdgeCell[list]	Copies the second element of list to the last element of the left processor and the second-to-last element of list to the first element of the right processor while simultaneously receiving the same from its neighbors.

Matrix and Vector Manipulation

The advanced functions module **304** can also include functions providing for linear algebra operations such as, for example, parallelized versions of basic linear algebra on structures partitioned on many nodes. Such linear algebra operations can reorganize data as needed to perform matrix and vector multiplication or other operations such as determinants, trace, and the like. Matrices are partitioned and stored in processors across the cluster. These calls manipulate these matrices in common ways.

TABLE K

Call	Description
ParallelTranspose[matrix]	Like Transpose[] except that it transposes matrix that is in fact represented across the cluster, rather than on just one processor. It returns the portion of the transposed matrix meant for that processor.
ParallelProduct[matrix, vector]	Evaluates the product of matrix and vector, as it would on one processor, except that matrix is represented across the cluster.
ParallelDimensions[matrix]	Like Dimensions[] except that matrix is represented across the cluster, rather than on just one processor. It returns a list of each dimension.
ParallelTr[matrix]	Like Tr[] except that the matrix is represented across the cluster, rather than on just one processor. It returns the trace of this matrix.
ParallelIdentity[rank]	Like Identity[], it generates a new identity matrix, except that the matrix is represented across the cluster, rather than on just one processor. It returns the portion of the new matrix for this processor.
ParallelOuter[f, vector1, vector2]	Like Outer[f, vector1, vector2] except that the answer becomes a matrix represented across the cluster, rather than on just one processor. It returns the portion of the new matrix for this processor.
ParallelInverse[matrix]	Like Inverse[] except that the matrix is represented across the cluster, rather than on just one processor. It returns the inverse of the matrix.

Element Management

In one embodiment, the advanced functions module **304** includes element management operations. For example, a large bin of elements or particles cut up in space across the nodes may need to migrate from node to node based on rules or criteria (such as their spatial coordinate). Such operations would migrate the data from one node to another. Besides the divide-and-conquer approach, a list of elements can also be partitioned in arbitrary ways. This is useful if elements need to be organized or sorted onto multiple processors. For example, particles of a system may drift out of the space of one processor into another, so their data would need to be redistributed periodically.

TABLE L

Call	Description
ElementManage[list, switch]	Selects which elements of list will be sent to which processors according to the function switch[] is evaluated on each element of list. If switch is a function, switch[] should return the ID of the processor that element should be sent. If switch is an integer, the call assumes that each elements is itself a list, whose first element is a number ranging from 0 to the passed argument. This call returns a list of the elements, from any processor, that is switch selected for this processor.
ElementManage[list]	Each element of list can be a list of two elements, the first being the ID of the processor where the element should be sent, while the second is arbitrary data to send. This call returns those list elements, from any and all processors, whose first element is this processors ID in a list. This call is used internally by the two-argument version of ElementManage[].

Fourier Transform

In one embodiment, the advanced functions module **304** includes program code for implementing large-scale parallel fast Fourier transforms ("FFTs"). For example, such functions can perform FFTs in one, two, and/or three dimensions on large amounts of data that are not stored on one node and that are instead stored on many nodes. Fourier transforms of very large arrays can be difficult to manage, not the least of which is the memory requirements. Parallelizing the Fourier transform makes it possible to make use of all the memory available on the entire cluster, making it possible to manipulate problem sizes that no one processor could possibly do alone.

TABLE M

Call	Description
ParallelFourier[list]	Like Fourier[] except that list is a two- or three-dimensional list represented across the cluster, like for matrices, above. It returns the portion of the Fourier-transformed array meant for that processor.

Parallel Disk I/O

In one embodiment, the advanced functions module **304** includes parallel disk input and output calls. For example, data may need to be read in and out of the cluster in such a way that the data is distributed across the cluster evenly. The calls in the following table enable the saving data from one or more processors to storage and the retrieval data from storage.

TABLE N

Call	Description
ParallelPut[expr, filename] ParallelPut[expr, filename root] ParallelPut[expr, filename, root, comm]	Puts expr into the file with the name filename in order on processor 0. The third argument specifies that the file be written on the processor whose ID is root. The fourth uses the communicator world comm.
ParallelGet[filename] ParallelGet[filename, root] ParallelGet[filename, root, comm]	Reads and returns data from the file with the name filename on processor 0 partitioned into each processor on the cluster. The second argument specifies that the file is to be read on the processor whose ID is root. The third uses the communicator world comm.
ParallelBinaryPut[expr, type, filename] ParallelBinaryPut[expr, filename, root] ParallelBinaryPut[expr, filename, root, comm]	Puts expr into the file with the binary format type with the name filename in order on processor 0. The fourth argument specifies that the file be written on the processor whose ID is root. The fifth uses the communicator world comm.
ParallelBinaryGet[type, filename] ParallelBinaryGet[type, filename, root] ParallelBinaryGet[type, filename, root, comm]	Reads and returns data in the binary format type from the file with the name filename on processor 0 partitioned into each processor on the cluster. The third argument specifies that the file is to be read on the processor whose ID is root. The fourth uses the communicator world comm.
ParallelGetPerProcessor [expr, filename] ParallelGetPerProcessor [filename, root] ParallelGetPerProcessor [filename, root, comm]	Puts expr into the file with the name filename in order on processor 0, one line per processor. The third argument specifies that the file be written on the processor whose ID is root. The fourth uses the communicator world comm.

TABLE N-continued

Call	Description
ParallelGetPerProcessor [filename] ParallelGetPerProcessor [filename, root] ParallelGetPerProcessr [filename, root, comm]	Reads and returns data from the file with the name filename on processor 0, one line for each processor. The second argument specifies that the file is to be read on the processor whose ID is root. The third uses the communicator world comm.

Automatic Load Balancing

Some function calls can take an inconsistent amount of processing time to complete. For example, in Mathematica, the call f[20] could in general take much longer to evaluate than f[19]. Moreover, if one or more processors within the cluster are of different speeds (for example, if some operate at a core frequency of 2.6 GHz while other operate at less than one 1 GHz), one processor may finish a task sooner than another processor.

In some embodiments, the advanced functions module **304** includes a call that can improve the operation of the computer cluster **100** in such situations. In some embodiments, the root processor assigns a small subset of the possible calls for a function to each processor on the cluster **100**. Whichever processor returns its results first is assigned a second small subset of the possible calls. The root processor will continue to assign small subsets of the possible calls as results are received until an evaluation is complete. The order in which the processors finish can vary every time an expression is evaluated, but the root processor will continue assigning additional work to processors as they become available.

In one illustrative example, there are 4 processors and f[1] to f[100] to evaluate. One could implement this by assigning f[1], f[2], f[3], f [4] to each of processors 0 (the root can assign to oneself) through 3. If the f[2] result came back first, then processor 1 would be assigned f[5]. If the f[4] result is returned next, f[6] would be assigned to processor 3. The assignments continue until all results are calculated. The results are organized for output back to the user.

In alternative embodiments, the subsets of possible calls can be assigned in any order, rather than sequentially, or in batches (for example, f[1], f[5], f[9] assigned to processor 1, etc.). Also, the subsets could be organized by delegation. For example, one processor node may not necessarily be in direct control of the other processors. Instead, a large subset could be assigned to a processor, which would in turn assign subsets of its work to other processors. The result would create a hierarchy of assignments like a vast army.

TABLE O

Call	Description
LoadBalanceFunctionToList[f, count] LoadBalanceFunctionToList[f, count, root]	Evaluates the function f[i] from 1 to count, but across the cluster using load-balancing techniques, and returns these results in a list. The third argument has it gather this list into the processor whose ID is root.

3. Received Message Queue **306**

In one embodiment, the cluster node module **204** includes a received message queue **306**. The received message queue **306** includes a data structure for storing messages received from other cluster node modules. Related data pertaining to the messages received, such as whether an expression has been completed, may also be stored in the received message queue **306**. The received message queue **306** may include a queue and/or another type of data structure such as, for example, a stack, a linked list, an array, a tree, etc.

4. Message Receiving Queue **308**

In one embodiment, the cluster node module **204** includes a message receiving queue **308**. The message receiving queue **308** includes a data structure for storing information about the location to which an expression is expected to be sent and the processor from which the expression is expected. The message receiving queue **308** may include a queue and/or another type of data structure such as, for example, a stack, a linked list, an array, a tree, etc.

B. Cluster Configuration Module **208**

Cluster configuration module **208** includes program code for initializing a plurality of cluster node modules to add cluster computing support to computer systems **110**, **120**, **130**. U.S. Pat. No. 7,136,924, issued to Dauger (the "'924 patent"), the entirety of which is hereby incorporated by reference and made a part of this specification, discloses a method and system for parallel operation and control of computer clusters. One method generally includes obtaining one or more personal computers having an operating system with discoverable network services. In some embodiments, the method includes obtaining one or more processors or processor cores on which a kernel module can run. As described in the '924 patent, a cluster node control and interface (CNCI) group of software applications is copied to each node. When the CNCI applications are running on a node, the cluster configuration module **208** can permit a cluster node module **204**, in combination with a kernel module **206**, to use the node's processing resources to perform a parallel computation task as part of a computer cluster. The cluster configuration module **208** allows extensive automation of the cluster creation process in connection with the present disclosure.

C. User Interface Module **202**

In some embodiments, computer cluster **100** includes a user interface module **202**, such as, for example a Mathematica Front End or a command line interface, that includes program code for a kernel module **206** to provide graphical output, accept graphical input, and provide other methods of user communication that a graphical user interface or a command-line interface provides. To support a user interface module **202**, the behavior of a cluster node module **204***a* is altered in some embodiments. Rather than sending output to and accepting input from the user directly, the user interface module **202** activates the cluster node module **204***a* to which it is connected and specifies parameters to form a connection, such as a MathLink connection, between the cluster node module **204***a* and the user interface module **202**. The user interface module's activation of the cluster node module **204***a* can initiate the execution of instructions to activate the remaining cluster node modules **204***b-e* on the cluster and to complete the sequence to start all kernel modules **206***a-e* on the cluster. Packets from the user interface

module **202**, normally intended for a kernel module **206***a*, are accepted by the cluster node module **204***a* as a user command. Output from the kernel module **206***a* associated with the cluster node module **204***a* can be forwarded back to the user interface module **202** for display to a user. Any of the cluster node modules **204***a-e* can be configured to communicate with a user interface module **202**.

D. Kernel Module **206**

A kernel module **206** typically includes program code for interpreting high-level code, commands, and/or instructions supplied by a user or a script into low-level code, such as, for example, machine language or assembly language. In one embodiment, each cluster node module **204***a-e* is connected to all other cluster node modules, while each kernel module **206***a-e* is allocated and connected only to one cluster node module **204**. In one embodiment, there is one cluster node module-kernel module pair per processor. For example, in an embodiment of a computer cluster **100** including single-processor computer systems, each cluster node module-kernel module pair could reside on a single-processor computer. If a computer contains multiple processors or processing cores, it may contain multiple cluster node module-kernel module pairs, but the pairs can still communicate over the cluster node module's network connections.

IV. Cluster Computing Methods

In one embodiment, the computer cluster **100** includes a cluster initialization process, a method of cluster node module operation, and a cluster shut down process.

A. Cluster Initialization Process

In one embodiment, a cluster configuration module **202** initializes one or more cluster node modules **204** in order to provide cluster computing support to one or more kernel modules **206**, as shown in FIG. **4**.

At **402**, cluster node modules are launched on the computer cluster **100**. In one embodiment, the cluster node module **204***a* running on a first processor **112***a* (for example, where the user is located) accesses the other processors **112***b*, **122***a-b*, **132** on the computer cluster **100** via the cluster configuration module **208** to launch cluster node modules **204***b-e* onto the entire cluster. In an alternative embodiment, the cluster configuration module **208** searches for processors **112***a-b*, **122***a-b*, **132** connected to one another via communications network **102** and launches cluster node modules **204***a-e* on each of the processors **112***a-b*, **122***a-b*, **132**.

The cluster node modules **204***a-e* establish communication with one another at **404**. In one embodiment, each of the cluster node modules **204***a-e* establish direct connections using the MPI_Init command with other cluster node modules **204***a-e* launched on the computer cluster **100** by the cluster configuration module **208**.

At **406**, each cluster node module **204** attempts to connect to a kernel module **206**. In one embodiment, each instance of the cluster node modules **204***a-e* locates, launches, and connects with a local kernel module via MathLink connections and/or similar connection tools, for example, built into the kernel module **206**.

At **408**, the cluster node modules **204** that are unconnected to a kernel module **206** are shut down. In one embodiment, each cluster node module **204** determines whether the local kernel module cannot be found or connected to. In one embodiment, each cluster node module **204** reports the failure to connect to a kernel module **206** to the other cluster node modules on computer cluster **100** and quits.

Processor identification numbers are assigned to the remaining cluster node modules **204** at **410**. In one embodiment, each remaining cluster node module **204** calculates the total number of active processors (N) and determines identification numbers describing the remaining subset of active cluster node modules **204***a-e* and kernel modules **206***a-e*. This new set of cluster node module-kernel module pairs may be numbered 0 through N−1, for example.

Message passing support is initialized on the kernel modules **206***a-e* at **412**. In one embodiment, each cluster node module **204** supplies initialization code (for example, Mathematica initialization code) to the local kernel module **206** to support message passing.

Finally, at **414**, the cluster node modules **204***a-e* enter a loop to accept user entry. In one embodiment, a main loop (for example, a cluster operation loop) begins execution after the cluster node module **204***a* on the first processor **112***a* returns to user control while each of the other cluster node modules **204** waits for messages from all other cluster node modules **204***a-e* connected to the network **102**.

The initialization process creates a structure enabling a way for the kernel modules **206***a-e* to send messages to one another. In some embodiments, any kernel module can send data to and receive data from any other kernel module within the cluster when initialization is complete. The cluster node module creates an illusion that a kernel module is communicating directly with the other kernel modules. The initialization process can create a relationship among kernel modules on a computer cluster **100** such as the one shown by way of example in FIG. **2**.

B. Cluster Node Module Operation

In one embodiment, a cluster node module **204** implements cluster computing support for a kernel module **206** during a main loop, as shown in FIG. **5**.

At **502**, cluster node modules **204** wait for user commands or messages from other cluster node modules. In one embodiment, the cluster node module **204***a* connected to the user interface module **202** waits for a user command, while the other cluster node modules **204***b-e* continue checking for messages.

Once a command or message is received, the method proceeds to **504**. At **504**, the cluster node module **204***a* determines whether the message received is a quit command. If a quit command is received, the cluster node module **204***a* exits the loop and proceeds to a cluster node module shut down process at **505**. If the message received is not a quit command, the process continues to **506**.

At **506**, received commands are communicated to all cluster node modules **204***a-e* on the computer cluster **100**. In one embodiment, when a user enters a command in the user interface module **202**, the cluster node module **204***a* connected to the user interface module **202** submits the user command to all other cluster node modules **204***b-e* in the computer cluster **100**. The user commands can be simple (for example, "1+1"), but can also be entire subroutines and sequences of code (such as, for example, Mathematica code), including calls to MPI from within the user interface module **202** (for example, the Mathematica Front End) to perform message passing between kernel modules **206***a-e* (for example, Mathematica kernels). These include the fundamental MPI calls, which are implemented using specially identified messages between a cluster node module **204** and its local kernel module **206**.

The message (or user command) is communicated to the kernel modules **206***a-e* at **508**. In one embodiment, the cluster node module **204***a* connected to the user interface module **202** submits the user command to the kernel module **206***a* to which it is connected. Each of the other cluster node modules **204***b-e*, after receiving the message, submits the command to the respective kernel module **206***b-e* to which it is connected.

At **510**, a cluster node module **204** receives a result from a kernel module **206**. In one embodiment, once the kernel module **206** completes its evaluation, it returns the kernel module's output to the cluster node module **204** to which it is connected. Depending on the nature of the result from the kernel module, the cluster node module **204** can report the result to a local computer system or pass the result as a message to another cluster node module **204**. For example, the cluster node module **204***a* running on the first processor **112***a* reports the output on its local computer system **110**. For example, on the first processor **112***a*, cluster node module **204***a* only directly reports the output of kernel module **206***a*.

Messages from other cluster node modules **204** are responded to at **512**. In one embodiment, each cluster node module (for example, the cluster node module **204***a*) checks for and responds to messages from other cluster node modules **204***b-e* and from the kernel module **206***a* repeatedly until those are exhausted. In one embodiment, output messages from the kernel module **206** are forwarded to output on the local computer system. Messages from other cluster node modules **204** are forwarded to a received message queue **306** ("RMQ"). Data from each entry in the message receiving queue **308** ("MRQ") is matched with entries in the RMQ **306** (see, for example, description of the mpilRecv[ ] call, above). If found, data from the MRQ **308** are combined into those in the RMQ **306** and marked as "completed" (see, for example, description of the mpiTest[ ] call, above). This process provides the peer-to-peer behavior of the cluster node modules **204***a-e*. Via this mechanism, code running within multiple, simultaneously running kernel modules (for example, Mathematica kernels) can interact on a pair-wise or collective basis, performing calculations, processing, or other work on a scale larger and/or faster than one kernel could have done alone. In this manner, user-entered instructions and data specifying what work will be done via user commands can be executed more quickly and/or reliably. Once responding to messages has completed, the process returns to **502**.

In some embodiments, a computer system includes software, such as an operating system, that divides memory and/or other system resources into a user space, a kernel space, an application space (for example, a portion of the user space allocated to an application program), and/or an operating system space (for example, a portion of the user space allocated to an operating system). In some embodiments, some or all of the cluster node modules **204***a-e* are implemented in the application space of a computer system. In further embodiments, at least some of the cluster node modules **204***a-e* are implemented in the operating system space of a computer system. For example, some cluster node modules in a computer cluster may operate in the application space while others operate in the operating system space.

In some embodiments, some or all of the functionality of the cluster node modules **204***a-e* is incorporated into or integrated with the operating system. The operating system can add cluster computing functionality to application programs, for example, by implementing at least some of the methods, modules, data structures, commands, functions, and processes discussed herein. Other suitable variations of the techniques described herein can be employed, as would be recognized by one skilled in the art.

In some embodiments, the operating system or components of the operating system can identify and launch the front end **202** and the kernels **206**. The operating system or its components can connect the front end **202** and kernels **206** to one another in the same manner as a cluster node module **204** would or by a variation of one of the techniques described previously. The operating system can also be responsible for maintaining the communications network **102** that connects the modules to one another. In some embodiments, the operating system implements at least some MPI-style calls, such as, for example, collective MPI-style calls. In some embodiments, the operating system includes an application programming interface (API) library of cluster subroutine calls that is exposed to application programs. Applications programs can use the API library to assist with launching and operating the computer cluster.

C. Cluster Shut Down Process

In one embodiment, a computer cluster **100** includes a procedure to shut down the system. If the operation process (or main loop) on the cluster node module **204***a* connected to the user interface module **202** detects a "Quit" or "Exit" command or otherwise receives a message from the user indicating a shut down, the sequence to shut down the cluster node modules **204***a-e* and the kernel modules **206***a-e* is activated. In one embodiment, the cluster node module **204***a* connected to the user interface module **202** sends a quit message to all other cluster node modules **204***b-e*. Each cluster node module **204** forwards the quit command to its local kernel module **206**. Once its Mathematica kernel has quit, each cluster node module **204** proceeds to tear down its communication network with other cluster node modules (for example, see description of the MPI_Finalize command, above). At the conclusion of the process, each cluster node module **204** exits execution.

V. Example Operation

For purposes of illustration, sample scenarios are discussed in which the computer cluster system is used in operation. In these sample scenarios, examples of Mathematica code are given, and descriptions of how the code would be executed by a cluster system are provided.

Basic MPI

Fundamental data available to each node includes the node's identification number and total processor count.

```
In[1]:= {$IdProc, $NProc}
Out[1]:= {0, 2}
```

The first element should be unique for each processor, while the second is generally the same for all. Processor 0 can see what other values are using a collective (see below) communications call such as mpiGather[ ].

```
In[2]:= mpiGather[{$IdProc, $NProc},list,0]; list
Out[2]:= {{0, 2}, {1, 2}}
```

Peer-to-Peer MPI

The mpiSend and mpiRecv commands make possible basic message passing, but one needs to define which

processor to target. The following defines a new variable, targetProc, so that each pair of processors will point to each other.

```
In[3]:= targetProc=If[1==Mod[$IdProc, 2],$IdProc-1,$IdProc+l]
Out[3]:= 1
```

In this example, the even processors target its "right" processor, while the odd ones point its "left." For example, if the processors were lined up in a row and numbered in order, every even-numbered processor would pair with the processor following it in the line, and every odd-numbered processor would pair with the processor preceding it. Then a message can be sent:

```
In[4]:= If [ 1==Mod[ $IdProc , 2],mpiSend[N[Pi,22],targetProc,
    mpiCommWorld,d], mpiRecv[a,targetProc,mpiCommWorld,d]]
```

The If [ ] statement causes the processors to evaluate different code: the odd processor sends 22 digits of Pi, while the even receives that message. Note that these MPI calls return nothing. The received message is in the variable a:

```
In[5]:= a
Out[5]:= 3.1415926535897932384626
In[6]:= Clear[a]
```

The variable a on the odd processors would have no definition. Moreover, if $NProc is 8, processor 3 sent Pi to processor 2, processor 5 sent Pi to processor 4, and so on. These messages were not sent through processor 0, but they communicated on their own.

The mpilSend and mpilRecv commands have a letter "I" to indicate asynchronous behavior, making it possible to do other work while messages are being sent and received, or if the other processor is busy. So, the above example could be done asynchronously:

```
In[7]:= If[1==Mod[$IdProc, 2],mpiISend[N[Pi,22],targetProc,
    mpiCommWorld,d,e],
    mpiIRecv[a,targetProc,mpiCommWorld,d,e]]
```

The variable e has important data identifying the message, and mpiTest[e] can return True before the expressions are to be accessed. At this point, many other evaluations can be performed. Then, one can check using mpiTest when the data is needed:

```
In[29]:= mpiTest[e]
Out[29]:= True
In[30]:= a
Out[30]:= 3.1415926535897932384626
In[31]:= Clear[a,e]
```

The mpiWait[e] command could have also have been used, which does not return until mpiTest[e] returns True. The power of using these peer-to-peer calls is that it becomes possible to construct any message-passing pattern for any problem.

Collective MPI

In some cases, such explicit control is not required and a commonly used communication pattern is sufficient. Suppose processor 0 has an expression in b that all processors are meant to have. A broadcast MPI call would do:

In[8]:=mpiBcast[b, 0, mpiCommWorld]

The second argument specifies which processor is the "root" of this broadcast; all others have their b overwritten. To collect values from all processors, use mpiGatherD:

In[9]:=mpiGather[b, c, 0, mpiCommWorld]

The variable c of processor 0 is written with a list of all the b of all the processors in mpiCommWorld. The temporal opposite is mpiScatter:

```
In[10]:= Clear[b]; a = {2, 4, 5, 6}; mpiScatter[a, b, 0,
    mpiCommWorld]; b
Out[10]:= {2, 4}
```

The mpiScatter command cuts up the variable a into even pieces (when possible) and scatters them to the processors. This is the result if $NProc=2, but if $NProc=4, b would only have {2}.

MPI provides reduction operations to perform simple computations mixed with messaging. Consider the following:

```
In[11]:= a = {{2 + $IdProc, 45[ ],3,{1 + $IdProc,$NProc[ ][ ];
    mpiReduce [a,d,mpiSum, 0,mpiCommWorld ]
In[12]:= d
Out[12]:= {{5, 90}, 6, {3, 4}}
```

The mpiSum constant indicates that variable a of every processor will be summed. In this case, $NProc is 2, so those elements that were not identical result in odd sums, while those that were the same are even.

Most of these calls have default values if not all are specified. For example each of the following calls will have the equivalent effect as the above mpiGather[ ] call:

```
mpiGather[b, c, 0]
mpiGather[b, c]
c = mpiGather[b]
```

High-Level Calls

High-level calls can include convenient parallel versions of commonly used application program calls (for example, Mathematica calls). For example, ParallelTable[ ] is like Table[ ], except that the evaluations are automatically performed in a distributed manner:

```
In[13]:= ParallelTable[i,{i,100},0]
Out[13]:= {1,2,3,4,5, ••• ,99,100}
```

The third argument specifies that the answers are collated back to processor 0. This is a useful, simple way to parallelize many calls to a complex function. One could define a complicated function and evaluate it over a large range of inputs:

```
In[14]:= g[x_] := Gamma[2 + 0.5*(x-1)];
    ParallelTable[g[i],{i,100},0]
Out[14]:= {1, 1.32934, 2., 3.32335, 6., 11.6317, 24., 52.3428,
    120., 287.885, 720}
```

ParallelFunctionToList[ ] also provides a simplified way to perform this form of parallelism.

Operations with Non-Trivial Communication

Matrix Operations

In some embodiments, one or more functions can help solve matrix calculations in parallel:

```
In[15]:= a = Table[i+ 3* $IdProc + 2 j, {i, 2}, {j,4}]
Out[15]:= {{3, 5, 7, 9}, {4, 6, 8, 10}}
In[16]:= t = ParallelTranspose[a]
Out[16]:= {{3, 4, 6, 7}, {5, 6, 8, 9}}
```

Fourier Transforms

A Fourier transform of a large array can be solved faster in parallel, or made possible on a cluster because it can all be held in memory. A two-dimensional Fourier transform of the above example follows:

```
In[17]:= f = ParallelFourier[a]
Out[17]:= {{32. + 0. I, -4. - 4. I, -4., -4. + 4. I}, {-3. -
    3. I, 0. + 0. I, 0., 0. + 0. I}}
```

Edge Cell Management

Many problems require interactions between partitions, but only on the edge elements. Maintaining these edges can be performed using EdgeCell[ ].

```
In[18]:= a = {2, 4, 5, 6, 7}+8*$IdProc
Out[18]:= {2, 4, 5, 6, 7}
In[19]:= EdgeCell[a]; a
Out[19]:= {14, 4, 5, 6, 12}
```

Element Management

In particle-based problems, items can drift through space, sometimes outside the partition of a particular processor. This can be solved with ElementManage[1:

```
In[20]:= list={{0,4},{1,3},{1,4},{0,5}}; fcn[x_]:=x[[1]]
In[21]:= ElementManage[list, fcn]
Out[21]:= {{0, 4}, {0, 5}, {0, 4}, {0, 5}}
In[21]:= ElementManage[list, 2]
Out[21]:= {{0, 4}, {0, 5}, {0, 4}, {0, 5}}
```

The second argument of ElementManage describes how to test elements of a list. The fcn identifier returns which processor is the "home" of that element. Passing an integer assumes that each element is itself a list, whose first element is a number ranging from 0 to the passed argument.

While the examples above involve Mathematica software and specific embodiments of MPI calls and cluster commands, it is recognized that these embodiments are used only to illustrate features of various embodiments of the systems and methods.

VI. Additional Embodiments

Although cluster computing techniques, modules, calls, and functions are disclosed with reference to certain embodiments, the disclosure is not intended to be limited thereby. Rather, a skilled artisan will recognize from the disclosure herein a wide number of alternatives for the exact selection of cluster calls, functions, and management systems. For example, single-node kernels can be managed using a variety of management tools and/or can be managed manually by a user, as described herein. As another example, a cluster node module can contain additional calls and procedures, including calls and procedures unrelated to cluster computing, that are not disclosed herein.

Other embodiments will be apparent to those of ordinary skill in the art from the disclosure herein. Moreover, the described embodiments have been presented by way of example only, and are not intended to limit the scope of the disclosure. Indeed, the novel methods and systems described herein can be embodied in a variety of other forms without departing from the spirit thereof. Accordingly, other combinations, omissions, substitutions and modifications will be apparent to the skilled artisan in view of the disclosure herein. Thus, the present disclosure is not intended to be limited by the disclosed embodiments, but is to be defined by reference to the appended claims. The accompanying claims and their equivalents are intended to cover forms or modifications as would fall within the scope and spirit of the inventions.

What is claimed is:

1. A computer cluster comprising:

a plurality of nodes, wherein each of the plurality of nodes comprises a hardware processor, wherein one or more of the nodes are configured to receive a command to start a cluster initialization process for the computer cluster, and wherein each of the nodes is configured to access a non-transitory computer-readable medium comprising program code for a single-node kernel that, when executed, is capable of causing the hardware processor to evaluate mathematical expressions; and

a mechanism for the nodes to communicate results of mathematical expression evaluation with each other using a peer-to-peer architecture;

wherein the plurality of nodes comprises:

a first node comprising a first hardware processor configured to access a first memory comprising program code for a user interface and program code for a first single-node kernel, the first single-node kernel configured to interpret user instructions and distribute calls to at least one of a plurality of other nodes for execution; and

a second node comprising a second hardware processor with a plurality of processing cores, wherein the second node is configured to receive calls from the first node, execute at least a first mathematical expression evaluation, and communicate a result of the first mathematical expression evaluation to a third node;

wherein the third node comprises a third hardware processor with a plurality of processing cores, wherein the third node is configured to receive the result of the first mathematical expression evaluation from the second node, execute at least a second mathematical expression evaluation using the received result, and communicate the result of the second mathematical expression evaluation to the first node;

wherein the first node is configured to return the result of the second mathematical expression evaluation to the user interface;

wherein one or more of the nodes are configured to:

accept user instructions;

after accepting user instructions, communicate at least some of the user instructions using the mechanism for the nodes to communicate with each other; and

after communicating at least some of the user instructions using the mechanism, communicate at least some of the user instructions to one or more single-node kernels.

2. The computer cluster of claim **1**, wherein the single-node kernel comprises a Mathematica kernel.

3. The computer cluster of claim **1**, wherein the mechanism comprises a message passing interface.

4. The computer cluster of claim **1**, wherein each of the nodes comprises one or more cluster node modules.

5. The computer cluster of claim **1**, wherein each single-node kernel is stored in a non-transitory computer-readable medium and configured to accept and execute a request.

6. The computer cluster of claim **5**, wherein each of the nodes comprises one or more cluster node modules, wherein each of the cluster node modules comprises instructions stored in a non-transitory computer-readable medium, and wherein the instructions, when executed by the hardware processor, cause the cluster node module to communicate with the single-node kernel and with one or more other cluster node modules.

7. The computer cluster of claim **6**, wherein the plurality of cluster node modules act as a cluster.

8. The computer cluster of claim **6**, wherein the plurality of cluster node modules communicate with one another to act as a cluster.

9. The computer cluster of claim **8**, wherein the computer cluster includes the user interface.

10. The computer cluster of claim **9**, wherein each cluster node module accepts instructions from the user interface and interprets one or more of the instructions.

11. The computer cluster of claim **1**, wherein the cluster initialization process comprises establishing communication among two or more of the nodes.

12. The computer cluster of claim **1**, wherein the cluster initialization process comprises configuring access by one or more of the nodes to a computer-readable medium comprising program code for the single-node kernel.

13. The computer cluster of claim **12**, wherein the cluster initialization process comprises establishing message-passing support among the nodes in the cluster.

14. The computer cluster of claim **12**, wherein the cluster initialization process comprises launching cluster node modules by the cluster.

15. The computer cluster of claim **14**, wherein the cluster initialization process comprises assigning a processor identification number to each of the cluster node modules.

16. The computer cluster of claim **1**, wherein the cluster initialization process comprises:

launching cluster node modules by the cluster; and

after launching the cluster node modules, configuring access by one or more of the nodes to a non-transitory computer-readable medium comprising program code for the single-node kernel.

17. The computer cluster of claim **1**, wherein the cluster initialization process comprises:

launching cluster node modules by the cluster;

after launching the cluster node modules, establishing communication among two or more of the nodes; and

after establishing communication among two or more of the nodes, assigning a processor identification number to each of the cluster node modules.

18. The computer cluster of claim **1**, wherein one or more of the nodes are configured to communicate at least some of the user instructions.

19. The computer cluster of claim **18**, wherein one or more of the nodes are configured to communicate at least some of the user instructions to one or more single-node kernels.

20. The computer cluster of claim **1**, wherein one or more of the nodes are configured to accept user instructions via one or more of the nodes.

21. The computer cluster of claim **20**, wherein one or more of the nodes are configured to communicate at least some of the user instructions using the mechanism for the nodes to communicate with each other.

22. The computer cluster of claim **1**, wherein one or more of the nodes are configured to transmit at least some of the user instructions that originate from a user interface.

23. The computer cluster of claim **1**, wherein one or more of the nodes are configured to parallelize at least some of the user instructions before communicating at least some of the user instructions to one or more single-node kernels.

24. The computer cluster of claim **1**, wherein one or more of the nodes are configured to accept user instructions before communicating at least some of the user instructions to one or more single-node kernels.

25. The computer cluster of claim **1**, wherein the plurality of nodes are configured to communicate with one another to interpret and translate commands for execution by a plurality of single-node kernels.

26. A computer cluster comprising:

a plurality of nodes, wherein one or more of the nodes are configured to receive:

a command to start a cluster initialization process for the computer cluster, wherein the cluster initialization process comprises establishing communication among two or more of the nodes; and

an instruction from a user interface or a script;

and

a mechanism for the nodes to communicate results of mathematical expression evaluation with each other using asynchronous calls;

wherein each of the nodes is configured to access a non-transitory computer-readable medium comprising program code for a single-node kernel that, when executed, is capable of causing a hardware processor to evaluate mathematical expressions;

wherein the plurality of nodes comprises:

a first node comprising a first hardware processor configured to access a first memory comprising program code for a user interface and program code for a first single-node kernel, the first single-node kernel configured to interpret user instructions and distribute calls to at least one of a plurality of other nodes for execution; and

a second node comprising a second hardware processor with a plurality of processing cores, wherein the second node is configured to receive calls from the first node, execute at least a first mathematical expression evaluation, and communicate a result of mathematical expression evaluation to a third node;

wherein the third node comprises a third hardware processor with a plurality of processing cores, wherein the third node is configured to receive the result of mathematical expression evaluation from the second node, execute at least a second mathematical expression evaluation using the received result, and communicate the result of the second mathematical expression evaluation to the first node;

wherein the first node is configured to return the result of the second mathematical expression evaluation to the user interface or the script;

wherein one or more of the nodes are configured to:

accept user instructions;

after accepting user instructions, communicate at least some of the user instructions using the mechanism for the nodes to communicate with each other; and

after communicating at least some of the user instructions using the mechanism, communicate at least some of the user instructions to one or more single-node kernels.

27. The computer cluster of claim **26**, wherein the asynchronous calls comprise a first command to create a first packet containing:

an expression to be sent as payload; and

a target node where the expression should be sent;

wherein the first command is configured to be called from within a single-node kernel;

wherein the single-node kernel is configured to send the first packet to a local cluster node module; and

wherein the local cluster node module is configured to forward the expression to the target node.

28. The computer cluster of claim **27**, wherein the asynchronous calls comprise a second command to create a second packet containing:

a location where the expression is expected to be received; and

a sending node from which the expression is expected to be received;

wherein the second command is configured to be called from within a single-node kernel;

wherein the single-node kernel is configured to send the second packet to a local cluster node module; and

wherein the local cluster node module is configured to store the second packet contents in a message receiving queue.

29. A computer cluster comprising:

a plurality of nodes, wherein one or more of the nodes are configured to receive:

a command to start a cluster initialization process for the computer cluster, wherein the cluster initialization process comprises establishing communication among two or more of the nodes; and

an instruction from a user interface or a script;

and

a mechanism for the nodes to communicate results of mathematical expression evaluation with each other;

wherein each of the nodes is configured to access a non-transitory computer-readable medium comprising program code for a single-node kernel that, when executed, is capable of causing a hardware processor to evaluate mathematical expressions;

wherein the plurality of nodes comprises:

a first node comprising a first hardware processor configured to access a first memory comprising program code for a user interface and program code for a first single-node kernel, the first single-node kernel configured to interpret user instructions and distribute calls to at least one of a plurality of other nodes for execution; and

a second node comprising a second hardware processor with a plurality of processing cores, wherein the second node is configured to receive calls from the first node, execute at least a first mathematical expression evaluation, and communicate a result of mathematical expression evaluation to a third node;

wherein the third node comprises a third hardware processor with a plurality of processing cores, wherein the third node is configured to receive the result of mathematical expression evaluation from the second node, execute at least a second mathematical expression evaluation using the received result, and communicate the result of the second mathematical expression evaluation to the first node;

and

wherein the first node is configured to return the result of the second mathematical expression evaluation to the user interface or the script;

wherein one or more of the nodes are configured to:

accept user instructions;

after accepting user instructions, communicate at least some of the user instructions using the mechanism for the nodes to communicate with each other; and

after communicating at least some of the user instructions using the mechanism, communicate at least some of the user instructions to one or more single-node kernels.

30. The computer cluster of claim **4**, wherein each of the plurality of nodes implements asynchronous calls that enable the single-node kernel to perform computation tasks while the cluster node modules are simultaneously communicating with one another.

31. The computer cluster of claim **4**, wherein intercommunication among the plurality of single-node kernels during thread execution is enabled by the plurality of cluster node modules, and wherein the computer cluster is configured to permit exchange of information between nodes during the course of a parallel computation.

32. The computer cluster of claim **4**, wherein each of the single-node kernels are configured to call a send command involving creating a packet containing:

an expression to be sent as payload; and

where the expression should be sent;

wherein, upon reception of the send command by a local cluster node module associated with the single-node kernel, the local cluster node module is configured to decode the packet and forward a payload to the cluster node module specified in the packet;

wherein each of the single-node kernels is configured to call a receipt command involving creating a packet specifying which message to test for completion and to then wait for a reply expression to evaluate;

wherein, upon reception of the receipt command by the local cluster node module associated with the single-node kernel, the local cluster node module is configured to decode the packet and use a message specifier to search for any matching expressions listed as completed in a received message queue;

wherein, upon determining that such a completed expression is found, the local cluster node module is configured to send the completed expression to a local single-node kernel associated with the cluster node module in response to the receipt command, and

wherein each of the single-node kernels is configured to receive the completed expression and to update variables used by the single-node kernel during evaluation of expressions.

33. The computer cluster of claim **1**, wherein the plurality of nodes are configured to permit exchange of information between nodes during the course of parallel computation.

34. The computer cluster of claim **1**, wherein each of the plurality of nodes comprises instructions executable by the

hardware processor and configured to implement asynchronous behavior, wherein the instructions comprise:

a first instruction to asynchronously send a payload to another node;

a second instruction to asynchronously receive a payload from another node; and

a third instruction to search for a payload matching a message specifier.

35. A computer cluster node for evaluating expressions in parallel with other computer cluster nodes, the computer cluster node comprising:

a hardware processor configured to access one or more non-transitory memory devices comprising program code for a single-node kernel that, when executed, causes the hardware processor to interpret user instructions, to evaluate mathematical expressions, and to produce results of mathematical expression evaluation, wherein the hardware processor comprises multiple processor cores;

a user connection interface configured to receive a command to start a cluster initialization process for a computer cluster;

a mechanism to communicate results of evaluation with other computer cluster nodes using a peer-to-peer architecture; and

program code that, when executed, is capable of causing the hardware processor to:

receive calls from a second node comprising a second hardware processor configured to access a second memory comprising program code for a user interface and program code for a second single-node kernel, the second single-node kernel configured to interpret user instructions and distribute calls to at least one of a plurality of other nodes for execution;

execute, using the hardware processor, at least a first mathematical expression evaluation; and

communicate a result of the first mathematical expression evaluation to a third node comprising a third hardware processor with a plurality of processing cores, wherein the third node is configured to receive the result of mathematical expression evaluation from the computer cluster node, execute at least a second mathematical expression evaluation using the result of the first mathematical expression evaluation, and communicate a result of the second mathematical expression evaluation to the first node;

wherein the user connection interface is configured to return at least one result of mathematical expression evaluation to a user interface or a script; and

wherein the computer cluster node is configured to:

accept user instructions;

after accepting user instructions, communicate at least some of the user instructions using the mechanism for the nodes to communicate with each other; and

after communicating at least some of the user instructions using the mechanism, communicate at least some of the user instructions to the single-node kernel.

36. The computer cluster node of claim **35**, wherein the one or more non-transitory memory devices comprise program code for performing a parallel fast Fourier transform, wherein the program code causes the hardware processor to evaluate a command to perform a Fourier transform on an array comprising a first data portion that is stored on the computer cluster node and a second data portion that is not stored on the computer cluster node.

37. The computer cluster node of claim **35**, wherein the computer cluster node is configured to permit exchange of information with other computer cluster nodes during the course of parallel computation.

38. The computer cluster node of claim **35**, wherein the computer cluster node comprises instructions executable by the hardware processor and configured to implement asynchronous behavior, wherein the instructions comprise:

a first instruction to asynchronously send a payload to another node;

a second instruction to asynchronously receive a payload from another node; and

a third instruction to search for a payload matching a message specifier.

39. The computer cluster node of claim **35**, wherein the hardware processor comprises a special purpose microprocessor.

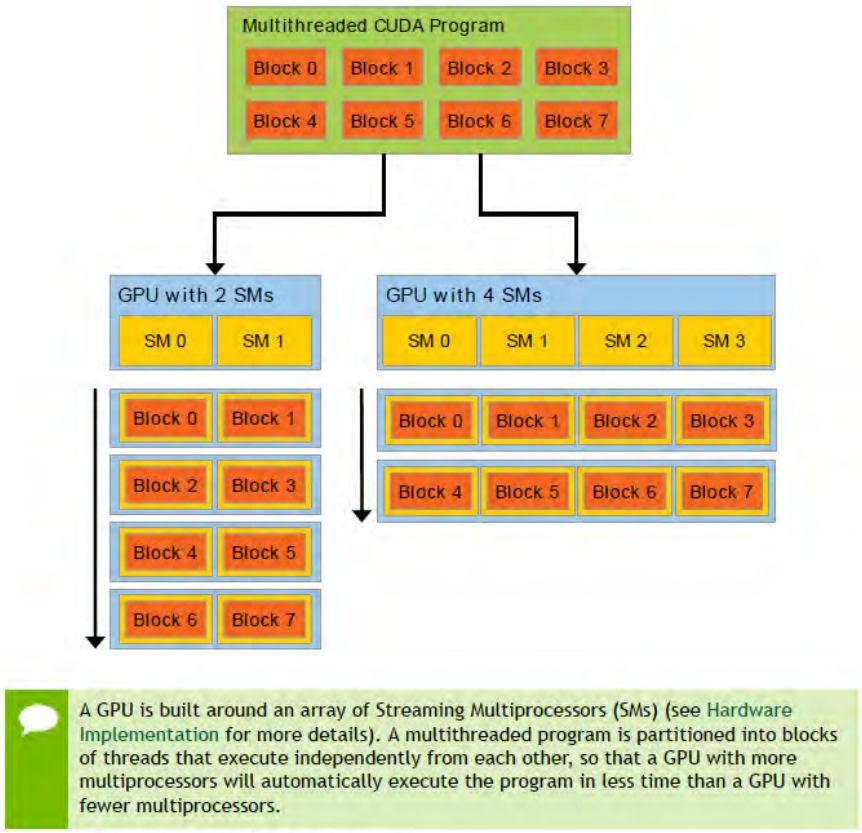
* * * * *

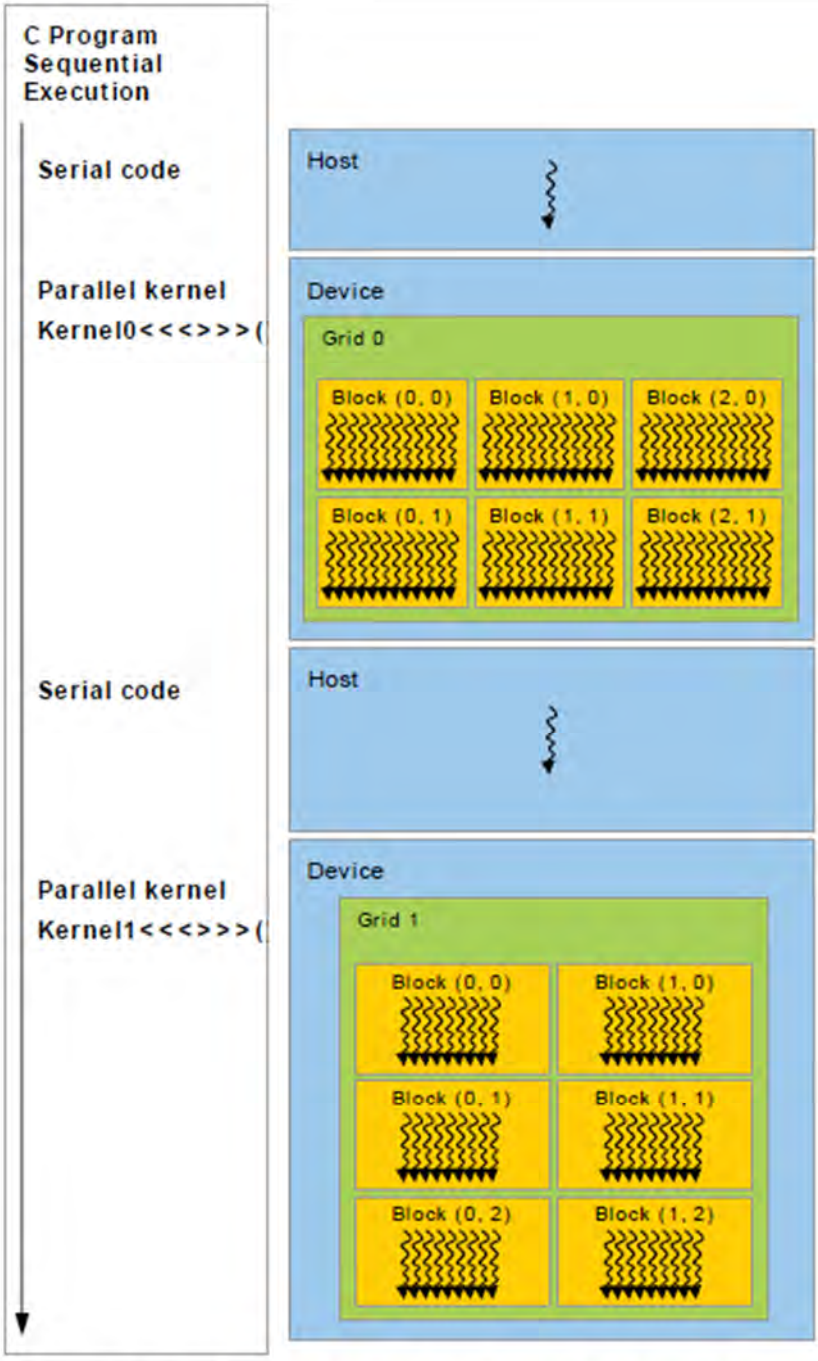
EXHIBIT E

EXHIBIT E – '289 Patent Exemplary Claim Chart

Claim 29 Language	NVIDIA's Infringing Products
A method of evaluating a command on a computer cluster comprising:	Each Accused Server Product practices a method of evaluating a command on a computer cluster as follows[1].
communicating a command from at least one of a user interface or a script to one or more cluster node modules within the computer cluster;	Each Accused Server Product practices a method comprising communicating a command from at least one of a user interface or a script to one or more cluster node modules within the computer cluster. For example, each Accused Server Product includes a CPU or host executing a program (user interface or script) that communicates commands to a cluster of GPU Accelerators, each of which include a cluster node module. 2.4. Heterogeneous Programming As illustrated by Figure 8, the CUDA programming model assumes that the CUDA threads execute on a physically separate *device* that operates as a coprocessor to the *host* running the C program. This is the case, for example, when the kernels execute on a GPU and the rest of the C program executes on a CPU.

[1] Each Accused Server Product and Accused GPU Product herein, including DGX Station, DGX-1, DGX-2, HGX-1, HGX-2, Tesla V100, and Tesla P100, uses or is configured to use the NVLink architecture and technology described in the references herein including NVIDIA's Tesla P100, Tesla V100, and NVLink whitepapers and CUDA C Programming Guide. *See* NVIDIA DGX Station: Maximize Your Data Science Team Productivity, https://www.nvidia.com/content/dam/en-zz/Solutions/Data-Center/dgx-station/nvidia-dgx-station-datasheet.pdf; NVIDIA DGX-1: Essential Instrument for AI Research, https://www.nvidia.com/content/dam/en-zz/Solutions/Data-Center/dgx-1/dgx-1-ai-supercomputer-datasheet-v4.pdf; NVIDIA DGX-2: The World's Most Powerful Deep Learning System for the Most Complex AI Challenges, https://www.nvidia.com/content/dam/en-zz/Solutions/Data-Center/dgx-2/dgx-2-print-datasheet-738070-nvidia-a4-web-uk.pdf; NVIDIA HGX-2: Powered by NVIDIA Tesla V100 GPUs and NVSwitch https://www.nvidia.com/en-us/data-center/hgx/; NVIDIA Tesla P100: GPU Accelerator, https://www.nvidia.com/content/dam/en-zz/Solutions/Data-Center/tesla-p100/pdf/nvidia-tesla-p100-datasheet.pdf; NVIDIA Tesla V100: GPU Accelerator, https://images.nvidia.com/content/technologies/volta/pdf/437317-Volta-V100-DS-NV-US-WEB.pdf.

	 A GPU is built around an array of Streaming Multiprocessors (SMs) (see Hardware Implementation for more details). A multithreaded program is partitioned into blocks of threads that execute independently from each other, so that a GPU with more multiprocessors will automatically execute the program in less time than a GPU with fewer multiprocessors. Figure 5 Automatic Scalability NVidia CUDA C Programming Guide pp. 5, 12 (v10.1 May 2019), https://docs.nvidia.com/pdf/CUDA_C_Programming_Guide.pdf.
for each of the one or more cluster node modules, communicating a message based on the command to a respective kernel module associated with the cluster node module;	Each Accused Server Product practices a method comprising for each of the one or more cluster node modules, communicating a message based on the command to a respective kernel module associated with the cluster node module. For example, each GPU Accelerator cluster node module communicates a message based on the command received from the CPU or host to a CUDA kernel associated with the cluster node module. ## 2.4. Heterogeneous Programming As illustrated by Figure 8, the CUDA programming model assumes that the CUDA threads execute on a physically separate *device* that operates as a coprocessor to the *host* running the C program. This is the case, for example, when the kernels execute on a GPU and the rest of the C program executes on a CPU.



Serial code executes on the host while parallel code executes on the device.

Figure 8 Heterogeneous Programming

NVidia CUDA C Programming Guide pp. 12-13 (v10.1 May 2019), https://docs.nvidia.com/pdf/CUDA_C_Programming_Guide.pdf.

for each of the one or more cluster node modules, receiving a result from the respective kernel module	Each Accused Server Product practices a method comprising, for each of the one or more cluster node modules, receiving a result from the respective kernel module associated with the cluster node module. For example, each GPU Accelerator cluster node module receives results from the CUDA kernel associated with the cluster node module.

associated with the cluster node module; and	2.1. Kernels
	
	CUDA C extends C by allowing the programmer to define C functions, called *kernels*, that, when called, are executed N times in parallel by N different *CUDA threads*, as opposed to only once like regular C functions.
	
	A kernel is defined using the **__global__** declaration specifier and the number of CUDA threads that execute that kernel for a given kernel call is specified using a new **<<<...>>>** *execution configuration* syntax (see C Language Extensions). Each thread that executes the kernel is given a unique *thread ID* that is accessible within the kernel through the built-in **threadIdx** variable.
	
	As an illustration, the following sample code adds two vectors *A* and *B* of size *N* and stores the result into vector *C*:
	
	`// Kernel definition`
	`__global__ void VecAdd(float* A, float* B, float* C)`
	`{`
	`  int i = threadIdx.x;`
	`  C[i] = A[i] + B[i];`
	`}`
	
	`int main()`
	`{`
	`  ...`
	`  // Kernel invocation with N threads`
	`  VecAdd<<<1, N>>>(A, B, C);`
	`  ...`
	`}`
	
	Here, each of the *N* threads that execute **VecAdd()** performs one pair-wise addition.
	
	NVidia CUDA C Programming Guide p. 7 (v10.1 May 2019), https://docs.nvidia.com/pdf/CUDA_C_Programming_Guide.pdf.

for at least one of the one or more cluster node modules, responding to messages from other cluster node modules.	Each Accused Server Product practices a method comprising, for at least one of the one or more cluster node modules, responding to messages from other cluster node modules. For example, at least one GPU Accelerator cluster node module responds to messages from other GPU Accelerator cluster node module using the NVLink peer-to-peer communication function. **Server Configuration with NVLink** In the following sections of this paper, we analyze the performance benefit of NVLink for several algorithms and applications by comparing model systems based on PCIe-interconnected next-gen GPUs to otherwise-identical systems with NVLink-interconnected GPUs. GPUs are connected to the CPU using existing PCIe connections, but the NVLink configurations augment this with interconnections among the GPUs for peer-to-peer communication. The following analyses assume future-generation GPUs with performance higher than that of today's GPUs, so as to better correspond with the GPUs that will be contemporary with NVLink.

The second scenario, shown in Figure 3, compares a pair of configurations with four GPUs each, referred to as *4-GPU-PCIe* and *4-GPU-NVLink*. 4-GPU-PCIe uses a PCIe tree topology, again mirroring a configuration used commonly today. 4-GPU-NVLink enhances this by providing point-to-point connections with either one or two NVLink connections per pair of GPUs, yielding 16 GB/s or 32 GB/s effective bandwidth per direction per pair, respectively.

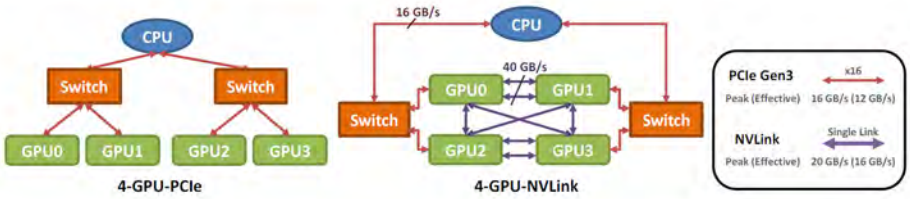


Figure 3: Comparing 4-GPU topologies with NVLink and PCIe. In 4-GPU-NVLink, GPU0 and GPU1 have 40 GB/s peak bandwidth between them, as do GPU2 and GPU3. The other peer-to-peer connections have 20 GB/s peak bandwidth.

The exchange algorithm begins with data evenly distributed across multiple GPUs as shown below.

Original Data

8	3	7	4	6	1	9	2	7	5	3	8	2	0	1	9

In this diagram, the color indicates which device the original data began on.

The first step is to hash the data to determine which GPU's bin each element should be placed in. For sorting, this might be as simple as extracting the highest order bits, but in theory we could apply any arbitrary hashing function to determine the keys (i.e., the destination bins). For this example, we will use the hash function `ceil(x/5)`, which produces the following keys given the data above.

Calculated Keys

1	0	1	0	1	0	1	0	1	1	0	1	0	0	0	1

Next we locally sort the data based on the hashed keys, yielding the following. Note that since we know these keys are in the range of `[0, NUM_GPUS)`, we can restrict the sorting to only a couple of bits.

Locally-sorted Data

3	4	1	2	8	7	6	9	3	2	0	1	7	5	8	9

Locally-sorted Keys

0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1

Next, we transfer all data with a key of 0 to device 0 and all data with a key of 1 to device 1. This becomes an all-to-all exchange among GPUs, producing the following.

Exchanged Data

3	4	1	2	3	2	0	1	8	7	6	9	7	5	8	9

Multi-GPU sorting is a simple extension of the above. After exchanging the data, we simply sort the data again locally to produce the final sorted list.

Fully-sorted Data

0	1	1	2	2	3	3	4	5	6	7	7	8	8	9	9

Here we can see the final list has been completely sorted. For many applications, the initial data exchange is all that is needed. Thus we will model both the exchange and the final sort separately.

Performance Considerations

The exchange algorithm can be easily pipelined by operating on small portions of the domain. The first step of the pipeline is to perform the local sort on the generated keys; the second is to perform the all-to-all exchange. This allows the initial sorting and the exchange to be completely overlapped. For the full multi-GPU sorting algorithm, the final local sort cannot begin until all data has arrived, so this portion cannot be pipelined with the other two.

If we assume the original data is randomly distributed among GPUs following a uniform random distribution, then each GPU must communicate `W*N/P` bytes of data to every other GPU, where W is the size of each element in bytes, N is the number of elements per GPU, and P is the number of GPUs.

In a PCIe tree topology, the communication is limited by the bisection bandwidth. Each GPU must communicate half of its data (N/2) across the top link in the system, and there are P/2 devices all trying to communicate across the same PCIe lanes in the same direction.

On an NVLink system, however, the communication can occur in parallel, because there are dedicated links between all pairs of GPUs.

NVIDIA NVLink High-Speed Interconnect: Application Performance p. 4-7, https://info.nvidianews.com/rs/nvidia/images/NVIDIA%20NVLink%20High-Speed%20Interconnect%20Application%20Performance%20Brief.pdf.

To address this issue, Tesla P100 features NVIDIA's new high-speed interface, NVLink, that provides GPU-to-GPU data transfers at up to 160 Gigabytes/second of bidirectional bandwidth—5x the bandwidth of PCIe Gen 3 x16. Figure 4 shows NVLink connecting eight Tesla P100 Accelerators in a Hybrid Cube Mesh Topology.

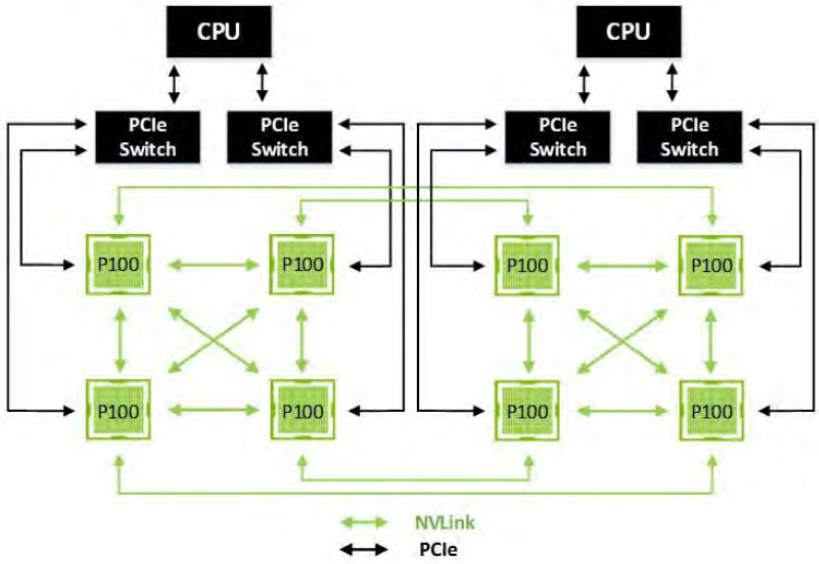


Figure 4. NVLink Connecting Eight Tesla P100 Accelerators in a Hybrid Cube Mesh Topology

NVIDIA Tesla P100: The Most Advanced Datacenter Accelerator Ever Built Featuring Pascal GP100, the World's Fastest GPU p. 7, https://images.nvidia.com/content/pdf/tesla/whitepaper/pascal-architecture-whitepaper.pdf.

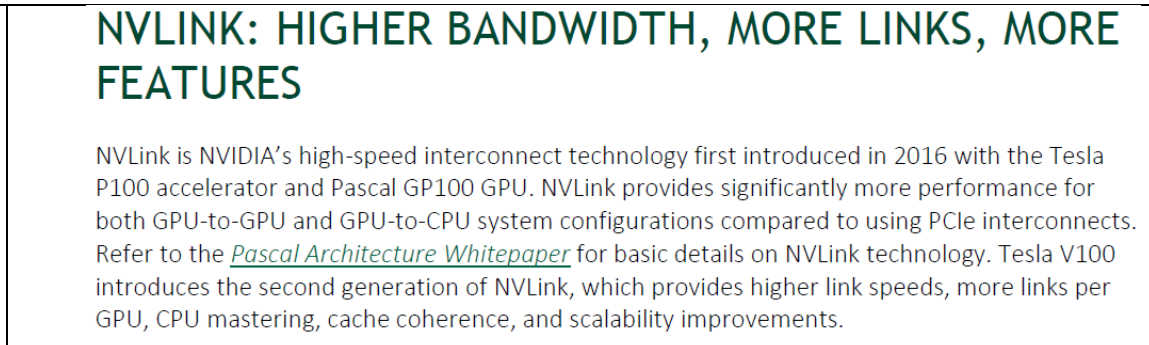
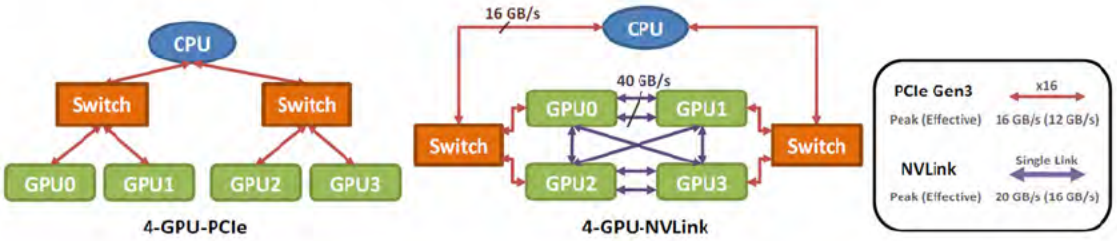
	 NVidia Tesla V100 GPU Architecture: The World's Most Advanced Data Center GPU, p. 19, https://images.nvidia.com/content/volta-architecture/pdf/volta-architecture-whitepaper.pdf.

EXHIBIT F

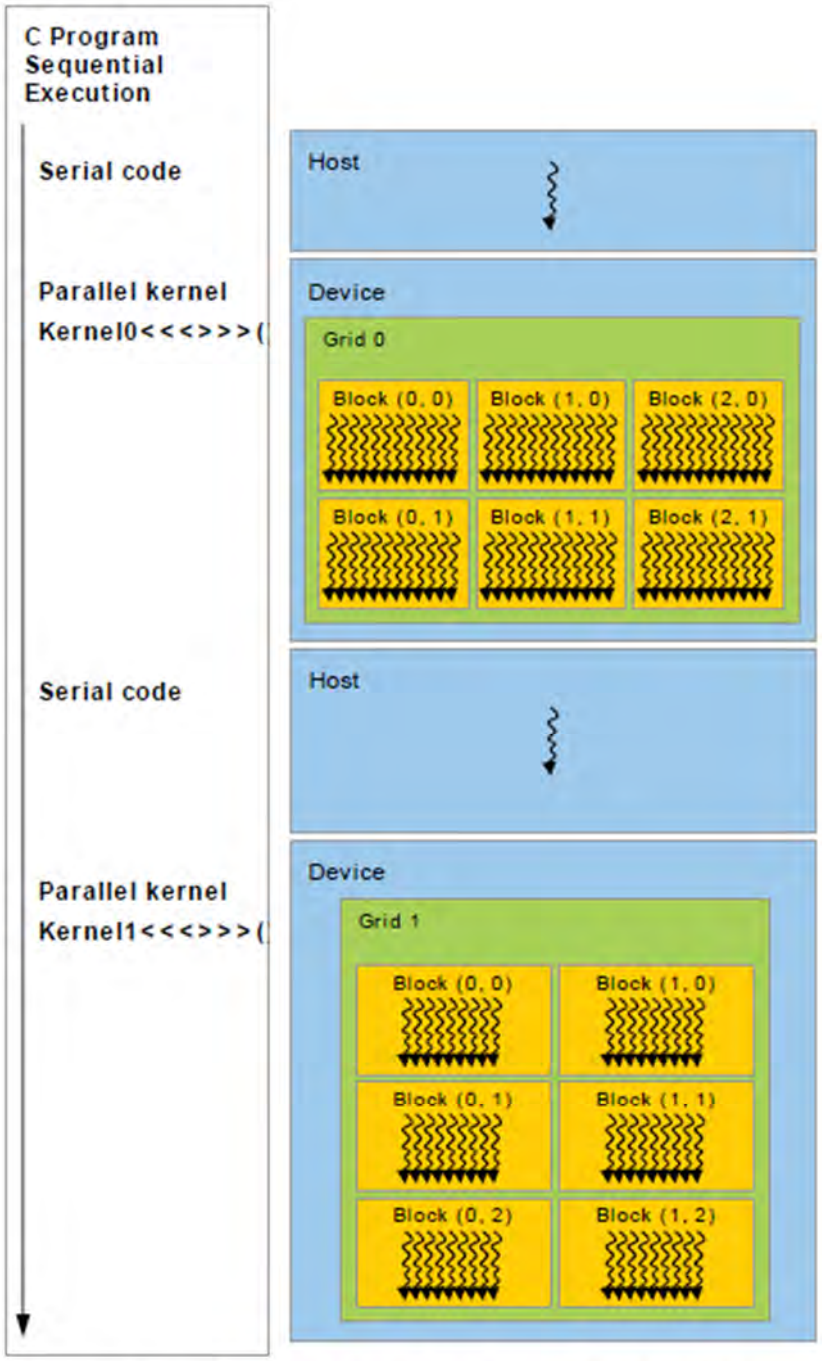
EXHIBIT F – '612 Patent Exemplary Claim Chart

Claim 1 Language	NVIDIA's Infringing Products
A computer cluster comprising:	Each Accused Server Product is a computer cluster as follows.[1]
a plurality of nodes, wherein each node is configured to access a computer-readable medium comprising program code for a single-node kernel module configured to interpret user instructions;	Each Accused Server Product comprises a plurality of nodes, wherein each node is configured to access a computer-readable medium comprising program code for a single-node kernel module configured to interpret user instructions. For example, each Accused Server Product includes a cluster of GPU Accelerators (nodes). Each GPU Accelerator is configured to access memory that stores the CUDA kernel associated with that GPU Accelerator. The CUDA kernel is configured to interpret instructions from the CPU or host . The second scenario, shown in Figure 3, compares a pair of configurations with four GPUs each, referred to as *4-GPU-PCIe* and *4-GPU-NVLink*. 4-GPU-PCIe uses a PCIe tree topology, again mirroring a configuration used commonly today. 4-GPU-NVLink enhances this by providing point-to-point connections with either one or two NVLink connections per pair of GPUs, yielding 16 GB/s or 32 GB/s effective bandwidth per direction per pair, respectively.  Figure 3: Comparing 4-GPU topologies with NVLink and PCIe. In 4-GPU-NVLink, GPU0 and GPU1 have 40 GB/s peak bandwidth between them, as do GPU2 and GPU3. The other peer-to-peer connections have 20 GB/s peak bandwidth. NVIDIA NVLink High-Speed Interconnect: Application Performance p. 5, https://info.nvidianews.com/rs/nvidia/images/NVIDIA%20NVLink%20High-Speed%20Interconnect%20Application%20Performance%20Brief.pdf.

[1] Each Accused Server Product and Accused GPU Product herein, including DGX Station, DGX-1, DGX-2, HGX-1, HGX-2, Tesla V100, and Tesla P100, uses or is configured to use the NVLink architecture and technology described in the references herein including NVIDIA's Tesla P100, Tesla V100, and NVLink whitepapers and CUDA C Programming Guide. See NVIDIA DGX Station: Maximize Your Data Science Team Productivity, https://www.nvidia.com/content/dam/en-zz/Solutions/Data-Center/dgx-station/nvidia-dgx-station-datasheet.pdf; NVIDIA DGX-1: Essential Instrument for AI Research, https://www.nvidia.com/content/dam/en-zz/Solutions/Data-Center/dgx-1/dgx-1-ai-supercomputer-datasheet-v4.pdf; NVIDIA DGX-2: The World's Most Powerful Deep Learning System for the Most Complex AI Challenges, https://www.nvidia.com/content/dam/en-zz/Solutions/Data-Center/dgx-2/dgx-2-print-datasheet-738070-nvidia-a4-web-uk.pdf; NVIDIA HGX-2: Powered by NVIDIA Tesla V100 GPUs and NVSwitch https://www.nvidia.com/en-us/data-center/hgx/; NVIDIA Tesla P100: GPU Accelerator, https://www.nvidia.com/content/dam/en-zz/Solutions/Data-Center/tesla-p100/pdf/nvidia-tesla-p100-datasheet.pdf; NVIDIA Tesla V100: GPU Accelerator, https://images.nvidia.com/content/technologies/volta/pdf/437317-Volta-V100-DS-NV-US-WEB.pdf.

2.4. Heterogeneous Programming

As illustrated by Figure 8, the CUDA programming model assumes that the CUDA threads execute on a physically separate *device* that operates as a coprocessor to the *host* running the C program. This is the case, for example, when the kernels execute on a GPU and the rest of the C program executes on a CPU.



Serial code executes on the host while parallel code executes on the device.

Figure 8 Heterogeneous Programming

NVidia CUDA C Programming Guide pp. 12-13 (v10.1 May 2019), https://docs.nvidia.com/pdf/CUDA_C_Programming_Guide.pdf.

The CUDA programming model also assumes that both the host and the device maintain their own separate memory spaces in DRAM, referred to as *host memory* and *device memory*, respectively. Therefore, a program manages the global, constant, and texture memory spaces visible to kernels through calls to the CUDA runtime (described in Programming Interface). This includes device memory allocation and deallocation as well as data transfer between host and device memory.

2.3. Memory Hierarchy

CUDA threads may access data from multiple memory spaces during their execution as illustrated by Figure 7. Each thread has private local memory. Each thread block has shared memory visible to all threads of the block and with the same lifetime as the block. All threads have access to the same global memory.

There are also two additional read-only memory spaces accessible by all threads: the constant and texture memory spaces. The global, constant, and texture memory spaces are optimized for different memory usages (see Device Memory Accesses). Texture

memory also offers different addressing modes, as well as data filtering, for some specific data formats (see Texture and Surface Memory).

The global, constant, and texture memory spaces are persistent across kernel launches by the same application.

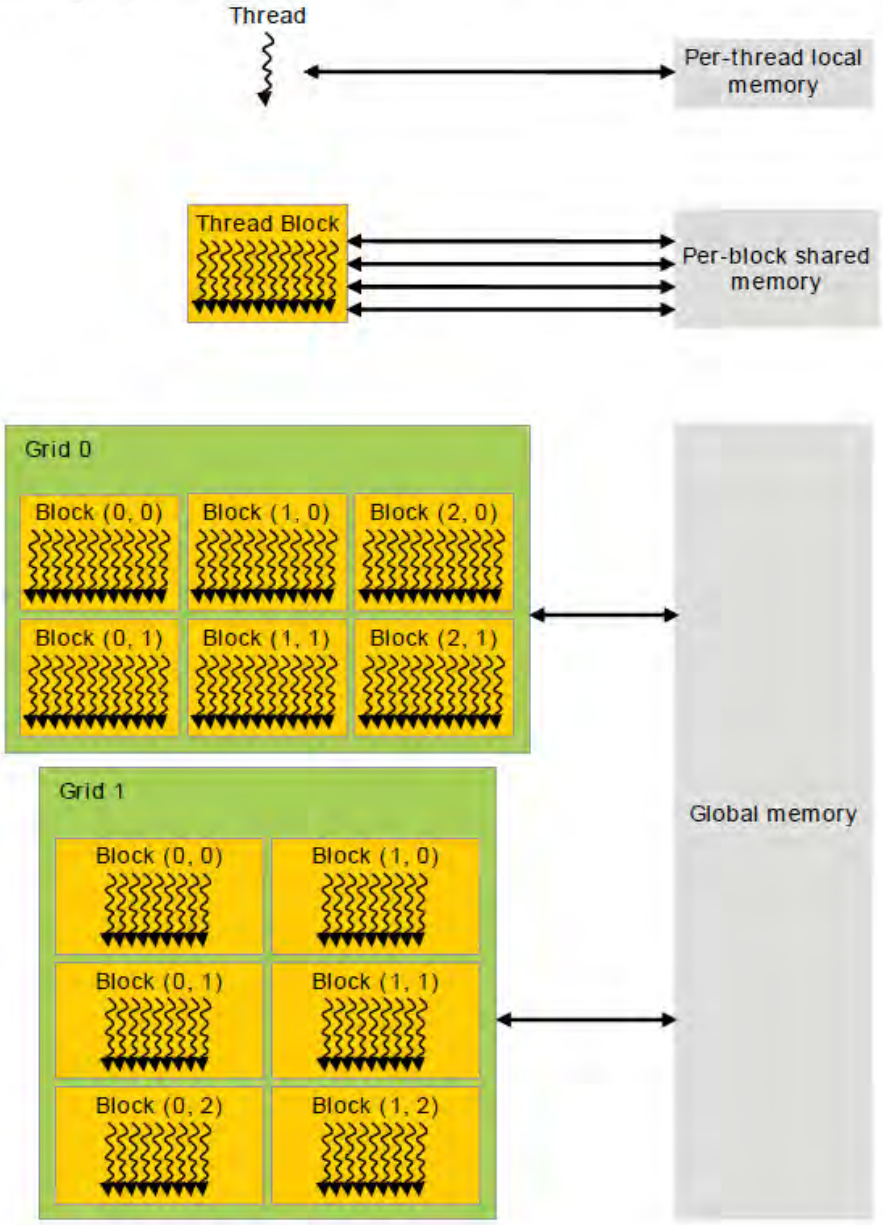


Figure 7 Memory Hierarchy

NVidia CUDA C Programming Guide pp. 7, 10-11 (v10.1 May 2019)
https://docs.nvidia.com/pdf/CUDA_C_Programming_Guide.pdf.

GP100 was built to be the highest performing parallel computing processor in the world to address the needs of the GPU accelerated computing markets serviced by our Tesla P100 accelerator platform. Like previous Tesla-class GPUs, GP100 is composed of an array of Graphics Processing Clusters (GPCs), Texture Processing Clusters (TPCs), Streaming Multiprocessors (SMs), and memory controllers. A full GP100 consists of six GPCs, 60 Pascal SMs, 30 TPCs (each including two SMs), and eight 512-bit memory controllers (4096 bits total).

Each GPC inside GP100 has ten SMs. Each SM has 64 CUDA Cores and four texture units. With 60 SMs, GP100 has a total of 3840 single precision CUDA Cores and 240 texture units. Each memory controller is attached to 512 KB of L2 cache, and each HBM2 DRAM stack is controlled by a pair of memory controllers. The full GPU includes a total of 4096 KB of L2 cache.

Figure 7 shows a full GP100 GPU with 60 SM units (different products can use different configurations of GP100). The Tesla P100 accelerator uses 56 SM units.

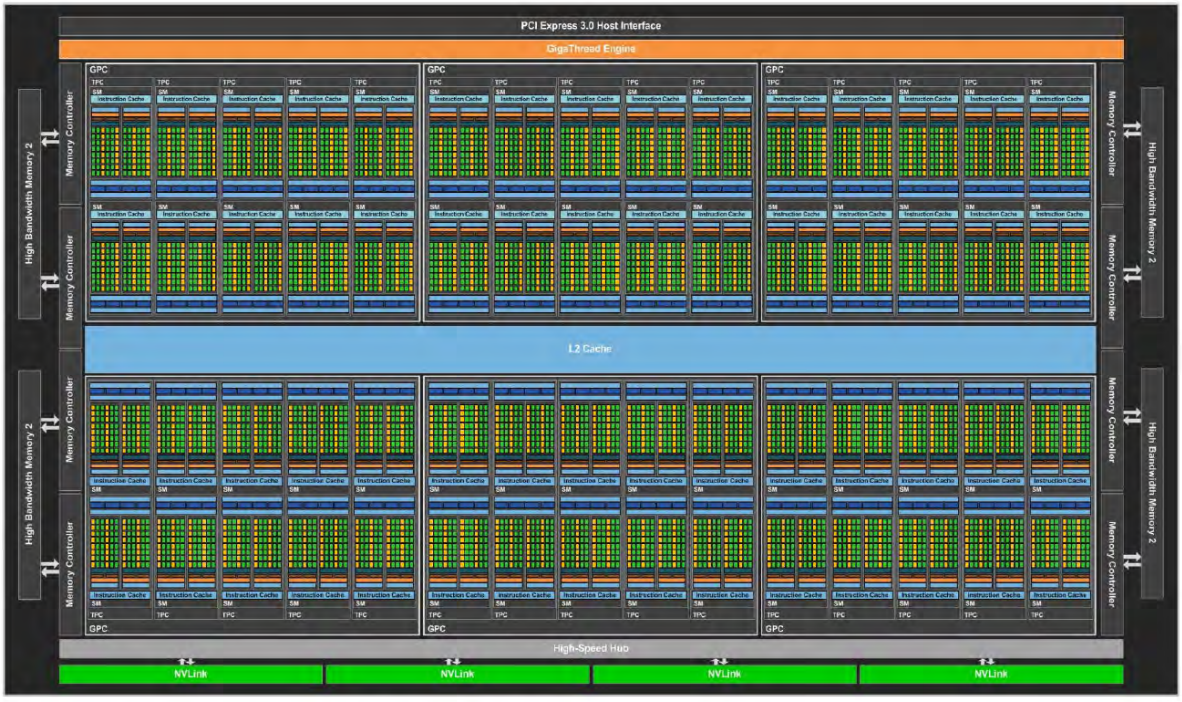


Figure 7. Pascal GP100 Full GPU with 60 SM Units

NVIDIA Tesla P100: The Most Advanced Datacenter Accelerator Ever Built Featuring Pascal GP100, the World's Fastest GPU p. 10, https://images.nvidia.com/content/pdf/tesla/whitepaper/pascal-architecture-whitepaper.pdf

Table 1. Comparison of NVIDIA Tesla GPUs

Tesla Product	Tesla K40	Tesla M40	Tesla P100	Tesla V100
GPU	GK180 (Kepler)	GM200 (Maxwell)	GP100 (Pascal)	GV100 (Volta)
SMs	15	24	56	80
TPCs	15	24	28	40
FP32 Cores / SM	192	128	64	64
FP32 Cores / GPU	2880	3072	3584	5120
FP64 Cores / SM	64	4	32	32
FP64 Cores / GPU	960	96	1792	2560
Tensor Cores / SM	NA	NA	NA	8
Tensor Cores / GPU	NA	NA	NA	640
GPU Boost Clock	810/875 MHz	1114 MHz	1480 MHz	1530 MHz
Peak FP32 TFLOPS[1]	5	6.8	10.6	15.7
Peak FP64 TFLOPS[1]	1.7	.21	5.3	7.8
Peak Tensor TFLOPS[1]	NA	NA	NA	125
Texture Units	240	192	224	320
Memory Interface	384-bit GDDR5	384-bit GDDR5	4096-bit HBM2	4096-bit HBM2
Memory Size	Up to 12 GB	Up to 24 GB	16 GB	16 GB
L2 Cache Size	1536 KB	3072 KB	4096 KB	6144 KB
Shared Memory Size / SM	16 KB/32 KB/48 KB	96 KB	64 KB	Configurable up to 96 KB
Register File Size / SM	256 KB	256 KB	256 KB	256KB
Register File Size / GPU	3840 KB	6144 KB	14336 KB	20480 KB
TDP	235 Watts	250 Watts	300 Watts	300 Watts
Transistors	7.1 billion	8 billion	15.3 billion	21.1 billion
GPU Die Size	551 mm²	601 mm²	610 mm²	815 mm²
Manufacturing Process	28 nm	28 nm	16 nm FinFET+	12 nm FFN

[1] Peak TFLOPS rates are based on GPU Boost Clock

NVidia Tesla V100 GPU Architecture: The World's Most Advanced Data Center GPU, p. 10, https://images.nvidia.com/content/volta-architecture/pdf/volta-architecture-whitepaper.pdf.

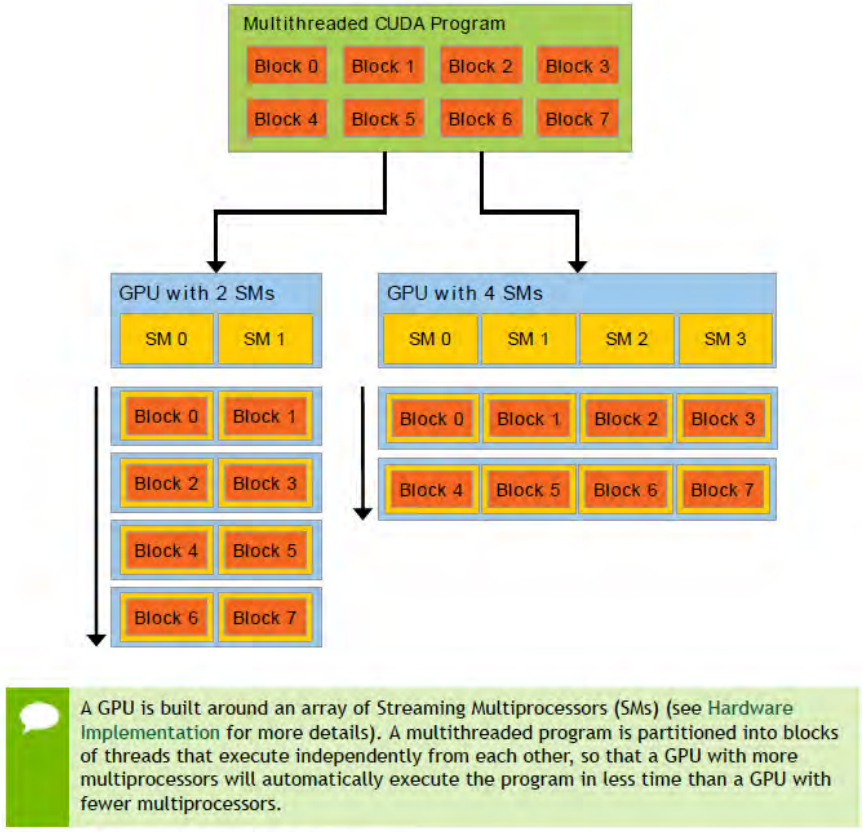
<table>
<tr>
<td></td>
<td>
<h2>2.4. Heterogeneous Programming</h2>
As illustrated by Figure 8, the CUDA programming model assumes that the CUDA threads execute on a physically separate device that operates as a coprocessor to the host running the C program. This is the case, for example, when the kernels execute on a GPU and the rest of the C program executes on a CPU.


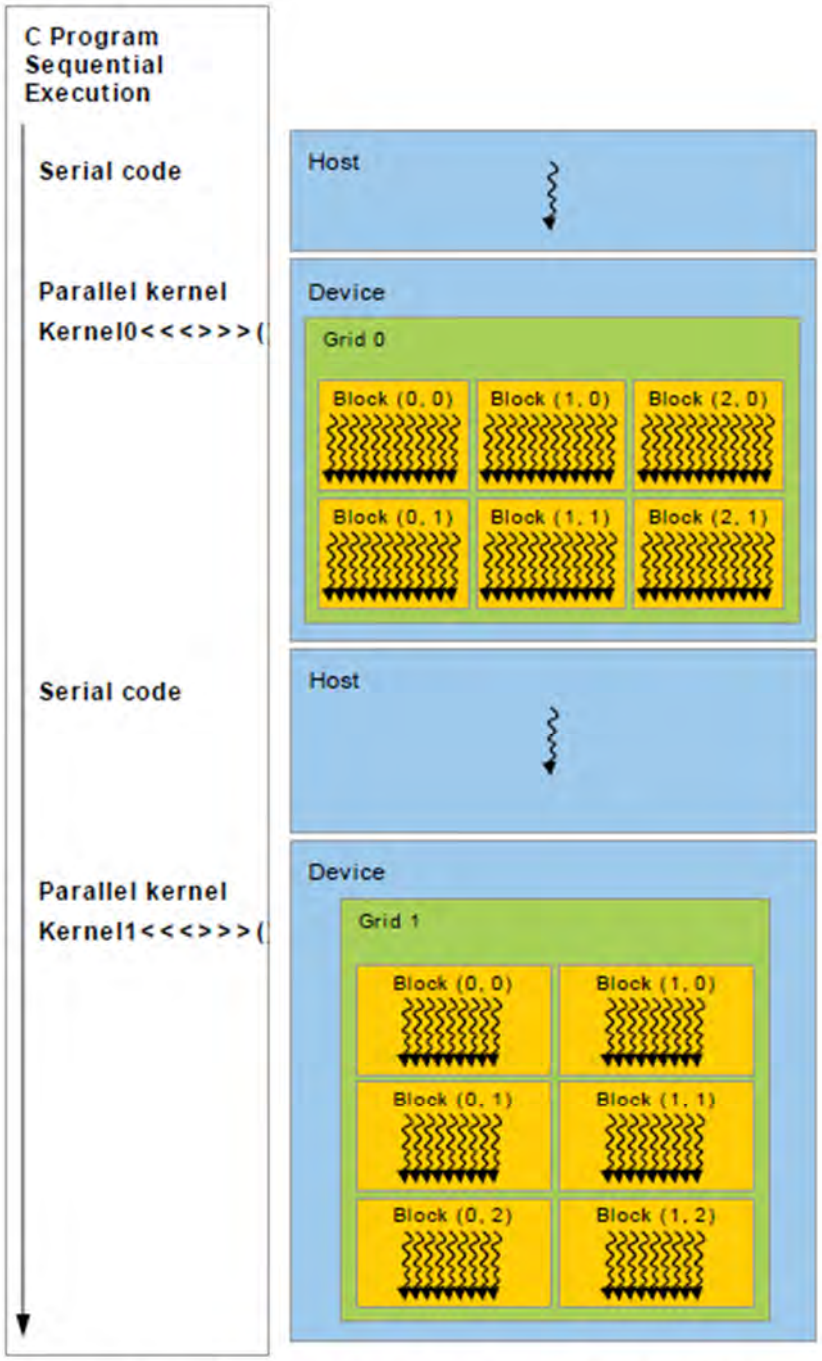
Figure 5 Automatic Scalability
NVidia CUDA C Programming Guide pp. 5, 12 (v10.1 May 2019), https://docs.nvidia.com/pdf/CUDA_C_Programming_Guide.pdf.
</td>
</tr>
<tr>
<td>a plurality of cluster node modules, wherein each cluster node module is stored in a computer-readable medium and configured to communicate with a single-node kernel and with one or more other cluster node modules, to accept instructions</td>
<td>Each Accused Server Product comprises a plurality of cluster node modules, wherein each cluster node module is stored in a computer-readable medium and configured to communicate with a single-node kernel and with one or more other cluster node modules, to accept instructions from a user interface or a script, and to interpret at least some of the user instructions such that the plurality of cluster node modules communicate with one another in order to act as a cluster.

For example, each GPU Accelerator has an associated cluster node module stored in memory. Each GPU Accelerator cluster node module is configured to communicate with a CUDA kernel associated with the cluster node module and configured to use the NVLink interconnect architecture to communicate with other cluster node modules associate with other GPU Accelerators. The GPU Accelerator cluster node modules accept instruction from a CPU or host executing a program (user interface or script), and interpret some of the user instructions such that the cluster node modules communicate with one another in order to act as a cluster.</td>
</tr>
</table>

from a user interface or a script, and to interpret at least some of the user instructions such that the plurality of cluster node modules communicate with one another in order to act as a cluster; and

2.4. Heterogeneous Programming

As illustrated by Figure 8, the CUDA programming model assumes that the CUDA threads execute on a physically separate *device* that operates as a coprocessor to the *host* running the C program. This is the case, for example, when the kernels execute on a GPU and the rest of the C program executes on a CPU.



Serial code executes on the host while parallel code executes on the device.

Figure 8 Heterogeneous Programming

NVidia CUDA C Programming Guide pp. 12-13 (v10.1 May 2019), https://docs.nvidia.com/pdf/CUDA_C_Programming_Guide.pdf.

Server Configuration with NVLink

In the following sections of this paper, we analyze the performance benefit of NVLink for several algorithms and applications by comparing model systems based on PCIe-interconnected next-gen GPUs to otherwise-identical systems with NVLink-interconnected GPUs. GPUs are connected to the CPU using existing PCIe connections, but the NVLink configurations augment this with interconnections among the GPUs for peer-to-peer communication. The following analyses assume future-generation GPUs with performance higher than that of today's GPUs, so as to better correspond with the GPUs that will be contemporary with NVLink.

The second scenario, shown in Figure 3, compares a pair of configurations with four GPUs each, referred to as *4-GPU-PCIe* and *4-GPU-NVLink*. 4-GPU-PCIe uses a PCIe tree topology, again mirroring a configuration used commonly today. 4-GPU-NVLink enhances this by providing point-to-point connections with either one or two NVLink connections per pair of GPUs, yielding 16 GB/s or 32 GB/s effective bandwidth per direction per pair, respectively.

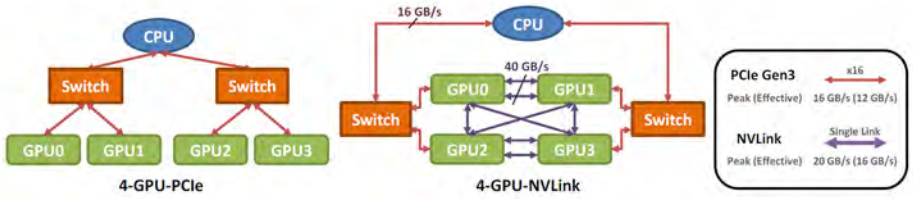


Figure 3: Comparing 4-GPU topologies with NVLink and PCIe. In 4-GPU-NVLink, GPU0 and GPU1 have 40 GB/s peak bandwidth between them, as do GPU2 and GPU3. The other peer-to-peer connections have 20 GB/s peak bandwidth.

The exchange algorithm begins with data evenly distributed across multiple GPUs as shown below.

Original Data

In this diagram, the color indicates which device the original data began on.

The first step is to hash the data to determine which GPU's bin each element should be placed in. For sorting, this might be as simple as extracting the highest order bits, but in theory we could apply any arbitrary hashing function to determine the keys (i.e., the destination bins). For this example, we will use the hash function `ceil(x/5)`, which produces the following keys given the data above.

Calculated Keys

Next we locally sort the data based on the hashed keys, yielding the following. Note that since we know these keys are in the range of `[0, NUM_GPUS)`, we can restrict the sorting to only a couple of bits.

Locally-sorted Data

Locally-sorted Keys

Next, we transfer all data with a key of 0 to device 0 and all data with a key of 1 to device 1. This becomes an all-to-all exchange among GPUs, producing the following.

Exchanged Data

Multi-GPU sorting is a simple extension of the above. After exchanging the data, we simply sort the data again locally to produce the final sorted list.

Fully-sorted Data

Here we can see the final list has been completely sorted. For many applications, the initial data exchange is all that is needed. Thus we will model both the exchange and the final sort separately.

Performance Considerations

The exchange algorithm can be easily pipelined by operating on small portions of the domain. The first step of the pipeline is to perform the local sort on the generated keys; the second is to perform the all-to-all exchange. This allows the initial sorting and the exchange to be completely overlapped. For the full multi-GPU sorting algorithm, the final local sort cannot begin until all data has arrived, so this portion cannot be pipelined with the other two.

If we assume the original data is randomly distributed among GPUs following a uniform random distribution, then each GPU must communicate `W*N/P` bytes of data to every other GPU, where W is the size of each element in bytes, N is the number of elements per GPU, and P is the number of GPUs.

In a PCIe tree topology, the communication is limited by the bisection bandwidth. Each GPU must communicate half of its data (N/2) across the top link in the system, and there are P/2 devices all trying to communicate across the same PCIe lanes in the same direction.

On an NVLink system, however, the communication can occur in parallel, because there are dedicated links between all pairs of GPUs.

NVIDIA NVLink High-Speed Interconnect: Application Performance p. 4-7, https://info.nvidianews.com/rs/nvidia/images/NVIDIA%20NVLink%20High-Speed%20Interconnect%20Application%20Performance%20Brief.pdf.

To address this issue, Tesla P100 features NVIDIA's new high-speed interface, NVLink, that provides GPU-to-GPU data transfers at up to 160 Gigabytes/second of bidirectional bandwidth—5x the bandwidth of PCIe Gen 3 x16. Figure 4 shows NVLink connecting eight Tesla P100 Accelerators in a Hybrid Cube Mesh Topology.

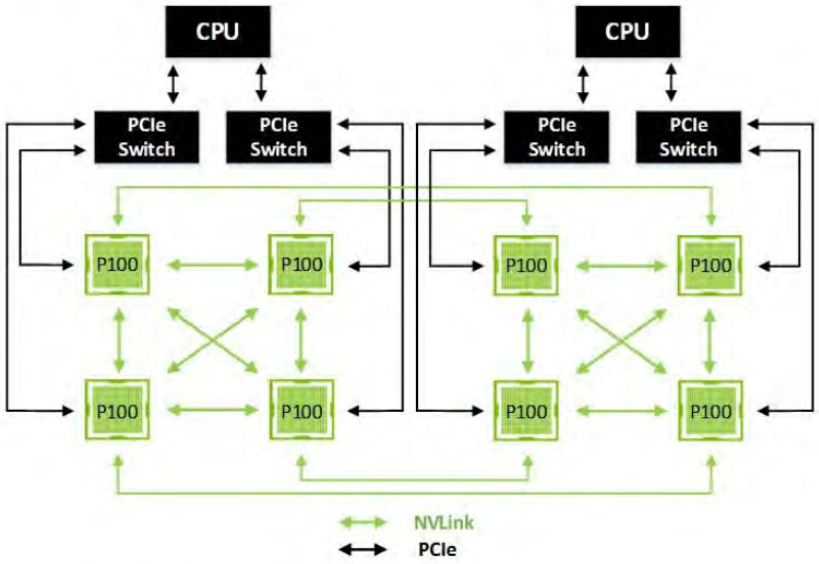


Figure 4. NVLink Connecting Eight Tesla P100 Accelerators in a Hybrid Cube Mesh Topology

NVIDIA Tesla P100: The Most Advanced Datacenter Accelerator Ever Built Featuring Pascal GP100, the World's Fastest GPU p. 7, https://images.nvidia.com/content/pdf/tesla/whitepaper/pascal-architecture-whitepaper.pdf.

NVLINK: HIGHER BANDWIDTH, MORE LINKS, MORE FEATURES

NVLink is NVIDIA's high-speed interconnect technology first introduced in 2016 with the Tesla P100 accelerator and Pascal GP100 GPU. NVLink provides significantly more performance for both GPU-to-GPU and GPU-to-CPU system configurations compared to using PCIe interconnects. Refer to the *Pascal Architecture Whitepaper* for basic details on NVLink technology. Tesla V100 introduces the second generation of NVLink, which provides higher link speeds, more links per GPU, CPU mastering, cache coherence, and scalability improvements.

NVidia Tesla V100 GPU Architecture: The World's Most Advanced Data Center GPU, p. 19, https://images.nvidia.com/content/volta-architecture/pdf/volta-architecture-whitepaper.pdf.

2.4. Heterogeneous Programming

As illustrated by Figure 8, the CUDA programming model assumes that the CUDA threads execute on a physically separate *device* that operates as a coprocessor to the *host* running the C program. This is the case, for example, when the kernels execute on a GPU and the rest of the C program executes on a CPU.

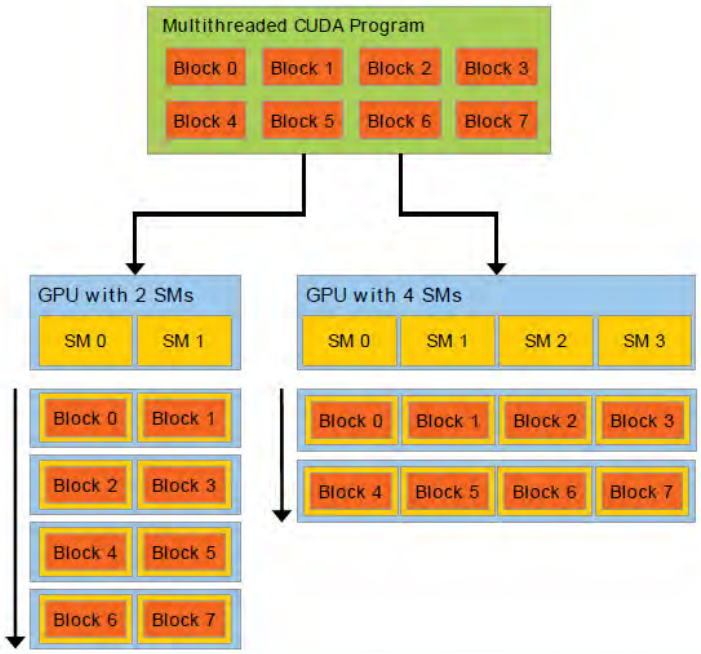
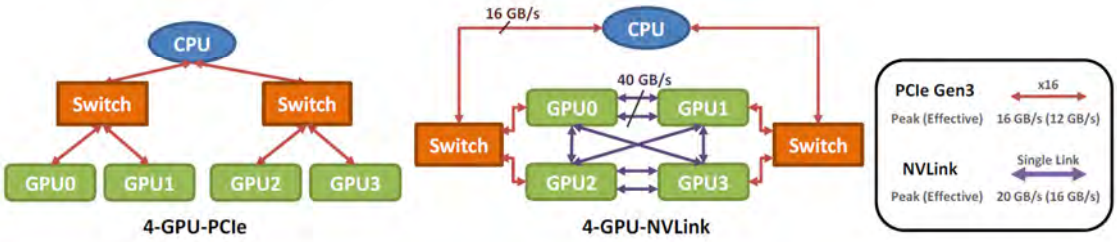
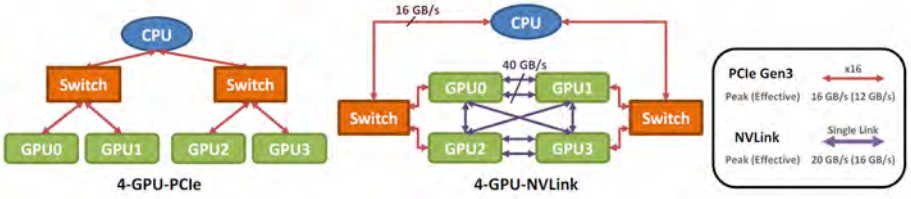


Figure 5 Automatic Scalability

NVidia CUDA C Programming Guide pp. 5, 12 (v10.1 May 2019), https://docs.nvidia.com/pdf/CUDA_C_Programming_Guide.pdf.

a communications network to connect the nodes;	Each Accused Server Product comprises a communications network to connect the nodes. For example, each Accused Server Product uses the NVLink interconnect architecture to connect the GPU Accelerators and enable peer-to-peer communications between GPU Accelerators.

	The second scenario, shown in Figure 3, compares a pair of configurations with four GPUs each, referred to as *4-GPU-PCIe* and *4-GPU-NVLink*. 4-GPU-PCIe uses a PCIe tree topology, again mirroring a configuration used commonly today. 4-GPU-NVLink enhances this by providing point-to-point connections with either one or two NVLink connections per pair of GPUs, yielding 16 GB/s or 32 GB/s effective bandwidth per direction per pair, respectively.  Figure 3: Comparing 4-GPU topologies with NVLink and PCIe. In 4-GPU-NVLink, GPU0 and GPU1 have 40 GB/s peak bandwidth between them, as do GPU2 and GPU3. The other peer-to-peer connections have 20 GB/s peak bandwidth. NVIDIA NVLink High-Speed Interconnect: Application Performance p. 5, https://info.nvidianews.com/rs/nvidia/images/NVIDIA%20NVLink%20High-Speed%20Interconnect%20Application%20Performance%20Brief.pdf.
wherein the plurality of cluster node modules are configured to cooperate to interpret, translate, or interpret and translate commands for execution by a plurality of single-node kernel modules, and wherein at least one of the plurality of cluster node modules returns a result to the user interface.	Each Accused Server Product comprises a plurality of cluster node modules configured to cooperate to interpret, translate, or interpret and translate commands for execution by a plurality of single-node kernel modules, and wherein at least one of the plurality of cluster node modules returns a result to the user interface. For example, the GPU Accelerator cluster node modules are configured to cooperate to interpret, translate, or interpret and translate command for execution by the CUDA kernels associated with each GPU Accelerator. At least one of the GPU Accelerator CUDA kernels returns a result to the CPU or host (user interface). **Server Configuration with NVLink** In the following sections of this paper, we analyze the performance benefit of NVLink for several algorithms and applications by comparing model systems based on PCIe-interconnected next-gen GPUs to otherwise-identical systems with NVLink-interconnected GPUs. GPUs are connected to the CPU using existing PCIe connections, but the NVLink configurations augment this with interconnections among the GPUs for peer-to-peer communication. The following analyses assume future-generation GPUs with performance higher than that of today's GPUs, so as to better correspond with the GPUs that will be contemporary with NVLink. The second scenario, shown in Figure 3, compares a pair of configurations with four GPUs each, referred to as *4-GPU-PCIe* and *4-GPU-NVLink*. 4-GPU-PCIe uses a PCIe tree topology, again mirroring a configuration used commonly today. 4-GPU-NVLink enhances this by providing point-to-point connections with either one or two NVLink connections per pair of GPUs, yielding 16 GB/s or 32 GB/s effective bandwidth per direction per pair, respectively.  Figure 3: Comparing 4-GPU topologies with NVLink and PCIe. In 4-GPU-NVLink, GPU0 and GPU1 have 40 GB/s peak bandwidth between them, as do GPU2 and GPU3. The other peer-to-peer connections have 20 GB/s peak bandwidth.

The exchange algorithm begins with data evenly distributed across multiple GPUs as shown below.

Original Data

In this diagram, the color indicates which device the original data began on.

The first step is to hash the data to determine which GPU's bin each element should be placed in. For sorting, this might be as simple as extracting the highest order bits, but in theory we could apply any arbitrary hashing function to determine the keys (i.e., the destination bins). For this example, we will use the hash function `ceil(x/5)`, which produces the following keys given the data above.

Calculated Keys

Next we locally sort the data based on the hashed keys, yielding the following. Note that since we know these keys are in the range of `[0, NUM_GPUS)`, we can restrict the sorting to only a couple of bits.

Locally-sorted Data

Locally-sorted Keys

Next, we transfer all data with a key of 0 to device 0 and all data with a key of 1 to device 1. This becomes an all-to-all exchange among GPUs, producing the following.

Exchanged Data

Multi-GPU sorting is a simple extension of the above. After exchanging the data, we simply sort the data again locally to produce the final sorted list.

Fully-sorted Data

Here we can see the final list has been completely sorted. For many applications, the initial data exchange is all that is needed. Thus we will model both the exchange and the final sort separately.

Performance Considerations

The exchange algorithm can be easily pipelined by operating on small portions of the domain. The first step of the pipeline is to perform the local sort on the generated keys; the second is to perform the all-to-all exchange. This allows the initial sorting and the exchange to be completely overlapped. For the full multi-GPU sorting algorithm, the final local sort cannot begin until all data has arrived, so this portion cannot be pipelined with the other two.

If we assume the original data is randomly distributed among GPUs following a uniform random distribution, then each GPU must communicate `W*N/P` bytes of data to every other GPU, where W is the size of each element in bytes, N is the number of elements per GPU, and P is the number of GPUs.

In a PCIe tree topology, the communication is limited by the bisection bandwidth. Each GPU must communicate half of its data (N/2) across the top link in the system, and there are P/2 devices all trying to communicate across the same PCIe lanes in the same direction.

On an NVLink system, however, the communication can occur in parallel, because there are dedicated links between all pairs of GPUs.

NVIDIA NVLink High-Speed Interconnect: Application Performance p. 4-7, https://info.nvidianews.com/rs/nvidia/images/NVIDIA%20NVLink%20High-Speed%20Interconnect%20Application%20Performance%20Brief.pdf.

To address this issue, Tesla P100 features NVIDIA's new high-speed interface, NVLink, that provides GPU-to-GPU data transfers at up to 160 Gigabytes/second of bidirectional bandwidth—5x the bandwidth of PCIe Gen 3 x16. Figure 4 shows NVLink connecting eight Tesla P100 Accelerators in a Hybrid Cube Mesh Topology.

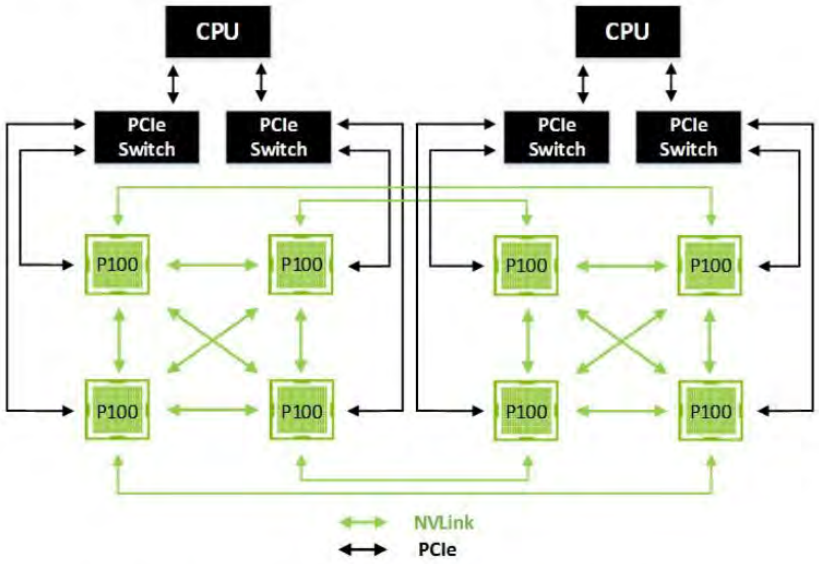


Figure 4. NVLink Connecting Eight Tesla P100 Accelerators in a Hybrid Cube Mesh Topology

NVIDIA Tesla P100: The Most Advanced Datacenter Accelerator Ever Built Featuring Pascal GP100, the World's Fastest GPU p. 7, https://images.nvidia.com/content/pdf/tesla/whitepaper/pascal-architecture-whitepaper.pdf.

NVLINK: HIGHER BANDWIDTH, MORE LINKS, MORE FEATURES

NVLink is NVIDIA's high-speed interconnect technology first introduced in 2016 with the Tesla P100 accelerator and Pascal GP100 GPU. NVLink provides significantly more performance for both GPU-to-GPU and GPU-to-CPU system configurations compared to using PCIe interconnects. Refer to the *Pascal Architecture Whitepaper* for basic details on NVLink technology. Tesla V100 introduces the second generation of NVLink, which provides higher link speeds, more links per GPU, CPU mastering, cache coherence, and scalability improvements.

NVidia Tesla V100 GPU Architecture: The World's Most Advanced Data Center GPU, p. 19, https://images.nvidia.com/content/volta-architecture/pdf/volta-architecture-whitepaper.pdf.

This first example combines two numbers together on the GPU with a per-thread ID and returns the values in an array. Without managed memory, both host- and device-side storage for the return values is required (**host_ret** and **ret** in the example), as is an explicit copy between the two using **cudaMemcpy()**.

Compare this with the Unified Memory version of the program, which allows direct access of GPU data from the host. Notice the **cudaMallocManaged()** routine, which returns a pointer valid from both host and device code. This allows **ret** to be used without a separate **host_ret** copy, greatly simplifying and reducing the size of the program.

```
__global__ void AplusB(int *ret, int a, int b) {
    ret[threadIdx.x] = a + b + threadIdx.x;
}
int main() {
    int *ret;
    cudaMallocManaged(&ret, 1000 * sizeof(int));
    AplusB<<< 1, 1000 >>>(ret, 10, 100);
    cudaDeviceSynchronize();
    for(int i = 0; i < 1000; i++)
        printf("%d: A+B = %d\n", i, ret[i]);
    cudaFree(ret);
    return 0;
}
```

It is worth a comment on the synchronization between host and device. Notice how in the non-managed example, the synchronous **cudaMemcpy()** routine is used both to synchronize the kernel (that is, to wait for it to finish running), and to transfer the data to the host. The Unified Memory examples do not call **cudaMemcpy()** and so require an explicit **cudaDeviceSynchronize()** before the host program can safely use the output from the GPU.

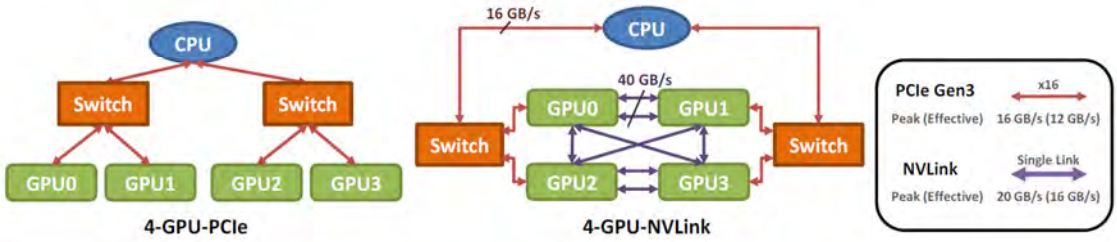
NVIDIA CUDA C Programming Guide pp. 281-82 (v10.1 May 2019), https://docs.nvidia.com/pdf/CUDA_C_Programming_Guide.pdf

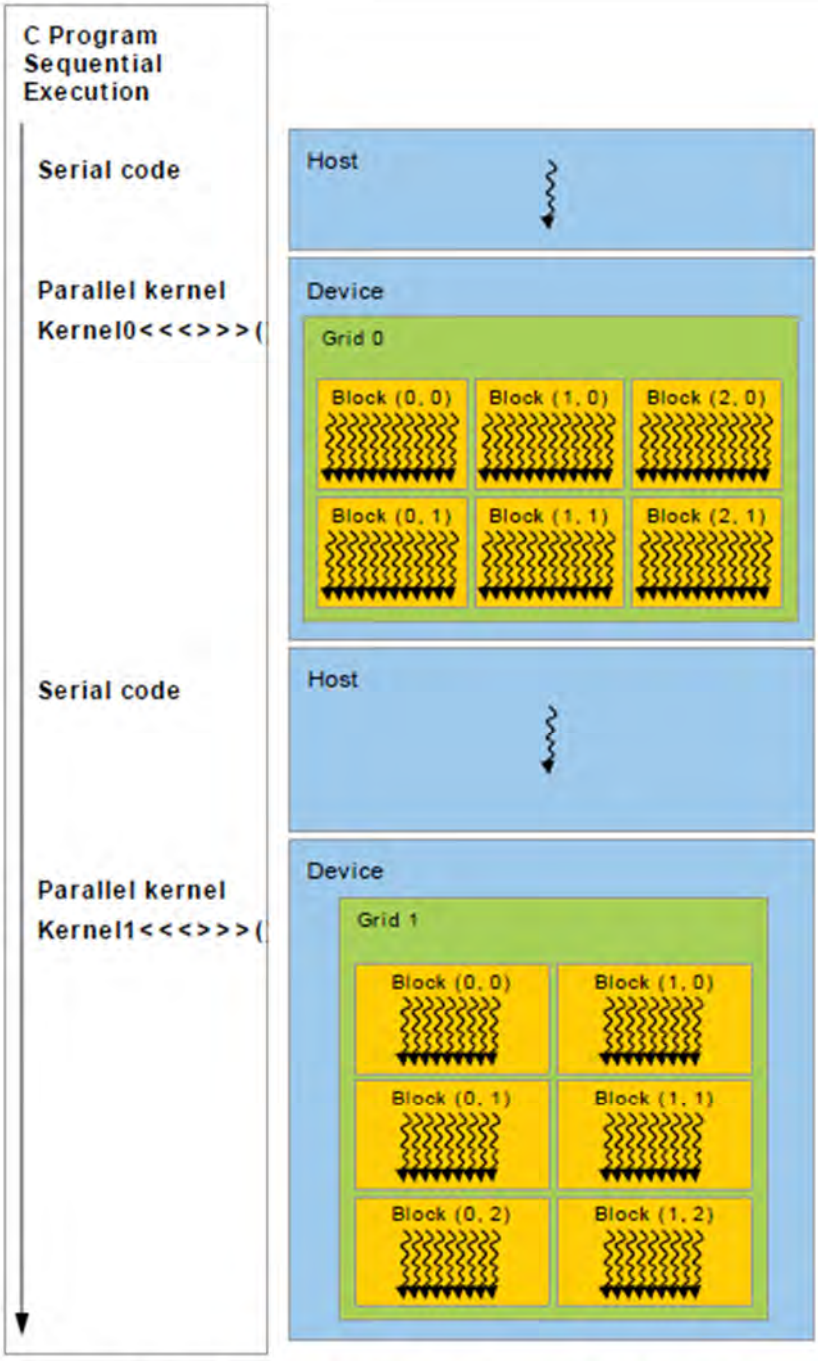
EXHIBIT G

EXHIBIT G – '877 Patent Exemplary Claim Chart

Claim 8 Language	NVIDIA's Infringing Products
A system for performing an instruction received from a front end by executing commands on one or more hardware processors, the system comprising:	Each Accused Server Product is a system for performing an instruction received from a front end by executing commands on one or more hardware processors as follows. [1]
a plurality of nodes, wherein each node is configured to access a computer-readable memory system comprising program code for a single-node kernel module, and wherein each single-node kernel module is configured to interpret instructions received by the single-node kernel module	Each Accused Server Product comprises a plurality of nodes, wherein each node is configured to access a computer-readable memory system comprising program code for a single-node kernel module, and wherein each single-node kernel module is configured to interpret instructions received by the single-node kernel module into commands that are executable by a special purpose microprocessor For example, each Accused Server Product includes a cluster of GPU Accelerator (nodes). Each GPU Accelerator is configured to access memory that stores the CUDA kernel associated with that GPU Accelerator. The CUDA kernel is configured to interpret instructions from the CPU or host into commands that are executable by CUDA processor cores within the GPU Accelerator.

[1] Each Accused Server Product and Accused GPU Product herein, including DGX Station, DGX-1, DGX-2, HGX-1, HGX-2, Tesla V100, and Tesla P100, uses or is configured to use the NVLink architecture and technology described in the references herein including NVIDIA's Tesla P100,Tesla V100, and NVLink whitepapers and CUDA C Programming Guide. See NVIDIA DGX Station: Maximize Your Data Science Team Productivity, https://www.nvidia.com/content/dam/en-zz/Solutions/Data-Center/dgx-station/nvidia-dgx-station-datasheet.pdf; NVIDIA DGX-1: Essential Instrument for AI Research, https://www.nvidia.com/content/dam/en-zz/Solutions/Data-Center/dgx-1/dgx-1-ai-supercomputer-datasheet-v4.pdf; NVIDIA DGX-2: The World's Most Powerful Deep Learning System for the Most Complex AI Challenges, https://www.nvidia.com/content/dam/en-zz/Solutions/Data-Center/dgx-2/dgx-2-print-datasheet-738070-nvidia-a4-web-uk.pdf; NVIDIA HGX-2: Powered by NVIDIA Tesla V100 GPUs and NVSwitch https://www.nvidia.com/en-us/data-center/hgx/; NVIDIA Tesla P100: GPU Accelerator, https://www.nvidia.com/content/dam/en-zz/Solutions/Data-Center/tesla-p100/pdf/nvidia-tesla-p100-datasheet.pdf; NVIDIA Tesla V100: GPU Accelerator, https://images.nvidia.com/content/technologies/volta/pdf/437317-Volta-V100-DS-NV-US-WEB.pdf.

into commands that are executable by a special purpose microprocessor;	The second scenario, shown in Figure 3, compares a pair of configurations with four GPUs each, referred to as *4-GPU-PCIe* and *4-GPU-NVLink*. 4-GPU-PCIe uses a PCIe tree topology, again mirroring a configuration used commonly today. 4-GPU-NVLink enhances this by providing point-to-point connections with either one or two NVLink connections per pair of GPUs, yielding 16 GB/s or 32 GB/s effective bandwidth per direction per pair, respectively.  Figure 3: Comparing 4-GPU topologies with NVLink and PCIe. In 4-GPU-NVLink, GPU0 and GPU1 have 40 GB/s peak bandwidth between them, as do GPU2 and GPU3. The other peer-to-peer connections have 20 GB/s peak bandwidth. NVIDIA NVLink High-Speed Interconnect: Application Performance p. 5, https://info.nvidianews.com/rs/nvidia/images/NVIDIA%20NVLink%20High-Speed%20Interconnect%20Application%20Performance%20Brief.pdf. ## 2.4. Heterogeneous Programming As illustrated by Figure 8, the CUDA programming model assumes that the CUDA threads execute on a physically separate *device* that operates as a coprocessor to the *host* running the C program. This is the case, for example, when the kernels execute on a GPU and the rest of the C program executes on a CPU.

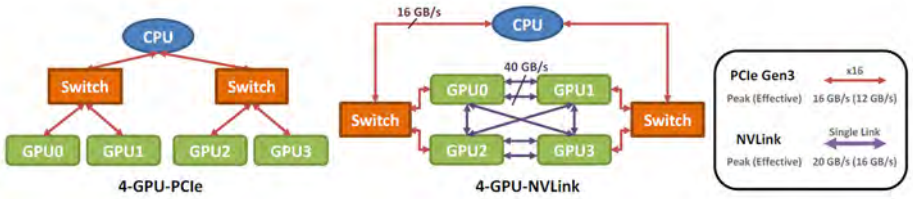


Serial code executes on the host while parallel code executes on the device.

Figure 8 Heterogeneous Programming

NVidia CUDA C Programming Guide pp. 12-13 (v10.1 May 2019), https://docs.nvidia.com/pdf/CUDA_C_Programming_Guide.pdf.

a plurality of cluster node modules, wherein each cluster node module is stored in a computer-readable	Each Accused Server Product comprises a plurality of cluster node modules, wherein each cluster node module is stored in a computer-readable memory system and configured to communicate with a single-node kernel and with one or more other cluster node modules, to accept instructions, and to interpret at least some of the instructions such that the plurality of cluster node modules communicate with one another in order to act as a cluster in executing commands using one or more hardware processors. For example, each GPU Accelerator has an associated cluster node module stored in

memory system and configured to communicate with a single-node kernel and with one or more other cluster node modules, to accept instructions, and to interpret at least some of the instructions such that the plurality of cluster node modules communicate with one another in order to act as a cluster in executing commands using one or more hardware processors; and	memory. Each GPU Accelerator cluster node module is configured to communicate with a CUDA kernel associated with that GPU Accelerator and to use the NVLink interconnect architecture to communicate with one or more other GPU Accelerator cluster node modules. Each GPU Accelerator cluster node module is configured to use NVLink to accept instructions, and to interpret at least some of the instructions such that the GPU Accelerator cluster node modules communicate with one another in order to act as a cluster in executing commands using one or more hardware processors. **Server Configuration with NVLink** In the following sections of this paper, we analyze the performance benefit of NVLink for several algorithms and applications by comparing model systems based on PCIe-interconnected next-gen GPUs to otherwise-identical systems with NVLink-interconnected GPUs. GPUs are connected to the CPU using existing PCIe connections, but the NVLink configurations augment this with interconnections among the GPUs for peer-to-peer communication. The following analyses assume future-generation GPUs with performance higher than that of today's GPUs, so as to better correspond with the GPUs that will be contemporary with NVLink. The second scenario, shown in Figure 3, compares a pair of configurations with four GPUs each, referred to as *4-GPU-PCIe* and *4-GPU-NVLink*. 4-GPU-PCIe uses a PCIe tree topology, again mirroring a configuration used commonly today. 4-GPU-NVLink enhances this by providing point-to-point connections with either one or two NVLink connections per pair of GPUs, yielding 16 GB/s or 32 GB/s effective bandwidth per direction per pair, respectively.  Figure 3: Comparing 4-GPU topologies with NVLink and PCIe. In 4-GPU-NVLink, GPU0 and GPU1 have 40 GB/s peak bandwidth between them, as do GPU2 and GPU3. The other peer-to-peer connections have 20 GB/s peak bandwidth.

The exchange algorithm begins with data evenly distributed across multiple GPUs as shown below.

Original Data

In this diagram, the color indicates which device the original data began on.

The first step is to hash the data to determine which GPU's bin each element should be placed in. For sorting, this might be as simple as extracting the highest order bits, but in theory we could apply any arbitrary hashing function to determine the keys (i.e., the destination bins). For this example, we will use the hash function `ceil(x/5)`, which produces the following keys given the data above.

Calculated Keys

Next we locally sort the data based on the hashed keys, yielding the following. Note that since we know these keys are in the range of `[0, NUM_GPUS)`, we can restrict the sorting to only a couple of bits.

Locally-sorted Data

Locally-sorted Keys

Next, we transfer all data with a key of 0 to device 0 and all data with a key of 1 to device 1. This becomes an all-to-all exchange among GPUs, producing the following.

Exchanged Data

Multi-GPU sorting is a simple extension of the above. After exchanging the data, we simply sort the data again locally to produce the final sorted list.

Fully-sorted Data

Here we can see the final list has been completely sorted. For many applications, the initial data exchange is all that is needed. Thus we will model both the exchange and the final sort separately.

Performance Considerations

The exchange algorithm can be easily pipelined by operating on small portions of the domain. The first step of the pipeline is to perform the local sort on the generated keys; the second is to perform the all-to-all exchange. This allows the initial sorting and the exchange to be completely overlapped. For the full multi-GPU sorting algorithm, the final local sort cannot begin until all data has arrived, so this portion cannot be pipelined with the other two.

If we assume the original data is randomly distributed among GPUs following a uniform random distribution, then each GPU must communicate `W*N/P` bytes of data to every other GPU, where W is the size of each element in bytes, N is the number of elements per GPU, and P is the number of GPUs.

In a PCIe tree topology, the communication is limited by the bisection bandwidth. Each GPU must communicate half of its data (N/2) across the top link in the system, and there are P/2 devices all trying to communicate across the same PCIe lanes in the same direction.

On an NVLink system, however, the communication can occur in parallel, because there are dedicated links between all pairs of GPUs.

NVIDIA NVLink High-Speed Interconnect: Application Performance p. 4-7, https://info.nvidianews.com/rs/nvidia/images/NVIDIA%20NVLink%20High-Speed%20Interconnect%20Application%20Performance%20Brief.pdf.

To address this issue, Tesla P100 features NVIDIA's new high-speed interface, NVLink, that provides GPU-to-GPU data transfers at up to 160 Gigabytes/second of bidirectional bandwidth—5x the bandwidth of PCIe Gen 3 x16. Figure 4 shows NVLink connecting eight Tesla P100 Accelerators in a Hybrid Cube Mesh Topology.

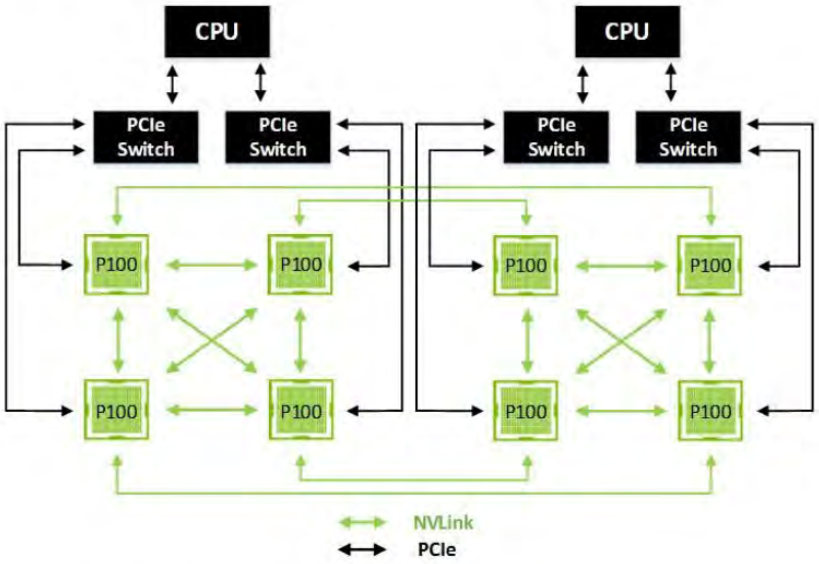
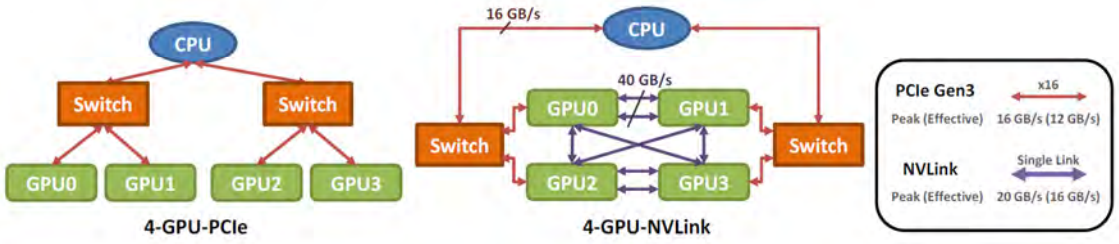
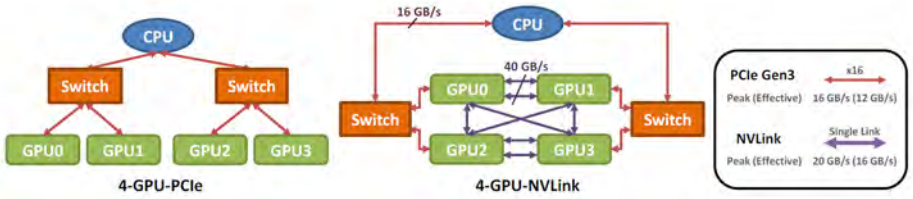


Figure 4. NVLink Connecting Eight Tesla P100 Accelerators in a Hybrid Cube Mesh Topology

NVIDIA Tesla P100: The Most Advanced Datacenter Accelerator Ever Built Featuring Pascal GP100, the World's Fastest GPU p. 7, https://images.nvidia.com/content/pdf/tesla/whitepaper/pascal-architecture-whitepaper.pdf.

	NVLINK: HIGHER BANDWIDTH, MORE LINKS, MORE FEATURES NVLink is NVIDIA's high-speed interconnect technology first introduced in 2016 with the Tesla P100 accelerator and Pascal GP100 GPU. NVLink provides significantly more performance for both GPU-to-GPU and GPU-to-CPU system configurations compared to using PCIe interconnects. Refer to the *Pascal Architecture Whitepaper* for basic details on NVLink technology. Tesla V100 introduces the second generation of NVLink, which provides higher link speeds, more links per GPU, CPU mastering, cache coherence, and scalability improvements. NVidia Tesla V100 GPU Architecture: The World's Most Advanced Data Center GPU, p. 19, https://images.nvidia.com/content/volta-architecture/pdf/volta-architecture-whitepaper.pdf.
a communications system configured to connect the plurality of nodes;	Each Accused Server Product comprises communications system configured to connect the plurality of nodes. For example, each Accused Server Product uses the NVLink interconnect architecture to connects the GPU Accelerators and enable peer-to-peer communications between GPU Accelerators. The second scenario, shown in Figure 3, compares a pair of configurations with four GPUs each, referred to as *4-GPU-PCIe* and *4-GPU-NVLink*. 4-GPU-PCIe uses a PCIe tree topology, again mirroring a configuration used commonly today. 4-GPU-NVLink enhances this by providing point-to-point connections with either one or two NVLink connections per pair of GPUs, yielding 16 GB/s or 32 GB/s effective bandwidth per direction per pair, respectively.  Figure 3: Comparing 4-GPU topologies with NVLink and PCIe. In 4-GPU-NVLink, GPU0 and GPU1 have 40 GB/s peak bandwidth between them, as do GPU2 and GPU3. The other peer-to-peer connections have 20 GB/s peak bandwidth. NVIDIA NVLink High-Speed Interconnect: Application Performance p. 5, https://info.nvidianews.com/rs/nvidia/images/NVIDIA%20NVLink%20High-Speed%20Interconnect%20Application%20Performance%20Brief.pdf.
wherein the plurality of cluster node modules cooperate to interpret and translate, as needed, the instruction for execution by a plurality of single-node kernel modules, and wherein at	Each Accused Server Product comprises a plurality of cluster node modules that cooperate to interpret and translate, as needed, the instruction for execution by a plurality of single-node kernel modules, and wherein at least one of the plurality of cluster node modules returns a result to the front end. For example, the GPU Accelerator cluster node modules cooperate, using the NVLink interconnect architecture, to interpret and translate the instruction for execution by kernels associated with each GPU Accelerator in the cluster. At least one of the GPU Accelerator cluster node modules returns a result to the CPU or host (front end).

least one of the plurality of cluster node modules returns a result to the front end.	**Server Configuration with NVLink** In the following sections of this paper, we analyze the performance benefit of NVLink for several algorithms and applications by comparing model systems based on PCIe-interconnected next-gen GPUs to otherwise-identical systems with NVLink-interconnected GPUs. GPUs are connected to the CPU using existing PCIe connections, but the NVLink configurations augment this with interconnections among the GPUs for peer-to-peer communication. The following analyses assume future-generation GPUs with performance higher than that of today's GPUs, so as to better correspond with the GPUs that will be contemporary with NVLink. The second scenario, shown in Figure 3, compares a pair of configurations with four GPUs each, referred to as *4-GPU-PCIe* and *4-GPU-NVLink*. 4-GPU-PCIe uses a PCIe tree topology, again mirroring a configuration used commonly today. 4-GPU-NVLink enhances this by providing point-to-point connections with either one or two NVLink connections per pair of GPUs, yielding 16 GB/s or 32 GB/s effective bandwidth per direction per pair, respectively.  Figure 3: Comparing 4-GPU topologies with NVLink and PCIe. In 4-GPU-NVLink, GPU0 and GPU1 have 40 GB/s peak bandwidth between them, as do GPU2 and GPU3. The other peer-to-peer connections have 20 GB/s peak bandwidth.

The exchange algorithm begins with data evenly distributed across multiple GPUs as shown below.

Original Data

In this diagram, the color indicates which device the original data began on.

The first step is to hash the data to determine which GPU's bin each element should be placed in. For sorting, this might be as simple as extracting the highest order bits, but in theory we could apply any arbitrary hashing function to determine the keys (i.e., the destination bins). For this example, we will use the hash function `ceil(x/5)`, which produces the following keys given the data above.

Calculated Keys

Next we locally sort the data based on the hashed keys, yielding the following. Note that since we know these keys are in the range of `[0, NUM_GPUS)`, we can restrict the sorting to only a couple of bits.

Locally-sorted Data

Locally-sorted Keys

Next, we transfer all data with a key of 0 to device 0 and all data with a key of 1 to device 1. This becomes an all-to-all exchange among GPUs, producing the following.

Exchanged Data

Multi-GPU sorting is a simple extension of the above. After exchanging the data, we simply sort the data again locally to produce the final sorted list.

Fully-sorted Data

Here we can see the final list has been completely sorted. For many applications, the initial data exchange is all that is needed. Thus we will model both the exchange and the final sort separately.

Performance Considerations

The exchange algorithm can be easily pipelined by operating on small portions of the domain. The first step of the pipeline is to perform the local sort on the generated keys; the second is to perform the all-to-all exchange. This allows the initial sorting and the exchange to be completely overlapped. For the full multi-GPU sorting algorithm, the final local sort cannot begin until all data has arrived, so this portion cannot be pipelined with the other two.

If we assume the original data is randomly distributed among GPUs following a uniform random distribution, then each GPU must communicate `W*N/P` bytes of data to every other GPU, where W is the size of each element in bytes, N is the number of elements per GPU, and P is the number of GPUs.

In a PCIe tree topology, the communication is limited by the bisection bandwidth. Each GPU must communicate half of its data (N/2) across the top link in the system, and there are P/2 devices all trying to communicate across the same PCIe lanes in the same direction.

On an NVLink system, however, the communication can occur in parallel, because there are dedicated links between all pairs of GPUs.

NVIDIA NVLink High-Speed Interconnect: Application Performance p. 4-7, https://info.nvidianews.com/rs/nvidia/images/NVIDIA%20NVLink%20High-Speed%20Interconnect%20Application%20Performance%20Brief.pdf.

To address this issue, Tesla P100 features NVIDIA's new high-speed interface, NVLink, that provides GPU-to-GPU data transfers at up to 160 Gigabytes/second of bidirectional bandwidth—5x the bandwidth of PCIe Gen 3 x16. Figure 4 shows NVLink connecting eight Tesla P100 Accelerators in a Hybrid Cube Mesh Topology.

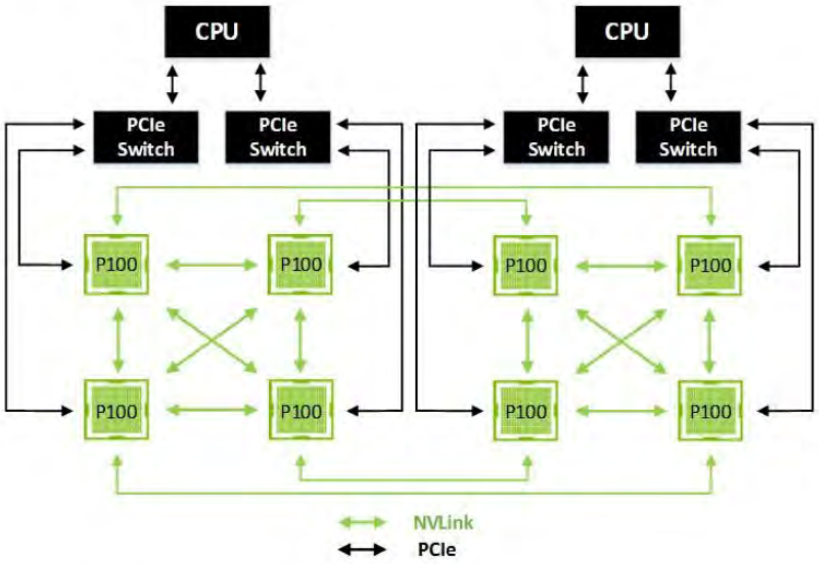


Figure 4. NVLink Connecting Eight Tesla P100 Accelerators in a Hybrid Cube Mesh Topology

NVIDIA Tesla P100: The Most Advanced Datacenter Accelerator Ever Built Featuring Pascal GP100, the World's Fastest GPU p. 7, https://images.nvidia.com/content/pdf/tesla/whitepaper/pascal-architecture-whitepaper.pdf.

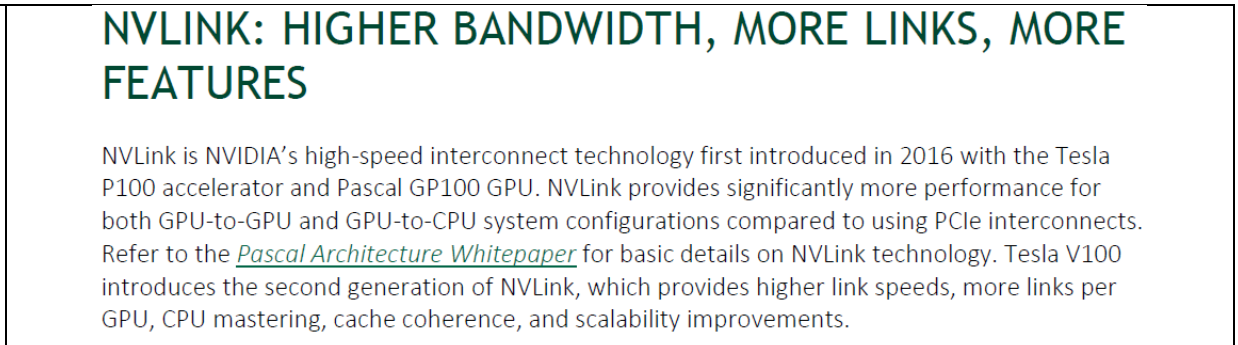
	 NVidia Tesla V100 GPU Architecture: The World's Most Advanced Data Center GPU, p. 19, https://images.nvidia.com/content/volta-architecture/pdf/volta-architecture-whitepaper.pdf.

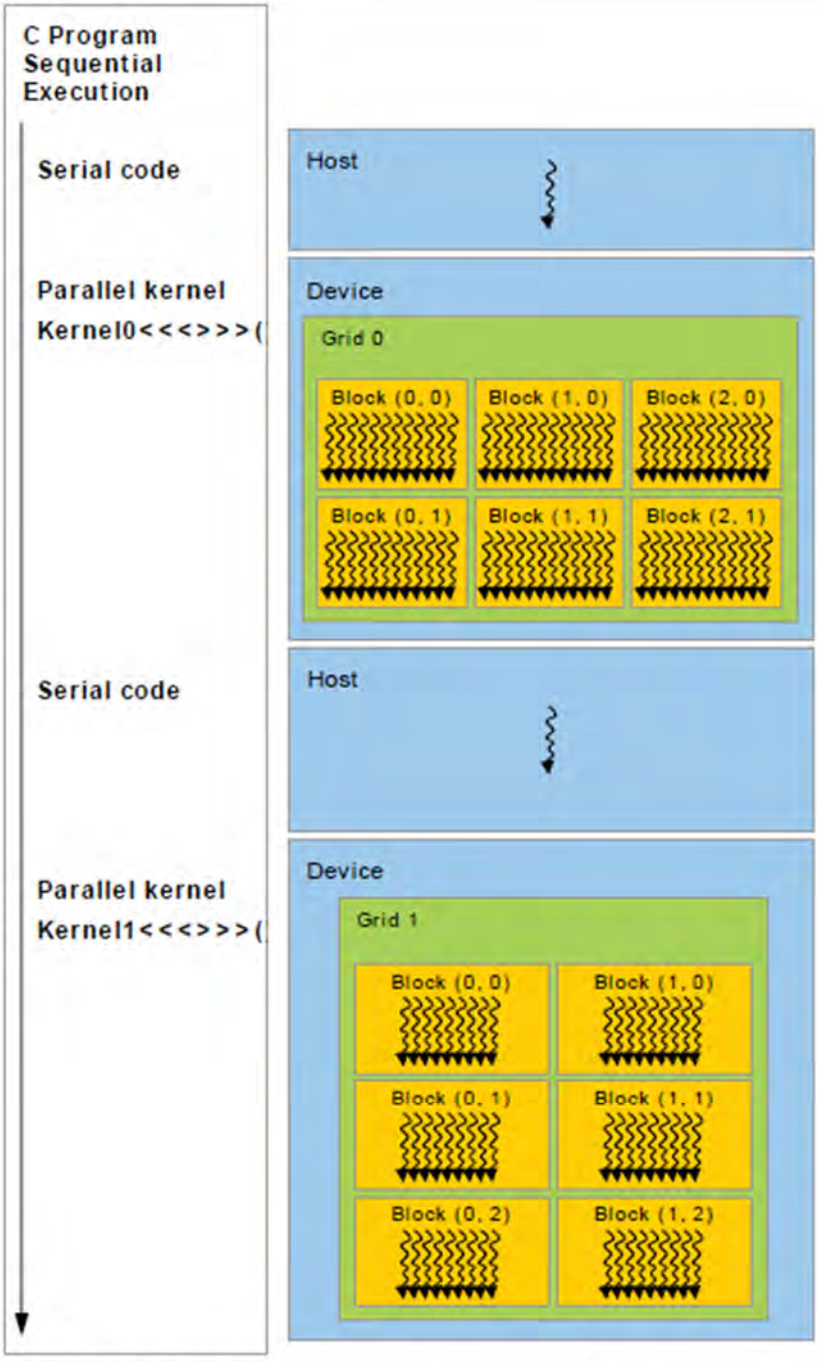
EXHIBIT H

EXHIBIT H – '768 Patent Exemplary Claim Chart

Claim 1 Language	NVIDIA's Infringing Products
A computer cluster comprising:	Each Accused Server Product is a computer cluster as follows. [1]
a plurality of nodes, wherein each of the plurality of nodes comprises a hardware processor, wherein one or more of the nodes are configured to receive a command to start a cluster initialization process for the computer cluster, and wherein each of the nodes is configured to access a non-transitory computer-readable medium comprising program code for a single-node kernel	Each Accused Server Product comprises a plurality of nodes, wherein each of the plurality of nodes comprises a hardware processor, wherein one or more of the nodes are configured to receive a command to start a cluster initialization process for the computer cluster, and wherein each of the nodes is configured to access a non-transitory computer-readable medium comprising program code for a single-node kernel that, when executed, is capable of causing the hardware processor to evaluate mathematical expressions. For example, each Accused Server Product comprises several GPU Accelerators (nodes) that each include a processor. One or more of the GPU Accelerators are configured to receive a command to start a cluster initialization process for the computer cluster. Each GPU Accelerator is configured to access memory with program code for a CUDA kernel that, when executed, is capable of causing the processor to evaluate mathematical expressions. Each of the GPU Accelerators is configured to access a memory storing the CUDA kernel that, when executed, is capable of causing the processor to evaluate mathematical expressions. As mentioned in Heterogeneous Programming, the CUDA programming model assumes a system composed of a host and a device, each with their own separate memory. Device Memory gives an overview of the runtime functions used to manage device memory.

[1] Each Accused Server Product and Accused GPU Product herein, including DGX Station, DGX-1, DGX-2, HGX-1, HGX-2, Tesla V100, and Tesla P100, uses or is configured to use the NVLink architecture and technology described in the references herein including NVIDIA's Tesla P100,Tesla V100, and NVLink whitepapers and CUDA C Programming Guide. See NVIDIA DGX Station: Maximize Your Data Science Team Productivity, https://www.nvidia.com/content/dam/en-zz/Solutions/Data-Center/dgx-station/nvidia-dgx-station-datasheet.pdf; NVIDIA DGX-1: Essential Instrument for AI Research, https://www.nvidia.com/content/dam/en-zz/Solutions/Data-Center/dgx-1/dgx-1-ai-supercomputer-datasheet-v4.pdf; NVIDIA DGX-2: The World's Most Powerful Deep Learning System for the Most Complex AI Challenges, https://www.nvidia.com/content/dam/en-zz/Solutions/Data-Center/dgx-2/dgx-2-print-datasheet-738070-nvidia-a4-web-uk.pdf; NVIDIA HGX-2: Powered by NVIDIA Tesla V100 GPUs and NVSwitch https://www.nvidia.com/en-us/data-center/hgx/; NVIDIA Tesla P100: GPU Accelerator, https://www.nvidia.com/content/dam/en-zz/Solutions/Data-Center/tesla-p100/pdf/nvidia-tesla-p100-datasheet.pdf; NVIDIA Tesla V100: GPU Accelerator, https://images.nvidia.com/content/technologies/volta/pdf/437317-Volta-V100-DS-NV-US-WEB.pdf.

that, when executed, is capable of causing the hardware processor to evaluate mathematical expressions; and	### 3.2.1. Initialization There is no explicit initialization function for the runtime; it initializes the first time a runtime function is called (more specifically any function other than functions from the device and version management sections of the reference manual). One needs to keep this in mind when timing runtime function calls and when interpreting the error code from the first call into the runtime. During initialization, the runtime creates a CUDA context for each device in the system (see Context for more details on CUDA contexts). This context is the *primary context* for this device and it is shared among all the host threads of the application. As part of this context creation, the device code is just-in-time compiled if necessary (see Just-in-Time Compilation) and loaded into device memory. This all happens under the hood and the runtime does not expose the primary context to the application. When a host thread calls `cudaDeviceReset()`, this destroys the primary context of the device the host thread currently operates on (i.e., the current device as defined in Device Selection). The next runtime function call made by any host thread that has this device as current will create a new primary context for this device. NVidia CUDA C Programming Guide pp. 18-19 (v10.1 May 2019), https://docs.nvidia.com/pdf/CUDA_C_Programming_Guide.pdf ## 2.4. Heterogeneous Programming As illustrated by Figure 8, the CUDA programming model assumes that the CUDA threads execute on a physically separate *device* that operates as a coprocessor to the *host* running the C program. This is the case, for example, when the kernels execute on a GPU and the rest of the C program executes on a CPU.

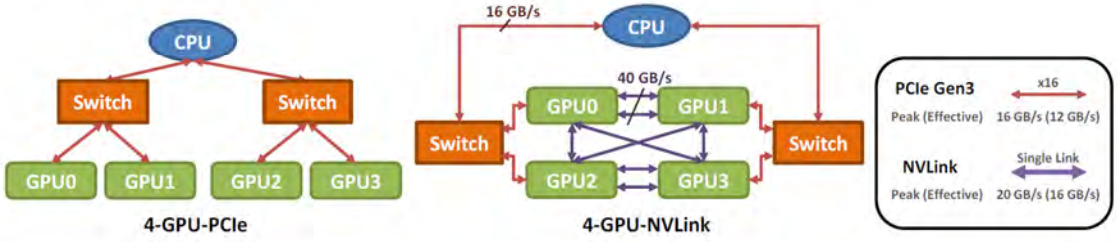


Serial code executes on the host while parallel code executes on the device.

Figure 8 Heterogeneous Programming

NVidia CUDA C Programming Guide pp. 12-13 (v10.1 May 2019), https://docs.nvidia.com/pdf/CUDA_C_Programming_Guide.pdf.

a mechanism for the nodes to communicate results of mathematical expression evaluation with	Each Accused Server Product comprises a mechanism for the nodes to communicate results of mathematical expression evaluation with each other using a peer-to-peer architecture. For example, each Accused Server Product GPU Accelerator (node) uses the NVLink interconnect architecture to communicate results of mathematical expression evaluation with other GPU Accelerators using peer-to-peer communications.

each other using a peer-to-peer architecture;	The second scenario, shown in Figure 3, compares a pair of configurations with four GPUs each, referred to as *4-GPU-PCIe* and *4-GPU-NVLink*. 4-GPU-PCIe uses a PCIe tree topology, again mirroring a configuration used commonly today. 4-GPU-NVLink enhances this by providing point-to-point connections with either one or two NVLink connections per pair of GPUs, yielding 16 GB/s or 32 GB/s effective bandwidth per direction per pair, respectively.  Figure 3: Comparing 4-GPU topologies with NVLink and PCIe. In 4-GPU-NVLink, GPU0 and GPU1 have 40 GB/s peak bandwidth between them, as do GPU2 and GPU3. The other peer-to-peer connections have 20 GB/s peak bandwidth. NVIDIA NVLink High-Speed Interconnect: Application Performance p. 5, https://info.nvidianews.com/rs/nvidia/images/NVIDIA%20NVLink%20High-Speed%20Interconnect%20Application%20Performance%20Brief.pdf.
wherein the plurality of nodes comprises: a first node comprising a first hardware processor configured to access a first memory comprising program code for a user interface and program code for a first single-node kernel, the first single-node kernel configured to interpret user instructions and distribute calls to at least one of a plurality of other nodes for execution;	Each Accused Server Product comprises a plurality of nodes comprising a first node comprising a first hardware processor configured to access a first memory comprising program code for a user interface and program code for a first single-node kernel, the first single-node kernel configured to interpret user instructions and distribute calls to at least one of a plurality of other nodes for execution. For example, each Accused Server Product implements CUDA code through executing Kernels on its GPU Accelerators. The GPU Accelerators engage in peer-to-peer communication to efficiently solve mathematical expressions and complete all threads generated by the CUDA code. The multiple GPU Accelerators (nodes) include a first GPU Accelerator, a second GPU Accelerator and a third GPU Accelerator. The first GPU Accelerator (first node) includes a processor configured to access a first memory that has program code for a user interface and program code for a first CUDA kernel, the first CUDA kernel configured to interpret user instructions and distribute calls to at least one of a plurality of other GPU Accelerators for execution. The CUDA programming model also assumes that both the host and the device maintain their own separate memory spaces in DRAM, referred to as *host memory* and *device memory*, respectively. Therefore, a program manages the global, constant, and texture memory spaces visible to kernels through calls to the CUDA runtime (described in Programming Interface). This includes device memory allocation and deallocation as well as data transfer between host and device memory. 2.3. Memory Hierarchy CUDA threads may access data from multiple memory spaces during their execution as illustrated by Figure 7. Each thread has private local memory. Each thread block has shared memory visible to all threads of the block and with the same lifetime as the block. All threads have access to the same global memory. There are also two additional read-only memory spaces accessible by all threads: the constant and texture memory spaces. The global, constant, and texture memory spaces are optimized for different memory usages (see Device Memory Accesses). Texture

memory also offers different addressing modes, as well as data filtering, for some specific data formats (see Texture and Surface Memory).

The global, constant, and texture memory spaces are persistent across kernel launches by the same application.

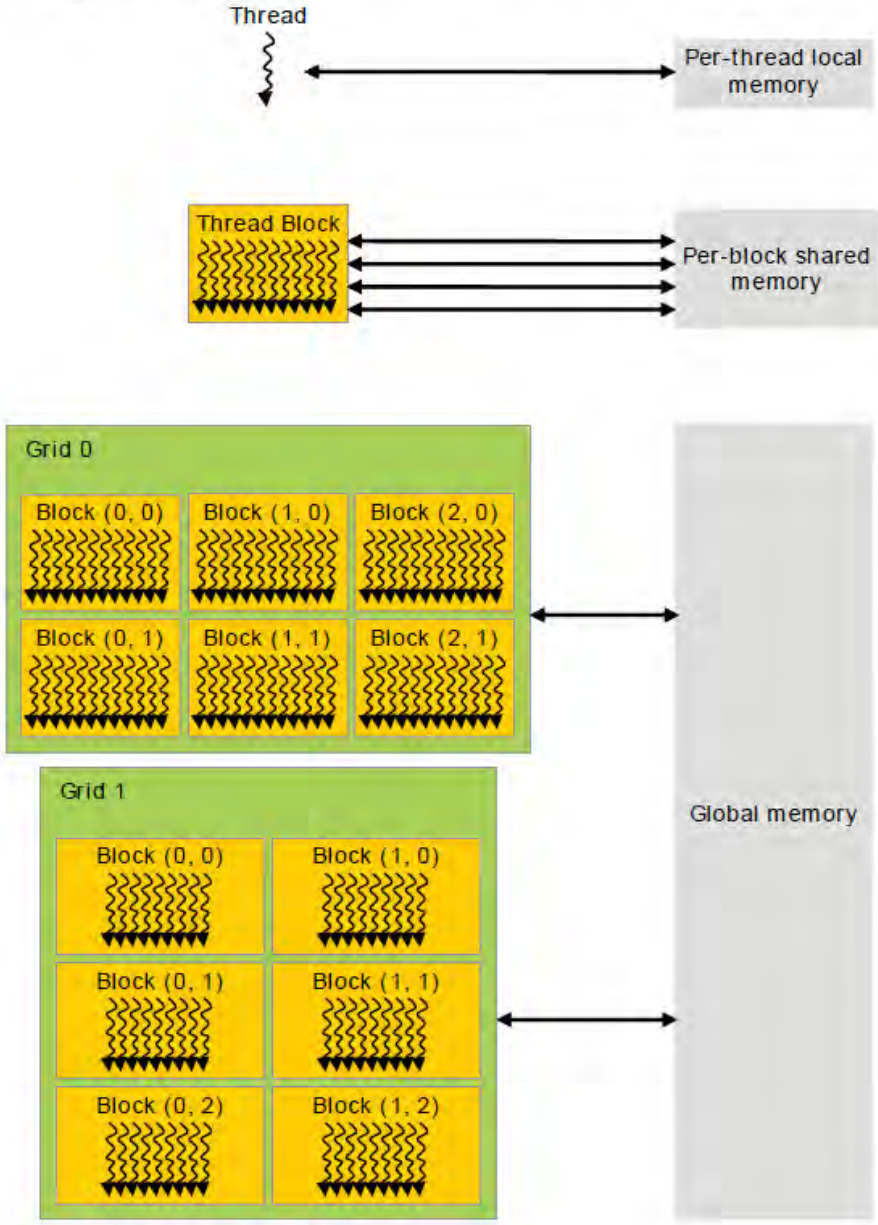


Figure 7 Memory Hierarchy

NVidia CUDA C Programming Guide pp. 7, 10-11 (v10.1 May 2019)
https://docs.nvidia.com/pdf/CUDA_C_Programming_Guide.pdf.

GP100 was built to be the highest performing parallel computing processor in the world to address the needs of the GPU accelerated computing markets serviced by our Tesla P100 accelerator platform. Like previous Tesla-class GPUs, GP100 is composed of an array of Graphics Processing Clusters (GPCs), Texture Processing Clusters (TPCs), Streaming Multiprocessors (SMs), and memory controllers. A full GP100 consists of six GPCs, 60 Pascal SMs, 30 TPCs (each including two SMs), and eight 512-bit memory controllers (4096 bits total).

Each GPC inside GP100 has ten SMs. Each SM has 64 CUDA Cores and four texture units. With 60 SMs, GP100 has a total of 3840 single precision CUDA Cores and 240 texture units. Each memory controller is attached to 512 KB of L2 cache, and each HBM2 DRAM stack is controlled by a pair of memory controllers. The full GPU includes a total of 4096 KB of L2 cache.

Figure 7 shows a full GP100 GPU with 60 SM units (different products can use different configurations of GP100). The Tesla P100 accelerator uses 56 SM units.

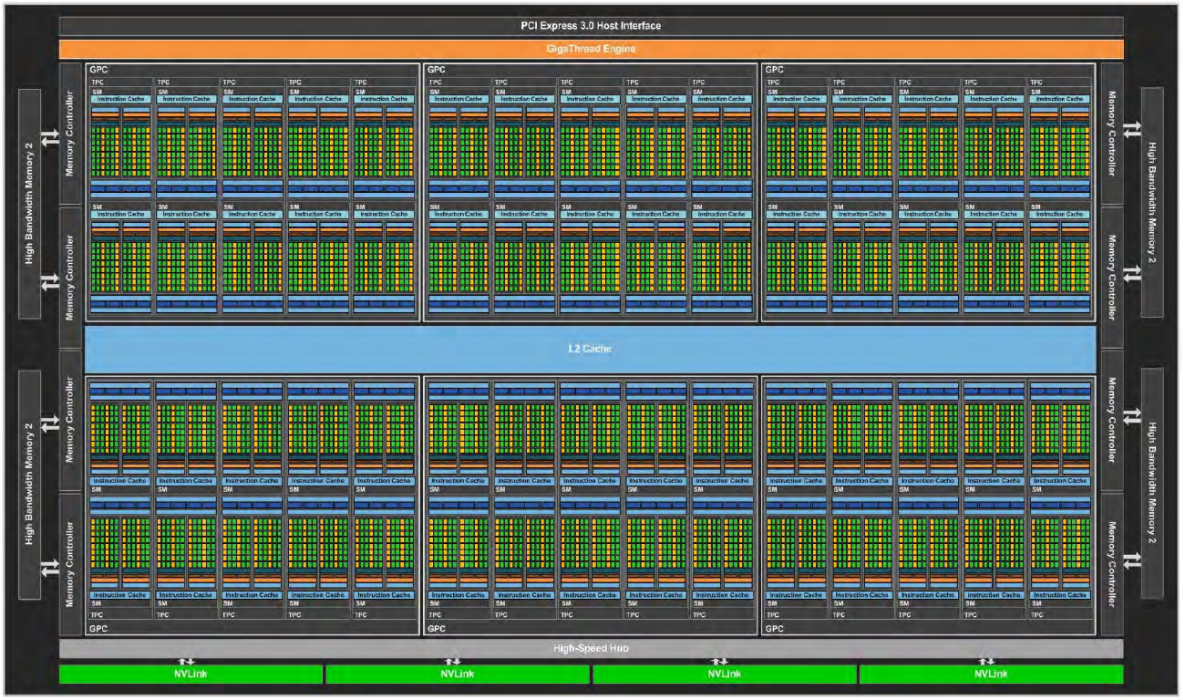


Figure 7. Pascal GP100 Full GPU with 60 SM Units

NVIDIA Tesla P100: The Most Advanced Datacenter Accelerator Ever Built Featuring Pascal GP100, the World's Fastest GPU p. 10, https://images.nvidia.com/content/pdf/tesla/whitepaper/pascal-architecture-whitepaper.pdf

Table 1. Comparison of NVIDIA Tesla GPUs

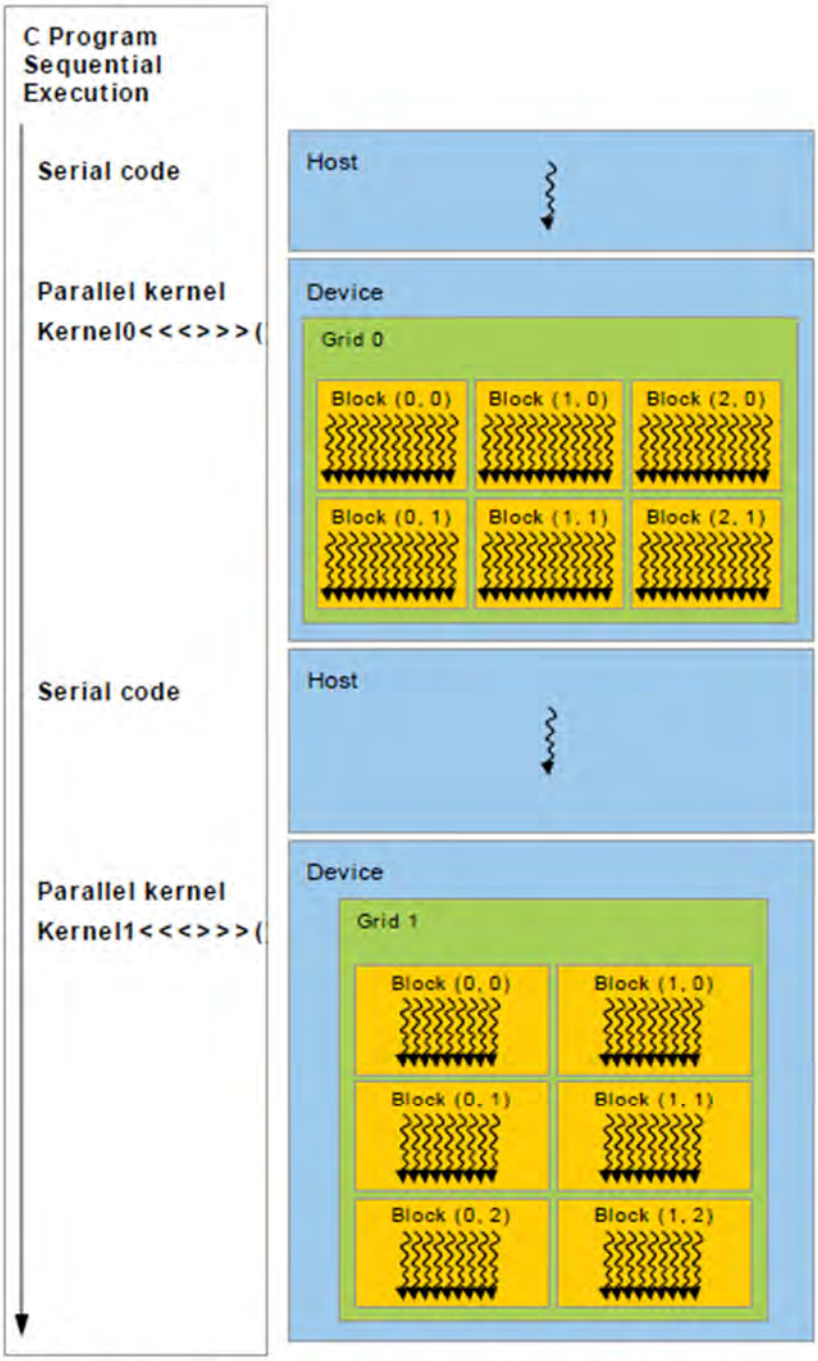
Tesla Product	Tesla K40	Tesla M40	Tesla P100	Tesla V100
GPU	GK180 (Kepler)	GM200 (Maxwell)	GP100 (Pascal)	GV100 (Volta)
SMs	15	24	56	80
TPCs	15	24	28	40
FP32 Cores / SM	192	128	64	64
FP32 Cores / GPU	2880	3072	3584	5120
FP64 Cores / SM	64	4	32	32
FP64 Cores / GPU	960	96	1792	2560
Tensor Cores / SM	NA	NA	NA	8
Tensor Cores / GPU	NA	NA	NA	640
GPU Boost Clock	810/875 MHz	1114 MHz	1480 MHz	1530 MHz
Peak FP32 TFLOPS[1]	5	6.8	10.6	15.7
Peak FP64 TFLOPS[1]	1.7	.21	5.3	7.8
Peak Tensor TFLOPS[1]	NA	NA	NA	125
Texture Units	240	192	224	320
Memory Interface	384-bit GDDR5	384-bit GDDR5	4096-bit HBM2	4096-bit HBM2
Memory Size	Up to 12 GB	Up to 24 GB	16 GB	16 GB
L2 Cache Size	1536 KB	3072 KB	4096 KB	6144 KB
Shared Memory Size / SM	16 KB/32 KB/48 KB	96 KB	64 KB	Configurable up to 96 KB
Register File Size / SM	256 KB	256 KB	256 KB	256KB
Register File Size / GPU	3840 KB	6144 KB	14336 KB	20480 KB
TDP	235 Watts	250 Watts	300 Watts	300 Watts
Transistors	7.1 billion	8 billion	15.3 billion	21.1 billion
GPU Die Size	551 mm²	601 mm²	610 mm²	815 mm²
Manufacturing Process	28 nm	28 nm	16 nm FinFET+	12 nm FFN

[1] Peak TFLOPS rates are based on GPU Boost Clock

NVidia Tesla V100 GPU Architecture: The World's Most Advanced Data Center GPU, p. 10, https://images.nvidia.com/content/volta-architecture/pdf/volta-architecture-whitepaper.pdf.

2.4. Heterogeneous Programming

As illustrated by Figure 8, the CUDA programming model assumes that the CUDA threads execute on a physically separate *device* that operates as a coprocessor to the *host* running the C program. This is the case, for example, when the kernels execute on a GPU and the rest of the C program executes on a CPU.



Serial code executes on the host while parallel code executes on the device.

Figure 8 Heterogeneous Programming

NVidia CUDA C Programming Guide pp. 12-13 (v10.1 May 2019), https://docs.nvidia.com/pdf/CUDA_C_Programming_Guide.pdf.

Server Configuration with NVLink

In the following sections of this paper, we analyze the performance benefit of NVLink for several algorithms and applications by comparing model systems based on PCIe-interconnected next-gen GPUs to otherwise-identical systems with NVLink-interconnected GPUs. GPUs are connected to the CPU using existing PCIe connections, but the NVLink configurations augment this with interconnections among the GPUs for peer-to-peer communication. The following analyses assume future-generation GPUs with performance higher than that of today's GPUs, so as to better correspond with the GPUs that will be contemporary with NVLink.

The second scenario, shown in Figure 3, compares a pair of configurations with four GPUs each, referred to as *4-GPU-PCIe* and *4-GPU-NVLink*. 4-GPU-PCIe uses a PCIe tree topology, again mirroring a configuration used commonly today. 4-GPU-NVLink enhances this by providing point-to-point connections with either one or two NVLink connections per pair of GPUs, yielding 16 GB/s or 32 GB/s effective bandwidth per direction per pair, respectively.

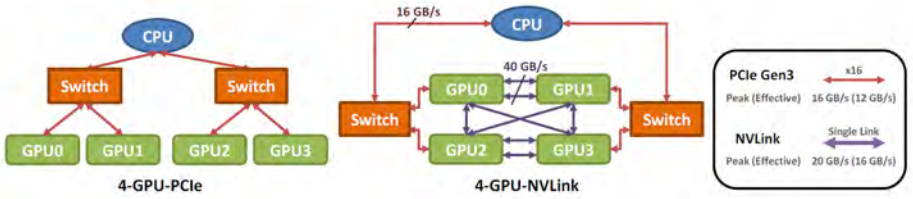


Figure 3: Comparing 4-GPU topologies with NVLink and PCIe. In 4-GPU-NVLink, GPU0 and GPU1 have 40 GB/s peak bandwidth between them, as do GPU2 and GPU3. The other peer-to-peer connections have 20 GB/s peak bandwidth.

The exchange algorithm begins with data evenly distributed across multiple GPUs as shown below.

Original Data

In this diagram, the color indicates which device the original data began on.

The first step is to hash the data to determine which GPU's bin each element should be placed in. For sorting, this might be as simple as extracting the highest order bits, but in theory we could apply any arbitrary hashing function to determine the keys (i.e., the destination bins). For this example, we will use the hash function `ceil(x/5)`, which produces the following keys given the data above.

Calculated Keys

Next we locally sort the data based on the hashed keys, yielding the following. Note that since we know these keys are in the range of `[0, NUM_GPUS)`, we can restrict the sorting to only a couple of bits.

Locally-sorted Data

Locally-sorted Keys

Next, we transfer all data with a key of 0 to device 0 and all data with a key of 1 to device 1. This becomes an all-to-all exchange among GPUs, producing the following.

Exchanged Data

Multi-GPU sorting is a simple extension of the above. After exchanging the data, we simply sort the data again locally to produce the final sorted list.

Fully-sorted Data

Here we can see the final list has been completely sorted. For many applications, the initial data exchange is all that is needed. Thus we will model both the exchange and the final sort separately.

Performance Considerations

The exchange algorithm can be easily pipelined by operating on small portions of the domain. The first step of the pipeline is to perform the local sort on the generated keys; the second is to perform the all-to-all exchange. This allows the initial sorting and the exchange to be completely overlapped. For the full multi-GPU sorting algorithm, the final local sort cannot begin until all data has arrived, so this portion cannot be pipelined with the other two.

If we assume the original data is randomly distributed among GPUs following a uniform random distribution, then each GPU must communicate `W*N/P` bytes of data to every other GPU, where W is the size of each element in bytes, N is the number of elements per GPU, and P is the number of GPUs.

In a PCIe tree topology, the communication is limited by the bisection bandwidth. Each GPU must communicate half of its data (N/2) across the top link in the system, and there are P/2 devices all trying to communicate across the same PCIe lanes in the same direction.

On an NVLink system, however, the communication can occur in parallel, because there are dedicated links between all pairs of GPUs.

NVIDIA NVLink High-Speed Interconnect: Application Performance p. 4-7, https://info.nvidianews.com/rs/nvidia/images/NVIDIA%20NVLink%20High-Speed%20Interconnect%20Application%20Performance%20Brief.pdf.

To address this issue, Tesla P100 features NVIDIA's new high-speed interface, NVLink, that provides GPU-to-GPU data transfers at up to 160 Gigabytes/second of bidirectional bandwidth—5x the bandwidth of PCIe Gen 3 x16. Figure 4 shows NVLink connecting eight Tesla P100 Accelerators in a Hybrid Cube Mesh Topology.

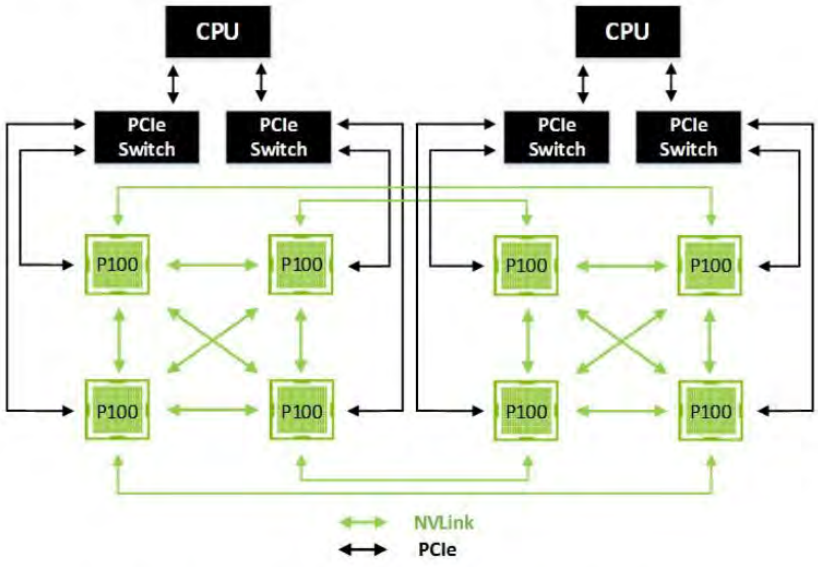
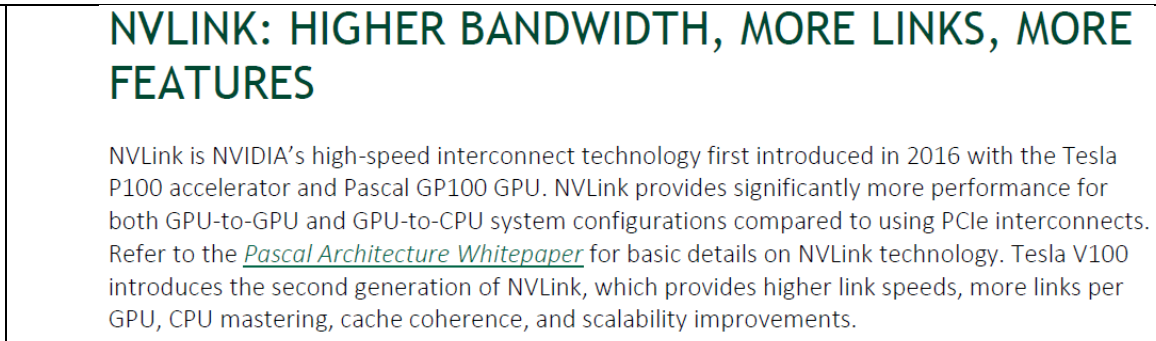
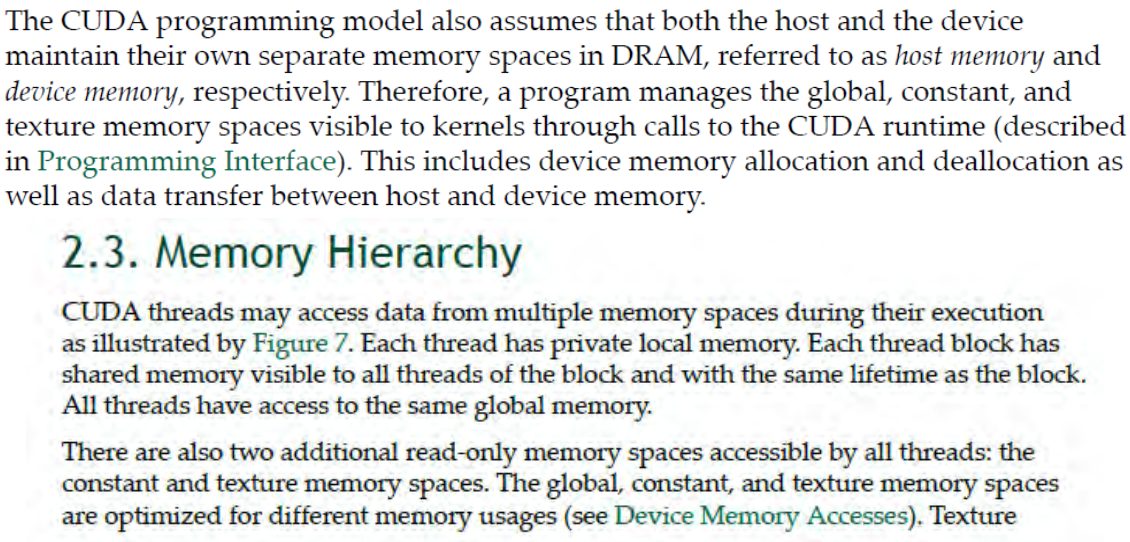


Figure 4. NVLink Connecting Eight Tesla P100 Accelerators in a Hybrid Cube Mesh Topology

NVIDIA Tesla P100: The Most Advanced Datacenter Accelerator Ever Built Featuring Pascal GP100, the World's Fastest GPU p. 7, https://images.nvidia.com/content/pdf/tesla/whitepaper/pascal-architecture-whitepaper.pdf.

	 NVidia Tesla V100 GPU Architecture: The World's Most Advanced Data Center GPU, p. 19, https://images.nvidia.com/content/volta-architecture/pdf/volta-architecture-whitepaper.pdf.
and a second node comprising a second hardware processor with a plurality of processing cores, wherein the second node is configured to receive calls from the first node, execute at least a first mathematical expression evaluation, and communicate a result of the first mathematical expression evaluation to a third node;	Each Accused Server Product comprises a plurality of nodes comprising a second node comprising a second hardware processor with a plurality of processing cores, wherein the second node is configured to receive calls from the first node, execute at least a first mathematical expression evaluation, and communicate a result of the first mathematical expression evaluation to a third node. For example, each Accused Server Product implements CUDA code through executing Kernels on its GPU Accelerators. The GPU Accelerators engage in peer-to-peer communication to efficiently solve mathematical expressions and complete all threads generated by the CUDA code. The second GPU Accelerator (second node) includes a processor with a plurality of CUDA processing cores. The second GPU Accelerator is configured to receive calls from the first GPU Accelerator, execute at least a first mathematical expression evaluation, and communicate a result of the first mathematical expression evaluation to a third GPU Accelerator. 

memory also offers different addressing modes, as well as data filtering, for some specific data formats (see Texture and Surface Memory).

The global, constant, and texture memory spaces are persistent across kernel launches by the same application.

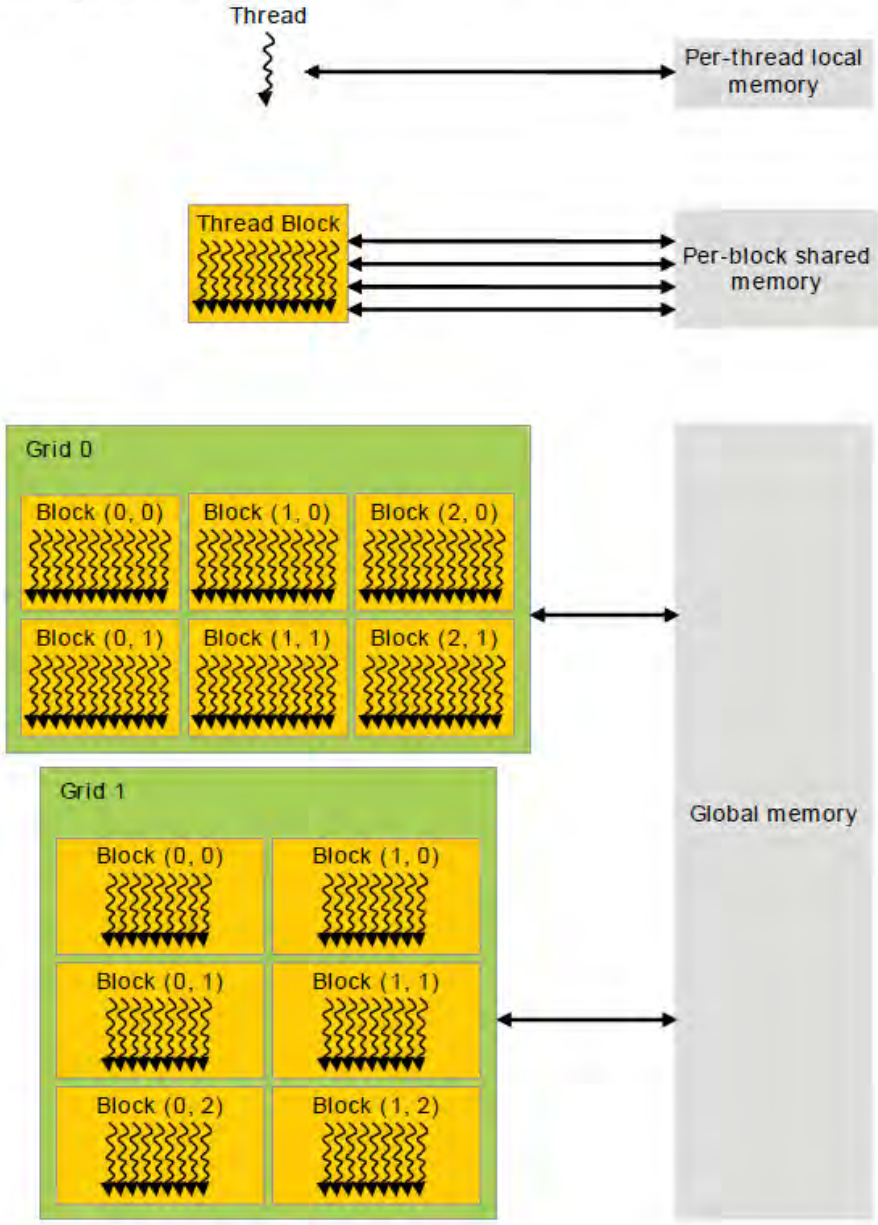


Figure 7 Memory Hierarchy

NVidia CUDA C Programming Guide pp. 7, 10-11 (v10.1 May 2019)
https://docs.nvidia.com/pdf/CUDA_C_Programming_Guide.pdf.

GP100 was built to be the highest performing parallel computing processor in the world to address the needs of the GPU accelerated computing markets serviced by our Tesla P100 accelerator platform. Like previous Tesla-class GPUs, GP100 is composed of an array of Graphics Processing Clusters (GPCs), Texture Processing Clusters (TPCs), Streaming Multiprocessors (SMs), and memory controllers. A full GP100 consists of six GPCs, 60 Pascal SMs, 30 TPCs (each including two SMs), and eight 512-bit memory controllers (4096 bits total).

Each GPC inside GP100 has ten SMs. Each SM has 64 CUDA Cores and four texture units. With 60 SMs, GP100 has a total of 3840 single precision CUDA Cores and 240 texture units. Each memory controller is attached to 512 KB of L2 cache, and each HBM2 DRAM stack is controlled by a pair of memory controllers. The full GPU includes a total of 4096 KB of L2 cache.

Figure 7 shows a full GP100 GPU with 60 SM units (different products can use different configurations of GP100). The Tesla P100 accelerator uses 56 SM units.

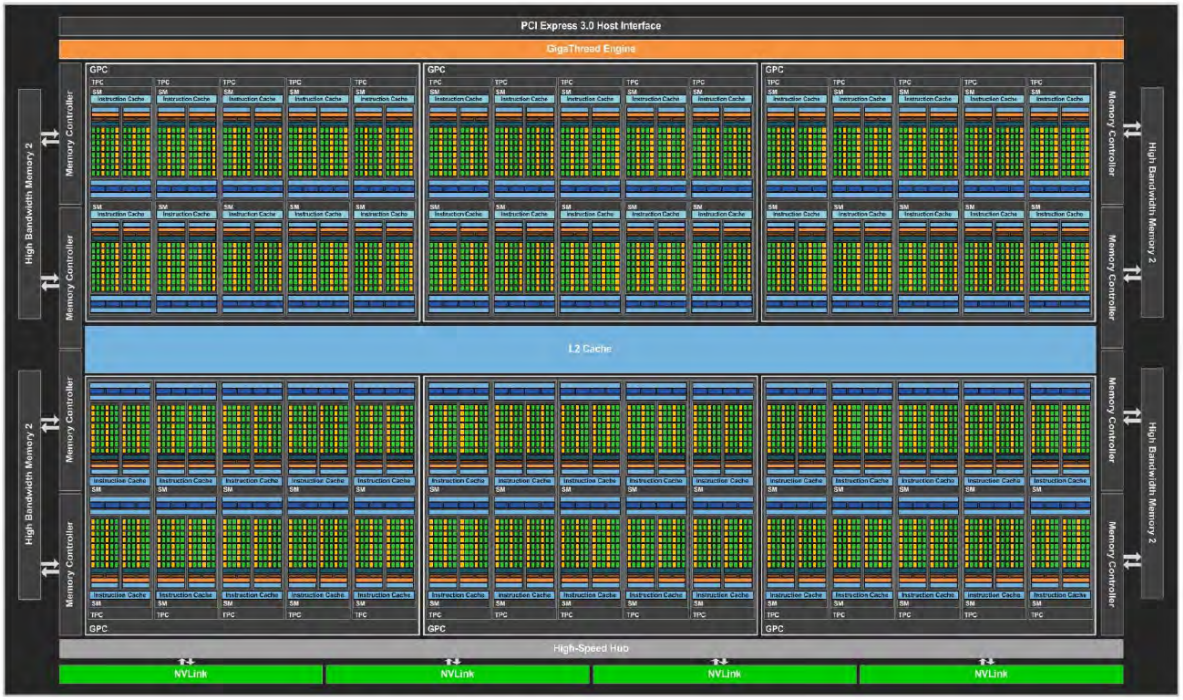


Figure 7. Pascal GP100 Full GPU with 60 SM Units

NVIDIA Tesla P100: The Most Advanced Datacenter Accelerator Ever Built Featuring Pascal GP100, the World's Fastest GPU p. 10, https://images.nvidia.com/content/pdf/tesla/whitepaper/pascal-architecture-whitepaper.pdf

Table 1. Comparison of NVIDIA Tesla GPUs

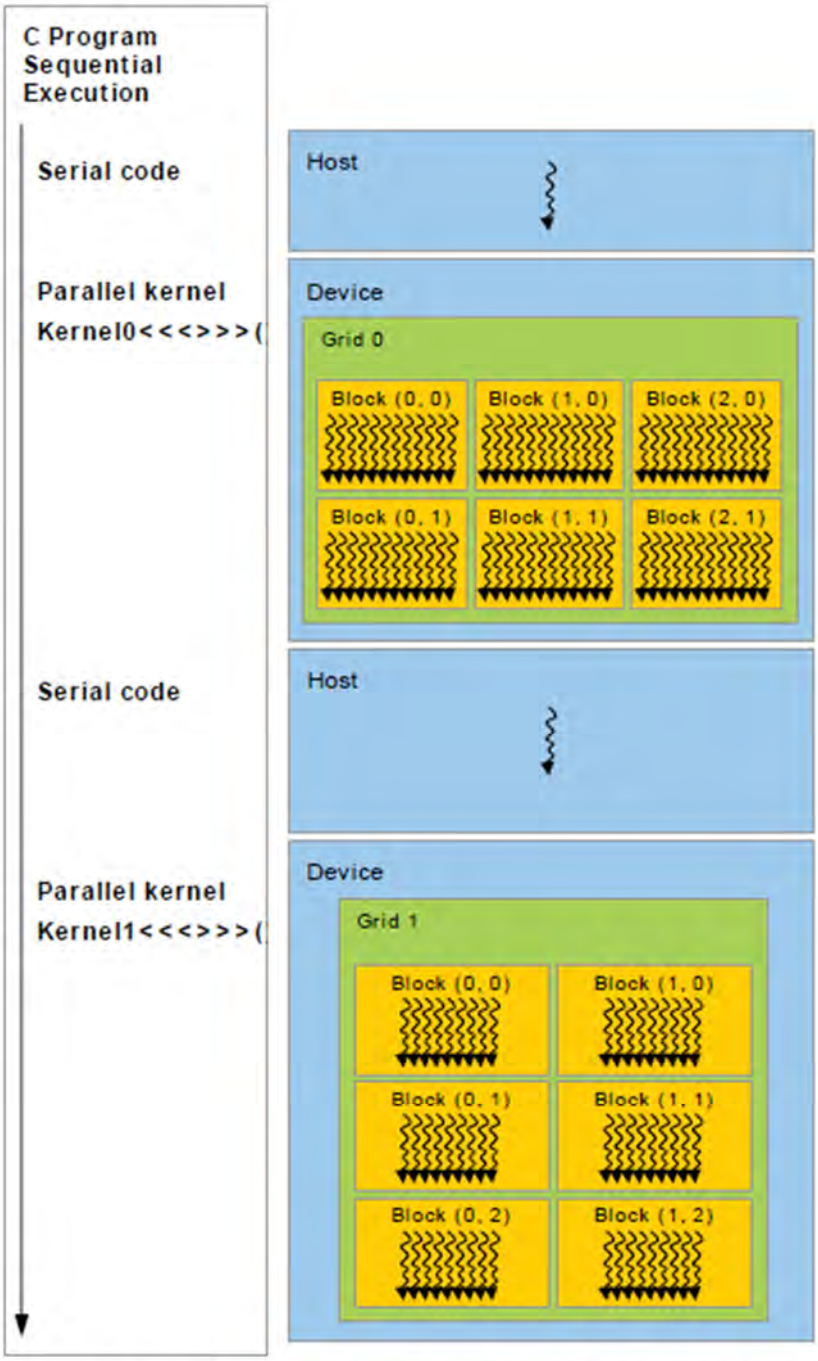
Tesla Product	Tesla K40	Tesla M40	Tesla P100	Tesla V100
GPU	GK180 (Kepler)	GM200 (Maxwell)	GP100 (Pascal)	GV100 (Volta)
SMs	15	24	56	80
TPCs	15	24	28	40
FP32 Cores / SM	192	128	64	64
FP32 Cores / GPU	2880	3072	3584	5120
FP64 Cores / SM	64	4	32	32
FP64 Cores / GPU	960	96	1792	2560
Tensor Cores / SM	NA	NA	NA	8
Tensor Cores / GPU	NA	NA	NA	640
GPU Boost Clock	810/875 MHz	1114 MHz	1480 MHz	1530 MHz
Peak FP32 TFLOPS[1]	5	6.8	10.6	15.7
Peak FP64 TFLOPS[1]	1.7	.21	5.3	7.8
Peak Tensor TFLOPS[1]	NA	NA	NA	125
Texture Units	240	192	224	320
Memory Interface	384-bit GDDR5	384-bit GDDR5	4096-bit HBM2	4096-bit HBM2
Memory Size	Up to 12 GB	Up to 24 GB	16 GB	16 GB
L2 Cache Size	1536 KB	3072 KB	4096 KB	6144 KB
Shared Memory Size / SM	16 KB/32 KB/48 KB	96 KB	64 KB	Configurable up to 96 KB
Register File Size / SM	256 KB	256 KB	256 KB	256KB
Register File Size / GPU	3840 KB	6144 KB	14336 KB	20480 KB
TDP	235 Watts	250 Watts	300 Watts	300 Watts
Transistors	7.1 billion	8 billion	15.3 billion	21.1 billion
GPU Die Size	551 mm²	601 mm²	610 mm²	815 mm²
Manufacturing Process	28 nm	28 nm	16 nm FinFET+	12 nm FFN

[1] Peak TFLOPS rates are based on GPU Boost Clock

NVidia Tesla V100 GPU Architecture: The World's Most Advanced Data Center GPU, p. 10, https://images.nvidia.com/content/volta-architecture/pdf/volta-architecture-whitepaper.pdf.

2.4. Heterogeneous Programming

As illustrated by Figure 8, the CUDA programming model assumes that the CUDA threads execute on a physically separate *device* that operates as a coprocessor to the *host* running the C program. This is the case, for example, when the kernels execute on a GPU and the rest of the C program executes on a CPU.



Serial code executes on the host while parallel code executes on the device.

Figure 8 Heterogeneous Programming

NVidia CUDA C Programming Guide pp. 12-13 (v10.1 May 2019), https://docs.nvidia.com/pdf/CUDA_C_Programming_Guide.pdf.

Server Configuration with NVLink

In the following sections of this paper, we analyze the performance benefit of NVLink for several algorithms and applications by comparing model systems based on PCIe-interconnected next-gen GPUs to otherwise-identical systems with NVLink-interconnected GPUs. GPUs are connected to the CPU using existing PCIe connections, but the NVLink configurations augment this with interconnections among the GPUs for peer-to-peer communication. The following analyses assume future-generation GPUs with performance higher than that of today's GPUs, so as to better correspond with the GPUs that will be contemporary with NVLink.

The second scenario, shown in Figure 3, compares a pair of configurations with four GPUs each, referred to as *4-GPU-PCIe* and *4-GPU-NVLink*. 4-GPU-PCIe uses a PCIe tree topology, again mirroring a configuration used commonly today. 4-GPU-NVLink enhances this by providing point-to-point connections with either one or two NVLink connections per pair of GPUs, yielding 16 GB/s or 32 GB/s effective bandwidth per direction per pair, respectively.

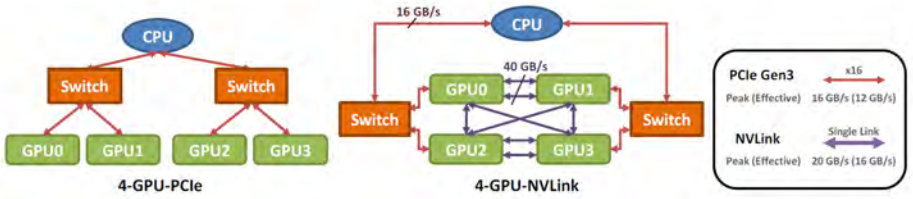


Figure 3: Comparing 4-GPU topologies with NVLink and PCIe. In 4-GPU-NVLink, GPU0 and GPU1 have 40 GB/s peak bandwidth between them, as do GPU2 and GPU3. The other peer-to-peer connections have 20 GB/s peak bandwidth.

The exchange algorithm begins with data evenly distributed across multiple GPUs as shown below.

Original Data

In this diagram, the color indicates which device the original data began on.

The first step is to hash the data to determine which GPU's bin each element should be placed in. For sorting, this might be as simple as extracting the highest order bits, but in theory we could apply any arbitrary hashing function to determine the keys (i.e., the destination bins). For this example, we will use the hash function `ceil(x/5)`, which produces the following keys given the data above.

Calculated Keys

Next we locally sort the data based on the hashed keys, yielding the following. Note that since we know these keys are in the range of `[0, NUM_GPUS)`, we can restrict the sorting to only a couple of bits.

Locally-sorted Data

Locally-sorted Keys

Next, we transfer all data with a key of 0 to device 0 and all data with a key of 1 to device 1. This becomes an all-to-all exchange among GPUs, producing the following.

Exchanged Data

Multi-GPU sorting is a simple extension of the above. After exchanging the data, we simply sort the data again locally to produce the final sorted list.

Fully-sorted Data

Here we can see the final list has been completely sorted. For many applications, the initial data exchange is all that is needed. Thus we will model both the exchange and the final sort separately.

Performance Considerations

The exchange algorithm can be easily pipelined by operating on small portions of the domain. The first step of the pipeline is to perform the local sort on the generated keys; the second is to perform the all-to-all exchange. This allows the initial sorting and the exchange to be completely overlapped. For the full multi-GPU sorting algorithm, the final local sort cannot begin until all data has arrived, so this portion cannot be pipelined with the other two.

If we assume the original data is randomly distributed among GPUs following a uniform random distribution, then each GPU must communicate `W*N/P` bytes of data to every other GPU, where W is the size of each element in bytes, N is the number of elements per GPU, and P is the number of GPUs.

In a PCIe tree topology, the communication is limited by the bisection bandwidth. Each GPU must communicate half of its data (N/2) across the top link in the system, and there are P/2 devices all trying to communicate across the same PCIe lanes in the same direction.

On an NVLink system, however, the communication can occur in parallel, because there are dedicated links between all pairs of GPUs.

NVIDIA NVLink High-Speed Interconnect: Application Performance p. 4-7, https://info.nvidianews.com/rs/nvidia/images/NVIDIA%20NVLink%20High-Speed%20Interconnect%20Application%20Performance%20Brief.pdf.

To address this issue, Tesla P100 features NVIDIA's new high-speed interface, NVLink, that provides GPU-to-GPU data transfers at up to 160 Gigabytes/second of bidirectional bandwidth—5x the bandwidth of PCIe Gen 3 x16. Figure 4 shows NVLink connecting eight Tesla P100 Accelerators in a Hybrid Cube Mesh Topology.

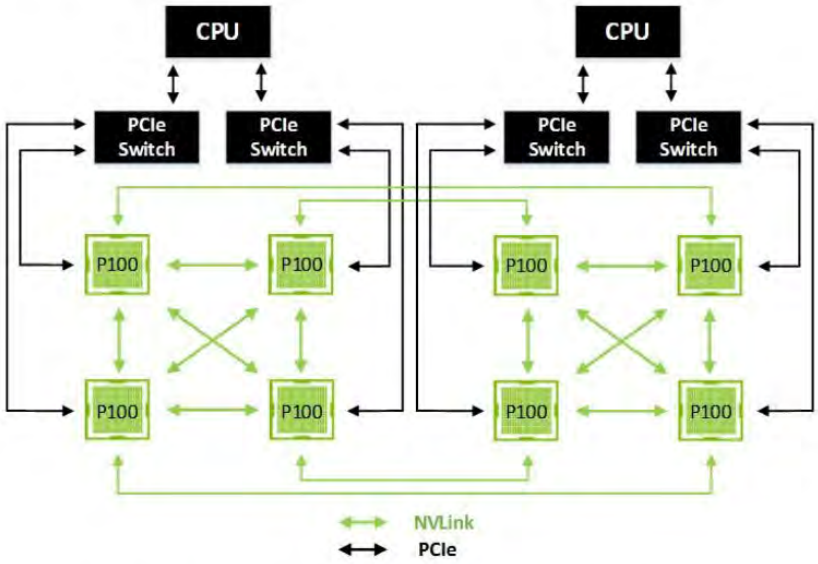
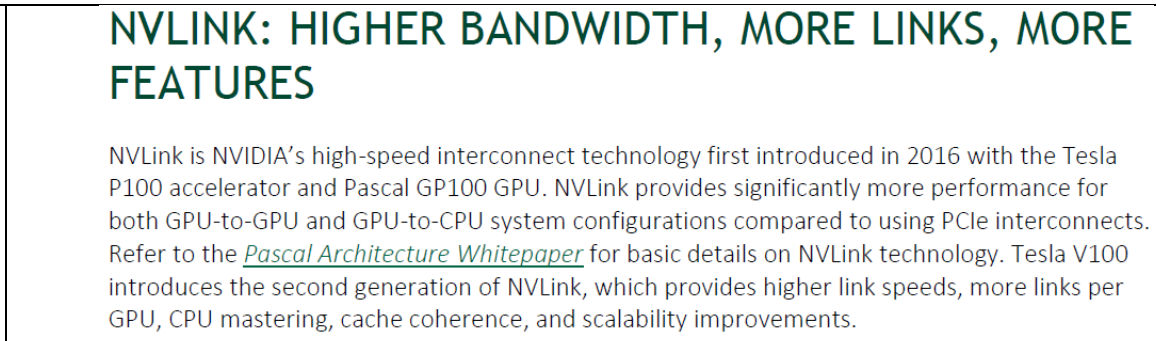
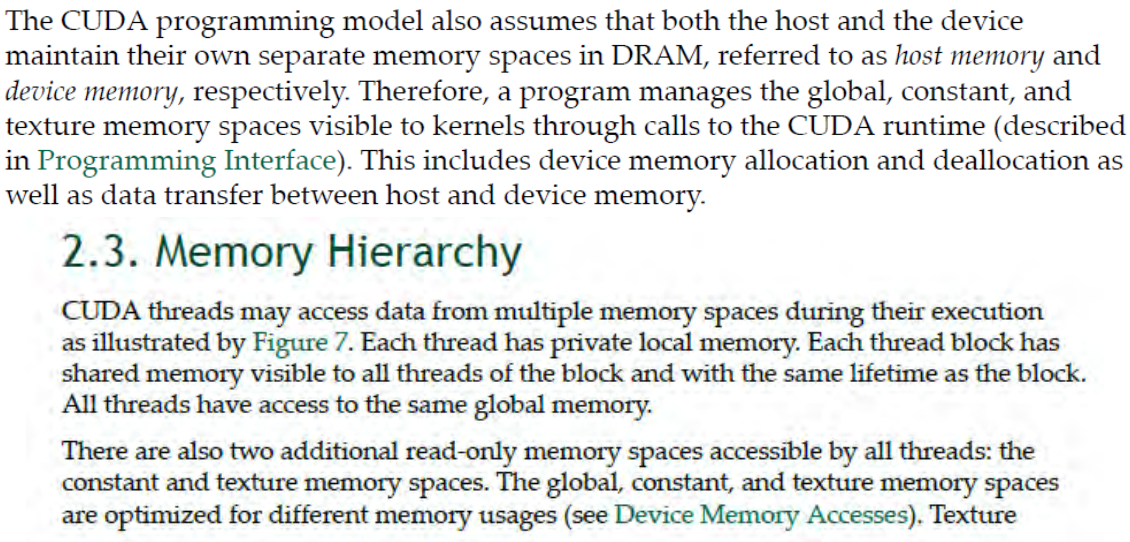


Figure 4. NVLink Connecting Eight Tesla P100 Accelerators in a Hybrid Cube Mesh Topology

NVIDIA Tesla P100: The Most Advanced Datacenter Accelerator Ever Built Featuring Pascal GP100, the World's Fastest GPU p. 7, https://images.nvidia.com/content/pdf/tesla/whitepaper/pascal-architecture-whitepaper.pdf.

	 NVidia Tesla V100 GPU Architecture: The World's Most Advanced Data Center GPU, p. 19, https://images.nvidia.com/content/volta-architecture/pdf/volta-architecture-whitepaper.pdf.
wherein the third node comprises a third hardware processor with a plurality of processing cores, wherein the third node is configured to receive the result of the first mathematical expression evaluation from the second node, execute at least a second mathematical expression evaluation using the received result, and communicate the result of the second mathematical expression evaluation to the first node;	Each Accused Server Product comprises a plurality of nodes wherein the third node comprises a third hardware processor with a plurality of processing cores, wherein the third node is configured to receive the result of the first mathematical expression evaluation from the second node, execute at least a second mathematical expression evaluation using the received result, and communicate the result of the second mathematical expression evaluation to the first node. For example, each Accused Server Product implements CUDA code through executing Kernels on its GPU Accelerators. The GPU Accelerators engage in peer-to-peer communication to efficiently solve mathematical expressions and complete all threads generated by the CUDA code. The third GPU Accelerator includes a processor with a plurality of CUDA processing cores. The third GPU Accelerator is configured receive the result of the first mathematical expression evaluation from the second GPU Accelerator, execute at least a second mathematical expression evaluation using the received result, and communicate the result of the second mathematical expression evaluation to the first GPU Accelerator. 

memory also offers different addressing modes, as well as data filtering, for some specific data formats (see Texture and Surface Memory).

The global, constant, and texture memory spaces are persistent across kernel launches by the same application.

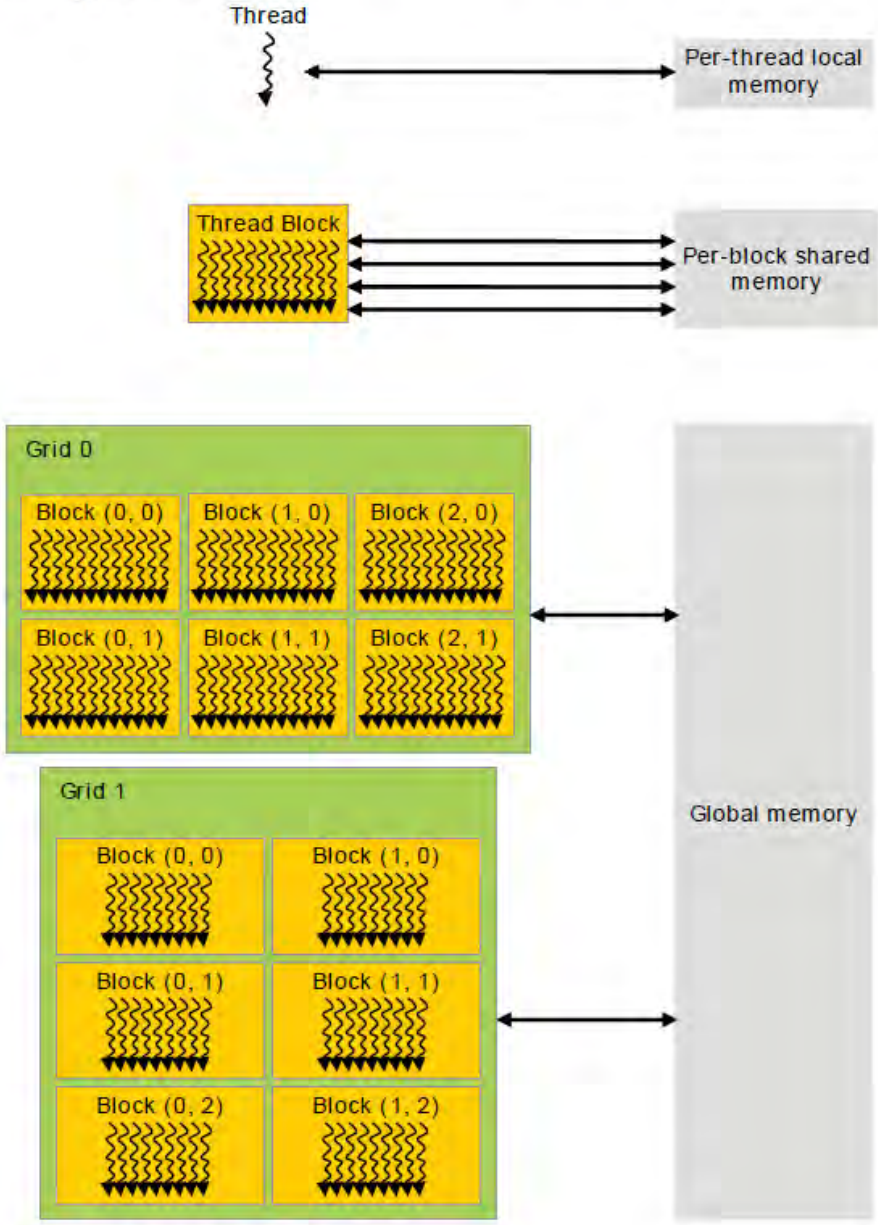


Figure 7 Memory Hierarchy

NVidia CUDA C Programming Guide pp. 7, 10-11 (v10.1 May 2019)
https://docs.nvidia.com/pdf/CUDA_C_Programming_Guide.pdf.

GP100 was built to be the highest performing parallel computing processor in the world to address the needs of the GPU accelerated computing markets serviced by our Tesla P100 accelerator platform. Like previous Tesla-class GPUs, GP100 is composed of an array of Graphics Processing Clusters (GPCs), Texture Processing Clusters (TPCs), Streaming Multiprocessors (SMs), and memory controllers. A full GP100 consists of six GPCs, 60 Pascal SMs, 30 TPCs (each including two SMs), and eight 512-bit memory controllers (4096 bits total).

Each GPC inside GP100 has ten SMs. Each SM has 64 CUDA Cores and four texture units. With 60 SMs, GP100 has a total of 3840 single precision CUDA Cores and 240 texture units. Each memory controller is attached to 512 KB of L2 cache, and each HBM2 DRAM stack is controlled by a pair of memory controllers. The full GPU includes a total of 4096 KB of L2 cache.

Figure 7 shows a full GP100 GPU with 60 SM units (different products can use different configurations of GP100). The Tesla P100 accelerator uses 56 SM units.

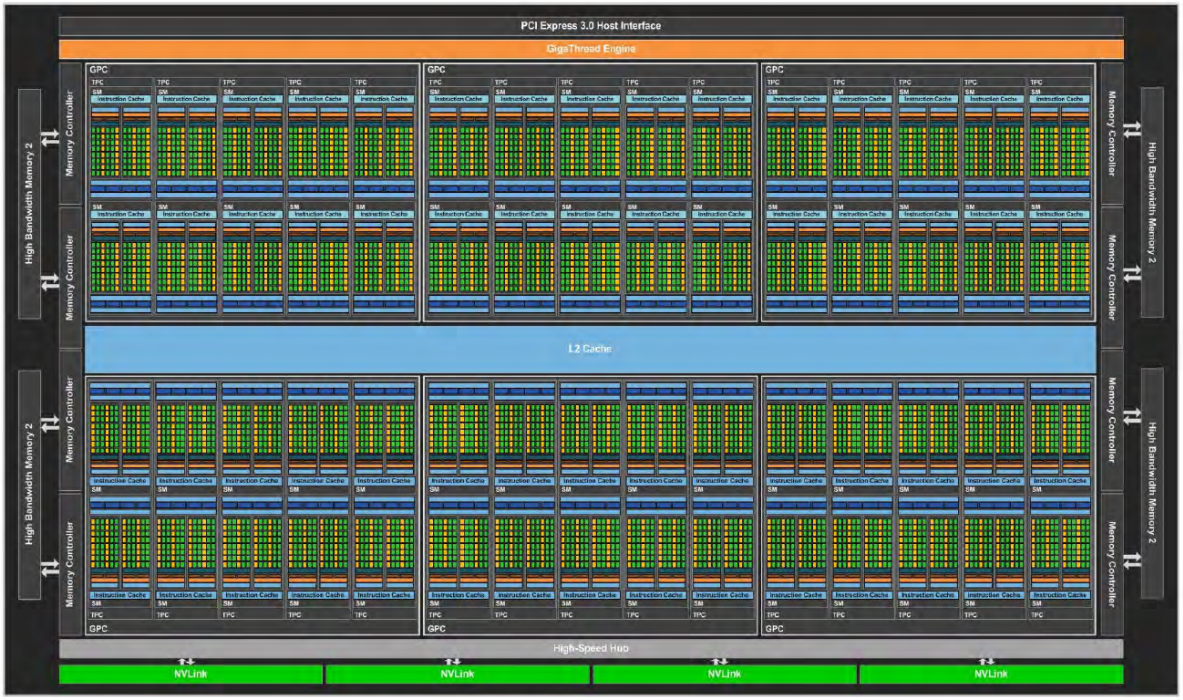


Figure 7. Pascal GP100 Full GPU with 60 SM Units

NVIDIA Tesla P100: The Most Advanced Datacenter Accelerator Ever Built Featuring Pascal GP100, the World's Fastest GPU p. 10, https://images.nvidia.com/content/pdf/tesla/whitepaper/pascal-architecture-whitepaper.pdf

Table 1. Comparison of NVIDIA Tesla GPUs

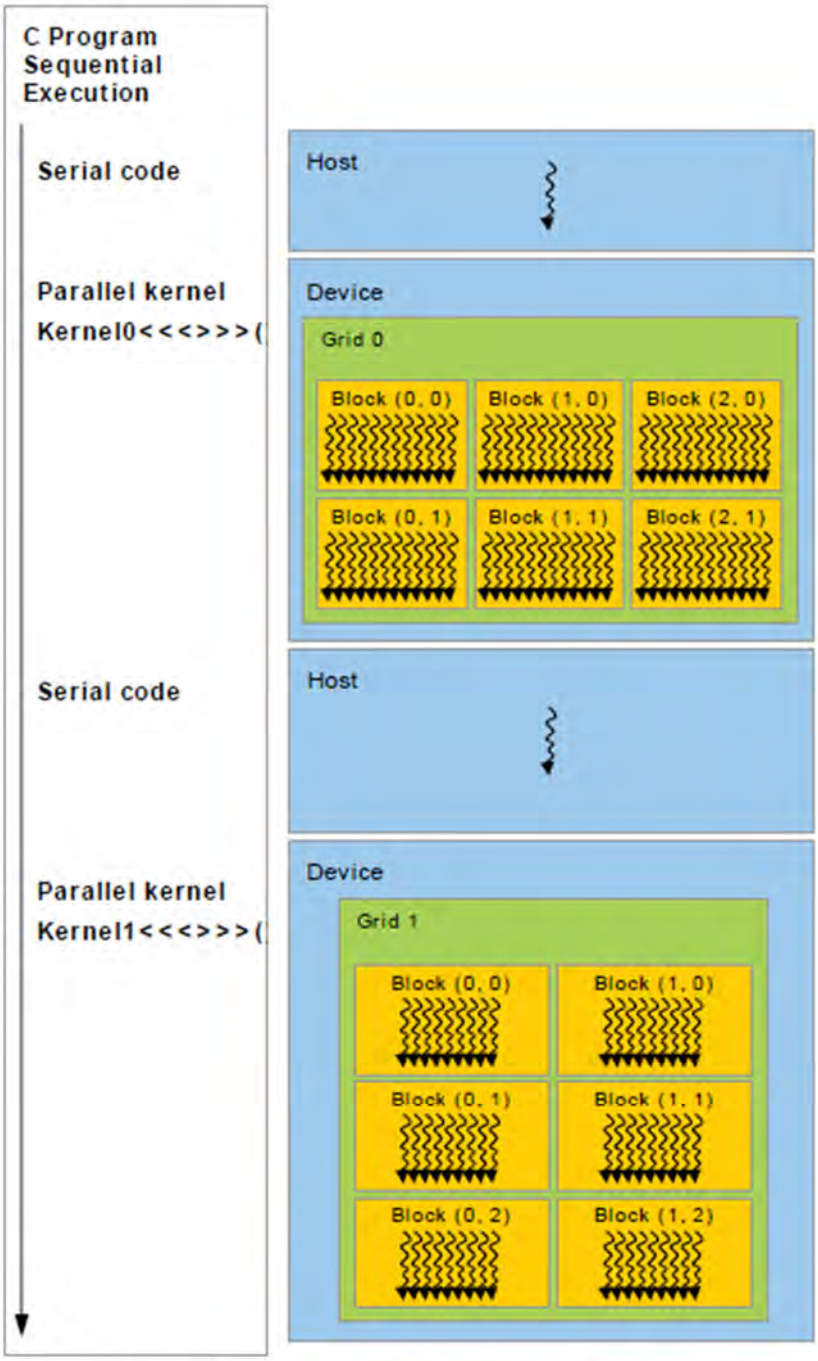
Tesla Product	Tesla K40	Tesla M40	Tesla P100	Tesla V100
GPU	GK180 (Kepler)	GM200 (Maxwell)	GP100 (Pascal)	GV100 (Volta)
SMs	15	24	56	80
TPCs	15	24	28	40
FP32 Cores / SM	192	128	64	64
FP32 Cores / GPU	2880	3072	3584	5120
FP64 Cores / SM	64	4	32	32
FP64 Cores / GPU	960	96	1792	2560
Tensor Cores / SM	NA	NA	NA	8
Tensor Cores / GPU	NA	NA	NA	640
GPU Boost Clock	810/875 MHz	1114 MHz	1480 MHz	1530 MHz
Peak FP32 TFLOPS[1]	5	6.8	10.6	15.7
Peak FP64 TFLOPS[1]	1.7	.21	5.3	7.8
Peak Tensor TFLOPS[1]	NA	NA	NA	125
Texture Units	240	192	224	320
Memory Interface	384-bit GDDR5	384-bit GDDR5	4096-bit HBM2	4096-bit HBM2
Memory Size	Up to 12 GB	Up to 24 GB	16 GB	16 GB
L2 Cache Size	1536 KB	3072 KB	4096 KB	6144 KB
Shared Memory Size / SM	16 KB/32 KB/48 KB	96 KB	64 KB	Configurable up to 96 KB
Register File Size / SM	256 KB	256 KB	256 KB	256KB
Register File Size / GPU	3840 KB	6144 KB	14336 KB	20480 KB
TDP	235 Watts	250 Watts	300 Watts	300 Watts
Transistors	7.1 billion	8 billion	15.3 billion	21.1 billion
GPU Die Size	551 mm²	601 mm²	610 mm²	815 mm²
Manufacturing Process	28 nm	28 nm	16 nm FinFET+	12 nm FFN

[1] Peak TFLOPS rates are based on GPU Boost Clock

NVidia Tesla V100 GPU Architecture: The World's Most Advanced Data Center GPU, p. 10, https://images.nvidia.com/content/volta-architecture/pdf/volta-architecture-whitepaper.pdf.

2.4. Heterogeneous Programming

As illustrated by Figure 8, the CUDA programming model assumes that the CUDA threads execute on a physically separate *device* that operates as a coprocessor to the *host* running the C program. This is the case, for example, when the kernels execute on a GPU and the rest of the C program executes on a CPU.



Serial code executes on the host while parallel code executes on the device.

Figure 8 Heterogeneous Programming

NVidia CUDA C Programming Guide pp. 12-13 (v10.1 May 2019), https://docs.nvidia.com/pdf/CUDA_C_Programming_Guide.pdf.

Server Configuration with NVLink

In the following sections of this paper, we analyze the performance benefit of NVLink for several algorithms and applications by comparing model systems based on PCIe-interconnected next-gen GPUs to otherwise-identical systems with NVLink-interconnected GPUs. GPUs are connected to the CPU using existing PCIe connections, but the NVLink configurations augment this with interconnections among the GPUs for peer-to-peer communication. The following analyses assume future-generation GPUs with performance higher than that of today's GPUs, so as to better correspond with the GPUs that will be contemporary with NVLink.

The second scenario, shown in Figure 3, compares a pair of configurations with four GPUs each, referred to as *4-GPU-PCIe* and *4-GPU-NVLink*. 4-GPU-PCIe uses a PCIe tree topology, again mirroring a configuration used commonly today. 4-GPU-NVLink enhances this by providing point-to-point connections with either one or two NVLink connections per pair of GPUs, yielding 16 GB/s or 32 GB/s effective bandwidth per direction per pair, respectively.

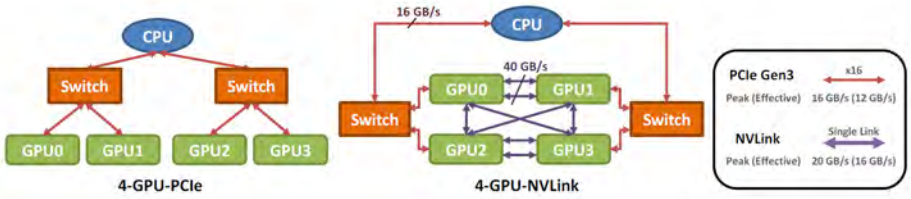


Figure 3: Comparing 4-GPU topologies with NVLink and PCIe. In 4-GPU-NVLink, GPU0 and GPU1 have 40 GB/s peak bandwidth between them, as do GPU2 and GPU3. The other peer-to-peer connections have 20 GB/s peak bandwidth.

The exchange algorithm begins with data evenly distributed across multiple GPUs as shown below.

Original Data

In this diagram, the color indicates which device the original data began on.

The first step is to hash the data to determine which GPU's bin each element should be placed in. For sorting, this might be as simple as extracting the highest order bits, but in theory we could apply any arbitrary hashing function to determine the keys (i.e., the destination bins). For this example, we will use the hash function `ceil(x/5)`, which produces the following keys given the data above.

Calculated Keys

Next we locally sort the data based on the hashed keys, yielding the following. Note that since we know these keys are in the range of `[0, NUM_GPUS)`, we can restrict the sorting to only a couple of bits.

Locally-sorted Data

Locally-sorted Keys

Next, we transfer all data with a key of 0 to device 0 and all data with a key of 1 to device 1. This becomes an all-to-all exchange among GPUs, producing the following.

Exchanged Data

Multi-GPU sorting is a simple extension of the above. After exchanging the data, we simply sort the data again locally to produce the final sorted list.

Fully-sorted Data

Here we can see the final list has been completely sorted. For many applications, the initial data exchange is all that is needed. Thus we will model both the exchange and the final sort separately.

Performance Considerations

The exchange algorithm can be easily pipelined by operating on small portions of the domain. The first step of the pipeline is to perform the local sort on the generated keys; the second is to perform the all-to-all exchange. This allows the initial sorting and the exchange to be completely overlapped. For the full multi-GPU sorting algorithm, the final local sort cannot begin until all data has arrived, so this portion cannot be pipelined with the other two.

If we assume the original data is randomly distributed among GPUs following a uniform random distribution, then each GPU must communicate `W*N/P` bytes of data to every other GPU, where W is the size of each element in bytes, N is the number of elements per GPU, and P is the number of GPUs.

In a PCIe tree topology, the communication is limited by the bisection bandwidth. Each GPU must communicate half of its data (N/2) across the top link in the system, and there are P/2 devices all trying to communicate across the same PCIe lanes in the same direction.

On an NVLink system, however, the communication can occur in parallel, because there are dedicated links between all pairs of GPUs.

NVIDIA NVLink High-Speed Interconnect: Application Performance p. 4-7, https://info.nvidianews.com/rs/nvidia/images/NVIDIA%20NVLink%20High-Speed%20Interconnect%20Application%20Performance%20Brief.pdf.

To address this issue, Tesla P100 features NVIDIA's new high-speed interface, NVLink, that provides GPU-to-GPU data transfers at up to 160 Gigabytes/second of bidirectional bandwidth—5x the bandwidth of PCIe Gen 3 x16. Figure 4 shows NVLink connecting eight Tesla P100 Accelerators in a Hybrid Cube Mesh Topology.

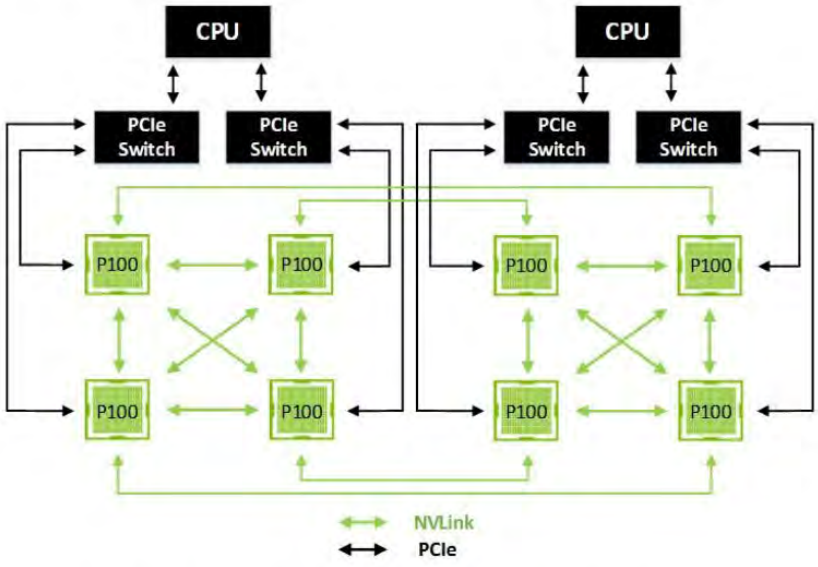


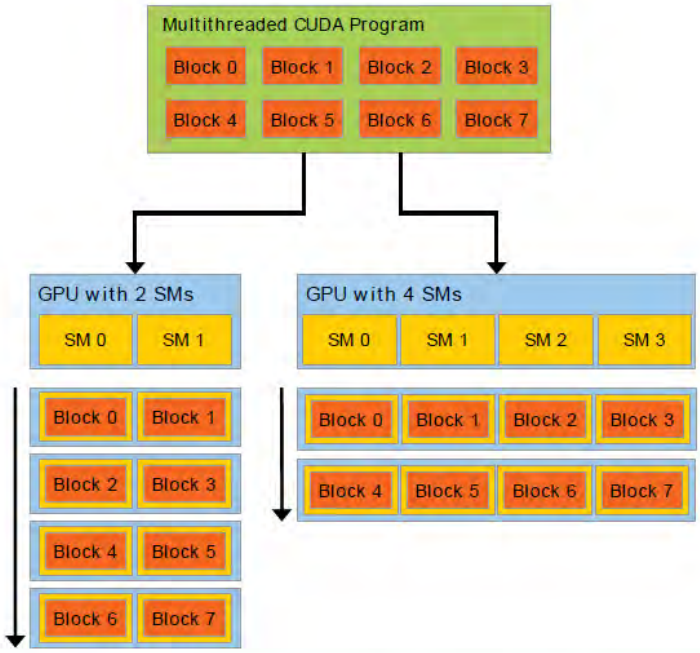
Figure 4. NVLink Connecting Eight Tesla P100 Accelerators in a Hybrid Cube Mesh Topology

NVIDIA Tesla P100: The Most Advanced Datacenter Accelerator Ever Built Featuring Pascal GP100, the World's Fastest GPU p. 7, https://images.nvidia.com/content/pdf/tesla/whitepaper/pascal-architecture-whitepaper.pdf.

	NVLINK: HIGHER BANDWIDTH, MORE LINKS, MORE FEATURES NVLink is NVIDIA's high-speed interconnect technology first introduced in 2016 with the Tesla P100 accelerator and Pascal GP100 GPU. NVLink provides significantly more performance for both GPU-to-GPU and GPU-to-CPU system configurations compared to using PCIe interconnects. Refer to the *Pascal Architecture Whitepaper* for basic details on NVLink technology. Tesla V100 introduces the second generation of NVLink, which provides higher link speeds, more links per GPU, CPU mastering, cache coherence, and scalability improvements. NVidia Tesla V100 GPU Architecture: The World's Most Advanced Data Center GPU, p. 19, https://images.nvidia.com/content/volta-architecture/pdf/volta-architecture-whitepaper.pdf.

wherein the first node is configured to return the result of the second mathematical expression evaluation to the user interface;	Each Accused Server Product comprises a first node configured to return the result of the second mathematical expression to the user interface.
	
	For example, the first GPU Accelerator is configured to return the result of the second mathematical expression evaluation to the CPU or host (user interface).
	
	This first example combines two numbers together on the GPU with a per-thread ID and returns the values in an array. Without managed memory, both host- and device-side storage for the return values is required (**host_ret** and **ret** in the example), as is an explicit copy between the two using **cudaMemcpy()**.
	
	Compare this with the Unified Memory version of the program, which allows direct access of GPU data from the host. Notice the **cudaMallocManaged()** routine, which returns a pointer valid from both host and device code. This allows **ret** to be used without a separate **host_ret** copy, greatly simplifying and reducing the size of the program.
	`__global__ void AplusB(int *ret, int a, int b) {`
	`  ret[threadIdx.x] = a + b + threadIdx.x;`
	`}`
	`int main() {`
	`  int *ret;`
	`  cudaMallocManaged(&ret, 1000 * sizeof(int));`
	`  AplusB<<< 1, 1000 >>>(ret, 10, 100);`
	`  cudaDeviceSynchronize();`
	`  for(int i = 0; i < 1000; i++)`
	`  printf("%d: A+B = %d\n", i, ret[i]);`
	`  cudaFree(ret);`
	`  return 0;`
	`}`
	
	It is worth a comment on the synchronization between host and device. Notice how in the non-managed example, the synchronous **cudaMemcpy()** routine is used both to synchronize the kernel (that is, to wait for it to finish running), and to transfer the data to the host. The Unified Memory examples do not call **cudaMemcpy()** and so require an explicit **cudaDeviceSynchronize()** before the host program can safely use the output from the GPU.
	
	NVIDIA CUDA C Programming Guide pp. 281-82 (v10.1 May 2019), https://docs.nvidia.com/pdf/CUDA_C_Programming_Guide.pdf

wherein one or more of the nodes are configured to: accept user instructions; after accepting user instructions,	Each Accused Server Product comprises one or more of the nodes configured to: accept user instructions; after accepting user instructions, communicate at least some of the user instructions using the mechanism for the nodes to communicate with each other; and after communicating at least some of the user instructions using the mechanism, communicate at least some of the user instructions to one or more single-node kernels. For example, one or more of the GPU Accelerators are configured to: accept user instructions; after accepting user instructions, communicate at least some of the user instructions using the NVLink interconnect architecture; and after communicating at least some of the user

communicate at least some of the user instructions using the mechanism for the nodes to communicate with each other; and after communicating at least some of the user instructions using the mechanism, communicate at least some of the user instructions to one or more single-node kernels.	instructions using NVLink, communicate at least some of the user instructions to one or more CUDA kernels. ## 2.4. Heterogeneous Programming As illustrated by Figure 8, the CUDA programming model assumes that the CUDA threads execute on a physically separate *device* that operates as a coprocessor to the *host* running the C program. This is the case, for example, when the kernels execute on a GPU and the rest of the C program executes on a CPU.  A GPU is built around an array of Streaming Multiprocessors (SMs) (see Hardware Implementation for more details). A multithreaded program is partitioned into blocks of threads that execute independently from each other, so that a GPU with more multiprocessors will automatically execute the program in less time than a GPU with fewer multiprocessors. Figure 5 Automatic Scalability NVidia CUDA C Programming Guide pp. 5, 12 (v10.1 May 2019), https://docs.nvidia.com/pdf/CUDA_C_Programming_Guide.pdf. **Server Configuration with NVLink** In the following sections of this paper, we analyze the performance benefit of NVLink for several algorithms and applications by comparing model systems based on PCIe-interconnected next-gen GPUs to otherwise-identical systems with NVLink-interconnected GPUs. GPUs are connected to the CPU using existing PCIe connections, but the NVLink configurations augment this with interconnections among the GPUs for peer-to-peer communication. The following analyses assume future-generation GPUs with performance higher than that of today's GPUs, so as to better correspond with the GPUs that will be contemporary with NVLink.

The second scenario, shown in Figure 3, compares a pair of configurations with four GPUs each, referred to as *4-GPU-PCIe* and *4-GPU-NVLink*. 4-GPU-PCIe uses a PCIe tree topology, again mirroring a configuration used commonly today. 4-GPU-NVLink enhances this by providing point-to-point connections with either one or two NVLink connections per pair of GPUs, yielding 16 GB/s or 32 GB/s effective bandwidth per direction per pair, respectively.

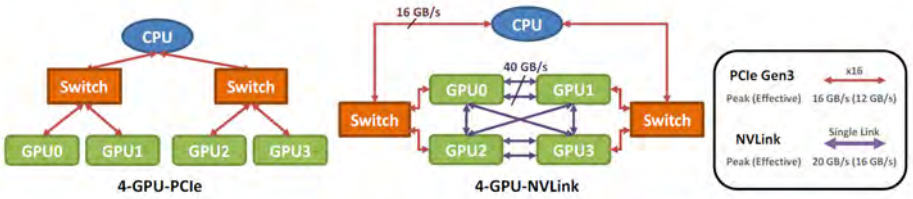


Figure 3: Comparing 4-GPU topologies with NVLink and PCIe. In 4-GPU-NVLink, GPU0 and GPU1 have 40 GB/s peak bandwidth between them, as do GPU2 and GPU3. The other peer-to-peer connections have 20 GB/s peak bandwidth.

The exchange algorithm begins with data evenly distributed across multiple GPUs as shown below.

Original Data

8	3	7	4	6	1	9	2	7	5	3	8	2	0	1	9

In this diagram, the color indicates which device the original data began on.

The first step is to hash the data to determine which GPU's bin each element should be placed in. For sorting, this might be as simple as extracting the highest order bits, but in theory we could apply any arbitrary hashing function to determine the keys (i.e., the destination bins). For this example, we will use the hash function `ceil(x/5)`, which produces the following keys given the data above.

Calculated Keys

1	0	1	0	1	0	1	0	1	1	0	1	0	0	0	1

Next we locally sort the data based on the hashed keys, yielding the following. Note that since we know these keys are in the range of `[0, NUM_GPUS)`, we can restrict the sorting to only a couple of bits.

Locally-sorted Data

3	4	1	2	8	7	6	9	3	2	0	1	7	5	8	9

Locally-sorted Keys

0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1

Next, we transfer all data with a key of 0 to device 0 and all data with a key of 1 to device 1. This becomes an all-to-all exchange among GPUs, producing the following.

Exchanged Data

3	4	1	2	3	2	0	1	8	7	6	9	7	5	8	9

Multi-GPU sorting is a simple extension of the above. After exchanging the data, we simply sort the data again locally to produce the final sorted list.

Fully-sorted Data

0	1	1	2	2	3	3	4	5	6	7	7	8	8	9	9

Here we can see the final list has been completely sorted. For many applications, the initial data exchange is all that is needed. Thus we will model both the exchange and the final sort separately.

Performance Considerations

The exchange algorithm can be easily pipelined by operating on small portions of the domain. The first step of the pipeline is to perform the local sort on the generated keys; the second is to perform the all-to-all exchange. This allows the initial sorting and the exchange to be completely overlapped. For the full multi-GPU sorting algorithm, the final local sort cannot begin until all data has arrived, so this portion cannot be pipelined with the other two.

If we assume the original data is randomly distributed among GPUs following a uniform random distribution, then each GPU must communicate `W*N/P` bytes of data to every other GPU, where W is the size of each element in bytes, N is the number of elements per GPU, and P is the number of GPUs.

In a PCIe tree topology, the communication is limited by the bisection bandwidth. Each GPU must communicate half of its data (N/2) across the top link in the system, and there are P/2 devices all trying to communicate across the same PCIe lanes in the same direction.

On an NVLink system, however, the communication can occur in parallel, because there are dedicated links between all pairs of GPUs.

NVIDIA NVLink High-Speed Interconnect: Application Performance p. 4-7, https://info.nvidianews.com/rs/nvidia/images/NVIDIA%20NVLink%20High-Speed%20Interconnect%20Application%20Performance%20Brief.pdf.

To address this issue, Tesla P100 features NVIDIA's new high-speed interface, NVLink, that provides GPU-to-GPU data transfers at up to 160 Gigabytes/second of bidirectional bandwidth—5x the bandwidth of PCIe Gen 3 x16. Figure 4 shows NVLink connecting eight Tesla P100 Accelerators in a Hybrid Cube Mesh Topology.

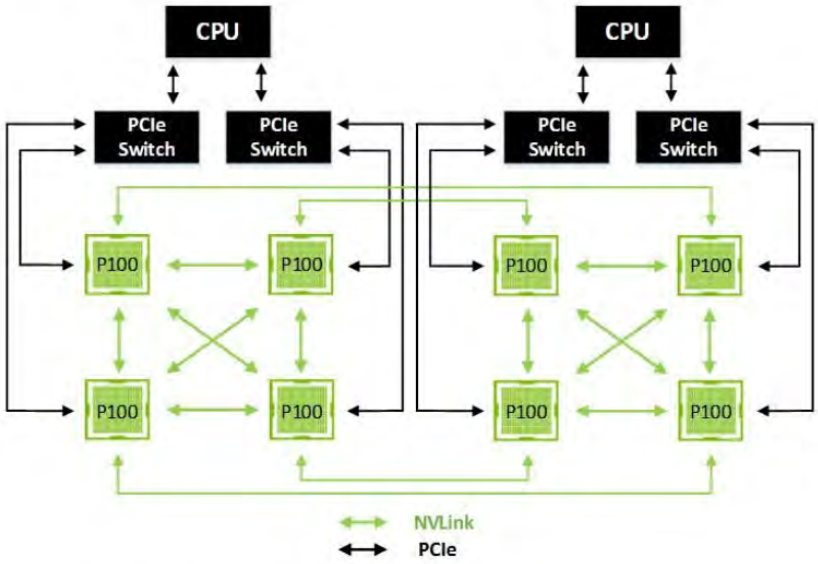


Figure 4. NVLink Connecting Eight Tesla P100 Accelerators in a Hybrid Cube Mesh Topology

NVIDIA Tesla P100: The Most Advanced Datacenter Accelerator Ever Built Featuring Pascal GP100, the World's Fastest GPU p. 7, https://images.nvidia.com/content/pdf/tesla/whitepaper/pascal-architecture-whitepaper.pdf.

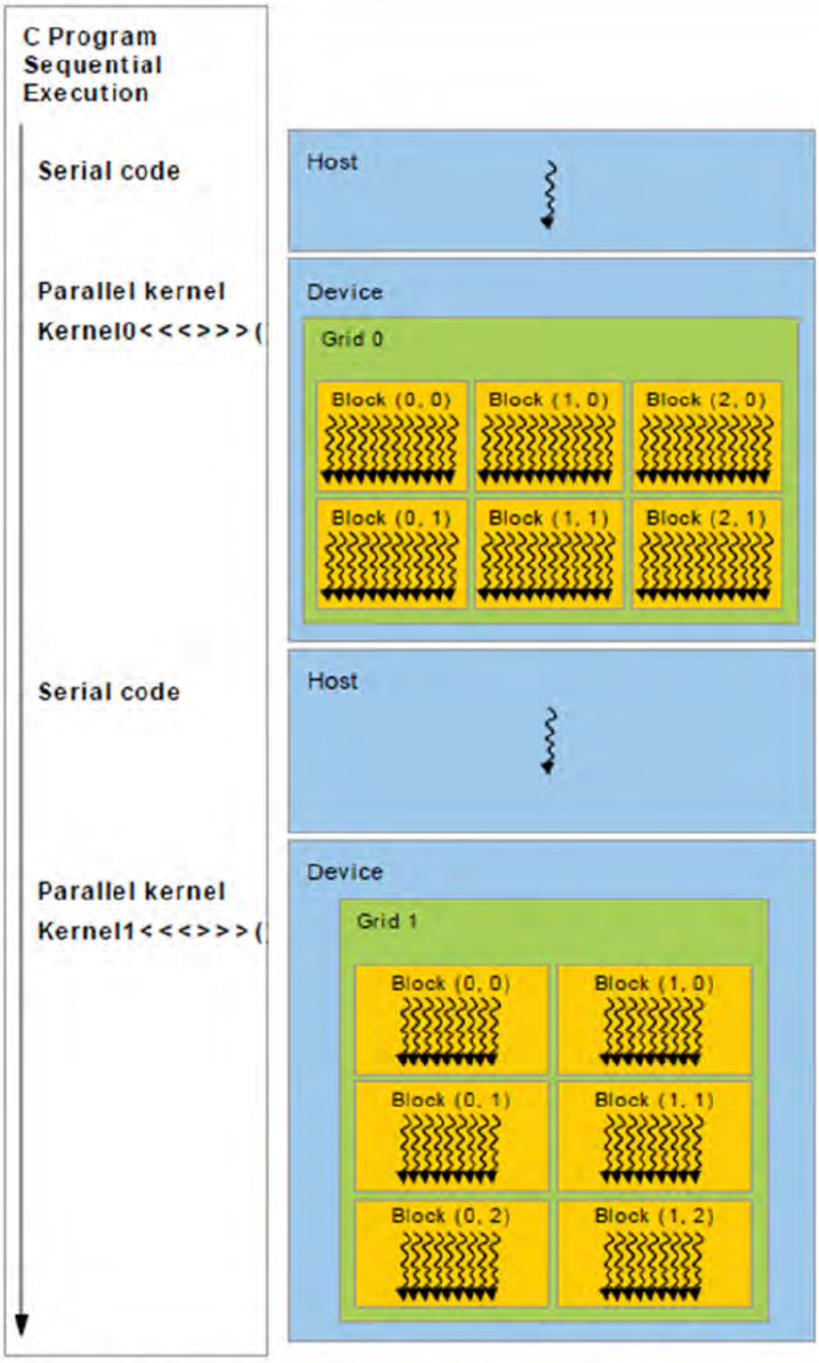
NVLINK: HIGHER BANDWIDTH, MORE LINKS, MORE FEATURES

NVLink is NVIDIA's high-speed interconnect technology first introduced in 2016 with the Tesla P100 accelerator and Pascal GP100 GPU. NVLink provides significantly more performance for both GPU-to-GPU and GPU-to-CPU system configurations compared to using PCIe interconnects. Refer to the *Pascal Architecture Whitepaper* for basic details on NVLink technology. Tesla V100 introduces the second generation of NVLink, which provides higher link speeds, more links per GPU, CPU mastering, cache coherence, and scalability improvements.

NVidia Tesla V100 GPU Architecture: The World's Most Advanced Data Center GPU, p. 19, https://images.nvidia.com/content/volta-architecture/pdf/volta-architecture-whitepaper.pdf.

2.4. Heterogeneous Programming

As illustrated by Figure 8, the CUDA programming model assumes that the CUDA threads execute on a physically separate *device* that operates as a coprocessor to the *host* running the C program. This is the case, for example, when the kernels execute on a GPU and the rest of the C program executes on a CPU.

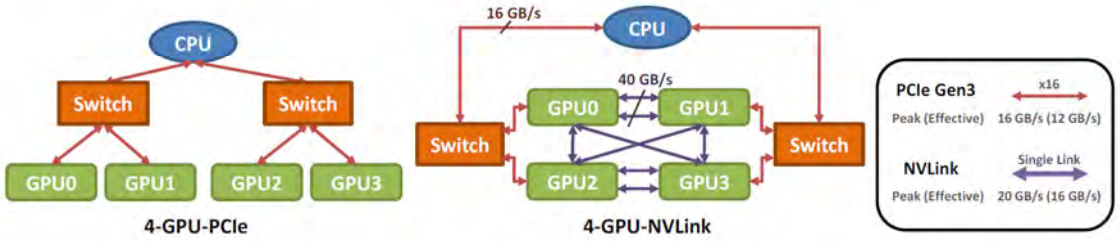


Serial code executes on the host while parallel code executes on the device.

Figure 8 Heterogeneous Programming

NVidia CUDA C Programming Guide pp. 12-13 (v10.1 May 2019), https://docs.nvidia.com/pdf/CUDA_C_Programming_Guide.pdf.

Claim 35 Language	NVLink Application
A computer cluster node for evaluating expressions in parallel with	Each Accused Server Product includes an Accused GPU Product. Each Accused GPU Product is a computer cluster node for evaluating expressions in parallel with other computer cluster nodes. For example, each GPU Accelerator is capable of evaluating expressions in parallel with other

other computer cluster nodes, the computer cluster node comprising:	GPU Accelerators. The second scenario, shown in Figure 3, compares a pair of configurations with four GPUs each, referred to as *4-GPU-PCIe* and *4-GPU-NVLink*. 4-GPU-PCIe uses a PCIe tree topology, again mirroring a configuration used commonly today. 4-GPU-NVLink enhances this by providing point-to-point connections with either one or two NVLink connections per pair of GPUs, yielding 16 GB/s or 32 GB/s effective bandwidth per direction per pair, respectively.  Figure 3: Comparing 4-GPU topologies with NVLink and PCIe. In 4-GPU-NVLink, GPU0 and GPU1 have 40 GB/s peak bandwidth between them, as do GPU2 and GPU3. The other peer-to-peer connections have 20 GB/s peak bandwidth. NVIDIA NVLink High-Speed Interconnect: Application Performance p. 5, https://info.nvidianews.com/rs/nvidia/images/NVIDIA%20NVLink%20High-Speed%20Interconnect%20Application%20Performance%20Brief.pdf.
a hardware processor configured to access one or more non-transitory memory devices comprising program code for a single-node kernel that, when executed, causes the hardware processor to interpret user instructions, to evaluate mathematical expressions, and to produce results of mathematical expression evaluation, wherein the hardware processor comprises	Each Accused GPU Product and Accused Server Product comprises a hardware processor configured to access one or more non-transitory memory devices comprising program code for a single-node kernel that, when executed, causes the hardware processor to interpret user instructions, to evaluate mathematical expressions, and to produce results of mathematical expression evaluation, wherein the hardware processor comprises multiple processor cores. For example, each GPU Accelerator has a processor configured to access memory containing a CUDA kernel that, when executed, causes the processor to interpret user instructions, to evaluate mathematical expressions, and to produce results of mathematical expression evaluation, wherein the processor comprises multiple processor cores. The CUDA programming model also assumes that both the host and the device maintain their own separate memory spaces in DRAM, referred to as *host memory* and *device memory*, respectively. Therefore, a program manages the global, constant, and texture memory spaces visible to kernels through calls to the CUDA runtime (described in Programming Interface). This includes device memory allocation and deallocation as well as data transfer between host and device memory. **2.3. Memory Hierarchy** CUDA threads may access data from multiple memory spaces during their execution as illustrated by Figure 7. Each thread has private local memory. Each thread block has shared memory visible to all threads of the block and with the same lifetime as the block. All threads have access to the same global memory. There are also two additional read-only memory spaces accessible by all threads: the constant and texture memory spaces. The global, constant, and texture memory spaces are optimized for different memory usages (see Device Memory Accesses). Texture

multiple processor cores;

memory also offers different addressing modes, as well as data filtering, for some specific data formats (see Texture and Surface Memory).

The global, constant, and texture memory spaces are persistent across kernel launches by the same application.

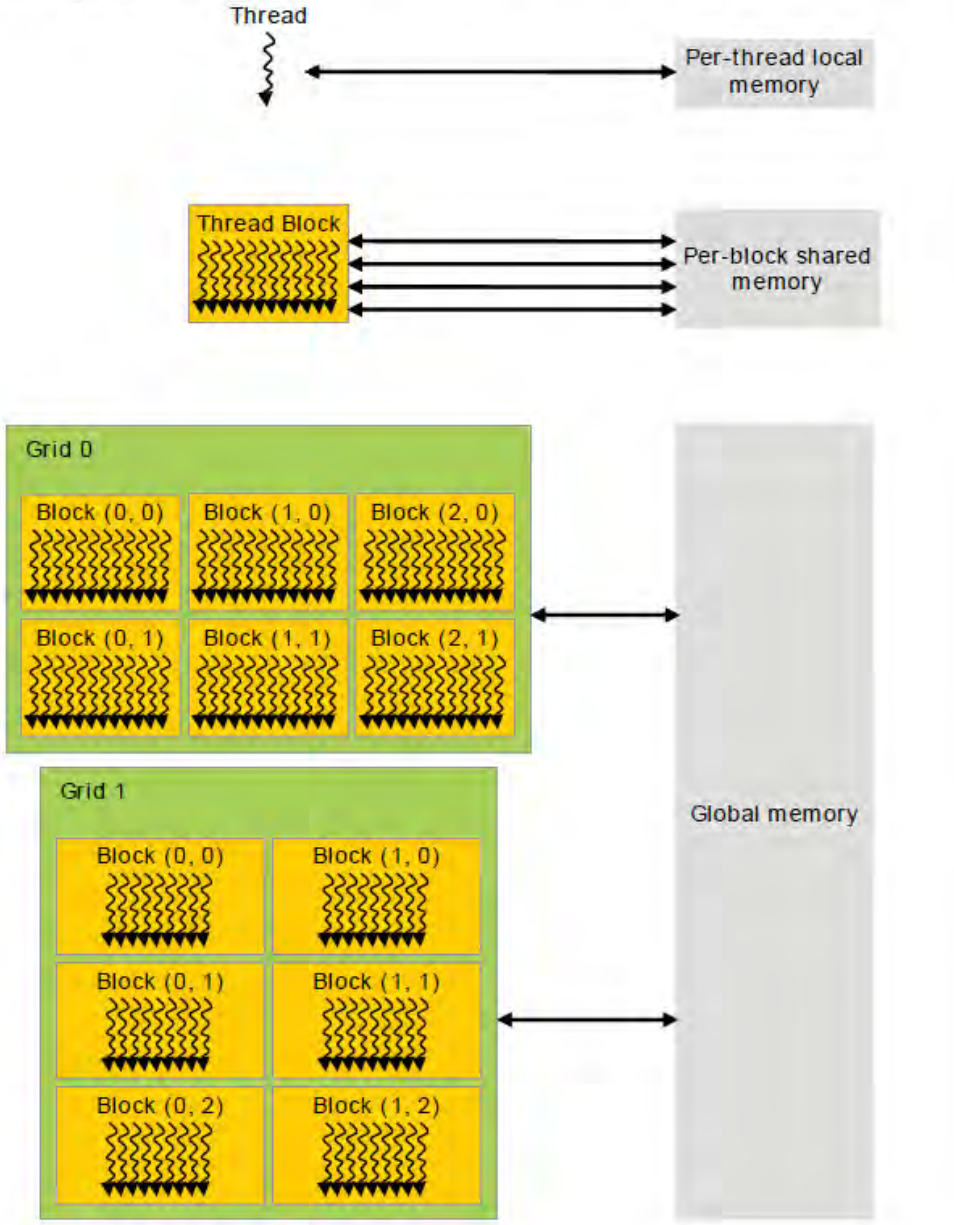


Figure 7 Memory Hierarchy

NVidia CUDA C Programming Guide pp. 7, 10-11 (v10.1 May 2019)
https://docs.nvidia.com/pdf/CUDA_C_Programming_Guide.pdf.

GP100 was built to be the highest performing parallel computing processor in the world to address the needs of the GPU accelerated computing markets serviced by our Tesla P100 accelerator platform. Like previous Tesla-class GPUs, GP100 is composed of an array of Graphics Processing Clusters (GPCs), Texture Processing Clusters (TPCs), Streaming Multiprocessors (SMs), and memory controllers. A full GP100 consists of six GPCs, 60 Pascal SMs, 30 TPCs (each including two SMs), and eight 512-bit memory controllers (4096 bits total).

Each GPC inside GP100 has ten SMs. Each SM has 64 CUDA Cores and four texture units. With 60 SMs, GP100 has a total of 3840 single precision CUDA Cores and 240 texture units. Each memory controller is attached to 512 KB of L2 cache, and each HBM2 DRAM stack is controlled by a pair of memory controllers. The full GPU includes a total of 4096 KB of L2 cache.

Figure 7 shows a full GP100 GPU with 60 SM units (different products can use different configurations of GP100). The Tesla P100 accelerator uses 56 SM units.

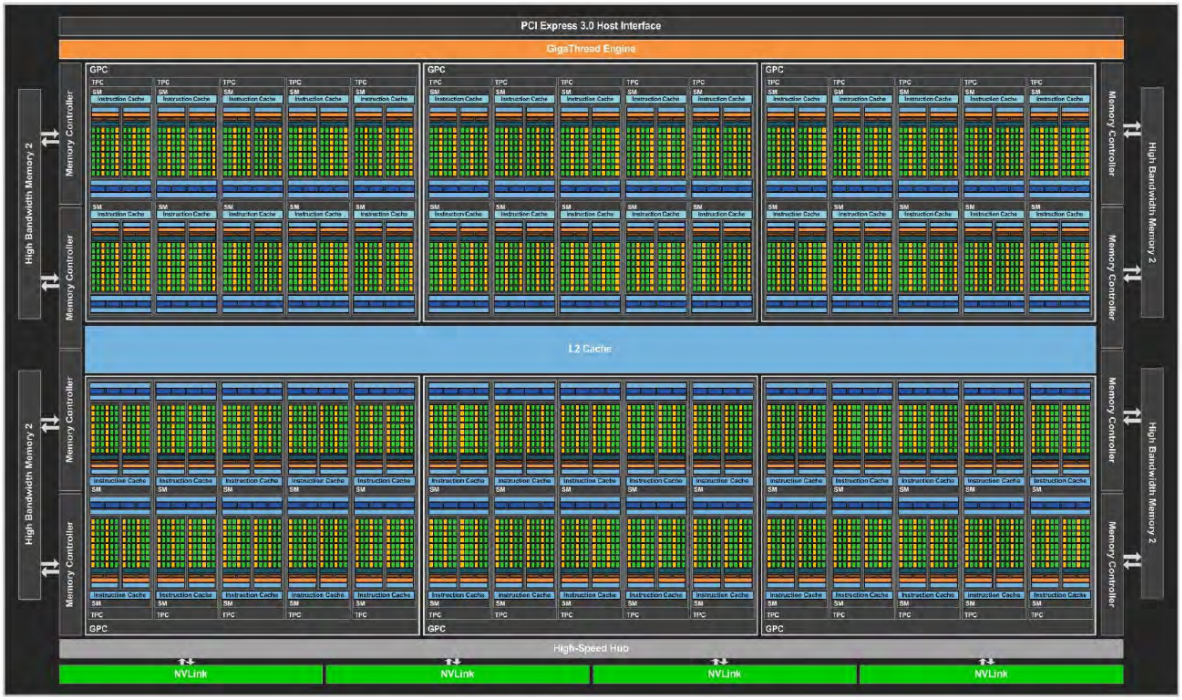


Figure 7. Pascal GP100 Full GPU with 60 SM Units

NVIDIA Tesla P100: The Most Advanced Datacenter Accelerator Ever Built Featuring Pascal GP100, the World's Fastest GPU p. 10, https://images.nvidia.com/content/pdf/tesla/whitepaper/pascal-architecture-whitepaper.pdf

Table 1. Comparison of NVIDIA Tesla GPUs

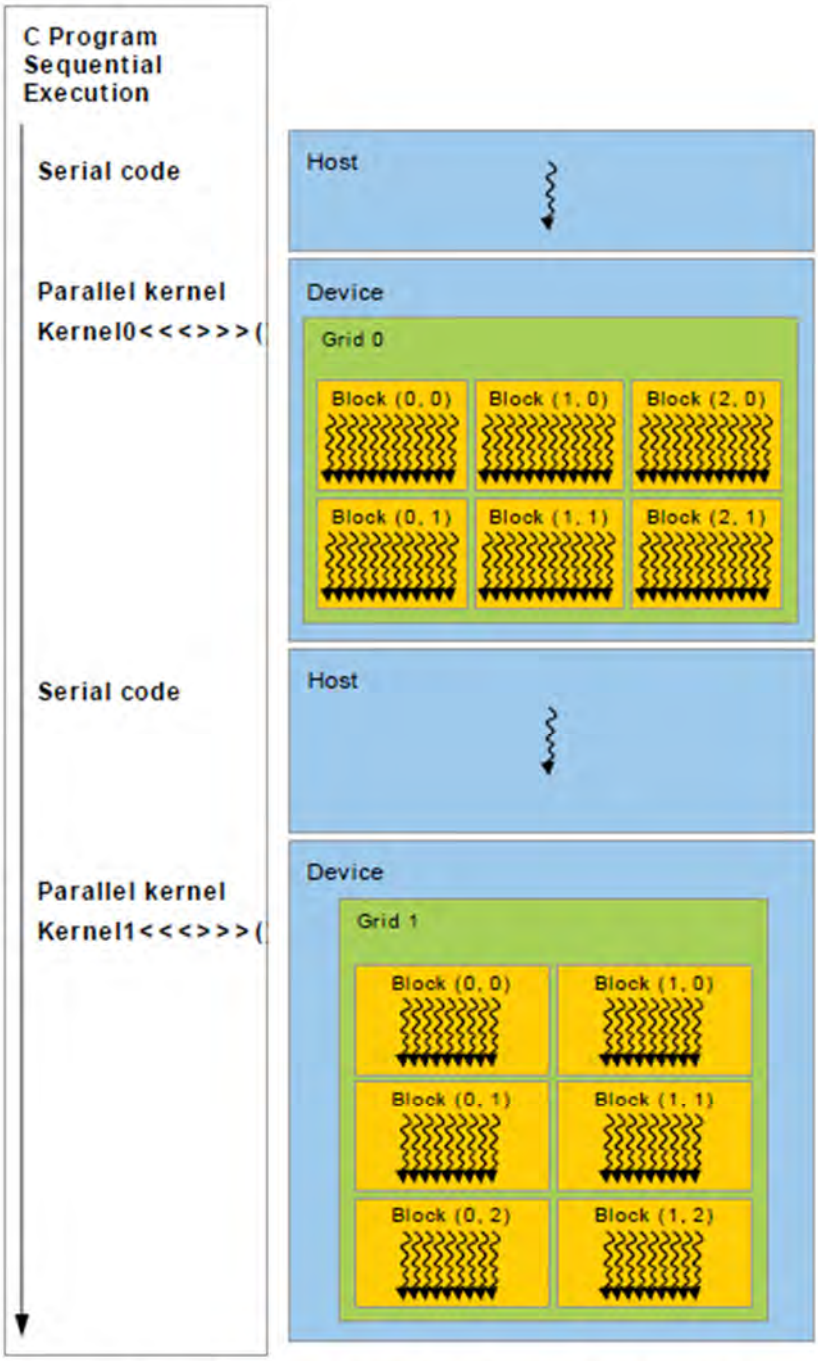
Tesla Product	Tesla K40	Tesla M40	Tesla P100	Tesla V100
GPU	GK180 (Kepler)	GM200 (Maxwell)	GP100 (Pascal)	GV100 (Volta)
SMs	15	24	56	80
TPCs	15	24	28	40
FP32 Cores / SM	192	128	64	64
FP32 Cores / GPU	2880	3072	3584	5120
FP64 Cores / SM	64	4	32	32
FP64 Cores / GPU	960	96	1792	2560
Tensor Cores / SM	NA	NA	NA	8
Tensor Cores / GPU	NA	NA	NA	640
GPU Boost Clock	810/875 MHz	1114 MHz	1480 MHz	1530 MHz
Peak FP32 TFLOPS[1]	5	6.8	10.6	15.7
Peak FP64 TFLOPS[1]	1.7	.21	5.3	7.8
Peak Tensor TFLOPS[1]	NA	NA	NA	125
Texture Units	240	192	224	320
Memory Interface	384-bit GDDR5	384-bit GDDR5	4096-bit HBM2	4096-bit HBM2
Memory Size	Up to 12 GB	Up to 24 GB	16 GB	16 GB
L2 Cache Size	1536 KB	3072 KB	4096 KB	6144 KB
Shared Memory Size / SM	16 KB/32 KB/48 KB	96 KB	64 KB	Configurable up to 96 KB
Register File Size / SM	256 KB	256 KB	256 KB	256KB
Register File Size / GPU	3840 KB	6144 KB	14336 KB	20480 KB
TDP	235 Watts	250 Watts	300 Watts	300 Watts
Transistors	7.1 billion	8 billion	15.3 billion	21.1 billion
GPU Die Size	551 mm²	601 mm²	610 mm²	815 mm²
Manufacturing Process	28 nm	28 nm	16 nm FinFET+	12 nm FFN

[1] Peak TFLOPS rates are based on GPU Boost Clock

NVidia Tesla V100 GPU Architecture: The World's Most Advanced Data Center GPU, p. 10, https://images.nvidia.com/content/volta-architecture/pdf/volta-architecture-whitepaper.pdf.

2.4. Heterogeneous Programming

As illustrated by Figure 8, the CUDA programming model assumes that the CUDA threads execute on a physically separate *device* that operates as a coprocessor to the *host* running the C program. This is the case, for example, when the kernels execute on a GPU and the rest of the C program executes on a CPU.



Serial code executes on the host while parallel code executes on the device.

Figure 8 Heterogeneous Programming

NVidia CUDA C Programming Guide pp. 12-13 (v10.1 May 2019), https://docs.nvidia.com/pdf/CUDA_C_Programming_Guide.pdf.

a user connection interface	Each Accused GPU Product and Accused Server Product comprises a user connection interface configured to receive a command to start a cluster initialization process for a computer cluster.

configured to receive a command to start a cluster initialization process for a computer cluster;	For example, each GPU Accelerator has a user connection interface configured to receive a command to start a cluster initialization process for a computer cluster. As mentioned in Heterogeneous Programming, the CUDA programming model assumes a system composed of a host and a device, each with their own separate memory. Device Memory gives an overview of the runtime functions used to manage device memory. **3.2.1. Initialization** There is no explicit initialization function for the runtime; it initializes the first time a runtime function is called (more specifically any function other than functions from the device and version management sections of the reference manual). One needs to keep this in mind when timing runtime function calls and when interpreting the error code from the first call into the runtime. During initialization, the runtime creates a CUDA context for each device in the system (see Context for more details on CUDA contexts). This context is the *primary context* for this device and it is shared among all the host threads of the application. As part of this context creation, the device code is just-in-time compiled if necessary (see Just-in-Time Compilation) and loaded into device memory. This all happens under the hood and the runtime does not expose the primary context to the application. When a host thread calls `cudaDeviceReset()`, this destroys the primary context of the device the host thread currently operates on (i.e., the current device as defined in Device Selection). The next runtime function call made by any host thread that has this device as current will create a new primary context for this device. NVidia CUDA C Programming Guide pp. 18-19 (v10.1 May 2019), https://docs.nvidia.com/pdf/CUDA_C_Programming_Guide.pdf
a mechanism to communicate results of evaluation with other computer cluster nodes using a peer-to-peer architecture; and	Each Accused GPU Product and Accused Server Product comprises a mechanism to communicate results of evaluation with other computer cluster nodes using a peer-to-peer architecture. For example, each GPU Accelerator includes NVLink to communicate results of evaluation with other computer GPU Accelerators using a peer-to-peer architecture . **Server Configuration with NVLink** In the following sections of this paper, we analyze the performance benefit of NVLink for several algorithms and applications by comparing model systems based on PCIe-interconnected next-gen GPUs to otherwise-identical systems with NVLink-interconnected GPUs. GPUs are connected to the CPU using existing PCIe connections, but the NVLink configurations augment this with interconnections among the GPUs for peer-to-peer communication. The following analyses assume future-generation GPUs with performance higher than that of today's GPUs, so as to better correspond with the GPUs that will be contemporary with NVLink.

The second scenario, shown in Figure 3, compares a pair of configurations with four GPUs each, referred to as *4-GPU-PCIe* and *4-GPU-NVLink*. 4-GPU-PCIe uses a PCIe tree topology, again mirroring a configuration used commonly today. 4-GPU-NVLink enhances this by providing point-to-point connections with either one or two NVLink connections per pair of GPUs, yielding 16 GB/s or 32 GB/s effective bandwidth per direction per pair, respectively.

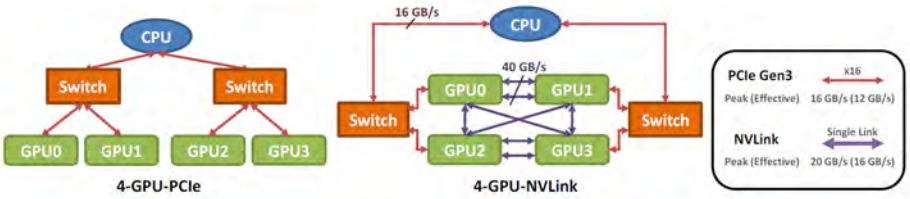


Figure 3: Comparing 4-GPU topologies with NVLink and PCIe. In 4-GPU-NVLink, GPU0 and GPU1 have 40 GB/s peak bandwidth between them, as do GPU2 and GPU3. The other peer-to-peer connections have 20 GB/s peak bandwidth.

The exchange algorithm begins with data evenly distributed across multiple GPUs as shown below.

Original Data

8	3	7	4	6	1	9	2	7	5	3	8	2	0	1	9

In this diagram, the color indicates which device the original data began on.

The first step is to hash the data to determine which GPU's bin each element should be placed in. For sorting, this might be as simple as extracting the highest order bits, but in theory we could apply any arbitrary hashing function to determine the keys (i.e., the destination bins). For this example, we will use the hash function `ceil(x/5)`, which produces the following keys given the data above.

Calculated Keys

1	0	1	0	1	0	1	0	1	1	0	1	0	0	0	1

Next we locally sort the data based on the hashed keys, yielding the following. Note that since we know these keys are in the range of `[0, NUM_GPUS)`, we can restrict the sorting to only a couple of bits.

Locally-sorted Data

3	4	1	2	8	7	6	9	3	2	0	1	7	5	8	9

Locally-sorted Keys

0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1

Next, we transfer all data with a key of 0 to device 0 and all data with a key of 1 to device 1. This becomes an all-to-all exchange among GPUs, producing the following.

Exchanged Data

3	4	1	2	3	2	0	1	8	7	6	9	7	5	8	9

Multi-GPU sorting is a simple extension of the above. After exchanging the data, we simply sort the data again locally to produce the final sorted list.

Fully-sorted Data

0	1	1	2	2	3	3	4	5	6	7	7	8	8	9	9

Here we can see the final list has been completely sorted. For many applications, the initial data exchange is all that is needed. Thus we will model both the exchange and the final sort separately.

Performance Considerations

The exchange algorithm can be easily pipelined by operating on small portions of the domain. The first step of the pipeline is to perform the local sort on the generated keys; the second is to perform the all-to-all exchange. This allows the initial sorting and the exchange to be completely overlapped. For the full multi-GPU sorting algorithm, the final local sort cannot begin until all data has arrived, so this portion cannot be pipelined with the other two.

If we assume the original data is randomly distributed among GPUs following a uniform random distribution, then each GPU must communicate `W*N/P` bytes of data to every other GPU, where W is the size of each element in bytes, N is the number of elements per GPU, and P is the number of GPUs.

In a PCIe tree topology, the communication is limited by the bisection bandwidth. Each GPU must communicate half of its data (N/2) across the top link in the system, and there are P/2 devices all trying to communicate across the same PCIe lanes in the same direction.

On an NVLink system, however, the communication can occur in parallel, because there are dedicated links between all pairs of GPUs.

NVIDIA NVLink High-Speed Interconnect: Application Performance p. 4-7, https://info.nvidianews.com/rs/nvidia/images/NVIDIA%20NVLink%20High-Speed%20Interconnect%20Application%20Performance%20Brief.pdf.

To address this issue, Tesla P100 features NVIDIA's new high-speed interface, NVLink, that provides GPU-to-GPU data transfers at up to 160 Gigabytes/second of bidirectional bandwidth—5x the bandwidth of PCIe Gen 3 x16. Figure 4 shows NVLink connecting eight Tesla P100 Accelerators in a Hybrid Cube Mesh Topology.

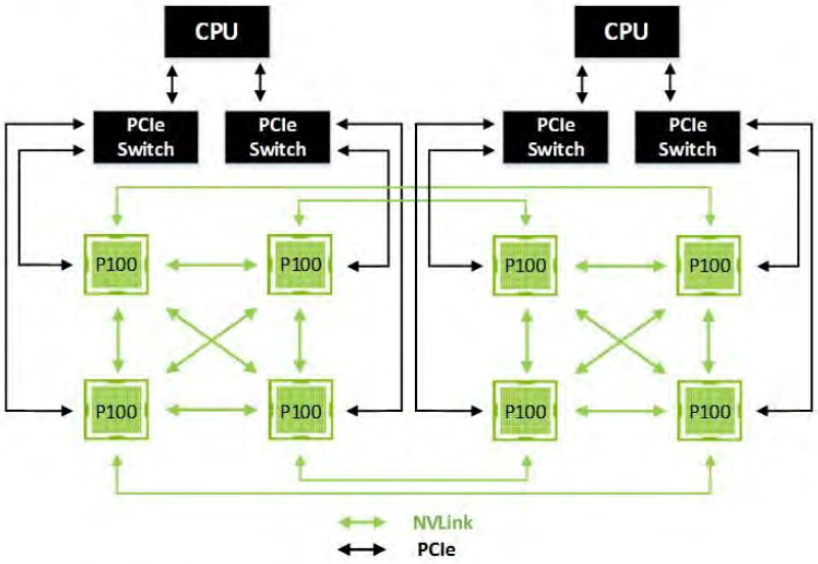
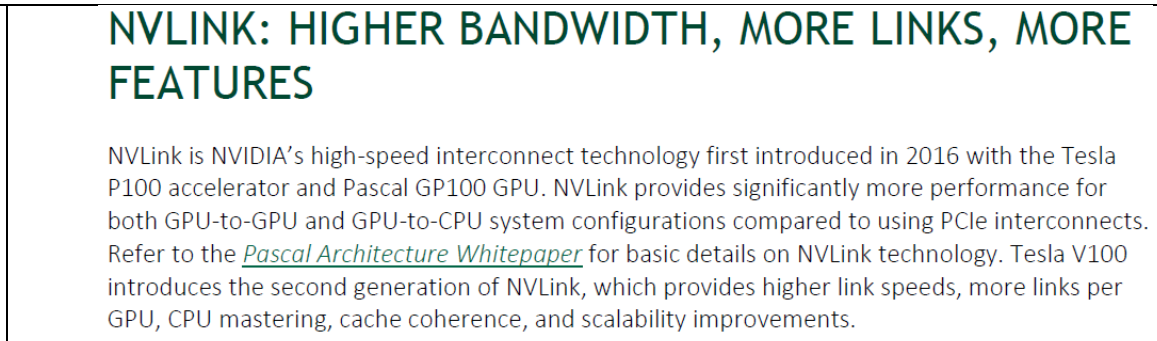
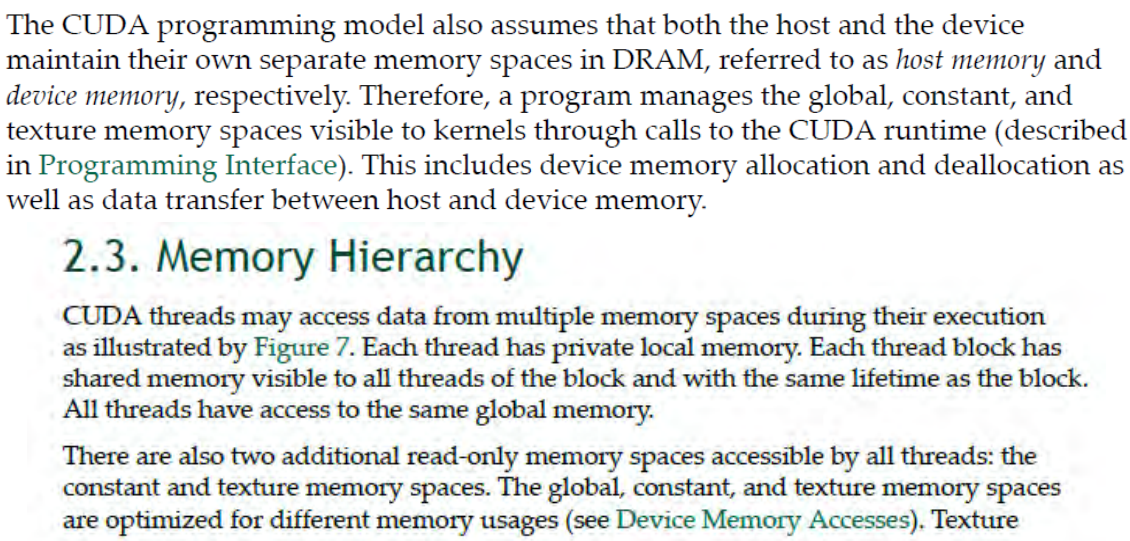


Figure 4. NVLink Connecting Eight Tesla P100 Accelerators in a Hybrid Cube Mesh Topology

NVIDIA Tesla P100: The Most Advanced Datacenter Accelerator Ever Built Featuring Pascal GP100, the World's Fastest GPU p. 7, https://images.nvidia.com/content/pdf/tesla/whitepaper/pascal-architecture-whitepaper.pdf.

	 NVidia Tesla V100 GPU Architecture: The World's Most Advanced Data Center GPU, p. 19, https://images.nvidia.com/content/volta-architecture/pdf/volta-architecture-whitepaper.pdf.
program code that, when executed, is capable of causing the hardware processor to: receive calls from a second node comprising a second hardware processor configured to access a second memory comprising program code for a user interface and program code for a second single-node kernel, the second single-node kernel configured to interpret user instructions and distribute calls to at least one of a plurality of other nodes for execution.	Each Accused GPU Product and Accused Server Product comprises program code that, when executed, is capable of causing the hardware processor to: receive calls from a second node comprising a second hardware processor configured to access a second memory comprising program code for a user interface and program code for a second single-node kernel, the second single-node kernel configured to interpret user instructions and distribute calls to at least one of a plurality of other nodes for execution. For example, each Accused GPU Product and the GPU Accelerators of each Accused Server Product execute CUDA program code that is capable of causing the hardware processor to receive calls from a second GPU Accelerator (node) that has a second hardware processor configured to access a second memory comprising CUDA program code for a user interface and a CUDA kernel, the second CUDA kernel configured to interpret user instructions and distribute calls to at least one of a plurality of other nodes for execution. 

memory also offers different addressing modes, as well as data filtering, for some specific data formats (see Texture and Surface Memory).

The global, constant, and texture memory spaces are persistent across kernel launches by the same application.

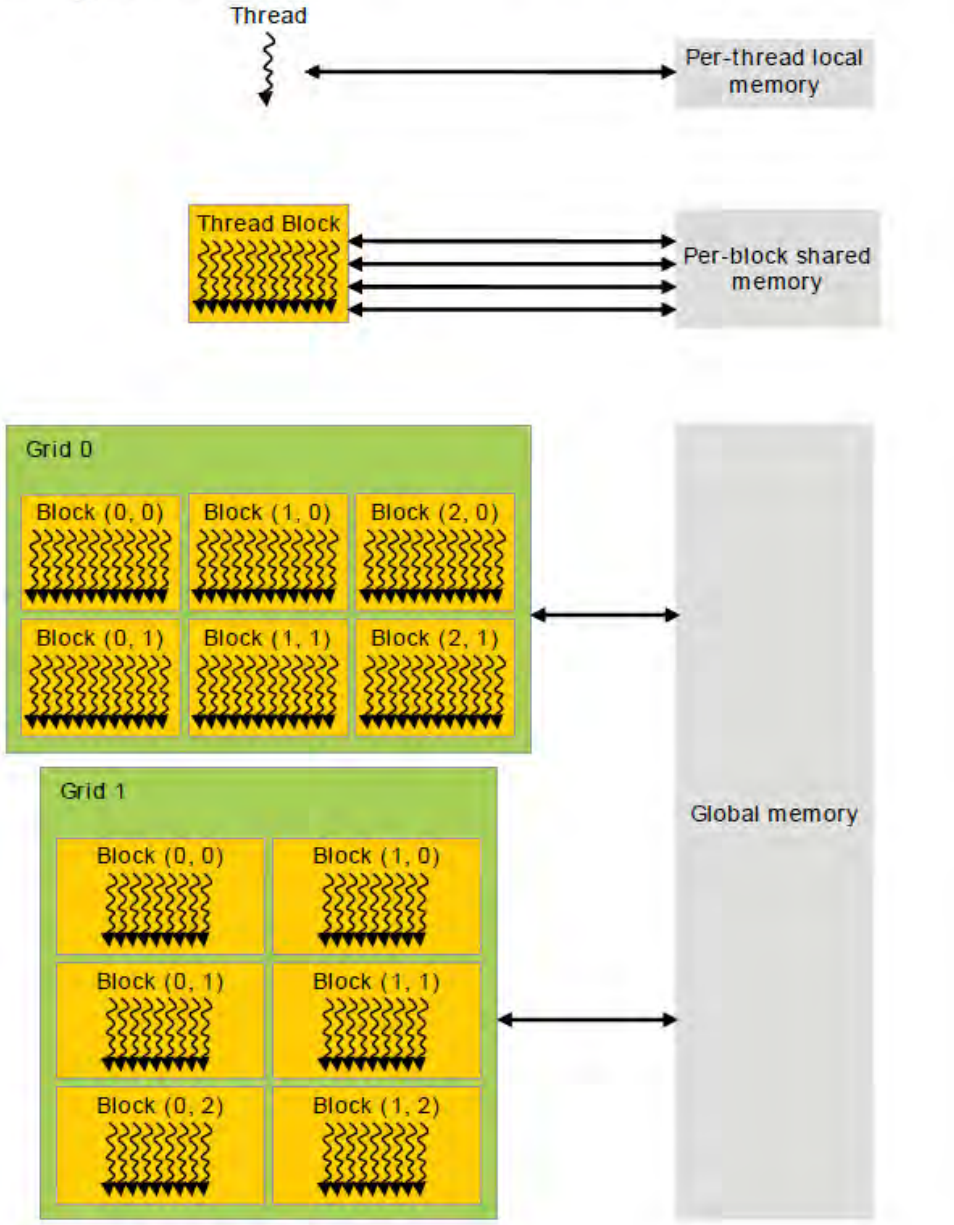


Figure 7 Memory Hierarchy

NVidia CUDA C Programming Guide pp. 7, 10-11 (v10.1 May 2019)
https://docs.nvidia.com/pdf/CUDA_C_Programming_Guide.pdf.

GP100 was built to be the highest performing parallel computing processor in the world to address the needs of the GPU accelerated computing markets serviced by our Tesla P100 accelerator platform. Like previous Tesla-class GPUs, GP100 is composed of an array of Graphics Processing Clusters (GPCs), Texture Processing Clusters (TPCs), Streaming Multiprocessors (SMs), and memory controllers. A full GP100 consists of six GPCs, 60 Pascal SMs, 30 TPCs (each including two SMs), and eight 512-bit memory controllers (4096 bits total).

Each GPC inside GP100 has ten SMs. Each SM has 64 CUDA Cores and four texture units. With 60 SMs, GP100 has a total of 3840 single precision CUDA Cores and 240 texture units. Each memory controller is attached to 512 KB of L2 cache, and each HBM2 DRAM stack is controlled by a pair of memory controllers. The full GPU includes a total of 4096 KB of L2 cache.

Figure 7 shows a full GP100 GPU with 60 SM units (different products can use different configurations of GP100). The Tesla P100 accelerator uses 56 SM units.

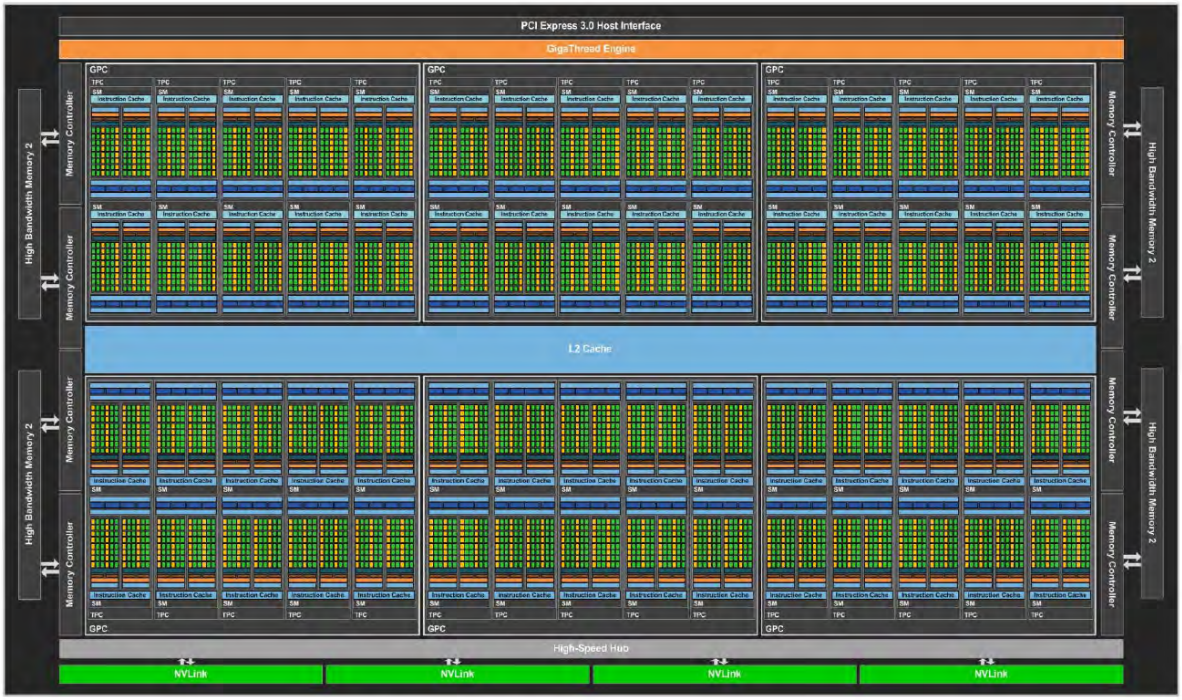


Figure 7. Pascal GP100 Full GPU with 60 SM Units

NVIDIA Tesla P100: The Most Advanced Datacenter Accelerator Ever Built Featuring Pascal GP100, the World's Fastest GPU p. 10, https://images.nvidia.com/content/pdf/tesla/whitepaper/pascal-architecture-whitepaper.pdf

Table 1. Comparison of NVIDIA Tesla GPUs

Tesla Product	Tesla K40	Tesla M40	Tesla P100	Tesla V100
GPU	GK180 (Kepler)	GM200 (Maxwell)	GP100 (Pascal)	GV100 (Volta)
SMs	15	24	56	80
TPCs	15	24	28	40
FP32 Cores / SM	192	128	64	64
FP32 Cores / GPU	2880	3072	3584	5120
FP64 Cores / SM	64	4	32	32
FP64 Cores / GPU	960	96	1792	2560
Tensor Cores / SM	NA	NA	NA	8
Tensor Cores / GPU	NA	NA	NA	640
GPU Boost Clock	810/875 MHz	1114 MHz	1480 MHz	1530 MHz
Peak FP32 TFLOPS[1]	5	6.8	10.6	15.7
Peak FP64 TFLOPS[1]	1.7	.21	5.3	7.8
Peak Tensor TFLOPS[1]	NA	NA	NA	125
Texture Units	240	192	224	320
Memory Interface	384-bit GDDR5	384-bit GDDR5	4096-bit HBM2	4096-bit HBM2
Memory Size	Up to 12 GB	Up to 24 GB	16 GB	16 GB
L2 Cache Size	1536 KB	3072 KB	4096 KB	6144 KB
Shared Memory Size / SM	16 KB/32 KB/48 KB	96 KB	64 KB	Configurable up to 96 KB
Register File Size / SM	256 KB	256 KB	256 KB	256KB
Register File Size / GPU	3840 KB	6144 KB	14336 KB	20480 KB
TDP	235 Watts	250 Watts	300 Watts	300 Watts
Transistors	7.1 billion	8 billion	15.3 billion	21.1 billion
GPU Die Size	551 mm²	601 mm²	610 mm²	815 mm²
Manufacturing Process	28 nm	28 nm	16 nm FinFET+	12 nm FFN

[1] Peak TFLOPS rates are based on GPU Boost Clock

NVidia Tesla V100 GPU Architecture: The World's Most Advanced Data Center GPU, p. 10, https://images.nvidia.com/content/volta-architecture/pdf/volta-architecture-whitepaper.pdf.

Server Configuration with NVLink

In the following sections of this paper, we analyze the performance benefit of NVLink for several algorithms and applications by comparing model systems based on PCIe-interconnected next-gen GPUs to otherwise-identical systems with NVLink-interconnected GPUs. GPUs are connected to the CPU using existing PCIe connections, but the NVLink configurations augment this with interconnections among the GPUs for peer-to-peer communication. The following analyses assume future-generation GPUs with performance higher than that of today's GPUs, so as to better correspond with the GPUs that will be contemporary with NVLink.

The second scenario, shown in Figure 3, compares a pair of configurations with four GPUs each, referred to as *4-GPU-PCIe* and *4-GPU-NVLink*. 4-GPU-PCIe uses a PCIe tree topology, again mirroring a configuration used commonly today. 4-GPU-NVLink enhances this by providing point-to-point connections with either one or two NVLink connections per pair of GPUs, yielding 16 GB/s or 32 GB/s effective bandwidth per direction per pair, respectively.

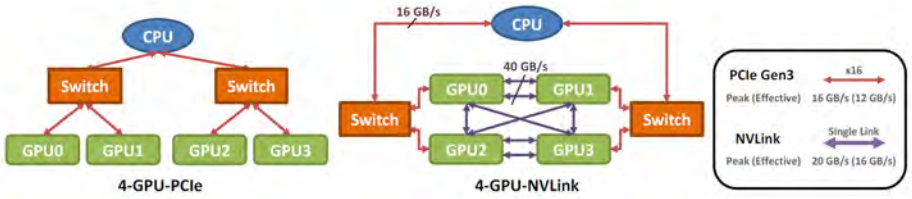


Figure 3: Comparing 4-GPU topologies with NVLink and PCIe. In 4-GPU-NVLink, GPU0 and GPU1 have 40 GB/s peak bandwidth between them, as do GPU2 and GPU3. The other peer-to-peer connections have 20 GB/s peak bandwidth.

The exchange algorithm begins with data evenly distributed across multiple GPUs as shown below.

Original Data

In this diagram, the color indicates which device the original data began on.

The first step is to hash the data to determine which GPU's bin each element should be placed in. For sorting, this might be as simple as extracting the highest order bits, but in theory we could apply any arbitrary hashing function to determine the keys (i.e., the destination bins). For this example, we will use the hash function `ceil(x/5)`, which produces the following keys given the data above.

Calculated Keys

Next we locally sort the data based on the hashed keys, yielding the following. Note that since we know these keys are in the range of `[0, NUM_GPUS)`, we can restrict the sorting to only a couple of bits.

Locally-sorted Data

Locally-sorted Keys

Next, we transfer all data with a key of 0 to device 0 and all data with a key of 1 to device 1. This becomes an all-to-all exchange among GPUs, producing the following.

Exchanged Data

Multi-GPU sorting is a simple extension of the above. After exchanging the data, we simply sort the data again locally to produce the final sorted list.

Fully-sorted Data

Here we can see the final list has been completely sorted. For many applications, the initial data exchange is all that is needed. Thus we will model both the exchange and the final sort separately.

Performance Considerations

The exchange algorithm can be easily pipelined by operating on small portions of the domain. The first step of the pipeline is to perform the local sort on the generated keys; the second is to perform the all-to-all exchange. This allows the initial sorting and the exchange to be completely overlapped. For the full multi-GPU sorting algorithm, the final local sort cannot begin until all data has arrived, so this portion cannot be pipelined with the other two.

If we assume the original data is randomly distributed among GPUs following a uniform random distribution, then each GPU must communicate `W*N/P` bytes of data to every other GPU, where W is the size of each element in bytes, N is the number of elements per GPU, and P is the number of GPUs.

In a PCIe tree topology, the communication is limited by the bisection bandwidth. Each GPU must communicate half of its data (N/2) across the top link in the system, and there are P/2 devices all trying to communicate across the same PCIe lanes in the same direction.

On an NVLink system, however, the communication can occur in parallel, because there are dedicated links between all pairs of GPUs.

NVIDIA NVLink High-Speed Interconnect: Application Performance p. 4-7, https://info.nvidianews.com/rs/nvidia/images/NVIDIA%20NVLink%20High-Speed%20Interconnect%20Application%20Performance%20Brief.pdf.

To address this issue, Tesla P100 features NVIDIA's new high-speed interface, NVLink, that provides GPU-to-GPU data transfers at up to 160 Gigabytes/second of bidirectional bandwidth—5x the bandwidth of PCIe Gen 3 x16. Figure 4 shows NVLink connecting eight Tesla P100 Accelerators in a Hybrid Cube Mesh Topology.

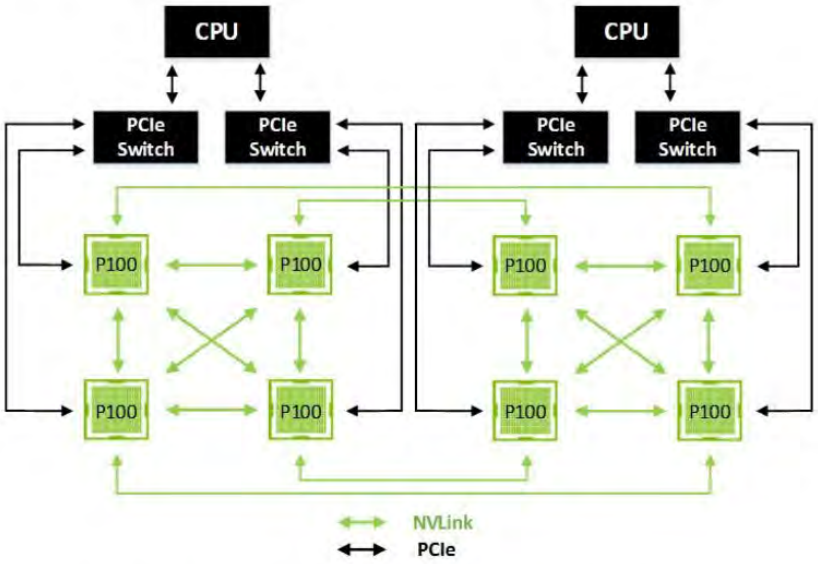
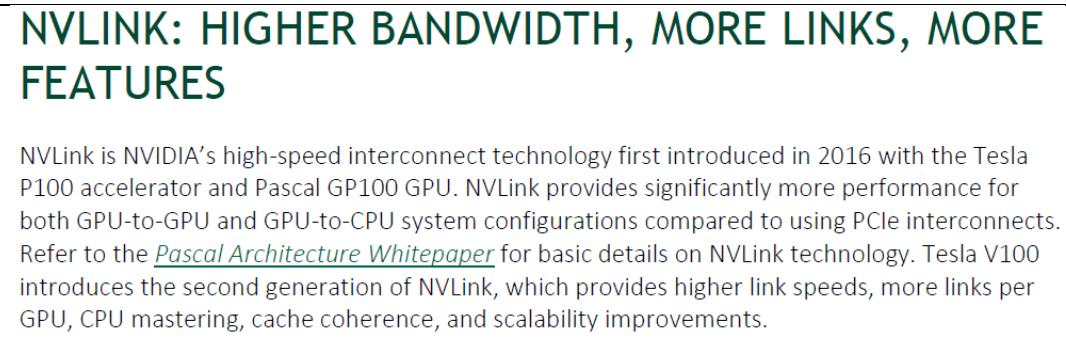


Figure 4. NVLink Connecting Eight Tesla P100 Accelerators in a Hybrid Cube Mesh Topology

NVIDIA Tesla P100: The Most Advanced Datacenter Accelerator Ever Built Featuring Pascal GP100, the World's Fastest GPU p. 7, https://images.nvidia.com/content/pdf/tesla/whitepaper/pascal-architecture-whitepaper.pdf.

<table>
<tr><td></td><td>


NVidia Tesla V100 GPU Architecture: The World's Most Advanced Data Center GPU, p. 19, https://images.nvidia.com/content/volta-architecture/pdf/volta-architecture-whitepaper.pdf.</td></tr>
<tr><td>execute, using the hardware processor, at least a first mathematical expression evaluation; and</td><td>Each Accused GPU Product and Accused Server Product comprises program code that, when executed, is capable of causing the hardware processor to execute, using the hardware processor, at least a first mathematical expression evaluation.

For example, each Accused GPU Product and the GPU Accelerators of each Accused Server Product execute CUDA program code that is capable of causing the hardware processor to execute mathematical expressions.

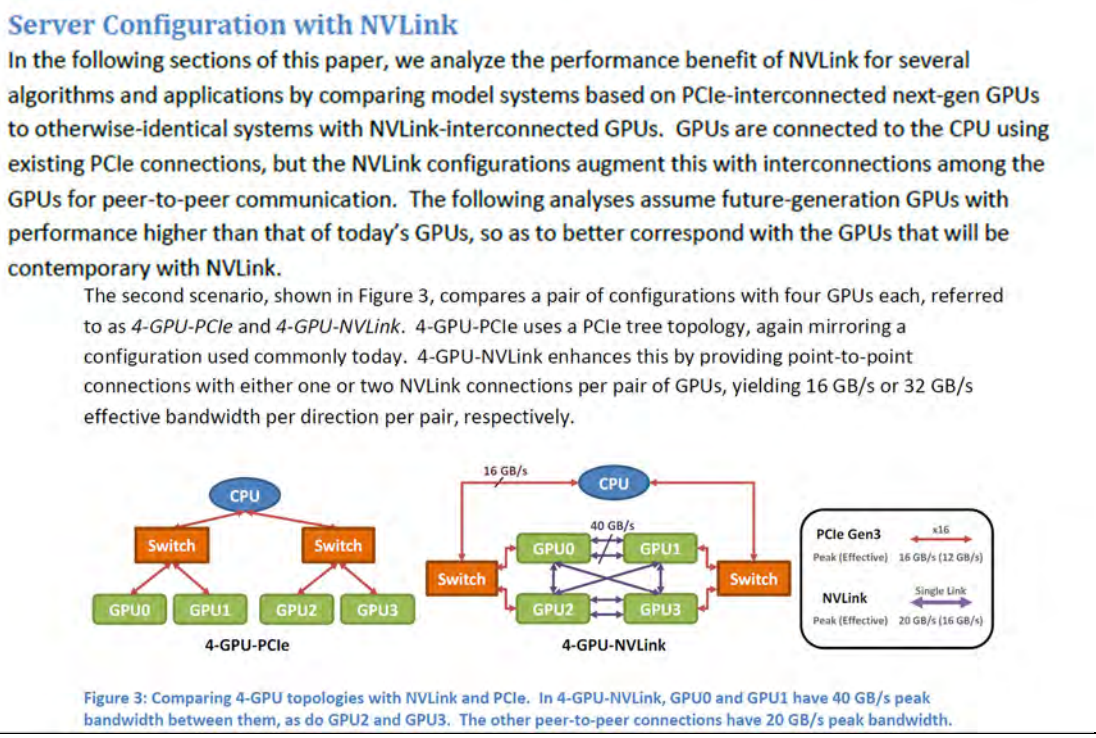


Figure 3: Comparing 4-GPU topologies with NVLink and PCIe. In 4-GPU-NVLink, GPU0 and GPU1 have 40 GB/s peak bandwidth between them, as do GPU2 and GPU3. The other peer-to-peer connections have 20 GB/s peak bandwidth.</td></tr>
</table>

The exchange algorithm begins with data evenly distributed across multiple GPUs as shown below.

Original Data

In this diagram, the color indicates which device the original data began on.

The first step is to hash the data to determine which GPU's bin each element should be placed in. For sorting, this might be as simple as extracting the highest order bits, but in theory we could apply any arbitrary hashing function to determine the keys (i.e., the destination bins). For this example, we will use the hash function `ceil(x/5)`, which produces the following keys given the data above.

Calculated Keys

Next we locally sort the data based on the hashed keys, yielding the following. Note that since we know these keys are in the range of `[0, NUM_GPUS)`, we can restrict the sorting to only a couple of bits.

Locally-sorted Data

Locally-sorted Keys

Next, we transfer all data with a key of 0 to device 0 and all data with a key of 1 to device 1. This becomes an all-to-all exchange among GPUs, producing the following.

Exchanged Data

Multi-GPU sorting is a simple extension of the above. After exchanging the data, we simply sort the data again locally to produce the final sorted list.

Fully-sorted Data

Here we can see the final list has been completely sorted. For many applications, the initial data exchange is all that is needed. Thus we will model both the exchange and the final sort separately.

Performance Considerations

The exchange algorithm can be easily pipelined by operating on small portions of the domain. The first step of the pipeline is to perform the local sort on the generated keys; the second is to perform the all-to-all exchange. This allows the initial sorting and the exchange to be completely overlapped. For the full multi-GPU sorting algorithm, the final local sort cannot begin until all data has arrived, so this portion cannot be pipelined with the other two.

If we assume the original data is randomly distributed among GPUs following a uniform random distribution, then each GPU must communicate `W*N/P` bytes of data to every other GPU, where W is the size of each element in bytes, N is the number of elements per GPU, and P is the number of GPUs.

In a PCIe tree topology, the communication is limited by the bisection bandwidth. Each GPU must communicate half of its data (N/2) across the top link in the system, and there are P/2 devices all trying to communicate across the same PCIe lanes in the same direction.

On an NVLink system, however, the communication can occur in parallel, because there are dedicated links between all pairs of GPUs.

NVIDIA NVLink High-Speed Interconnect: Application Performance p. 4-7, https://info.nvidianews.com/rs/nvidia/images/NVIDIA%20NVLink%20High-Speed%20Interconnect%20Application%20Performance%20Brief.pdf.

To address this issue, Tesla P100 features NVIDIA's new high-speed interface, NVLink, that provides GPU-to-GPU data transfers at up to 160 Gigabytes/second of bidirectional bandwidth—5x the bandwidth of PCIe Gen 3 x16. Figure 4 shows NVLink connecting eight Tesla P100 Accelerators in a Hybrid Cube Mesh Topology.

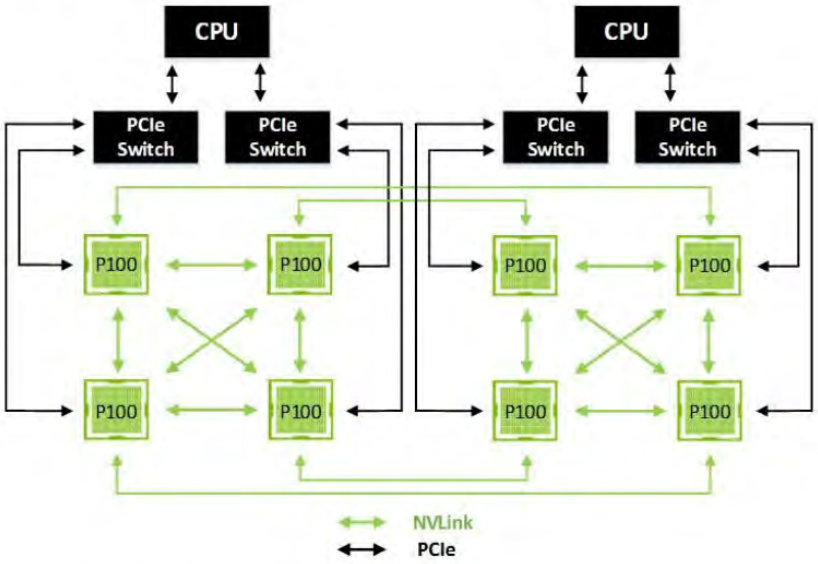
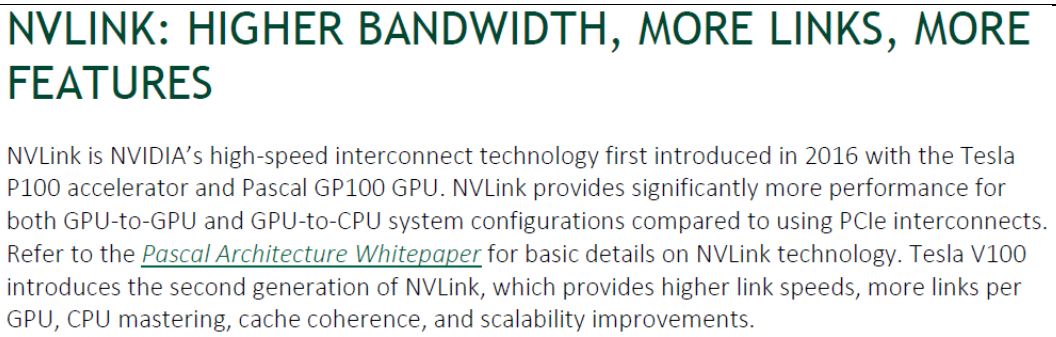
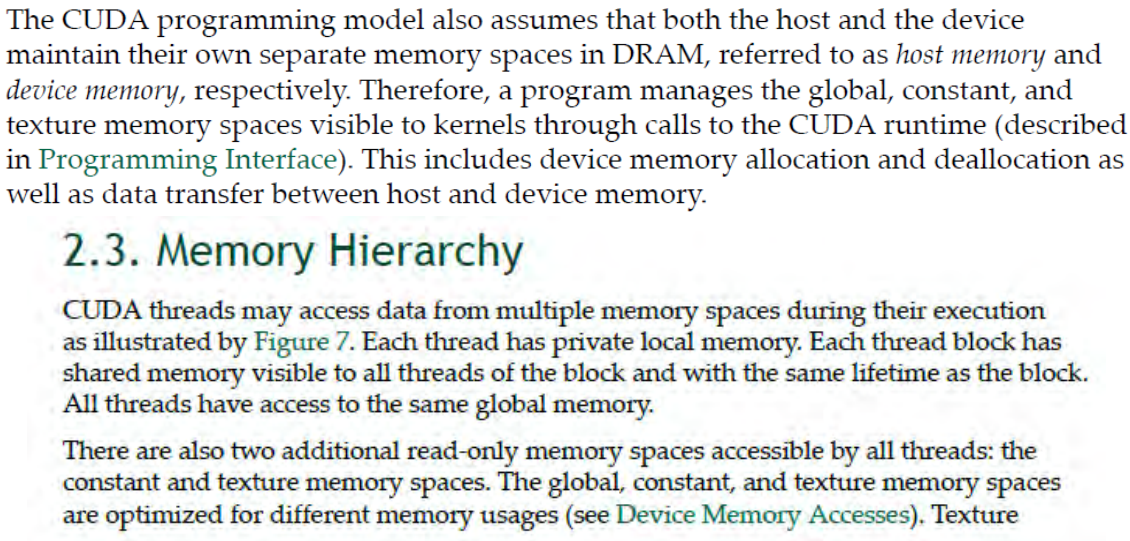


Figure 4. NVLink Connecting Eight Tesla P100 Accelerators in a Hybrid Cube Mesh Topology

NVIDIA Tesla P100: The Most Advanced Datacenter Accelerator Ever Built Featuring Pascal GP100, the World's Fastest GPU p. 7, https://images.nvidia.com/content/pdf/tesla/whitepaper/pascal-architecture-whitepaper.pdf.

	 NVidia Tesla V100 GPU Architecture: The World's Most Advanced Data Center GPU, p. 19, https://images.nvidia.com/content/volta-architecture/pdf/volta-architecture-whitepaper.pdf.
communicate a result of the first mathematical expression evaluation to a third node comprising a third hardware processor with a plurality of processing cores, wherein the third node is configured to receive the result of mathematical expression evaluation from the computer cluster node, execute at least a second mathematical expression evaluation using the result of the first mathematical expression evaluation, and communicate a result of the second mathematical expression evaluation to the first node·	Each Accused GPU Product and Accused Server Product comprises program code that, when executed, is capable of causing the hardware processor to communicate a result of the first mathematical expression evaluation to a third node comprising a third hardware processor with a plurality of processing cores, wherein the third node is configured to receive the result of mathematical expression evaluation from the computer cluster node, execute at least a second mathematical expression evaluation using the result of the first mathematical expression evaluation, and communicate a result of the second mathematical expression evaluation to the first node. For example, each Accused GPU Product and the GPU Accelerators of each Accused Server Product execute CUDA program code that is capable of causing the hardware processor to receive calls from a second GPU Accelerator (node) that has a second hardware processor configured to access a second memory comprising CUDA program code for a user interface and a CUDA kernel, the second CUDA kernel configured to interpret user instructions and distribute calls to at least one of a plurality of other nodes for execution. 

memory also offers different addressing modes, as well as data filtering, for some specific data formats (see Texture and Surface Memory).

The global, constant, and texture memory spaces are persistent across kernel launches by the same application.

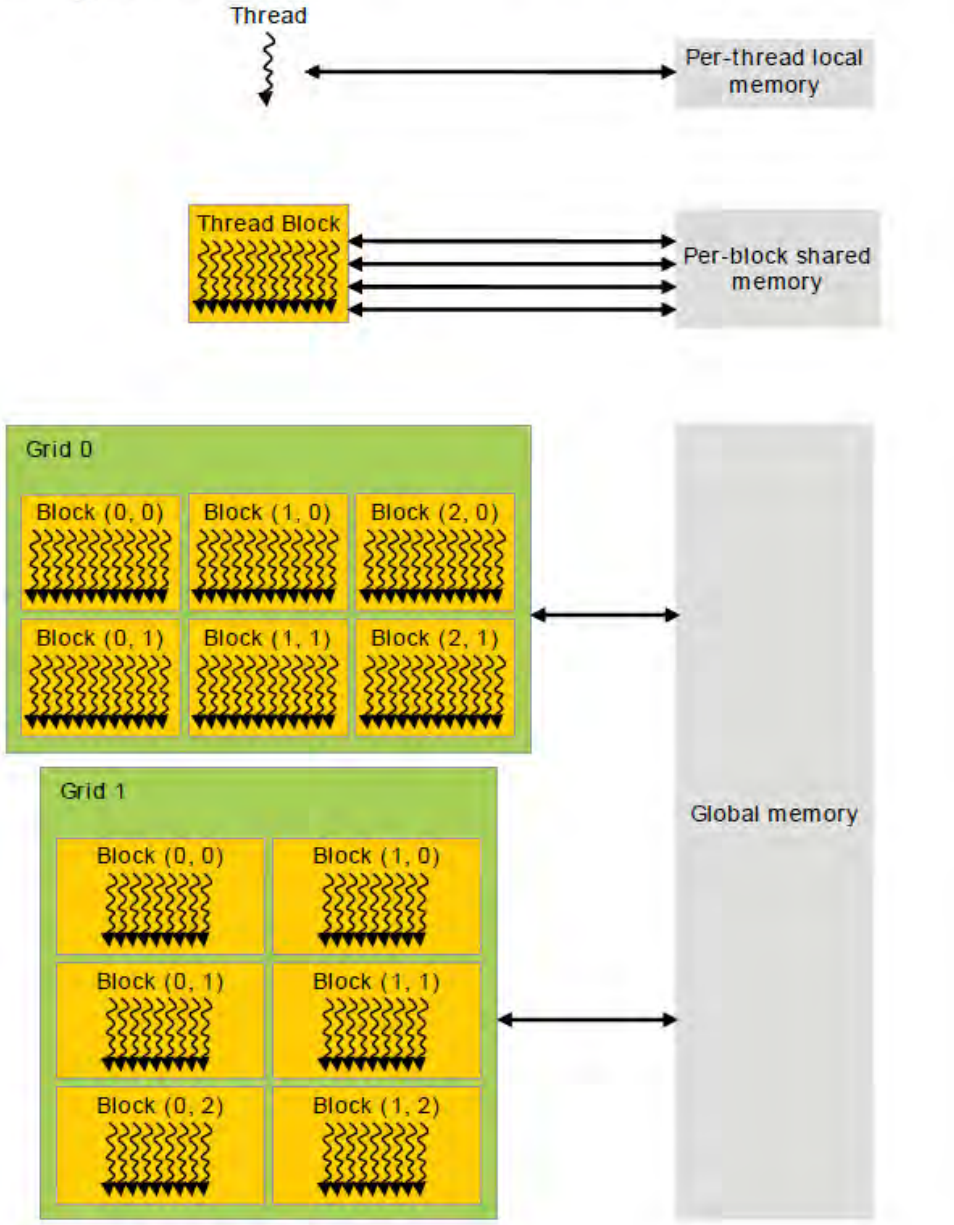


Figure 7 Memory Hierarchy

NVidia CUDA C Programming Guide pp. 7, 10-11 (v10.1 May 2019)
https://docs.nvidia.com/pdf/CUDA_C_Programming_Guide.pdf.

GP100 was built to be the highest performing parallel computing processor in the world to address the needs of the GPU accelerated computing markets serviced by our Tesla P100 accelerator platform. Like previous Tesla-class GPUs, GP100 is composed of an array of Graphics Processing Clusters (GPCs), Texture Processing Clusters (TPCs), Streaming Multiprocessors (SMs), and memory controllers. A full GP100 consists of six GPCs, 60 Pascal SMs, 30 TPCs (each including two SMs), and eight 512-bit memory controllers (4096 bits total).

Each GPC inside GP100 has ten SMs. Each SM has 64 CUDA Cores and four texture units. With 60 SMs, GP100 has a total of 3840 single precision CUDA Cores and 240 texture units. Each memory controller is attached to 512 KB of L2 cache, and each HBM2 DRAM stack is controlled by a pair of memory controllers. The full GPU includes a total of 4096 KB of L2 cache.

Figure 7 shows a full GP100 GPU with 60 SM units (different products can use different configurations of GP100). The Tesla P100 accelerator uses 56 SM units.

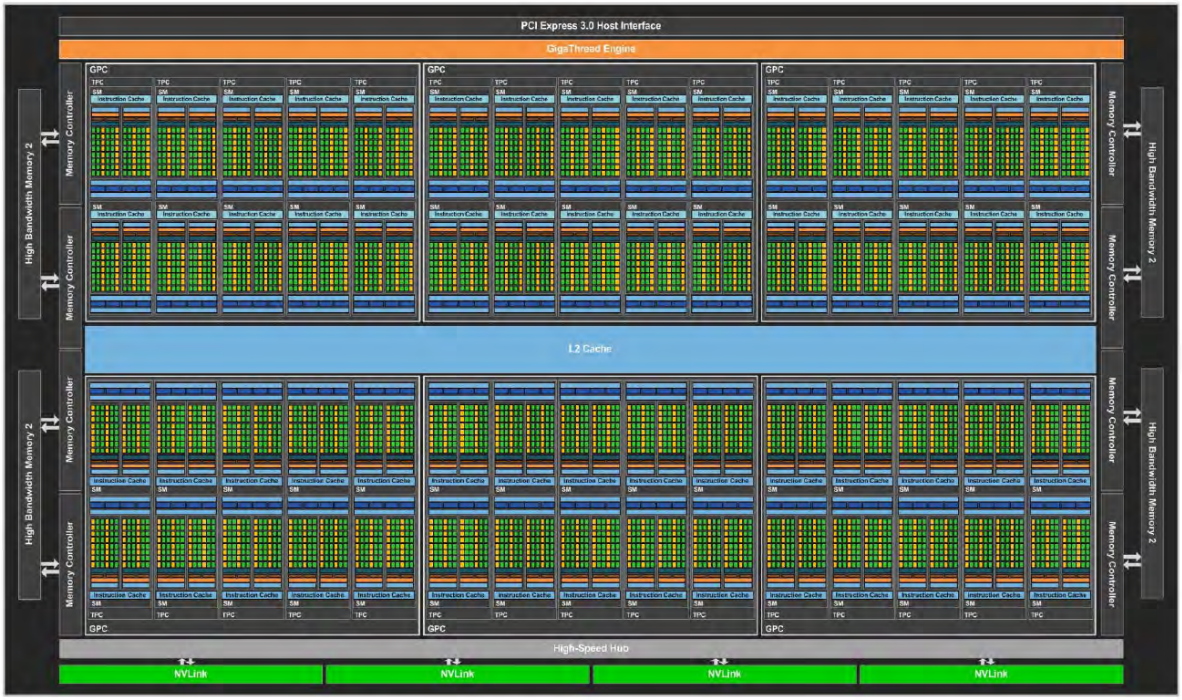


Figure 7. Pascal GP100 Full GPU with 60 SM Units

NVIDIA Tesla P100: The Most Advanced Datacenter Accelerator Ever Built Featuring Pascal GP100, the World's Fastest GPU p. 10, https://images.nvidia.com/content/pdf/tesla/whitepaper/pascal-architecture-whitepaper.pdf

Table 1. Comparison of NVIDIA Tesla GPUs

Tesla Product	Tesla K40	Tesla M40	Tesla P100	Tesla V100
GPU	GK180 (Kepler)	GM200 (Maxwell)	GP100 (Pascal)	GV100 (Volta)
SMs	15	24	56	80
TPCs	15	24	28	40
FP32 Cores / SM	192	128	64	64
FP32 Cores / GPU	2880	3072	3584	5120
FP64 Cores / SM	64	4	32	32
FP64 Cores / GPU	960	96	1792	2560
Tensor Cores / SM	NA	NA	NA	8
Tensor Cores / GPU	NA	NA	NA	640
GPU Boost Clock	810/875 MHz	1114 MHz	1480 MHz	1530 MHz
Peak FP32 TFLOPS[1]	5	6.8	10.6	15.7
Peak FP64 TFLOPS[1]	1.7	.21	5.3	7.8
Peak Tensor TFLOPS[1]	NA	NA	NA	125
Texture Units	240	192	224	320
Memory Interface	384-bit GDDR5	384-bit GDDR5	4096-bit HBM2	4096-bit HBM2
Memory Size	Up to 12 GB	Up to 24 GB	16 GB	16 GB
L2 Cache Size	1536 KB	3072 KB	4096 KB	6144 KB
Shared Memory Size / SM	16 KB/32 KB/48 KB	96 KB	64 KB	Configurable up to 96 KB
Register File Size / SM	256 KB	256 KB	256 KB	256KB
Register File Size / GPU	3840 KB	6144 KB	14336 KB	20480 KB
TDP	235 Watts	250 Watts	300 Watts	300 Watts
Transistors	7.1 billion	8 billion	15.3 billion	21.1 billion
GPU Die Size	551 mm²	601 mm²	610 mm²	815 mm²
Manufacturing Process	28 nm	28 nm	16 nm FinFET+	12 nm FFN

[1] Peak TFLOPS rates are based on GPU Boost Clock

NVidia Tesla V100 GPU Architecture: The World's Most Advanced Data Center GPU, p. 10, https://images.nvidia.com/content/volta-architecture/pdf/volta-architecture-whitepaper.pdf.

Server Configuration with NVLink

In the following sections of this paper, we analyze the performance benefit of NVLink for several algorithms and applications by comparing model systems based on PCIe-interconnected next-gen GPUs to otherwise-identical systems with NVLink-interconnected GPUs. GPUs are connected to the CPU using existing PCIe connections, but the NVLink configurations augment this with interconnections among the GPUs for peer-to-peer communication. The following analyses assume future-generation GPUs with performance higher than that of today's GPUs, so as to better correspond with the GPUs that will be contemporary with NVLink.

The second scenario, shown in Figure 3, compares a pair of configurations with four GPUs each, referred to as *4-GPU-PCIe* and *4-GPU-NVLink*. 4-GPU-PCIe uses a PCIe tree topology, again mirroring a configuration used commonly today. 4-GPU-NVLink enhances this by providing point-to-point connections with either one or two NVLink connections per pair of GPUs, yielding 16 GB/s or 32 GB/s effective bandwidth per direction per pair, respectively.

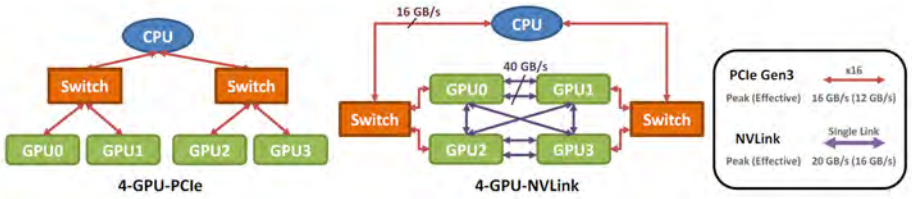


Figure 3: Comparing 4-GPU topologies with NVLink and PCIe. In 4-GPU-NVLink, GPU0 and GPU1 have 40 GB/s peak bandwidth between them, as do GPU2 and GPU3. The other peer-to-peer connections have 20 GB/s peak bandwidth.

The exchange algorithm begins with data evenly distributed across multiple GPUs as shown below.

Original Data

In this diagram, the color indicates which device the original data began on.

The first step is to hash the data to determine which GPU's bin each element should be placed in. For sorting, this might be as simple as extracting the highest order bits, but in theory we could apply any arbitrary hashing function to determine the keys (i.e., the destination bins). For this example, we will use the hash function `ceil(x/5)`, which produces the following keys given the data above.

Calculated Keys

Next we locally sort the data based on the hashed keys, yielding the following. Note that since we know these keys are in the range of `[0, NUM_GPUS)`, we can restrict the sorting to only a couple of bits.

Locally-sorted Data

Locally-sorted Keys

Next, we transfer all data with a key of 0 to device 0 and all data with a key of 1 to device 1. This becomes an all-to-all exchange among GPUs, producing the following.

Exchanged Data

Multi-GPU sorting is a simple extension of the above. After exchanging the data, we simply sort the data again locally to produce the final sorted list.

Fully-sorted Data

Here we can see the final list has been completely sorted. For many applications, the initial data exchange is all that is needed. Thus we will model both the exchange and the final sort separately.

Performance Considerations

The exchange algorithm can be easily pipelined by operating on small portions of the domain. The first step of the pipeline is to perform the local sort on the generated keys; the second is to perform the all-to-all exchange. This allows the initial sorting and the exchange to be completely overlapped. For the full multi-GPU sorting algorithm, the final local sort cannot begin until all data has arrived, so this portion cannot be pipelined with the other two.

If we assume the original data is randomly distributed among GPUs following a uniform random distribution, then each GPU must communicate `W*N/P` bytes of data to every other GPU, where W is the size of each element in bytes, N is the number of elements per GPU, and P is the number of GPUs.

In a PCIe tree topology, the communication is limited by the bisection bandwidth. Each GPU must communicate half of its data (N/2) across the top link in the system, and there are P/2 devices all trying to communicate across the same PCIe lanes in the same direction.

On an NVLink system, however, the communication can occur in parallel, because there are dedicated links between all pairs of GPUs.

NVIDIA NVLink High-Speed Interconnect: Application Performance p. 4-7, https://info.nvidianews.com/rs/nvidia/images/NVIDIA%20NVLink%20High-Speed%20Interconnect%20Application%20Performance%20Brief.pdf.

To address this issue, Tesla P100 features NVIDIA's new high-speed interface, NVLink, that provides GPU-to-GPU data transfers at up to 160 Gigabytes/second of bidirectional bandwidth—5x the bandwidth of PCIe Gen 3 x16. Figure 4 shows NVLink connecting eight Tesla P100 Accelerators in a Hybrid Cube Mesh Topology.

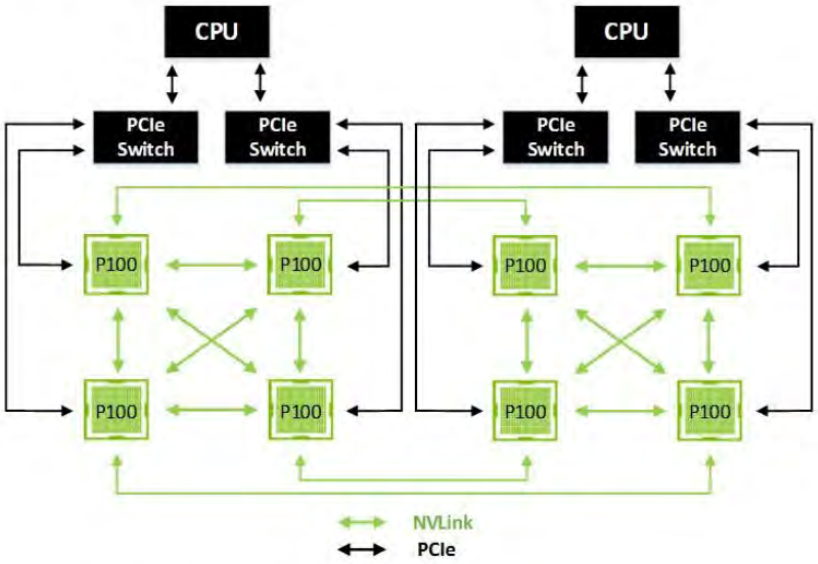
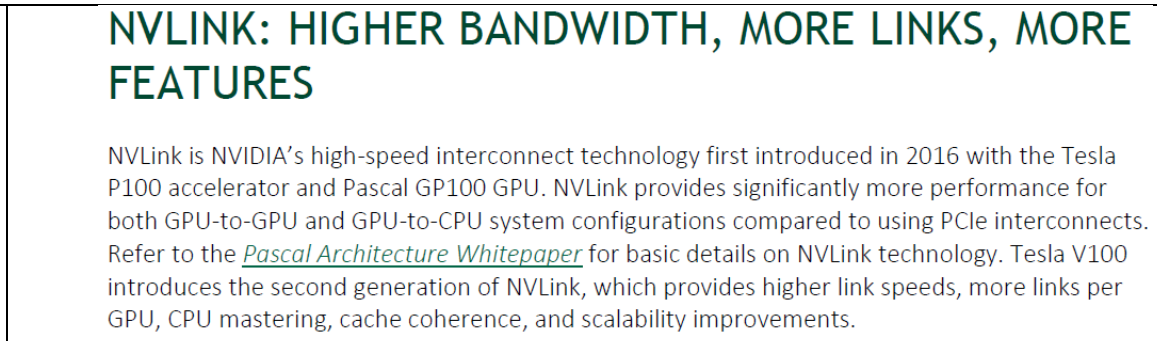
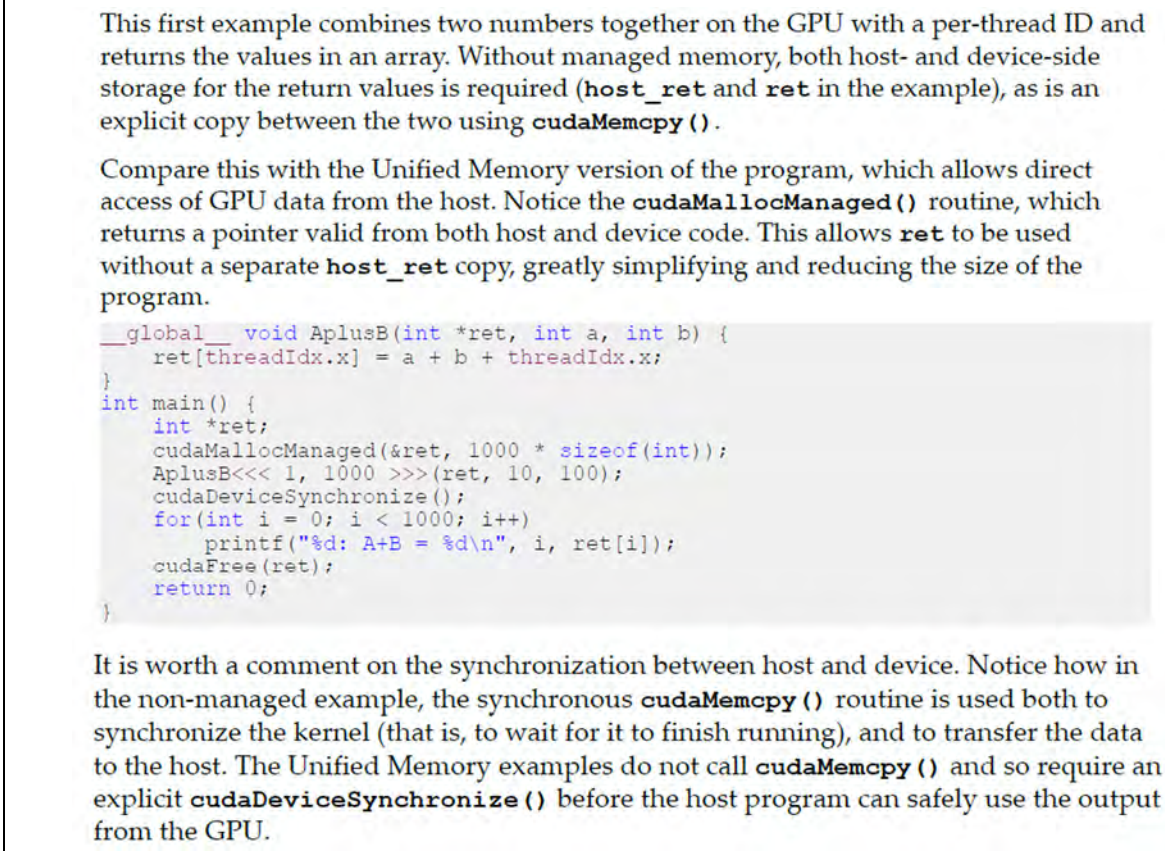


Figure 4. NVLink Connecting Eight Tesla P100 Accelerators in a Hybrid Cube Mesh Topology

NVIDIA Tesla P100: The Most Advanced Datacenter Accelerator Ever Built Featuring Pascal GP100, the World's Fastest GPU p. 7, https://images.nvidia.com/content/pdf/tesla/whitepaper/pascal-architecture-whitepaper.pdf.

	 NVidia Tesla V100 GPU Architecture: The World's Most Advanced Data Center GPU, p. 19, https://images.nvidia.com/content/volta-architecture/pdf/volta-architecture-whitepaper.pdf.
wherein the user connection interface is configured to return at least one result of mathematical expression evaluation to a user interface or a script; and	The user connection interface of each Accused GPU Product and Accused Server Product is configured to return at least one result of mathematical expression evaluation to a user interface or a script. For example, each Accused GPU Product and the GPU Accelerators of each Accused Server Product are configured to return a result to a user interface.  NVIDIA CUDA C Programming Guide pp. 281-82 (v10.1 May 2019), https://docs.nvidia.com/pdf/CUDA_C_Programming_Guide.pdf
wherein the computer cluster node is configured to: accept user instructions;	Each Accused GPU Product and Accused Server Product is configured to accept user instructions. For example, each Accused GPU Product and the GPU Accelerators of each Accused Server Product is capable of implementing CUDA code received from a CPU.

2.4. Heterogeneous Programming

As illustrated by Figure 8, the CUDA programming model assumes that the CUDA threads execute on a physically separate *device* that operates as a coprocessor to the *host* running the C program. This is the case, for example, when the kernels execute on a GPU and the rest of the C program executes on a CPU.

A GPU is built around an array of Streaming Multiprocessors (SMs) (see Hardware Implementation for more details). A multithreaded program is partitioned into blocks of threads that execute independently from each other, so that a GPU with more multiprocessors will automatically execute the program in less time than a GPU with fewer multiprocessors.

Figure 5 Automatic Scalability

NVidia CUDA C Programming Guide pp. 5, 12 (v10.1 May 2019), https://docs.nvidia.com/pdf/CUDA_C_Programming_Guide.pdf.

Server Configuration with NVLink

In the following sections of this paper, we analyze the performance benefit of NVLink for several algorithms and applications by comparing model systems based on PCIe-interconnected next-gen GPUs to otherwise-identical systems with NVLink-interconnected GPUs. GPUs are connected to the CPU using existing PCIe connections, but the NVLink configurations augment this with interconnections among the GPUs for peer-to-peer communication. The following analyses assume future-generation GPUs with performance higher than that of today's GPUs, so as to better correspond with the GPUs that will be contemporary with NVLink.

The second scenario, shown in Figure 3, compares a pair of configurations with four GPUs each, referred to as *4-GPU-PCIe* and *4-GPU-NVLink*. 4-GPU-PCIe uses a PCIe tree topology, again mirroring a configuration used commonly today. 4-GPU-NVLink enhances this by providing point-to-point connections with either one or two NVLink connections per pair of GPUs, yielding 16 GB/s or 32 GB/s effective bandwidth per direction per pair, respectively.

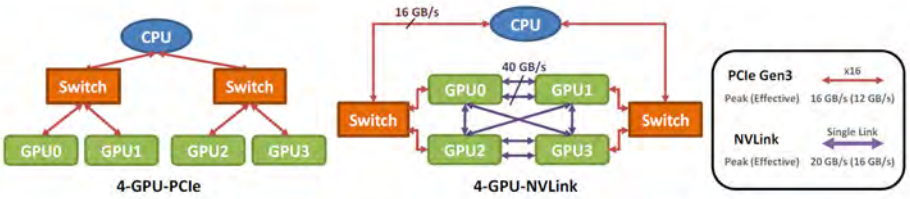


Figure 3: Comparing 4-GPU topologies with NVLink and PCIe. In 4-GPU-NVLink, GPU0 and GPU1 have 40 GB/s peak bandwidth between them, as do GPU2 and GPU3. The other peer-to-peer connections have 20 GB/s peak bandwidth.

The exchange algorithm begins with data evenly distributed across multiple GPUs as shown below.

Original Data

8	3	7	4	6	1	9	2	7	5	3	8	2	0	1	9

In this diagram, the color indicates which device the original data began on.

The first step is to hash the data to determine which GPU's bin each element should be placed in. For sorting, this might be as simple as extracting the highest order bits, but in theory we could apply any arbitrary hashing function to determine the keys (i.e., the destination bins). For this example, we will use the hash function `ceil(x/5)`, which produces the following keys given the data above.

Calculated Keys

1	0	1	0	1	0	1	0	1	1	0	1	0	0	0	1

Next we locally sort the data based on the hashed keys, yielding the following. Note that since we know these keys are in the range of `[0, NUM_GPUS)`, we can restrict the sorting to only a couple of bits.

Locally-sorted Data

3	4	1	2	8	7	6	9	3	2	0	1	7	5	8	9

Locally-sorted Keys

0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1

Next, we transfer all data with a key of 0 to device 0 and all data with a key of 1 to device 1. This becomes an all-to-all exchange among GPUs, producing the following.

Exchanged Data

3	4	1	2	3	2	0	1	8	7	6	9	7	5	8	9

Multi-GPU sorting is a simple extension of the above. After exchanging the data, we simply sort the data again locally to produce the final sorted list.

Fully-sorted Data

0	1	1	2	2	3	3	4	5	6	7	7	8	8	9	9

Here we can see the final list has been completely sorted. For many applications, the initial data exchange is all that is needed. Thus we will model both the exchange and the final sort separately.

Performance Considerations

The exchange algorithm can be easily pipelined by operating on small portions of the domain. The first step of the pipeline is to perform the local sort on the generated keys; the second is to perform the all-to-all exchange. This allows the initial sorting and the exchange to be completely overlapped. For the full multi-GPU sorting algorithm, the final local sort cannot begin until all data has arrived, so this portion cannot be pipelined with the other two.

If we assume the original data is randomly distributed among GPUs following a uniform random distribution, then each GPU must communicate `W*N/P` bytes of data to every other GPU, where W is the size of each element in bytes, N is the number of elements per GPU, and P is the number of GPUs.

In a PCIe tree topology, the communication is limited by the bisection bandwidth. Each GPU must communicate half of its data (N/2) across the top link in the system, and there are P/2 devices all trying to communicate across the same PCIe lanes in the same direction.

On an NVLink system, however, the communication can occur in parallel, because there are dedicated links between all pairs of GPUs.

NVIDIA NVLink High-Speed Interconnect: Application Performance p. 4-7, https://info.nvidianews.com/rs/nvidia/images/NVIDIA%20NVLink%20High-Speed%20Interconnect%20Application%20Performance%20Brief.pdf.

To address this issue, Tesla P100 features NVIDIA's new high-speed interface, NVLink, that provides GPU-to-GPU data transfers at up to 160 Gigabytes/second of bidirectional bandwidth—5x the bandwidth of PCIe Gen 3 x16. Figure 4 shows NVLink connecting eight Tesla P100 Accelerators in a Hybrid Cube Mesh Topology.

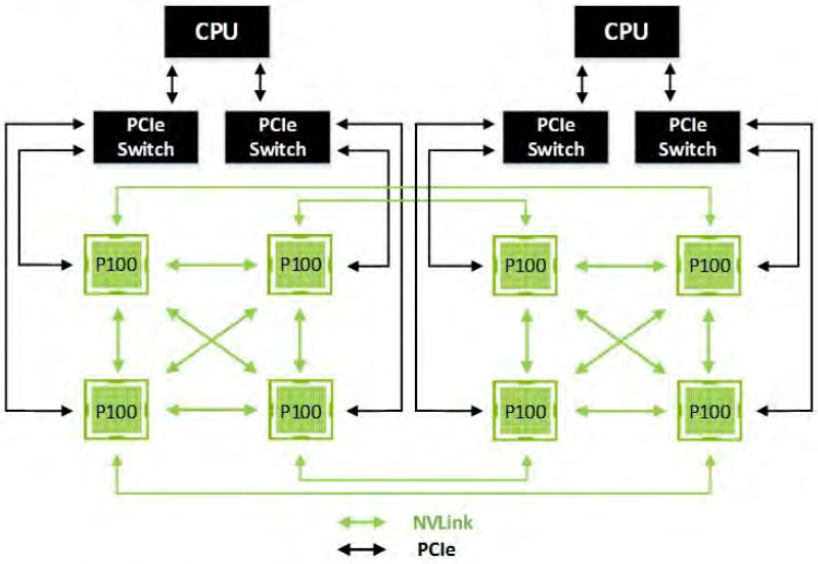


Figure 4. NVLink Connecting Eight Tesla P100 Accelerators in a Hybrid Cube Mesh Topology

NVIDIA Tesla P100: The Most Advanced Datacenter Accelerator Ever Built Featuring Pascal GP100, the World's Fastest GPU p. 7, https://images.nvidia.com/content/pdf/tesla/whitepaper/pascal-architecture-whitepaper.pdf.

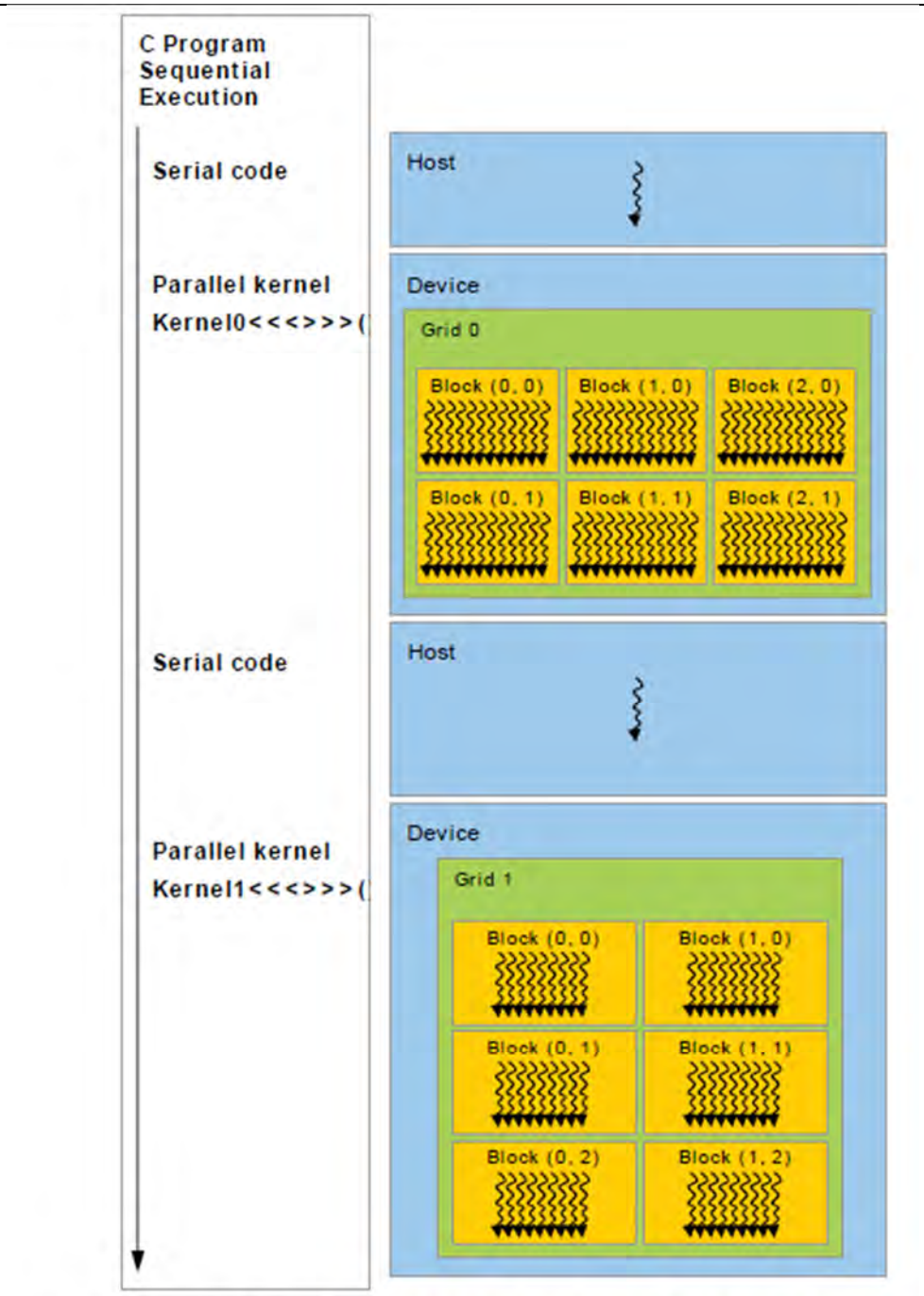
NVLINK: HIGHER BANDWIDTH, MORE LINKS, MORE FEATURES

NVLink is NVIDIA's high-speed interconnect technology first introduced in 2016 with the Tesla P100 accelerator and Pascal GP100 GPU. NVLink provides significantly more performance for both GPU-to-GPU and GPU-to-CPU system configurations compared to using PCIe interconnects. Refer to the *Pascal Architecture Whitepaper* for basic details on NVLink technology. Tesla V100 introduces the second generation of NVLink, which provides higher link speeds, more links per GPU, CPU mastering, cache coherence, and scalability improvements.

NVidia Tesla V100 GPU Architecture: The World's Most Advanced Data Center GPU, p. 19, https://images.nvidia.com/content/volta-architecture/pdf/volta-architecture-whitepaper.pdf.

2.4. Heterogeneous Programming

As illustrated by Figure 8, the CUDA programming model assumes that the CUDA threads execute on a physically separate *device* that operates as a coprocessor to the *host* running the C program. This is the case, for example, when the kernels execute on a GPU and the rest of the C program executes on a CPU.

<table>
<tr><td></td><td> 

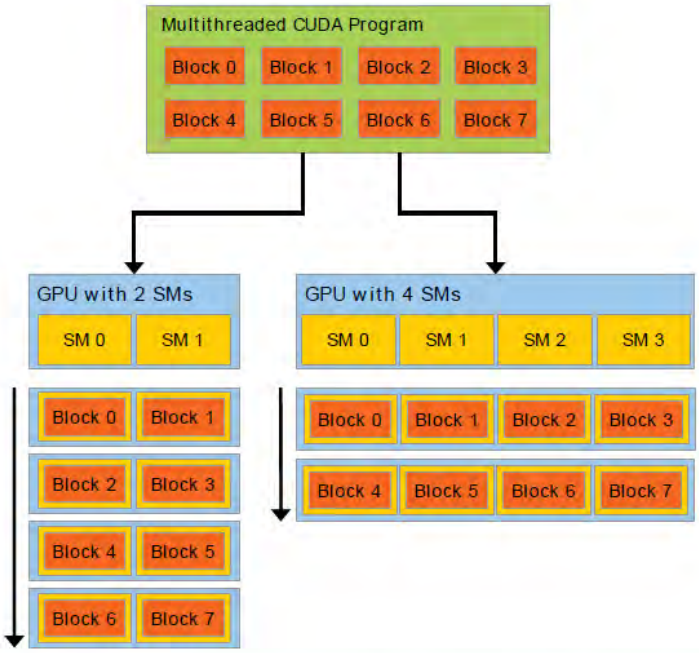
Serial code executes on the host while parallel code executes on the device.
Figure 8 Heterogeneous Programming
NVidia CUDA C Programming Guide pp. 12-13 (v10.1 May 2019), https://docs.nvidia.com/pdf/CUDA_C_Programming_Guide.pdf.</td></tr>
<tr><td>after accepting user instructions, communicate at least some of the user instructions using the</td><td>Each Accused GPU Product and Accused Server Product is configured to, after accepting user instructions, communicate at least some of the user instructions using the mechanism for the nodes to communicate with each other.

For example, each Accused GPU Product and the GPU Accelerators of each Accused Server Product are capable of implementing CUDA code through executing Kernels and engaging in peer-to-peer communication to efficiently solve mathematical expressions and complete all threads generated by the CUDA code.</td></tr>
</table>

mechanism for the nodes to communicate with each other; and

2.4. Heterogeneous Programming

As illustrated by Figure 8, the CUDA programming model assumes that the CUDA threads execute on a physically separate *device* that operates as a coprocessor to the *host* running the C program. This is the case, for example, when the kernels execute on a GPU and the rest of the C program executes on a CPU.



A GPU is built around an array of Streaming Multiprocessors (SMs) (see Hardware Implementation for more details). A multithreaded program is partitioned into blocks of threads that execute independently from each other, so that a GPU with more multiprocessors will automatically execute the program in less time than a GPU with fewer multiprocessors.

Figure 5 Automatic Scalability

NVidia CUDA C Programming Guide pp. 5, 12 (v10.1 May 2019), https://docs.nvidia.com/pdf/CUDA_C_Programming_Guide.pdf.

Server Configuration with NVLink

In the following sections of this paper, we analyze the performance benefit of NVLink for several algorithms and applications by comparing model systems based on PCIe-interconnected next-gen GPUs to otherwise-identical systems with NVLink-interconnected GPUs. GPUs are connected to the CPU using existing PCIe connections, but the NVLink configurations augment this with interconnections among the GPUs for peer-to-peer communication. The following analyses assume future-generation GPUs with performance higher than that of today's GPUs, so as to better correspond with the GPUs that will be contemporary with NVLink.

The second scenario, shown in Figure 3, compares a pair of configurations with four GPUs each, referred to as *4-GPU-PCIe* and *4-GPU-NVLink*. 4-GPU-PCIe uses a PCIe tree topology, again mirroring a configuration used commonly today. 4-GPU-NVLink enhances this by providing point-to-point connections with either one or two NVLink connections per pair of GPUs, yielding 16 GB/s or 32 GB/s effective bandwidth per direction per pair, respectively.

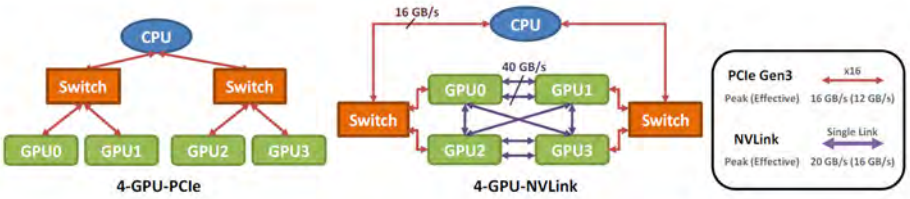


Figure 3: Comparing 4-GPU topologies with NVLink and PCIe. In 4-GPU-NVLink, GPU0 and GPU1 have 40 GB/s peak bandwidth between them, as do GPU2 and GPU3. The other peer-to-peer connections have 20 GB/s peak bandwidth.

The exchange algorithm begins with data evenly distributed across multiple GPUs as shown below.

Original Data

8	3	7	4	6	1	9	2	7	5	3	8	2	0	1	9

In this diagram, the color indicates which device the original data began on.

The first step is to hash the data to determine which GPU's bin each element should be placed in. For sorting, this might be as simple as extracting the highest order bits, but in theory we could apply any arbitrary hashing function to determine the keys (i.e., the destination bins). For this example, we will use the hash function `ceil(x/5)`, which produces the following keys given the data above.

Calculated Keys

1	0	1	0	1	0	1	0	1	1	0	1	0	0	0	1

Next we locally sort the data based on the hashed keys, yielding the following. Note that since we know these keys are in the range of `[0, NUM_GPUS)`, we can restrict the sorting to only a couple of bits.

Locally-sorted Data

3	4	1	2	8	7	6	9	3	2	0	1	7	5	8	9

Locally-sorted Keys

0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1

Next, we transfer all data with a key of 0 to device 0 and all data with a key of 1 to device 1. This becomes an all-to-all exchange among GPUs, producing the following.

Exchanged Data

3	4	1	2	3	2	0	1	8	7	6	9	7	5	8	9

Multi-GPU sorting is a simple extension of the above. After exchanging the data, we simply sort the data again locally to produce the final sorted list.

Fully-sorted Data

0	1	1	2	2	3	3	4	5	6	7	7	8	8	9	9

Here we can see the final list has been completely sorted. For many applications, the initial data exchange is all that is needed. Thus we will model both the exchange and the final sort separately.

Performance Considerations

The exchange algorithm can be easily pipelined by operating on small portions of the domain. The first step of the pipeline is to perform the local sort on the generated keys; the second is to perform the all-to-all exchange. This allows the initial sorting and the exchange to be completely overlapped. For the full multi-GPU sorting algorithm, the final local sort cannot begin until all data has arrived, so this portion cannot be pipelined with the other two.

If we assume the original data is randomly distributed among GPUs following a uniform random distribution, then each GPU must communicate `W*N/P` bytes of data to every other GPU, where W is the size of each element in bytes, N is the number of elements per GPU, and P is the number of GPUs.

In a PCIe tree topology, the communication is limited by the bisection bandwidth. Each GPU must communicate half of its data (N/2) across the top link in the system, and there are P/2 devices all trying to communicate across the same PCIe lanes in the same direction.

On an NVLink system, however, the communication can occur in parallel, because there are dedicated links between all pairs of GPUs.

NVIDIA NVLink High-Speed Interconnect: Application Performance p. 4-7, https://info.nvidianews.com/rs/nvidia/images/NVIDIA%20NVLink%20High-Speed%20Interconnect%20Application%20Performance%20Brief.pdf.

To address this issue, Tesla P100 features NVIDIA's new high-speed interface, NVLink, that provides GPU-to-GPU data transfers at up to 160 Gigabytes/second of bidirectional bandwidth—5x the bandwidth of PCIe Gen 3 x16. Figure 4 shows NVLink connecting eight Tesla P100 Accelerators in a Hybrid Cube Mesh Topology.

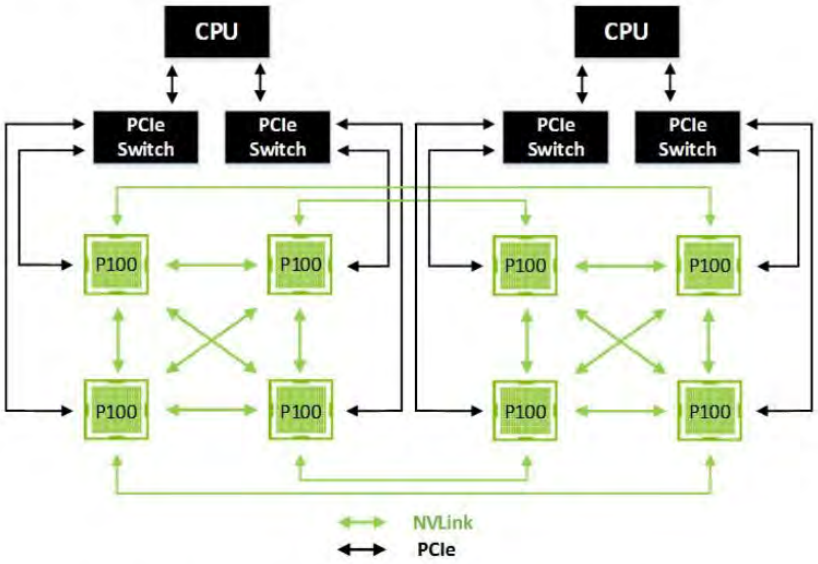


Figure 4. NVLink Connecting Eight Tesla P100 Accelerators in a Hybrid Cube Mesh Topology

NVIDIA Tesla P100: The Most Advanced Datacenter Accelerator Ever Built Featuring Pascal GP100, the World's Fastest GPU p. 7, https://images.nvidia.com/content/pdf/tesla/whitepaper/pascal-architecture-whitepaper.pdf.

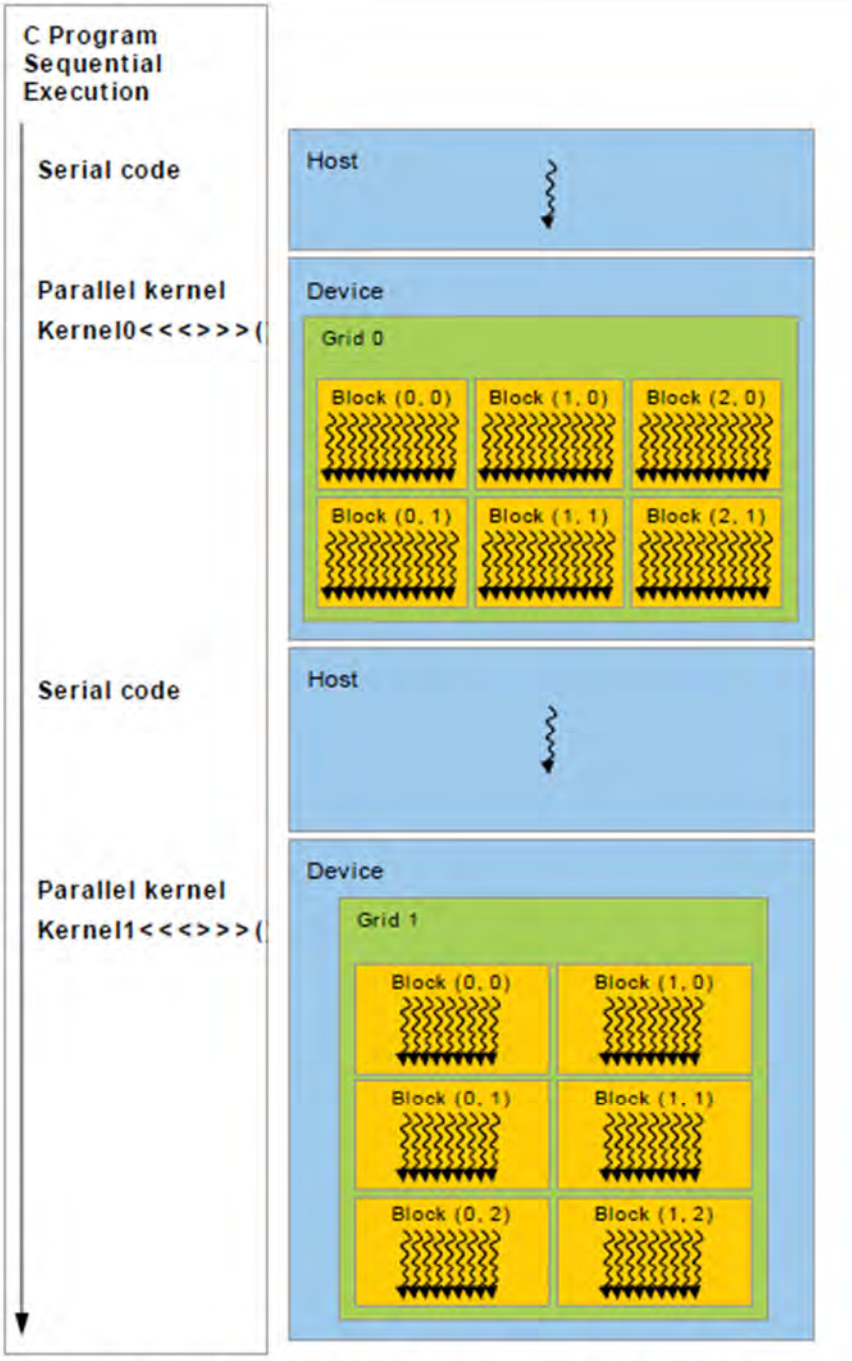
NVLINK: HIGHER BANDWIDTH, MORE LINKS, MORE FEATURES

NVLink is NVIDIA's high-speed interconnect technology first introduced in 2016 with the Tesla P100 accelerator and Pascal GP100 GPU. NVLink provides significantly more performance for both GPU-to-GPU and GPU-to-CPU system configurations compared to using PCIe interconnects. Refer to the *Pascal Architecture Whitepaper* for basic details on NVLink technology. Tesla V100 introduces the second generation of NVLink, which provides higher link speeds, more links per GPU, CPU mastering, cache coherence, and scalability improvements.

NVidia Tesla V100 GPU Architecture: The World's Most Advanced Data Center GPU, p. 19, https://images.nvidia.com/content/volta-architecture/pdf/volta-architecture-whitepaper.pdf.

2.4. Heterogeneous Programming

As illustrated by Figure 8, the CUDA programming model assumes that the CUDA threads execute on a physically separate *device* that operates as a coprocessor to the *host* running the C program. This is the case, for example, when the kernels execute on a GPU and the rest of the C program executes on a CPU.



Serial code executes on the host while parallel code executes on the device.

Figure 8 Heterogeneous Programming

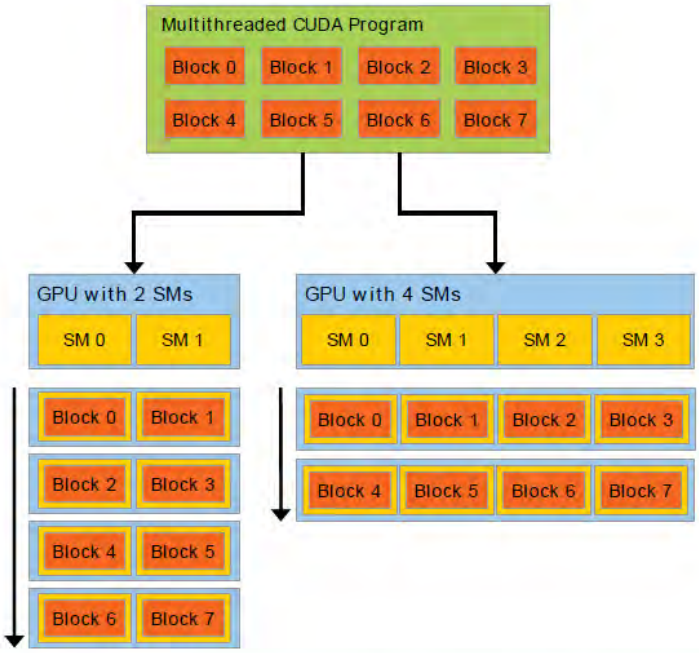
NVidia CUDA C Programming Guide pp. 12-13 (v10.1 May 2019), https://docs.nvidia.com/pdf/CUDA_C_Programming_Guide.pdf.

after communicating at least some of the user instructions using the mechanism, communicate	Each Accused GPU Product and Accused Server Product is configured to, after communicating at least some of the user instructions using the mechanism, communicate at least some of the user instructions to the single-node kernel. For example, each Accused GPU Product and the GPU Accelerators of each Accused Server Product are capable of implements CUDA code through executing Kernels and engaging in peer-to-peer communication to efficiently solve mathematical expressions and complete all threads generated by the CUDA code.

at least some of the user instructions to the single-node kernel.

2.4. Heterogeneous Programming

As illustrated by Figure 8, the CUDA programming model assumes that the CUDA threads execute on a physically separate *device* that operates as a coprocessor to the *host* running the C program. This is the case, for example, when the kernels execute on a GPU and the rest of the C program executes on a CPU.



A GPU is built around an array of Streaming Multiprocessors (SMs) (see Hardware Implementation for more details). A multithreaded program is partitioned into blocks of threads that execute independently from each other, so that a GPU with more multiprocessors will automatically execute the program in less time than a GPU with fewer multiprocessors.

Figure 5 Automatic Scalability

NVidia CUDA C Programming Guide pp. 5, 12 (v10.1 May 2019), https://docs.nvidia.com/pdf/CUDA_C_Programming_Guide.pdf.

Server Configuration with NVLink

In the following sections of this paper, we analyze the performance benefit of NVLink for several algorithms and applications by comparing model systems based on PCIe-interconnected next-gen GPUs to otherwise-identical systems with NVLink-interconnected GPUs. GPUs are connected to the CPU using existing PCIe connections, but the NVLink configurations augment this with interconnections among the GPUs for peer-to-peer communication. The following analyses assume future-generation GPUs with performance higher than that of today's GPUs, so as to better correspond with the GPUs that will be contemporary with NVLink.

The second scenario, shown in Figure 3, compares a pair of configurations with four GPUs each, referred to as *4-GPU-PCIe* and *4-GPU-NVLink*. 4-GPU-PCIe uses a PCIe tree topology, again mirroring a configuration used commonly today. 4-GPU-NVLink enhances this by providing point-to-point connections with either one or two NVLink connections per pair of GPUs, yielding 16 GB/s or 32 GB/s effective bandwidth per direction per pair, respectively.

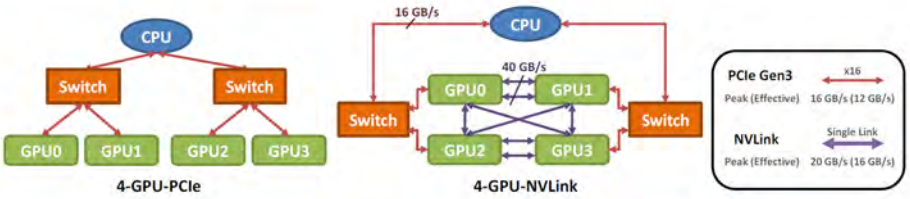


Figure 3: Comparing 4-GPU topologies with NVLink and PCIe. In 4-GPU-NVLink, GPU0 and GPU1 have 40 GB/s peak bandwidth between them, as do GPU2 and GPU3. The other peer-to-peer connections have 20 GB/s peak bandwidth.

The exchange algorithm begins with data evenly distributed across multiple GPUs as shown below.

Original Data

8	3	7	4	6	1	9	2	7	5	3	8	2	0	1	9

In this diagram, the color indicates which device the original data began on.

The first step is to hash the data to determine which GPU's bin each element should be placed in. For sorting, this might be as simple as extracting the highest order bits, but in theory we could apply any arbitrary hashing function to determine the keys (i.e., the destination bins). For this example, we will use the hash function `ceil(x/5)`, which produces the following keys given the data above.

Calculated Keys

1	0	1	0	1	0	1	0	1	1	0	1	0	0	0	1

Next we locally sort the data based on the hashed keys, yielding the following. Note that since we know these keys are in the range of `[0, NUM_GPUS)`, we can restrict the sorting to only a couple of bits.

Locally-sorted Data

3	4	1	2	8	7	6	9	3	2	0	1	7	5	8	9

Locally-sorted Keys

0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1

Next, we transfer all data with a key of 0 to device 0 and all data with a key of 1 to device 1. This becomes an all-to-all exchange among GPUs, producing the following.

Exchanged Data

3	4	1	2	3	2	0	1	8	7	6	9	7	5	8	9

Multi-GPU sorting is a simple extension of the above. After exchanging the data, we simply sort the data again locally to produce the final sorted list.

Fully-sorted Data

0	1	1	2	2	3	3	4	5	6	7	7	8	8	9	9

Here we can see the final list has been completely sorted. For many applications, the initial data exchange is all that is needed. Thus we will model both the exchange and the final sort separately.

Performance Considerations

The exchange algorithm can be easily pipelined by operating on small portions of the domain. The first step of the pipeline is to perform the local sort on the generated keys; the second is to perform the all-to-all exchange. This allows the initial sorting and the exchange to be completely overlapped. For the full multi-GPU sorting algorithm, the final local sort cannot begin until all data has arrived, so this portion cannot be pipelined with the other two.

If we assume the original data is randomly distributed among GPUs following a uniform random distribution, then each GPU must communicate `W*N/P` bytes of data to every other GPU, where W is the size of each element in bytes, N is the number of elements per GPU, and P is the number of GPUs.

In a PCIe tree topology, the communication is limited by the bisection bandwidth. Each GPU must communicate half of its data (N/2) across the top link in the system, and there are P/2 devices all trying to communicate across the same PCIe lanes in the same direction.

On an NVLink system, however, the communication can occur in parallel, because there are dedicated links between all pairs of GPUs.

NVIDIA NVLink High-Speed Interconnect: Application Performance p. 4-7, https://info.nvidianews.com/rs/nvidia/images/NVIDIA%20NVLink%20High-Speed%20Interconnect%20Application%20Performance%20Brief.pdf.

To address this issue, Tesla P100 features NVIDIA's new high-speed interface, NVLink, that provides GPU-to-GPU data transfers at up to 160 Gigabytes/second of bidirectional bandwidth—5x the bandwidth of PCIe Gen 3 x16. Figure 4 shows NVLink connecting eight Tesla P100 Accelerators in a Hybrid Cube Mesh Topology.

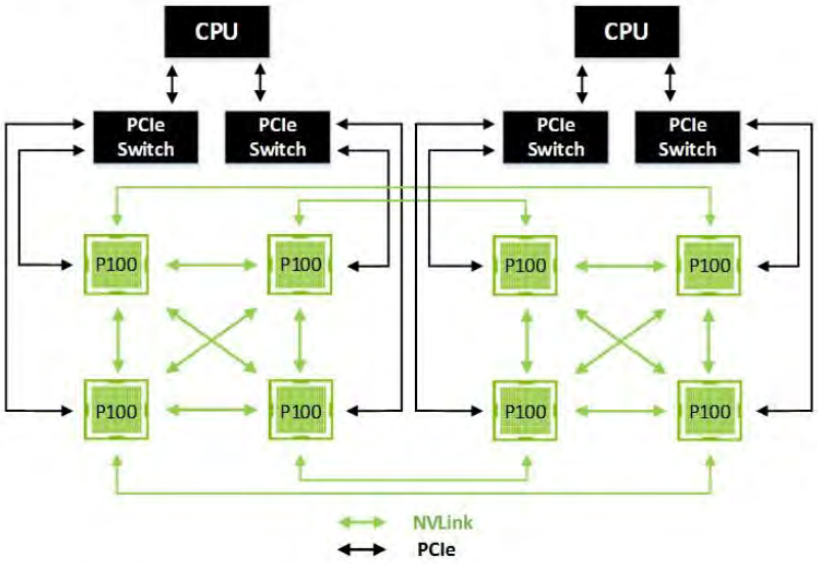


Figure 4. NVLink Connecting Eight Tesla P100 Accelerators in a Hybrid Cube Mesh Topology

NVIDIA Tesla P100: The Most Advanced Datacenter Accelerator Ever Built Featuring Pascal GP100, the World's Fastest GPU p. 7, https://images.nvidia.com/content/pdf/tesla/whitepaper/pascal-architecture-whitepaper.pdf.

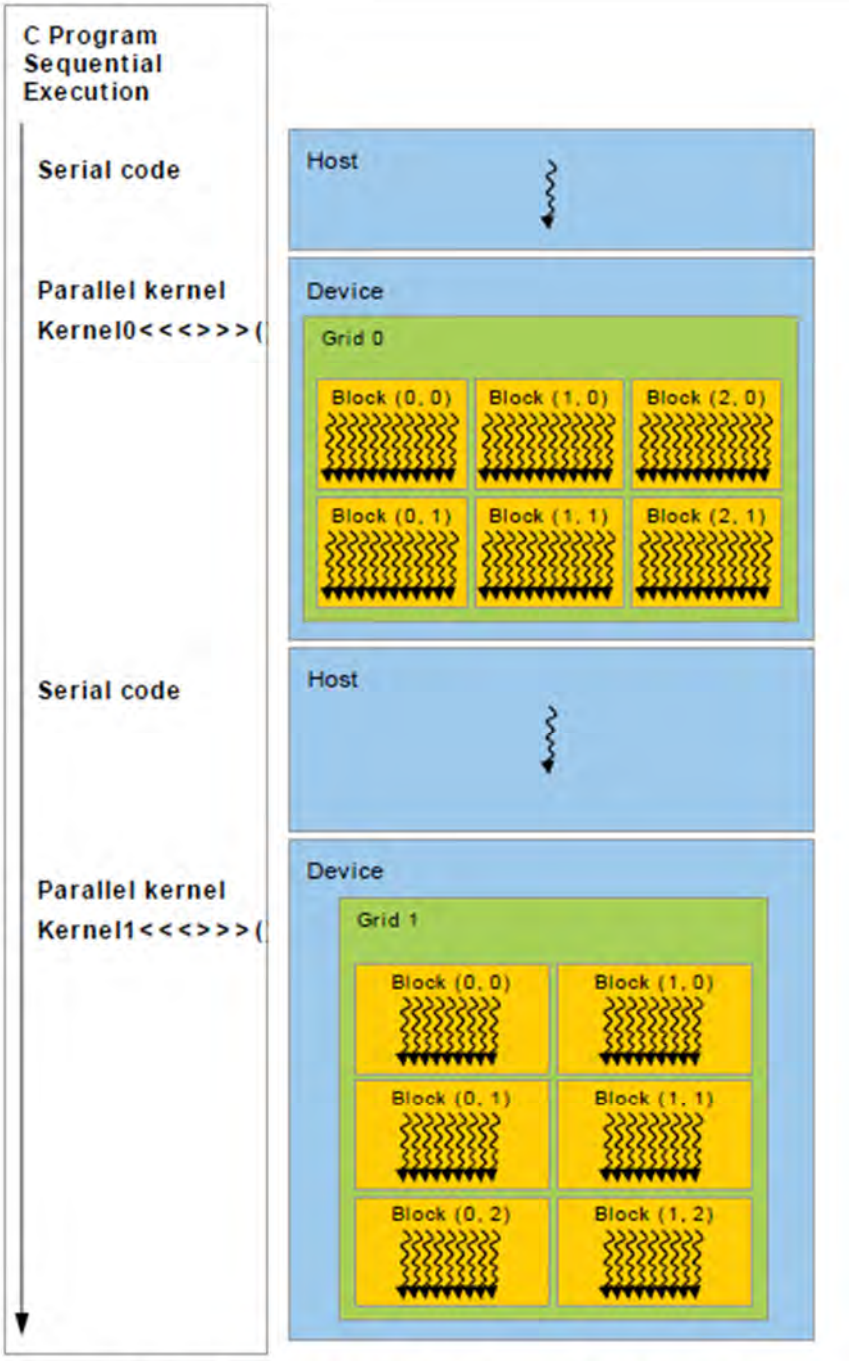
NVLINK: HIGHER BANDWIDTH, MORE LINKS, MORE FEATURES

NVLink is NVIDIA's high-speed interconnect technology first introduced in 2016 with the Tesla P100 accelerator and Pascal GP100 GPU. NVLink provides significantly more performance for both GPU-to-GPU and GPU-to-CPU system configurations compared to using PCIe interconnects. Refer to the *Pascal Architecture Whitepaper* for basic details on NVLink technology. Tesla V100 introduces the second generation of NVLink, which provides higher link speeds, more links per GPU, CPU mastering, cache coherence, and scalability improvements.

NVidia Tesla V100 GPU Architecture: The World's Most Advanced Data Center GPU, p. 19, https://images.nvidia.com/content/volta-architecture/pdf/volta-architecture-whitepaper.pdf.

2.4. Heterogeneous Programming

As illustrated by Figure 8, the CUDA programming model assumes that the CUDA threads execute on a physically separate *device* that operates as a coprocessor to the *host* running the C program. This is the case, for example, when the kernels execute on a GPU and the rest of the C program executes on a CPU.



Serial code executes on the host while parallel code executes on the device.

Figure 8 Heterogeneous Programming

NVidia CUDA C Programming Guide pp. 12-13 (v10.1 May 2019), https://docs.nvidia.com/pdf/CUDA_C_Programming_Guide.pdf.

31189898

EXHIBIT I

vdatetime	bol_id	bol_nbr	raw_cmd_desc	name	fname	mtons	kilos	est_value
1/20/2016 0:00	350858153	DO6911221868	COMPUTER GRAPHICS CARD	NVIDIA	NVIDIA SINGAPORE	1.28	1282	9006
6/12/2016 0:00	357523694	DO6911239739	COMPUTER GRAPHIC CARD	NVIDIA	NVIDIA SINGAPORE	1.38	1389	9758
12/11/2016 0:00	371177934	DO6911267146	COMPUTER GRAPHICS CARDS	NVIDIA	NVIDIA SINGAPORE	3.59	3596	25262
2/21/2017 0:00	373591079	DO6911277658	COMPUTER GRAPHICS CARDS	NVIDIA	NVIDIA SINGAPORE	3.84	3847	27025
5/8/2017 0:00	378457866	DO6911285643	COMPUTER GRAPHICS CARDS	NVIDIA	NVIDIA SINGAPORE	3.95	3950	27749

EXHIBIT J

a81a120d7db70422b5de2a3738b704cb270b1e5ae3d2ff32fa396e0498356fe4

NVIDIA SINGAPORE PTE LTD

(Incorporated in Singapore. Registration Number 200003831M)

ANNUAL REPORT

For the financial year ended 25 January 2015

NVIDIA SINGAPORE PTE LTD
(Incorporated in Singapore)

ANNUAL REPORT
For the financial year ended 25 January 2015

Contents

NVIDIA SINGAPORE PTE LTD

DIRECTORS' REPORT
For the financial year ended 25 January 2015

The directors present their report to the shareholder together with the audited financial statements for the financial year ended 25 January 2015.

Directors

The directors in office at the date of this report are as follows:

Lee Kay Beng
Karen Theresa Burns
Michael John Byron

Arrangements to enable directors to acquire shares and debentures

Neither at the end of nor at any time during the financial year was the Company a party to any arrangement whose object was to enable the directors of the Company to acquire benefits by means of the acquisition of shares in, or debentures of, the Company or any other body corporate.

Directors' interests in shares and debentures

None of the directors of the Company holding office at the end of the financial year has any interest in the shares or debentures of the Company or any related corporations.

Directors' contractual benefits

Since the end of the previous financial year, no director has received or become entitled to receive a benefit by reason of a contract made by the Company or a related corporation with the director or with a firm of which he is a member or with a company in which he has a substantial financial interest, except that certain directors received remuneration as a result of their employment with related corporations.

Share options

No options were granted during the financial year to subscribe for unissued shares of the Company.

No shares were issued during the financial year by virtue of the exercise of options to take up unissued shares of the Company.

There were no unissued shares of the Company under option at the end of the financial year.

d914130d7db7d422b5de2a3738b706cb27ddaeaa3d3ff331a398e0438356fe4

NVIDIA SINGAPORE PTE LTD

DIRECTORS' REPORT
For the financial year ended 25 January 2015

Independent auditor

The independent auditor, PricewaterhouseCoopers LLP, has expressed its willingness to accept re-appointment.

On behalf of the directors

MICHAEL JOHN BYRON
Director

KAREN THERESA BURNS
Director

25 JUN 2015

NVIDIA SINGAPORE PTE LTD

STATEMENT BY DIRECTORS
For the financial year ended 25 January 2015

In the opinion of the directors,

(a) the financial statements as set out on pages 6 to 27 are drawn up so as to give a true and fair view of the state of affairs of the Company for the financial year ended 25 January 2015 and of the results of the business, changes in equity and cash flows of the Company for the financial year then ended; and

(b) at the date of this statement, there are reasonable grounds to believe that the Company will be able to pay its debts as and when they fall due.

On behalf of the directors

MICHAEL JOHN BYRON
Director

KAREN THERESA BURNS
Director

25 JUN 2015

INDEPENDENT AUDITOR'S REPORT TO THE SHAREHOLDER OF NVIDIA SINGAPORE PTE LTD

Report on the Financial Statements

We have audited the accompanying financial statements of Nvidia Singapore Pte Ltd set out on pages 6 to 27, which comprise the balance sheet as at 25 January 2015, the statement of comprehensive income, statement of changes in equity and the statement of cash flows For the financial year ended 25 January 2015, and a summary of significant accounting policies and other explanatory information.

Management's Responsibility for the Financial Statements

Management is responsible for the preparation of financial statements that give a true and fair view in accordance with the provisions of the Singapore Companies Act (the "Act") and Singapore Financial Reporting Standards, and for devising and maintaining a system of internal accounting controls sufficient to provide a reasonable assurance that assets are safeguarded against loss from unauthorised use or disposition; and transactions are properly authorised and that they are recorded as necessary to permit the preparation of true and fair profit and loss accounts and balance sheets and to maintain accountability of assets.

Auditor's Responsibility

Our responsibility is to express an opinion on these financial statements based on our audit. We conducted our audit in accordance with Singapore Standards on Auditing. Those standards require that we comply with ethical requirements and plan and perform the audit to obtain reasonable assurance about whether the financial statements are free from material misstatement.

An audit involves performing procedures to obtain audit evidence about the amounts and disclosures in the financial statements. The procedures selected depend on the auditor's judgement, including the assessment of the risks of material misstatement of the financial statements, whether due to fraud or error. In making those risk assessments, the auditor considers internal control relevant to the entity's preparation of financial statements that give a true and fair view in order to design audit procedures that are appropriate in the circumstances, but not for the purpose of expressing an opinion on the effectiveness of the entity's internal control. An audit also includes evaluating the appropriateness of accounting policies used and the reasonableness of accounting estimates made by management, as well as evaluating the overall presentation of the financial statements.

We believe that the audit evidence we have obtained is sufficient and appropriate to provide a basis for our audit opinion.

INDEPENDENT AUDITOR'S REPORT TO THE SHAREHOLDER OF NVIDIA SINGAPORE PTE LTD (continued)

Basis for Qualified Opinion

As described in Note 19 to the financial statements, no expense has been recognised in the statement of comprehensive income in respect of the fair value of share options granted by the ultimate holding corporation to the Company's employees. This is not in accordance with Financial Reporting Standard 102 – Share-based Payments ("FRS 102") which requires the Company to measure and recognise the fair value of share options granted by the ultimate holding corporation to its employees in the statement of comprehensive income with a corresponding increase recognised in equity as a contribution from the ultimate holding corporation. In the absence of a valuation by management of these share options granted, we are unable to perform any other audit procedures to estimate the effect on the financial statements of this departure from FRS 102.

Qualified Opinion

In our opinion, except for the possible effects of the matter described in the Basis for Qualified Opinion paragraph, the financial statements of the Company are properly drawn up in accordance with the provisions of the Act and Singapore Financial Reporting Standards so as to give a true and fair view of the state of affairs of the Company as at 25 January 2015, and the results, changes in equity and cash flows of the Company for the financial year ended on that date.

Report on Other Legal and Regulatory Requirements

In our opinion, the accounting and other records required by the Act to be kept by the Company have been properly kept in accordance with the provisions of the Act.

PricewaterhouseCoopers LLP
Public Accountants and Chartered Accountants

Singapore,

25 JUN 2015

NVIDIA SINGAPORE PTE LTD

STATEMENT OF COMPREHENSIVE INCOME
For the financial year ended 25 January 2015

	Note	27/01/2014 to 25/01/2015 US$'000	28/01/2013 to 26/01/2014 US$'000
Revenue	3	**3,885,594**	3,461,212
Other income - net	3	**612**	527
Other gain - net	4	**19**	1,578
Expenses			
- Raw materials and other consumables consumed		**(3,715,120)**	(3,300,452)
- Employee compensation	6	**(6,250)**	(6,508)
- Travelling expenses		**(232)**	(588)
- Sales and marketing expenses		**(5,152)**	(6,541)
- Depreciation of plant and equipment		**(829)**	(765)
- Intercompany service expenses		**(147,245)**	(137,145)
- Rental on operating leases		**(404)**	(331)
- Other operating expenses	5	**(6,086)**	(4,600)
Total expenses		**(3,881,318)**	(3,456,930)
Profit before income tax		**4,907**	6,387
Income tax expense	7	**(1,282)**	(852)
Net profit after tax/total comprehensive income		**3,625**	5,535

The accompanying notes form an integral part of these financial statements.

NVIDIA SINGAPORE PTE LTD

BALANCE SHEET
As at 25 January 2015

	Note	25/01/2015 US$'000	26/01/2014 US$'000
ASSETS			
Current assets			
Cash and cash equivalents	8	**66,979**	198,035
Trade and other receivables	9	**711,098**	676,720
Inventories	10	**220,854**	191,712
Prepayment and deposits		**2,252**	3,856
		1,001,183	1,070,323
Non-current assets			
Plant and equipment	11	**744**	1,498
Investment in subsidiary	12	**85**	85
Loan to related party	13	**500**	500
		1,329	2,083
Total assets		**1,002,512**	1,072,406
LIABILITIES			
Current liabilities			
Trade and other payables	14	**968,867**	1,042,846
Current income tax liabilities	7	**746**	286
		969,613	1,043,132
Total liabilities		**969,613**	1,043,132
NET ASSETS		**32,899**	29,274
SHAREHOLDER'S EQUITY			
Share capital	15	**1**	1
Retained earnings		**32,898**	29,273
		32,899	29,274

The accompanying notes form an integral part of these financial statements.

03ea130d7db7d422b5de2a3732b704cb270b7e5aa3d2ff32fa398e0199356fe4

NVIDIA SINGAPORE PTE LTD

STATEMENT OF CHANGES IN EQUITY
For the financial year ended 25 January 2015

	Share capital US$'000	Retained earnings US$'000	Total equity US$'000
2015			
Beginning of financial year	1	29,273	29,274
Total comprehensive income	-	3,625	3,625
End of financial year	**1**	**32,898**	**32,899**
2014			
Beginning of financial year	1	23,738	23,739
Total comprehensive income	-	5,535	5,535
End of financial year	**1**	**29,273**	**29,274**

The accompanying notes form an integral part of these financial statements.

NVIDIA SINGAPORE PTE LTD

STATEMENT OF CASH FLOWS

For the financial year ended 25 January 2015

	Note	27/01/2014 to 25/01/2015 US$'000	28/01/2013 to 26/01/2014 US$'000
Cash flows from operating activities			
Net profit after tax		3,625	5,535
Adjustments for:			
- Tax		1,282	852
- Depreciation of plant and equipment		829	765
- Net loss on disposal of plant and equipment		-	2
- Interest income		(128)	(523)
Operating cash flow before working capital changes		5,608	6,631
Change in operating assets and liabilities			
- Trade and other receivables		(34,378)	286,510
- Inventories		(29,142)	29,377
- Trade and other payables		(73,979)	(321,784)
- Prepayments and deposits		1,604	(820)
Cash used in operations		(130,287)	(86)
Income tax paid		(822)	(1,206)
Net cash used in operating activities		(131,109)	(1,292)
Cash flows from investing activities			
Loan to related party		-	(500)
Purchases of plant and equipment		(75)	(818)
Proceeds from disposal of plant and equipment		-	7
Interest received		128	523
Net cash generated from/(used in) investing activities		53	(788)
Net decrease in cash and cash equivalents		(131,056)	(2,080)
Cash and cash equivalents at beginning of financial period		198,035	200,115
Cash and cash equivalents at end of financial period	8	66,979	198,035

The accompanying notes form an integral part of these financial statements.

NVIDIA SINGAPORE PTE LTD

NOTES TO THE FINANCIAL STATEMENTS
For the financial year ended 25 January 2015

These notes form an integral part of and should be read in conjunction with the accompanying financial statements.

1. General information

The Company is incorporated in Singapore. The address of its registered office is 112 Robinson Road, #05-01, Singapore 068902.

The principal activity of the Company consists of sales of graphics processors and media and communication devices.

The principal activities of its subsidiary are those relating to the solicitation and negotiation of orders and marketing on behalf of Nvidia Singapore Pte Ltd.

2. Significant accounting policies

(a) Basis of preparation

The financial statements have been prepared in accordance with Singapore Financial Reporting Standards ("FRS") under the historical cost convention, except as disclosed in the accounting policies below.

The preparation of these financial statements in conformity with FRS requires management to exercise its judgement in the process of applying the Company's accounting policies. It also requires the use of certain critical accounting estimates and assumptions. There are no significant areas involving a higher degree of judgement, estimates and assumption to these financial statements.

Interpretations and amendments to published standards effective in 2015

On 27 January 2014, the Company adopted the new or amended FRS and Interpretations to FRS ("INT FRS") that are mandatory for application for the financial year. Changes to the Company's accounting policies have been made as required, in accordance with the relevant transitional provisions in the respective FRS and INT FRS.

The adoption of the above new or revised FRS did not result in any substantial changes to the Company's accounting policies nor any significant impact on these financial statements.

2. Significant accounting policies (continued)

(b) Revenue recognition

Revenue comprises the fair value of the consideration received or receivable for the sales of goods in the ordinary course of the Company's activities. Revenue is presented net of goods and services tax, rebates and discounts. Revenue is recognised as follows:

(1) Sale of goods

Revenue from sale of graphics processors and related media and communication devices is recognised when the Company has delivered the products to the customer, the customer has accepted the products and collectability of the related receivables is reasonably assured.

A sales return allowance for estimated product returns is established at the time revenue is recognised, based primarily on historical return rates.

(2) Interest income

Interest income is recognised using the effective interest method.

(c) Group accounting

These financial statements are the separate financial statements of Nvidia Singapore Pte Ltd. The Company is exempted from the preparation of consolidated financial statements as the Company is a wholly-owned subsidiary of Nvidia Corporation, a US-incorporated company which produces consolidated financial statements available for public use. The basis on which the subsidiary is accounted for is disclosed in Note 2(f). The main office of Nvidia Corporation is as follows: 2701 San Tomas Expressway, Santa Clara, CA 95050, U.S.A..

(d) Employee compensation

(1) Defined contribution plans

Defined contribution plans are post-employment benefit plans under which the Company pays fixed contributions into separate entities such as Mandatory Provident Fund and Superannuation Fund established in Hong Kong, on a mandatory, contractual or voluntary basis. The Company has no further payment obligations once the contributions have been paid. The Company's contribution to defined contribution plans are recognised as employee compensation expense when they are due.

2. Significant accounting policies (continued)

(d) Employee compensation (continued)

(2) Employee leave entitlements

Employee entitlements to annual leave are recognised when they accrue to employees. A provision is made for the estimated liability for annual leave as a result of services rendered by employees up to the balance sheet date.

(e) Plant and equipment

(1) Measurement

Plant and equipment are initially recognised at cost and subsequently carried at cost less accumulated depreciation and accumulated impairment losses.

The cost of an item of plant and equipment initially recognised includes its purchase price and any cost that is directly attributable to bringing the asset to the location and condition necessary for it to be capable of operating in the manner intended by management. The projected cost of dismantlement, removal or restoration is recognised as part of the cost of plant and equipment if such obligation is incurred either when the item is acquired or as a consequence of using the asset during a particular period for purposes other than to produce inventories during that period.

(2) Depreciation

Depreciation on plant and equipment is calculated using the straight-line method to allocate their depreciable amounts over their estimated useful lives as follows:

	Useful lives
Machinery and equipment	3 to 5 years
Computer equipment	3 to 5 years
Office equipment	5 years
Furniture	5 years
Leasehold improvement	Lesser of estimated useful life or remaining lease term

The residual values, estimated useful lives and depreciation method of plant and equipment are reviewed, and adjusted as appropriate, at each balance sheet date. The effects of any revision are recognised in the statement of comprehensive income for the financial period in which the changes arise.

2. Significant accounting policies (continued)

(e) Plant and equipment (continued)

(3) Subsequent expenditure

Subsequent expenditure relating to plant and equipment that has already been recognised is added to the carrying amount of the asset only when it is probable that future economic benefits associated with the item, will flow to the Company and the cost can be reliably measured. All other repair and maintenance expense is recognised in the statement of comprehensive income when incurred.

(4) Disposal

On disposal of an item of plant and equipment, the difference between the net disposal proceeds and its carrying amount is recognised in the statement of comprehensive income.

(f) Investment in subsidiary

Investment in subsidiary is carried at cost less accumulated impairment losses in the Company's balance sheet. On disposal of investment in subsidiary, the difference between disposal proceeds and the carrying amount of the investment is recognised in the statement of comprehensive income.

(g) Loan to related party

Loan to related party is initially recognised at fair value plus transaction costs and subsequently carried at amortised cost using the effective interest method, less accumulated impairment losses. The Company assesses at each balance sheet date whether there is objective evidence that this financial asset is impaired and recognises an allowance for impairment when such evidence exists.

(h) Impairment of non-financial assets

Plant and equipment and investment in subsidiary are reviewed for impairment whenever there is any objective evidence or indication that these assets may be impaired.

For the purpose of impairment testing, the recoverable amount (i.e. the higher of the fair value less cost to sell and value in use) is determined on an individual asset basis unless the asset does not generate cash flows that are largely independent of those from other assets. If this is the case, the recoverable amount is determined for the cash-generating unit (CGU) to which the asset belongs.

2. Significant accounting policies (continued)

(h) Impairment of non-financial assets (continued)

If the recoverable amount of the asset (or CGU) is estimated to be less than its carrying amount, the carrying amount of the asset (or CGU) is reduced to its recoverable amount. The difference between the carrying amount and recoverable amount is recognised in the statement of comprehensive income.

An impairment loss for an asset is reversed if, and only if, there has been a change in the estimates used to determine the assets' recoverable amount since the last impairment loss was recognised. The carrying amount of the asset is increased to its revised recoverable amount, provided that this amount does not exceed the carrying amount that would have been determined (net of any accumulated amortisation or depreciation) had no impairment loss been recognised for the asset in prior years. A reversal of impairment loss for an asset is recognised in the statement of comprehensive income.

(i) Financial assets

Receivables are non-derivative financial assets with fixed or determinable payments that are not quoted in an active market. They are presented as current assets, except for those maturing later than 12 months after the balance sheet date which are presented as non-current assets. These financial assets are initially recognised at fair value plus transaction cost and subsequently carried at amortised cost using the effective interest method, less allowance for impairment.

The Company assesses at each balance sheet date whether there is objective evidence that these financial assets are impaired and recognises an allowance for impairment when such evidence exists.

The carrying amount of these assets is reduced through the use of an impairment allowance account which is calculated as the difference between the carrying amount and the present value of estimated future cash flows, discounted at the original effective interest rate.

(j) Trade and other payables

Trade and other payables are initially recognised at fair value and subsequently carried at amortised cost, using the effective interest method.

(k) Operating leases

Leases of assets where substantially all risks and rewards incidental to ownership are retained by the lessor are classified as operating leases. Payments made under operating leases (net of any incentives received from the lessor) are recognised in the statement of comprehensive income on a straight-line basis over the period of the lease.

2. Significant accounting policies (continued)

(k) Operating leases (continued)

When a lease is terminated before the lease period expires, any payment made by the Company as penalty is recognised as an expense in the period in which the termination takes place.

(l) Inventories

Inventories are carried at the lower of cost and net realisable value. Cost is determined using the first-in, first-out method. Net realisable value is the estimated selling price in the ordinary course of business, less applicable variable selling expenses.

(m) Income taxes

Current income tax for current and prior periods is recognised at the amount expected to be paid to or recovered from the tax authorities, using the tax rates and tax laws that have been enacted or substantively enacted by the balance sheet date.

Deferred income tax is recognised for all temporary differences arising between the tax bases of assets and liabilities and their carrying amounts in the financial statements except when the deferred income tax arises from the initial recognition of goodwill or an asset or liability in a transaction that is not a business combination and affects neither accounting nor taxable profit or loss at the time of the transaction.

Current and deferred income tax is recognised as income or expense in the statement of comprehensive income, except to the extent that the tax arises from a business combination or a transaction which is recognised directly in equity.

(n) Provisions for other liabilities and charges

Provisions for other liabilities and charges are recognised when the Company has a present legal or constructive obligation as a result of past events, it is more likely than not that an outflow of resources will be required to settle the obligation and the amount has been reliably estimated.

(o) Currency translation

The financial statements of the Company are presented in United States Dollars, which is the functional currency of the Company.

2. Significant accounting policies (continued)

(o) Currency translation (continued)

Transactions in a currency other than United States Dollar ("foreign currency") are translated into United States Dollar using the exchange rates prevailing at the dates of the transactions. Currency translation differences resulting from the settlement of such transactions and from the translation of monetary assets and liabilities denominated in foreign currencies at the closing rates at the balance sheet date are recognised in the statement of comprehensive income.

(p) Cash and cash equivalents

For the purpose of presentation in the cash flow statement, cash and cash equivalents include cash on hand and deposits with financial institutions which are subject to an insignificant risk of change in value.

(q) Share capital

Ordinary shares are classified as equity.

3. Income

	27/01/2014 to 25/01/2015 US$'000	28/01/2013 to 26/01/2014 US$'000
Revenue		
Sale of graphics processors and related media and communication devices	**3,885,594**	3,461,212
Other income – net:		
Interest income	**128**	523
Others	**484**	4
	3,886,206	3,461,739

4. Other gain - net

	27/01/2014 to 25/01/2015 US$'000	28/01/2013 to 26/01/2014 US$'000
Exchange gains - net	**19**	1,578

5. Other operating expenses

	27/01/2014 to 25/01/2015 US$'000	28/01/2013 to 26/01/2014 US$'000
Consulting and professional expense	**821**	1,897
Utilities expense	**110**	119
Repair and maintenance expense	**276**	293
Bank charges	**166**	164
Bad debt expense / (reversal)	**1,786**	(847)
Insurance for credit	**1,064**	858
Other expenses	**1,863**	2,116
Total	**6,086**	4,600

6. Employee compensation

	27/01/2014 to 25/01/2015 US$'000	28/01/2013 to 26/01/2014 US$'000
Wages and salaries	**5,822**	5,905
Employer's contribution to defined contribution plans	**428**	603
	6,250	6,508

The key management is directors of the Company and there is no key management personnel remuneration during the financial year (2014: US$Nil).

7. Income tax

(a) Income tax expense

	27/01/2014 to 25/01/2015 US$'000	28/01/2013 to 26/01/2014 US$'000
Income tax expense attributable to profit is made up of:		
- Current income tax	**1,282**	852

7. Income tax (continued)

(a) Income tax expense (continued)

The tax expense on profit differs from the amount that would arise using the Singapore standard rate of income tax as explained below:

	27/01/2014 to 25/01/2015 US$'000	28/01/2013 to 26/01/2014 US$'000
Profit before tax	**4,907**	6,387
Tax calculated at a tax rate of 17% (2014: 17%)	**834**	1,086
Effects of:		
- Expenses not deductible for tax purposes	**307**	50
- Income not subject to tax	**(12)**	(226)
- Different tax rates in other countries	**(24)**	(32)
- Others	**177**	(26)
Tax charge	**1,282**	852

(b) Movements in current tax liabilities

	25/01/2015 US$'000	26/01/2014 US$'000
Beginning of financial year	**286**	640
Tax expense on profit - current financial year	**1,282**	852
Income tax paid	**(822)**	(1,206)
End of the financial year	**746**	286

8. Cash and cash equivalents

	25/01/2015 US$'000	26/01/2014 US$'000
Cash at bank	**66,979**	198,035

9. Trade and other receivables

	25/01/2015 US$'000	26/01/2014 US$'000
Trade receivables:		
- Immediate holding company	48	48
- Non-related parties	419,802	387,524
Less: Allowance for impairment of receivables		
- Non-related parties	(2,546)	(760)
Trade receivables - net	417,304	386,812
Non-trade receivables:		
- Immediate holding company	190,696	193,332
- Other related companies	102,968	96,491
	293,664	289,823
Other receivables	130	85
	711,098	676,720

The non-trade receivables due from other related companies are unsecured, interest-free and are receivable on demand.

10. Inventories

	25/01/2015 US$'000	26/01/2014 US$'000
Finished goods	220,854	191,712

11. Plant and equipment

	Computer equipment US$'000	Machinery and equipment US$'000	Office equipment US$'000	Furniture US$'000	Leasehold improvement US$'000	Total US$'000
2015						
Cost						
Beginning of financial year	1,996	2,651	68	26	1,257	5,998
Additions	16	24	24	-	11	75
Disposals	(1,146)	(1,153)	(64)	-	(1,071)	(3,434)
End of financial year	866	1,522	28	26	197	2,639
Accumulated depreciation						
Beginning of financial year	1,330	1,996	59	7	1,108	4,500
Depreciation charge	315	443	10	6	55	829
Disposals	(1,146)	(1,153)	(64)	-	(1,071)	(3,434)
End of financial year	499	1,286	5	13	92	1,895
Net book value						
End of financial year	**367**	**236**	**23**	**13**	**105**	**744**
2014						
Cost						
Beginning of financial year	2,305	2,406	186	26	1,244	6,167
Additions	532	273	-	-	13	818
Disposals	(841)	(28)	(118)	-	-	(987)
End of financial year	1,996	2,651	68	26	1,257	5,998
Accumulated depreciation						
Beginning of financial year	1,968	1,521	167	2	1,055	4,713
Depreciation charge	202	496	9	5	53	765
Disposals	(840)	(21)	(117)	-	-	(978)
End of financial year	1,330	1,996	59	7	1,108	4,500
Net book value						
End of financial year	**666**	**655**	**9**	**19**	**149**	**1,498**

12. Investment in subsidiary

	25/01/2015 US$'000 $	26/01/2014 US$'000 $
Unquoted equity shares at cost	**85**	85

Details of investment in subsidiary are as follows:

Name of company	Principal activities	Country of incorporation	Equity holding 25/01/2015 %	Equity holding 26/01/2014 %	Cost of investment 25/01/2015 US$'000	Cost of investment 26/01/2014 US$'000
NVIDIA GK	Solicitation and negotiation of orders and working on behalf of Nvidia Singapore Pte Ltd	Japan	**100**	100	**85**	85

The subsidiary does not require an audit in accordance with the laws in the country of its incorporation.

13. Loan to related party

The loan to Nvidia Brazil is unsecured, denominated in the United States Dollar and is due in full on 7 August 2023. The contractual interest rate on the loan at balance sheet date is 3.0% (2014: 3.0%) per annum. The carrying amount of the loan approximate its fair value.

14. Trade and other payables

	25/01/2015 US$'000	26/01/2014 US$'000
Trade payables		
- Immediate holding company	**768,850**	847,647
	768,850	847,647
Non-trade payables		
- Immediate holding company	**4,157**	3,979
- Other related companies	**43,683**	33,163
	47,840	37,142
Accrued operating expenses	**146,686**	153,677
Other payables	**5,491**	4,380
	152,177	158,057
	968,867	1,042,846

The non-trade payables to immediate holding company and other related companies are unsecured, interest-free and are repayable on demand.

15. Share capital

The Company's share capital comprise fully paid-up 2 (2014: 2) ordinary shares with no par value, amounting to a total of US$1,000 (2014: US$1,000).

16. Commitments

Operating lease commitments

The future minimum lease payable under non-cancellable operating leases contracted for at the balance sheet date but not recognised as liabilities are as follows:

	25/01/2015 US$'000	26/01/2014 US$'000
Not later than one year	**433**	398
Later than one year but not later than five years	**578**	1,011
	1,011	1,409

17. Financial risk management

Financial risk factors

The Company's activities expose it to market risk (including currency risk, interest rate risk and price risk), credit risk and liquidity risk.

The management team is responsible for setting the objectives and underlying principles of financial risk management for the Company, and establishes detailed policies such as risk identification and measurement, exposure limits and hedging strategies. Financial risk management is carried out by finance personnel.

The finance personnel measure actual exposures against the limits set and prepare regular reports for the review of the management team. The information presented below is based on information received by key management.

(a) Market risk

(i) Currency risk

The Company's revenue, cost of operations and majority of the financial assets and liabilities are primarily in United States Dollar ("USD"). The Company is not exposed to any significant exchange rate risk. The Company does not purchase forward contracts to hedge this exposure.

17. Financial risk management (continued)

(a) Market risk (continued)

(ii) Price risk

The Company has insignificant exposure to equity price risk as it does not hold any equity financial assets.

(iii) Interest rate risk

The Company is exposed to interest rate risk on its bank deposits. The bank deposits are mainly deposits in USD.

At 25 January 2015, if the USD interest rates had increased/decreased by 0.05% (2014: 0.05%) with all other variables including tax being constant, the profit after tax would have been higher/lower by US$33,000 (2014: US$98,500) as a result of higher/lower interest income on these deposits.

The Company has insignificant financial liabilities that are exposed to interest rate risks.

(b) Credit risk

Credit risk refers to the risk that a counterparty will default on its contractual obligations resulting in financial loss to the Company. The major classes of financial assets of the Company are cash and cash equivalents and trade receivables. Cash and cash equivalents are deposits in banks with sound credit ratings, therefore, the Company does not expect to have high credit risk in this regard. For trade receivables, the Company adopts the policy of dealing only with customers of appropriate credit history, and obtaining sufficient collateral where appropriate to mitigate credit risk. In addition, the Company monitors its credit risk on an ongoing basis by reviewing the debtors' aging to minimise its exposure to credit risk.

(i) Financial assets that are neither past due nor impaired

Trade receivables that are neither past due nor impaired are substantially companies with a good collection track record with the Company.

17. Financial risk management (continued)

(b) Credit risk (continued)

(ii) Financial assets that are past due and/or impaired

There is no financial asset that is past due and/or impaired.

The carrying amount of trade receivables individually determined to be impaired and the movement in the related allowance for impairment are as follows:

	25/01/2015 US$'000	26/01/2014 US$'000
Gross amount	2,546	760
Less: Allowance for impairment	(2,546)	(760)
	-	-
Beginning of financial year	760	1,607
Allowance made/(utilised)	1,786	(847)
End of financial year (Note 9)	2,546	760

(c) Liquidity risk

The Company manage the liquidity risk by maintaining sufficient cash and bank balances to enable it to meet its operational requirements.

(d) Capital risk

The Company's objectives when managing capital are to ensure that the Company is adequately capitalised and to maintain an optimal capital structure.

The Company is not subject to any externally imposed capital requirements.

(e) Financial instruments by category

The carrying amounts of financial assets at fair value through profit or loss are disclosed on the face of the balance sheet. The aggregate carrying amounts of loans and receivables and financial liabilities at amortised cost are as follows:

	25/01/2015 2015 US$'000	26/01/2014 2014 US$'000
Loans and receivables	777,948	874,670
Financial liabilities at amortised cost	968,609	1,042,611

17. Financial risk management (continued)

(f) Offsetting financial assets and financial liabilities

(i) Financial assets

The Company has the following financial instrument subject to enforceable master netting arrangement or similar agreement as follows:

	Related amounts set off			Related amounts not set off	
	Gross amounts - financial assets	Gross amounts – financial liabilities	Net amounts – financial assets presented in the balance sheet	Financial assets	Net amount
As at 25 January 2015					
Trade and other receivables	**204,660**	**13,964**	**190,696**	**103,016**	**293,712**
Total	**204,660**	**13,964**	**190,696**	**103,016**	**293,712**
As at 26 January 2014					
Trade and other receivables	**204,666**	**11,334**	**193,332**	**96,539**	**289,871**
Total	**204,666**	**11,334**	**193,332**	**96,539**	**289,871**

(ii) Financial liabilities

The Company has the following financial instruments subject to enforceable master netting arrangements or similar agreement as follows:

	Related amounts set off			Related amounts not set off	
	Gross amounts - financial liabilities	Gross amounts – financial assets	Net amounts – financial liabilities presented in the balance sheet	Financial liabilities	Net amount
As at 25 January 2015					
Trade and other payables	**1,092,212**	**353,480**	**738,732**	**77,958**	**816,690**
Total	**1,092,212**	**353,480**	**738,732**	**77,958**	**816,690**
As at 26 January 2014					
Trade and other payables	837,299	337,880	499,419	385,370	884,789
Total	837,299	337,880	499,419	385,370	884,789

18. Immediate and ultimate holding corporations

The Company's immediate holding corporation is Nvidia International Inc., incorporated in the Cayman Islands. The Company's ultimate holding corporation is Nvidia Corporation, incorporated in United States of America.

19. Related party transactions

In addition to the information disclosed elsewhere in the financial statements, the following transactions took place between the Company and related parties at terms agreed between the parties:

	27/01/2014 to 25/01/2015 US$'000	28/01/2013 to 26/01/2014 US$'000
Purchases from immediate holding company	**3,483,981**	3,124,542
Intercompany services on marketing and administrative expenses paid and payable to other related corporations	**147,245**	137,145

No expense has been recorded in respect of the fair value of share options granted by the ultimate holding corporation to the Company's employees in the statement of comprehensive income. This is not in accordance with Financial Reporting Standard 102 – Share-based Payments which requires the Company to measure and recognise the fair value of share options granted by the ultimate holding corporation to its employees in the statement of comprehensive income with a corresponding increase recognised in equity as a contribution from the ultimate holding corporation.

The Company participates in the centralised cash pooling arrangement with Nvidia Corporation, its ultimate holding company. During the year, certain balances with related companies were settled and netted off through the cash pooling arrangement.

Share based payments have not been recorded as the directors are of the opinion that this process would involve expense or delay out of proportion to the value to the shareholder of the Company.

20. New or revised accounting standards and interpretations

The Company has not early adopted any mandatory standards, amendments and interpretations to existing standards that have been published but are only effective for the Company's accounting periods beginning on or after 26 January 2015. However, management anticipates that the adoption of these standards, amendments and interpretations will not have a material impact on the financial statements of the Company in the period of their initial adoption.

21. Authorisation of financial statements

These financial statements were authorised for issue in accordance with a resolution of the Board of Directors of Nvidia Singapore Pte Ltd on 25 JUN 2015

EXHIBIT K

NVIDIA SINGAPORE PTE LTD

(Incorporated in Singapore. Registration Number 200003831M)

FINANCIAL STATEMENT

For the financial year ended 31 January 2016

e23de91995aab7183b831efb1e7bc95aadd170a1b020a654280e86ae55096edb

NVIDIA SINGAPORE PTE LTD
(Incorporated in Singapore)

FINANCIAL STATEMENT
For the financial year ended 31 January 2016

Contents

NVIDIA SINGAPORE PTE LTD

DIRECTORS' STATEMENT
For the financial year ended 31 January 2016

The directors present their report to the shareholder together with the audited financial statements for the financial year ended 31 January 2016.

In the opinion of the directors,

(a) the financial statements as set out on pages 4 to 31 are drawn up so as to give a true and fair view of the financial position of the Company at 31 January 2016 and the financial performance, changes in equity and cash flows of the Company for the financial year covered by the financial statements; and

(b) at the date of this statement, there are reasonable grounds to believe that the Company will be able to pay its debts as and when they fall due.

Directors

The directors in office at the date of this report are as follows:

Lee Kay Beng
Karen Theresa Burns
Michael John Byron
Rebecca Peters (appointed on 31 May 2016)

Arrangements to enable directors to acquire shares and debentures

Neither at the end of nor at any time during the financial year was the Company a party to any arrangement whose object was to enable the directors of the Company to acquire benefits by means of the acquisition of shares in, or debentures of, the Company or any other body corporate.

Directors' interests in shares and debentures

According to the register of directors' shareholdings, none of the directors holding office at the end of the financial year had any interest in the shares or debentures of the Company or its related corporations.

Share options

No options were granted during the financial year to subscribe for unissued shares of the Company.

No shares were issued during the financial year by virtue of the exercise of options to take up unissued shares of the Company.

There were no unissued shares of the Company under option at the end of the financial year.

d236e91995e8b67183b831c1b1e7bd95aadd178e1b020ae5428de86ec5309ecdb

NVIDIA SINGAPORE PTE LTD

DIRECTORS' STATEMENT
For the financial year ended 31 January 2016

Independent auditor

The independent auditor, PricewaterhouseCoopers LLP, has expressed its willingness to accept re-appointment.

On behalf of the directors

MICHAEL JOHN BYRON
Director

KAREN THERESA BURNS
Director

13 JUL 2016

INDEPENDENT AUDITOR'S REPORT TO THE SHAREHOLDER OF NVIDIA SINGAPORE PTE LTD

Report on the Financial Statements

We have audited the accompanying financial statements of Nvidia Singapore Pte Ltd set out on pages 4 to 31, which comprise the balance sheet as at 31 January 2016, the statement of comprehensive income, statement of changes in equity and the statement of cash flows for the financial year ended 31 January 2016, and a summary of significant accounting policies and other explanatory information.

Management's Responsibility for the Financial Statements
Management is responsible for the preparation of financial statements that give a true and fair view in accordance with the provisions of the Singapore Companies Act (the "Act") and Singapore Financial Reporting Standards, and for devising and maintaining a system of internal accounting controls sufficient to provide a reasonable assurance that assets are safeguarded against loss from unauthorised use or disposition; and transactions are properly authorised and that they are recorded as necessary to permit the preparation of true and fair profit and loss accounts and balance sheets and to maintain accountability of assets.

Auditor's Responsibility
Our responsibility is to express an opinion on these financial statements based on our audit. We conducted our audit in accordance with Singapore Standards on Auditing. Those standards require that we comply with ethical requirements and plan and perform the audit to obtain reasonable assurance about whether the financial statements are free from material misstatement.

An audit involves performing procedures to obtain audit evidence about the amounts and disclosures in the financial statements. The procedures selected depend on the auditor's judgement, including the assessment of the risks of material misstatement of the financial statements, whether due to fraud or error. In making those risk assessments, the auditor considers internal control relevant to the entity's preparation of financial statements that give a true and fair view in order to design audit procedures that are appropriate in the circumstances, but not for the purpose of expressing an opinion on the effectiveness of the entity's internal control. An audit also includes evaluating the appropriateness of accounting policies used and the reasonableness of accounting estimates made by management, as well as evaluating the overall presentation of the financial statements.

We believe that the audit evidence we have obtained is sufficient and appropriate to provide a basis for our audit opinion.

Opinion
In our opinion, the financial statements of the Company are properly drawn up in accordance with the provisions of the Act and Singapore Financial Reporting Standards so as to give a true and fair view of the financial position of the Company as at 31 January 2016, and of the financial performance, changes in equity and cash flows of the Company for the year ended on that date.

Report on Other Legal and Regulatory Requirements

In our opinion, the accounting and other records required by the Act to be kept by the Company have been properly kept in accordance with the provisions of the Act.

PricewaterhouseCoopers LLP
Public Accountants and Chartered Accountants
Singapore,
13 JUL 2016

NVIDIA SINGAPORE PTE LTD

STATEMENT OF COMPREHENSIVE INCOME
For the financial year ended 31 January 2016

	Note	26/01/2015 to 31/01/2016 US$'000	27/01/2014 to 25/01/2015 US$'000
Revenue	3	**4,077,896**	3,885,594
Other income - net	3	**134**	612
Other gain - net	4	**4**	19
Expenses			
- Raw materials and other consumables consumed		**(3,896,460)**	(3,715,120)
- Employee compensation	6	**(7,356)**	(6,794)
- Travelling expenses		**(200)**	(232)
- Sales and marketing expenses		**(5,258)**	(5,152)
- Depreciation of plant and equipment		**(478)**	(829)
- Intercompany service expenses		**(159,282)**	(147,245)
- Rental on operating leases		**(439)**	(404)
- Other operating expenses	5	**(4,695)**	(6,086)
Total expenses		**(4,074,168)**	(3,881,862)
Profit before income tax		**3,866**	4,363
Income tax expense	7	**(843)**	(1,282)
Net profit after tax/ Total comprehensive income		**3,023**	3,081

The accompanying notes form an integral part of these financial statements.

NVIDIA SINGAPORE PTE LTD

BALANCE SHEET
As at 31 January 2016

	Note	31/01/2016 US$'000	25/01/2015 US$'000 (restated)	27/01/2014 US$'000 (restated)
ASSETS				
Current assets				
Cash and cash equivalents	8	**223,210**	66,979	198,035
Trade and other receivables	9	**704,405**	711,098	676,720
Inventories	10	**208,883**	220,854	191,712
Prepayment and deposits		**1,052**	2,252	3,856
		1,137,550	1,001,183	1,070,323
Non-current assets				
Plant and equipment	11	**316**	744	1,498
Investment in subsidiary	12	**85**	85	85
Loan to related party	13	**500**	500	500
		901	1,329	2,083
Total assets		**1,138,451**	1,002,512	1,072,406
LIABILITIES				
Current liabilities				
Trade and other payables	14	**1,101,716**	968,867	1,042,846
Current income tax liabilities	7	**70**	746	286
		1,101,786	969,613	1,043,132
Total liabilities		**1,101,786**	969,613	1,043,132
NET ASSETS		**36,665**	32,899	29,274
SHAREHOLDER'S EQUITY				
Share capital	15	**1**	1	1
Other reserves		**9,524**	8,781	8,237
Retained earnings		**27,140**	24,117	21,036
		36,665	32,899	29,274

The accompanying notes form an integral part of these financial statements.

NVIDIA SINGAPORE PTE LTD

STATEMENT OF CHANGES IN EQUITY

For the financial year ended 31 January 2016

	Share capital US$'000	Other reserves US$'000	Retained earnings US$'000	Total equity US$'000
2016				
Beginning of financial year - restated	**1**	**8,781**	**24,117**	**32,899**
Share-based compensation	-	743	-	743
Total comprehensive income	-	-	3,023	3,023
End of financial year	**1**	**9,524**	**27,140**	**36,665**
2015				
Beginning of financial year - restated	1	**8,237**	21,036	29,274
Share-based compensation	-	544	-	544
Total comprehensive income	-	-	3,081	3,081
End of financial year - restated	**1**	**8,781**	**24,117**	**32,899**

The accompanying notes form an integral part of these financial statements.

NVIDIA SINGAPORE PTE LTD

STATEMENT OF CASH FLOWS

For the financial year ended 31 January 2016

	Note	26/01/2015 to 31/01/2016 US$'000	27/01/2014 to 25/01/2015 US$'000 (restated)
Cash flows from operating activities			
Net profit after tax		3,023	3,081
Adjustments for:			
- Tax		843	1,282
- Depreciation of plant and equipment		478	829
- Share-based compensation		743	544
- Interest income		(119)	(128)
Operating cash flow before working capital changes		4,968	5,608
Change in operating assets and liabilities			
- Trade and other receivables		6,693	(34,378)
- Inventories		11,971	(29,142)
- Trade and other payables		132,849	(73,979)
- Prepayments and deposits		1,200	1,604
Cash generated/(used) in operations		157,681	(130,287)
Income tax paid		(1,519)	(822)
Net cash generated from/(used) in operating activities		156,162	(131,109)
Cash flows from investing activities			
Purchases of plant and equipment		(82)	(75)
Proceeds from disposal of plant and equipment		32	-
Interest received		119	128
Net cash generated from investing activities		69	53
Net increase/(decrease) in cash and cash equivalents		156,231	(131,056)
Cash and cash equivalents at beginning of financial period		66,979	198,035
Cash and cash equivalents at end of financial period	8	223,210	66,979

The accompanying notes form an integral part of these financial statements.

These notes form an integral part of and should be read in conjunction with the accompanying financial statements.

1. General information

The Company is incorporated in Singapore. The address of its registered office is 112 Robinson Road, #05-01, Singapore 068902.

The principal activity of the Company consists of sales of graphics processors and media and communication devices.

The principal activities of its subsidiary are those relating to the solicitation and negotiation of orders and marketing on behalf of Nvidia Singapore Pte Ltd.

2. Significant accounting policies

(a) Basis of preparation

The financial statements have been prepared in accordance with Singapore Financial Reporting Standards ("FRS") under the historical cost convention, except as disclosed in the accounting policies below.

The preparation of these financial statements in conformity with FRS requires management to exercise its judgement in the process of applying the Company's accounting policies. It also requires the use of certain critical accounting estimates and assumptions. There are no significant areas involving a higher degree of judgement, estimates and assumption to these financial statements.

Interpretations and amendments to published standards effective in 2016

On 26 January 2015, the Company adopted the new or amended FRS and Interpretations to FRS ("INT FRS") that are mandatory for application for the financial year. Changes to the Company's accounting policies have been made as required, in accordance with the relevant transitional provisions in the respective FRS and INT FRS.

The adoption of the above new or revised FRS did not result in any substantial changes to the Company's accounting policies nor any significant impact on these financial statements.

2. Significant accounting policies (continued)

(b) Revenue recognition

Revenue comprises the fair value of the consideration received or receivable for the sales of goods in the ordinary course of the Company's activities. Revenue is presented net of goods and services tax, rebates and discounts. Revenue is recognised as follows:

(1) Sale of goods

Revenue from sale of graphics processors and related media and communication devices is recognised when the Company has delivered the products to the customer, the customer has accepted the products and collectability of the related receivables is reasonably assured.

A sales return allowance for estimated product returns is established at the time revenue is recognised, based primarily on historical return rates.

(2) Interest income

Interest income is recognised using the effective interest method.

(c) Group accounting

These financial statements are the separate financial statements of Nvidia Singapore Pte Ltd. The Company is exempted from the preparation of consolidated financial statements as the Company is a wholly-owned subsidiary of Nvidia Corporation, a US-incorporated company which produces consolidated financial statements available for public use. The basis on which the subsidiary is accounted for is disclosed in Note 2(f). The main office of Nvidia Corporation is as follows: 2701 San Tomas Expressway, Santa Clara, CA 95050, U.S.A..

(d) Employee compensation

(1) Defined contribution plans

Defined contribution plans are post-employment benefit plans under which the Company pays fixed contributions into separate entities such as Mandatory Provident Fund and Superannuation Fund established in Hong Kong, on a mandatory, contractual or voluntary basis. The Company has no further payment obligations once the contributions have been paid. The Company's contribution to defined contribution plans are recognised as employee compensation expense when they are due.

2. Significant accounting policies (continued)

(d) Employee compensation (continued)

(2) Employee leave entitlements

Employee entitlements to annual leave are recognised when they accrue to employees. A provision is made for the estimated liability for annual leave as a result of services rendered by employees up to the balance sheet date.

(3) Share-based compensation

The employees of the Company are eligible for share-based compensation under equity-settled, shared based compensation plans operated by Nvidia Corporation, the ultimate holding corporation of the Company. Awards under these plans include stock options, restricted share units ("RSUs") and employee stock purchase plan ("ESPP").

The Company accounts for share based payments in accordance with FRS 102 – Share based payments. Share based compensation expense is measured at the grant date of the related equity awards, based on the fair value of the awards, and the expense is recognised over the vesting period, with a corresponding increase in the other reserve.

In case of exchange of employees' services, the equity-settled payment is measured at the fair value of equity instruments granted to employees. If exercisable at the time of grant, equity-settled payment is included in relevant cost or expenses at fair value at grant date and capital surplus is increased accordingly; if exercisable after service in waiting period is completed or specified performance conditions are met, the service obtained in current period is included in relevant cost and expenses at fair value based on the best estimation on quantity of exercisable equity instruments made by the Company in accordance with latest changes in the number of exercisable employees and subsequent information such as whether specified performance conditions are met, and capital surplus is increased accordingly.

Stock options

The total amount to be recognised over the vesting period is determined by reference to the fair value of the options on the date of the grant.

The fair value of stock options is determined using option pricing model. The fair value of stock appreciation right is determined using Bimonial option pricing model.

2. Significant accounting policies (continued)

(d) Employee compensation (continued)

(3) Share-based compensation (continued)

Stock options (continued)

At the balance sheet date, subsequent information indicates that the number of equity instruments which are different with previous estimates should be adjusted to the exercisable number of equity instruments on the vesting date. After vesting date, no further adjustment will be raised for relevant cost (expense) or total owner's equity which has been recognised.

Stock options granted to employees, subject to certain exceptions, vest over a four year period, subject to continued service, with 25% vesting on the anniversary of the hire date in the case of new hires or the anniversary of the date of grant in the case of grants to existing employees and 6.25% vesting at the end of each quarterly period thereafter. Stock options granted under the 2007 Plan generally expire ten years from the date of grant.

Restricted Stock Units ("RSU")

RSUs granted to employees vest four years, subject to continued service, with 25% vesting on a pre-determined date that is close to the anniversary of the date of grant and 12.5% vesting semi-annually thereafter until fully vested. The total amount to be recognised over the vesting period is determined by reference to the fair value of the ultimate holding corporation stock on the date of the grant.

Employee Stock Purchase Plan ("ESPP")

Employees are also eligible to participate in an offering to have up to 10% of their earnings withheld up to certain limitations and applied on specified dates determined by the Board of Directors of Nvidia Corporation to the purchase of shares of common stock. The Board of Directors of Nvidia Corporation may increase this percentage at its discretion, up to 15%. The price of common stock purchased under our ESPP will be equal to 85% of the lower of the fair market value of the common stock on the commencement date of each offering period and the purchase date of each offering period. Employees may end their participation in the ESPP at any time during the offering period, and participation ends automatically on termination of employment with us. In each case, the employee's contributions are refunded. The employee benefit is recognised in profit or loss in the period in which the employees purchase the shares under common stock under the ESPP.

2. Significant accounting policies (continued)

(e) Plant and equipment

(1) Measurement

Plant and equipment are initially recognised at cost and subsequently carried at cost less accumulated depreciation and accumulated impairment losses.

The cost of an item of plant and equipment initially recognised includes its purchase price and any cost that is directly attributable to bringing the asset to the location and condition necessary for it to be capable of operating in the manner intended by management. The projected cost of dismantlement, removal or restoration is recognised as part of the cost of plant and equipment if such obligation is incurred either when the item is acquired or as a consequence of using the asset during a particular period for purposes other than to produce inventories during that period.

(2) Depreciation

Depreciation on plant and equipment is calculated using the straight-line method to allocate their depreciable amounts over their estimated useful lives as follows:

	Useful lives
Machinery and equipment	3 - 5 years
Computer equipment	3 - 5 years
Office equipment	5 years
Furniture	5 years
Leasehold improvement	Lesser of estimated useful life or remaining lease term

The residual values, estimated useful lives and depreciation method of plant and equipment are reviewed, and adjusted as appropriate, at each balance sheet date. The effects of any revision are recognised in the statement of comprehensive income for the financial period in which the changes arise.

(3) Subsequent expenditure

Subsequent expenditure relating to plant and equipment that has already been recognised is added to the carrying amount of the asset only when it is probable that future economic benefits associated with the item, will flow to the Company and the cost can be reliably measured. All other repair and maintenance expense is recognised in the statement of comprehensive income when incurred.

2. Significant accounting policies (continued)

(e) Plant and equipment (continued)

(4) Disposal

On disposal of an item of plant and equipment, the difference between the net disposal proceeds and its carrying amount is recognised in the statement of comprehensive income.

(f) Investment in subsidiary

Investment in subsidiary is carried at cost less accumulated impairment losses in the Company's balance sheet. On disposal of investment in subsidiary, the difference between disposal proceeds and the carrying amount of the investment is recognised in the statement of comprehensive income.

(g) Loan to related party

Loan to related party is initially recognised at fair value plus transaction costs and subsequently carried at amortised cost using the effective interest method, less accumulated impairment losses. The Company assesses at each balance sheet date whether there is objective evidence that this financial asset is impaired and recognises an allowance for impairment when such evidence exists.

(h) Impairment of non-financial assets

Plant and equipment and investment in subsidiary are tested for impairment whenever there is any objective evidence or indication that these assets may be impaired.

For the purpose of impairment testing, the recoverable amount (i.e. the higher of the fair value less cost to sell and value in use) is determined on an individual asset basis unless the asset does not generate cash flows that are largely independent of those from other assets. If this is the case, the recoverable amount is determined for the cash-generating unit (CGU) to which the asset belongs.

If the recoverable amount of the asset (or CGU) is estimated to be less than its carrying amount, the carrying amount of the asset (or CGU) is reduced to its recoverable amount. The difference between the carrying amount and recoverable amount is recognised in the statement of comprehensive income.

2. Significant accounting policies (continued)

(h) Impairment of non-financial assets (continued)

An impairment loss for an asset is reversed if, and only if, there has been a change in the estimates used to determine the assets' recoverable amount since the last impairment loss was recognised. The carrying amount of this asset is increased to its revised recoverable amount, provided that this amount does not exceed the carrying amount that would have been determined (net of any accumulated amortisation or depreciation) had no impairment loss been recognised for the asset in prior years. A reversal of impairment loss for an asset is recognised in profit or loss.

(i) Loans and receivables
Bank balances

Receivables are non-derivative financial assets with fixed or determinable payments that are not quoted in an active market. They are presented as current assets, except for those maturing later than 12 months after the balance sheet date which are presented as non-current assets. These financial assets are initially recognised at fair value plus transaction cost and subsequently carried at amortised cost using the effective interest method, less allowance for impairment.

The Company assesses at each balance sheet date whether there is objective evidence that these financial assets are impaired and recognises an allowance for impairment when such evidence exists.

The carrying amount of these assets is reduced through the use of an impairment allowance account which is calculated as the difference between the carrying amount and the present value of estimated future cash flows, discounted at the original effective interest rate.

(j) Trade and other payables

Trade and other payables are initially recognised at fair value and subsequently carried at amortised cost, using the effective interest method.

(k) Operating leases

Leases of assets where substantially all risks and rewards incidental to ownership are retained by the lessor are classified as operating leases. Payments made under operating leases (net of any incentives received from the lessor) are recognised in profit or loss on a straight-line basis over the period of the lease.

When a lease is terminated before the lease period expires, any payment made by the Company as penalty is recognised as an expense in the period in which the termination takes place.

2. Significant accounting policies (continued)

(l) Inventories

Inventories are carried at the lower of cost and net realisable value. Cost is determined using the first-in, first-out method. Net realisable value is the estimated selling price in the ordinary course of business, less applicable variable selling expenses.

(m) Income taxes

Current income tax is recognised at the amount expected to be paid to or recovered from the tax authorities, using the tax rates and tax laws that have been enacted or substantively enacted by the balance sheet date.

Deferred income tax is recognised for all temporary differences except when the deferred income tax arises from the initial recognition of an asset or liability that affects neither accounting nor taxable profit or loss at the time of the transaction.

Current and deferred income tax is measured using the tax rates and tax laws that have been enacted or substantively enacted by the balance sheet date, and are recognised as income or expenses in profit or loss, except to the extent that the tax arises from a transaction which is recognised directly in equity.

(n) Provisions for other liabilities and charges

Provisions for other liabilities and charges are recognised when the Company has a present legal or constructive obligation as a result of past events, it is more likely than not that an outflow of resources will be required to settle the obligation and the amount has been reliably estimated.

(o) Currency translation

The financial statements of the Company are presented in United States Dollars, which is the functional currency of the Company.

Transactions in a currency other than United States Dollar ("foreign currency") are translated into United States Dollar using the exchange rates prevailing at the dates of the transactions. Currency translation differences resulting from the settlement of such transactions and from the translation of monetary assets and liabilities denominated in foreign currencies at the closing rates at the balance sheet date are recognised in profit or loss.

2. Significant accounting policies (continued)

(p) Cash and cash equivalents

For the purpose of presentation in the cash flow statement, cash and cash equivalents include cash on hand and deposits with financial institutions which are subject to an insignificant risk of change in value.

(q) Share capital

Ordinary shares are classified as equity.

3. Income

	26/01/2015 to 31/01/2016 US$'000	27/01/2014 to 25/01/2015 US$'000
Revenue		
Sale of graphics processors and related media and communication devices	**4,077,896**	3,885,594
Other income – net:		
Interest income	**119**	128
Others	**15**	484
	4,078,030	3,886,206

4. Other gain – net

	26/01/2015 to 31/01/2016 US$'000	27/01/2014 to 25/01/2015 US$'000
Exchange gains - net	**4**	19

5. Other operating expenses

	26/01/2015 to 31/01/2016 US$'000	27/01/2014 to 25/01/2015 US$'000
Consulting and professional expense	**1,344**	1,131
Utilities expense	**104**	110
Repair and maintenance expense	**283**	276
Bank charges	**117**	166
Bad debt expense/(reversal)	**157**	1,786
Insurance for credit	**1,180**	1,064
Engineering and testing expense	**713**	1,073
Restructuring expense	**340**	-
Other expenses	**457**	480
Total	**4,695**	6,086

6. Employee compensation

	26/01/2015 to 31/01/2016 US$'000	27/01/2014 to 25/01/2015 US$'000
Wages and salaries	**6,134**	5,822
Employer's contribution to defined contribution plans	**479**	428
Share-based compensation	**743**	544
	7,356	6,794

The key management is directors of the Company and there is no key management personnel remuneration during the financial year (2015: US$Nil).

7. Income tax

(a) Income tax expense

	26/01/2015 to 31/01/2016 US$'000	27/01/2014 to 25/01/2015 US$'000
Income tax expense attributable to profit is made up of:		
- Current income tax	**843**	1,282

7. Income tax (continued)

(a) Income tax expense (continued)

The tax expense on profit differs from the amount that would arise using the Singapore standard rate of income tax as explained below:

	26/01/2015 to 31/01/2016 US$'000	27/01/2014 to 25/01/2015 US$'000
Profit before tax	**3,866**	4,363
Tax calculated at a tax rate of 17% (2015: 17%)	**657**	742
Effects of:		
- Expenses not deductible for tax purposes	**207**	397
- Income not subject to tax	**(22)**	(12)
- Write off of bad debts	**(156)**	-
- Different tax rates in other country	**(19)**	(22)
- Others	**176**	177
Tax charge	**843**	1,282

(b) Movements in current tax liabilities

	31/01/2016 US$'000	25/01/2015 US$'000
Beginning of financial year	**746**	286
Tax expense on profit - current financial year	**843**	1,282
Income tax paid	**(1,519)**	(822)
End of the financial year	**70**	746

8. Cash and cash equivalents

	31/01/2016 US$'000	25/01/2015 US$'000
Cash at bank	**223,210**	66,979

9. Trade and other receivables

	31/01/2016 US$'000	25/01/2015 US$'000
Trade receivables:		
- Immediate holding company	48	48
- Non-related parties	427,874	419,802
Less: Allowance for impairment of receivables		
- Non-related parties	(1,756)	(2,546)
Trade receivables - net	426,166	417,304
Non-trade receivables:		
- Immediate holding company	185,717	190,696
- Other related companies	91,943	102,968
	277,660	293,664
Other receivables	579	130
	704,405	711,098

The non-trade receivables due from immediate holding company and other related companies are unsecured, interest-free and are repayable on demand.

10. Inventories

	31/01/2016 US$'000	25/01/2015 US$'000
Finished goods	208,883	220,854

11. Plant and equipment

	Computer equipment US$'000	Machinery and equipment US$'000	Office equipment US$'000	Furniture US$'000	Leasehold improvement US$'000	Total US$'000
2016						
Cost						
Beginning of financial year	866	1,522	28	26	197	2,639
Additions	65	12	-	-	5	82
Disposals	(410)	(1,212)	-	-	-	(1,622)
End of financial year	521	322	28	26	202	1,099
Accumulated depreciation						
Beginning of financial year	499	1,286	5	13	92	1,895
Depreciation charge	239	183	6	5	45	478
Disposals	(378)	(1,212)	-	-	-	(1,590)
End of financial year	360	257	11	18	137	783
Net book value						
End of financial year	**161**	**65**	**17**	**8**	**65**	**316**

11. Plant and equipment (continued)

	Computer equipment US$'000	Machinery and equipment US$'000	Office equipment US$'000	Furniture US$'000	Leasehold improvement US$'000	Total US$'000
2015						
Cost						
Beginning of financial year	1,996	2,651	68	26	1,257	5,998
Additions	16	24	24	-	11	75
Disposals	(1,146)	(1,153)	(64)	-	(1,071)	(3,434)
End of financial year	866	1,522	28	26	197	2,639
Accumulated depreciation						
Beginning of financial year	1,330	1,996	59	7	1,108	4,500
Depreciation charge	315	443	10	6	55	829
Disposals	(1,146)	(1,153)	(64)	-	(1,071)	(3,434)
End of financial year	499	1,286	5	13	92	1,895
Net book value						
End of financial year	**367**	**236**	**23**	**13**	**105**	**744**

12. Investment in subsidiary

	31/01/2016 US$'000	25/01/2015 US$'000
Unquoted equity shares at cost	**85**	85

Details of investment in subsidiary are as follows:

Name of company	Principal activities	Country of incorporation	Equity holding 31/01/2016 %	Equity holding 25/01/2015 %	Cost of investment 31/01/2016 US$'000	Cost of investment 25/01/2015 US$'000
NVIDIA GK	Solicitation and negotiation of orders and working on behalf of Nvidia Singapore Pte Ltd	Japan	**100**	100	**85**	85

The subsidiary does not require an audit in accordance with the laws in the country of its incorporation.

13. Loan to related party

The loan to Nvidia Brazil is unsecured, denominated in the United States Dollar and is due in full on 7 August 2023. The contractual interest rate on the loan at balance sheet date is 3.0% (2015: 3.0%) per annum. The carrying amount of the loan approximate its fair value.

14. Trade and other payables

	31/01/2016 US$'000	25/01/2015 US$'000
Trade payables		
- Immediate holding company	**864,103**	768,850
	864,103	768,850
Non-trade payables		
- Immediate holding company	**4,119**	4,157
- Other related companies	**57,657**	43,683
	61,776	47,840
Accrued operating expenses	**169,729**	146,686
Other payables	**6,108**	5,491
	175,837	152,177
	1,101,716	968,867

The non-trade payables to immediate holding company and other related companies are unsecured, interest-free and are repayable on demand.

15. Share capital

The Company's share capital comprise fully paid-up 2 (2015: 2) ordinary shares with no par value, amounting to a total of US$1,000 (2015: US$1,000).

16. Other reserves

	31/01/2016 US$'000	25/01/2015 US$'000 (restated)
Beginning of financial year	**8,781**	8,237
Employee share-based payment scheme		
- Value of employee services	**743**	544
End of financial year	**9,524**	8,781

17. Share based payments

The employees of the Company are eligible for share-based compensation under equity-settled, shared based compensation plans operated by Nvidia Corporation, the ultimate holding corporation of the Company. Awards include share options, restricted share units ("RSUs"), and employee share purchase plan ("ESPP"), are granted under the following plans:

Amended and Restated 2007 Equity Incentive Plan
The Nvidia Corporation 2007 Equity Incentive Plan was approved in 2007, and subsequently amended and restated in 2012, 2013 and 2014, or the Restated 2007 Plan. This plan authorizes the issuance of share options and restricted share units.

Share options granted to employees, subject to certain exceptions, vest over a four year period subject to continued service, and generally expire ten years from the date of grant.

Subject to certain exceptions, RSUs granted to employees vest over a four year period, subject to continued service.

1998 and 2012 Employee Share Purchase Plan
The Nvidia Corporation 2012 Employee Share Purchase Plan was approved in 2012, and subsequently amended and restated in 2014, or the Restated 2012 Plan, as the successor to the 1998 Employee Share Purchase Plan.

Employees who participate in an offering may have up to 10% of their earnings withheld to the purchase of shares of common stock. The Board may increase this percentage at its discretion, up to 15%. The price of common stock purchased under our ESPP will be equal to 85% of the lower of the fair market value of the common stock on the commencement date of each offering period and the purchase date of each offering period. Employees may end their participation in the ESPP at any time during the offering period, and participation ends automatically on termination of employment with us. In each case, the employee's contributions are refunded.

The fair value of share option awards on the date of grant is determined using the Binomial model. No options were granted during the financial years ended 25 January 2015 and 31 January 2016.

Fair value of RSUs is determined using the closing trading price of common stock on the date of grant, minus a dividend yield discount.

17. Share based payments (continued)

The following table summarises the number and weighted average fair value of ESPP granted during the period:

	2016 US$'000	2015 US$'000
Number of shares issued under the plan to participating Nvidia Singapore Pte Ltd's employees on 31 January 2016 (25 January 2015)	**32,576**	39,222

The weighted average market price during the year ended 31 January 2016 was US$22.21 (2015 US$18.89) and the shares had an average grant date fair value of US$16.10 (2015 US$12.73).

The fair value of shares issued under the ESPP has been estimated at the date of grant with the following assumptions, using the Black-Scholes model:

	2016	2015
Weighted average expected life (years)	**0.5-2.0**	0.5-2.0
Risk free interest rate	**0.1%-0.7%**	0.1%-0.5%
Volatility	**24%-34%**	23%-31%
Dividend yield	**1.5%-1.8%**	1.7%-1.9%

Nvidia Corporation uses the implied volatility as it is expected to be more reflective of market conditions and, therefore, could reasonably be expected to be a better indicator of expected volatility than historical volatility. The risk-free interest rate is based upon observed interest rates on Treasury bills appropriate for the term of employee share options.

Dividend yield is based on history and expectation of dividend pay-outs. RSU awards are not eligible for cash dividends prior to vesting; therefore, the fair value of RSUs is discounted by dividend yield. For awards granted on or subsequent to 8 November 2012, Nvidia Corporation uses a dividend yield at grant date based on the per share dividends declared during the most recent quarter.

17. Share based payments (continued)

A reconciliation of share options and RSU movements for the financial periods ended 31 January, 2016 and 25 January, 2015 is shown below (average exercise price and fair value reported in US dollars):

Share-based Payment Award Activity

The following table summarises activity for share options:

	Number of share options	Weighted average exercise price
31 January 2016		
Outstanding at beginning of the year	79,176	$14.28
Exercised*	(22,402)	$12.90
Forfeited/Cancelled	(5,761)	$14.23
Outstanding at end of the year	51,013	$14.90
Options exercisable at year end	40,245	$15.02

*The weighted average share price at the date of exercise of options exercised during the year ended 31 January 2016 was US$22.50.

	Number of share options	Weighted average exercise price
25 January 2015		
Outstanding at beginning of the year	134,929	$13.72
Exercised*	(55,003)	$12.85
Forfeited/Cancelled	(750)	$17.66
Outstanding at end of the year	79,176	$14.28
Options exercisable at year end	49,583	$14.23

*The weighted average share price at the date of exercise of options exercised during the year ended 25 January 2015 was US$19.73.

17. Share based payments (continued)

Range of exercise price details of awards outstanding as at 31 January 2016:

Range of exercise prices – US$	Outstanding options	Weighted average remaining contractual life (years)
US$10.01-US$15.00	30,854	6.36
US$15.01-US$20.00	20,159	5.32
	51,013	

Range of exercise price details of awards outstanding as at 25 January 2015:

Range of exercise prices – US$	Outstanding options	Weighted average remaining contractual life (years)
US$10.01-US$15.00	51,272	5.85
US$15.01-US$20.00	27,904	5.10
	79,176	

Restricted Share Units ("RSUs") Activity

The following table summarises activity for restricted share units:

	Number of RSUs	Weighted average grant date fair value
31 January 2016		
Outstanding at beginning of the year	79,388	$15.89
Granted	37,645	$24.12
Vested	(26,852)	$15.27
Forfeited/Cancelled	(11,783)	$16.50
Outstanding/non-vested at end of the year	78,398	$19.96

	Number of RSUs	Weighted average grant date fair value
25 January 2015		
Outstanding at beginning of the year	70,446	$13.91
Granted	38,778	$17.87
Vested	(28,765)	$13.77
Forfeited/Cancelled	(1,071)	$14.30
Outstanding/non-vested at end of the year	79,388	$15.89

18. Commitments

Operating lease commitments

The future minimum lease payable under non-cancellable operating leases contracted for at the balance sheet date but not recognised as liabilities are as follows:

	31/01/2016 US$'000	25/01/2015 US$'000
Not later than one year	**433**	433
Later than one year but not later than five years	**145**	578
	578	1,011

19. Financial risk management

Financial risk factors

The Company's activities expose it to market risk (including currency risk, interest rate risk and price risk), credit risk and liquidity risk.

The management team is responsible for setting the objectives and underlying principles of financial risk management for the Company, and establishes detailed policies such as risk identification and measurement, exposure limits and hedging strategies. Financial risk management is carried out by finance personnel.

The finance personnel measure actual exposures against the limits set and prepare regular reports for the review of the management team. The information presented below is based on information received by key management.

(a) Market risk

(i) Currency risk

The Company's revenue, cost of operations and majority of the financial assets and liabilities are primarily in United States Dollar ("USD"). The Company is not exposed to any significant exchange rate risk. The Company does not purchase forward contracts to hedge this exposure.

(ii) Price risk

The Company has insignificant exposure to equity price risk as it does not hold any equity financial assets.

19. Financial risk management (continued)

(a) Market risk (continued)

(iii) Interest rate risk

The Company is exposed to interest rate risk on its bank deposits. The bank deposits are mainly deposits in USD.

At 31 January 2016, if the USD interest rates had increased/decreased by 0.05% (2015: 0.05%) with all other variables including tax being constant, the profit after tax would have been higher/lower by US$111,000 (2015: US$33,000) as a result of higher/lower interest income on these deposits.

The Company has insignificant financial liabilities that are exposed to interest rate risks.

(b) Credit risk

Credit risk refers to the risk that a counterparty will default on its contractual obligations resulting in financial loss to the Company. The major classes of financial assets of the Company are cash and cash equivalents and trade receivables. Cash and cash equivalents are deposits in banks with sound credit ratings, therefore, the Company does not expect to have high credit risk in this regard. For trade receivables, the Company adopts the policy of dealing only with customers of appropriate credit history, and obtaining sufficient collateral where appropriate to mitigate credit risk. In addition, the Company monitors its credit risk on an ongoing basis by reviewing the debtors' aging to minimise its exposure to credit risk.

(i) Financial assets that are neither past due nor impaired

Trade receivables that are neither past due nor impaired are substantially companies with a good collection track record with the Company.

(ii) There is no other class of financial assets that is past due and/or impaired except for trade receivables

There is no financial asset that is past due and/or impaired.

19. Financial risk management (continued)

(b) Credit risk (continued)

(ii) There is no other class of financial assets that is past due and/or impaired except for trade receivables (continued)

The carrying amount of trade receivables individually determined to be impaired and the movement in the related allowance for impairment are as follows:

	31/01/2016 US$'000	25/01/2015 US$'000
Gross amount	1,756	2,546
Less: Allowance for impairment	(1,756)	(2,546)
	-	-
Beginning of financial year	2,546	760
Allowance made/(utilised)	(790)	1,786
End of financial year (Note 9)	1,756	2,546

(c) Liquidity risk

The Company manage the liquidity risk by maintaining sufficient cash and bank balances to enable it to meet its operational requirements.

(d) Capital risk

The Company's objectives when managing capital are to ensure that the Company is adequately capitalised and to maintain an optimal capital structure.

The Company is not subject to any externally imposed capital requirements.

(e) Financial instruments by category

The carrying amounts of financial assets at fair value through profit or loss are disclosed on the face of the balance sheet. The aggregate carrying amounts of loans and receivables and financial liabilities at amortised cost are as follows:

	31/01/2016 2016 US$'000	25/01/2015 2015 US$'000
Loans and receivables	927,036	777,948
Financial liabilities at amortised cost	1,101,449	968,609

19. Financial risk management (continued)

(f) Offsetting financial assets and financial liabilities

(i) Financial assets

The Company has the following financial instrument subject to enforceable master netting arrangement or similar agreement as follows:

	Related amounts set off			Related amounts not set off	
	Gross amounts - financial assets (a) US$'000	Gross amounts – financial liabilities (b) US$'000	Net amounts – financial assets presented in the balance sheet (c)=(a)-(b) US$'000	Financial assets (d) US$'000	Net amount (e)=(c)+(d) US$'000
As at 31 January 2016					
Trade and other receivables	**203,518**	**17,801**	**185,717**	**91,991**	**277,708**
Total	**203,518**	**17,801**	**185,717**	**91,991**	**277,708**
As at 25 January 2015					
Trade and other receivables	204,660	13,964	190,696	103,016	293,712
Total	204,660	13,964	190,696	103,016	293,712

(ii) Financial liabilities

The Company has the following financial instruments subject to enforceable master netting arrangements or similar agreement as follows:

	Related amounts set off			Related amounts not set off	
	Gross amounts – financial liabilities (a) US$'000	Gross amounts – financial assets (b) US$'000	Net amounts – financial liabilities presented in the balance sheet (c)=(a)-(b) US$'000	Financial liabilities (d) US$'000	Net amount (e)=(c)+(d) US$'000
As at 31 January 2016					
Trade and other payables	**1,206,644**	**367,730**	**838,914**	**86,965**	**925,879**
Total	**1,206,644**	**367,730**	**838,914**	**86,965**	**925,879**
As at 25 January 2015					
Trade and other payables	1,092,212	353,480	738,732	77,958	816,690
Total	1,092,212	353,480	738,732	77,958	816,690

20. Immediate and ultimate holding corporations

The Company's immediate holding corporation is Nvidia International Inc., incorporated in the Cayman Islands. The Company's ultimate holding corporation is Nvidia Corporation, incorporated in United States of America.

21. Related party transactions

In addition to the information disclosed elsewhere in the financial statements, the following transactions took place between the Company and related parties at terms agreed between the parties:

	26/01/2015 to 31/01/2016 US$'000	27/01/2014 to 25/01/2015 US$'000
Purchases from immediate holding company	**3,673,638**	3,483,981
Intercompany services on marketing and administrative expenses paid and payable to other related corporations	**159,282**	147,245

The Company participates in the centralised cash pooling arrangement with Nvidia Corporation, its ultimate holding company. During the year, certain balances with related companies were settled and netted off through the cash pooling arrangement.

22. New or revised accounting standards and interpretations

The Company has not early adopted any mandatory standards, amendments and Interpretations to existing standards that have been published but are only effective for the Company's accounting periods beginning on or after 1 February 2016. However, management anticipates that the adoption of these standards, amendments and interpretations will not have a material impact on the financial statements of the Company in the period of their initial adoption.

23. Comparative figures

In previous financial years, the Company has not recognised the expenses arising from share-based compensation in the financial statements. This has been corrected in the current financial year and comparative figures have been retrospectively adjusted, the financial statement line items affected are as follows:

Balance Sheet

	As previously reported US$'000	As restated US$'000	Impact US$'000
As at 25 January 2015			
Retained earnings	32,898	24,117	(8,781)
Other reserves	-	8,781	8,781
As at 27 January 2014			
Retained earnings	29,273	21,036	(8,237)
Other reserves	-	8,237	8,237

Statement of comprehensive income

	As previously reported US$'000	As restated US$'000	Impact US$'000
For the year ended 25 January 2015			
Employee compensation	6,250	6,794	544

24. Authorisation of financial statements

These financial statements were authorised for issue in accordance with a resolution of the Board of Directors of Nvidia Singapore Pte Ltd on 13 JUL 2016

EXHIBIT L

NVIDIA SINGAPORE PTE LTD

(Incorporated in Singapore. Registration Number 200003831M)

FINANCIAL STATEMENT

For the financial year ended 29 January 2017

NVIDIA SINGAPORE PTE LTD
(Incorporated in Singapore)

FINANCIAL STATEMENT
For the financial year ended 29 January 2017

Contents

NVIDIA SINGAPORE PTE LTD

DIRECTORS' STATEMENT
For the financial year ended 29 January 2017

The directors present their report to the shareholder together with the audited financial statements for the financial year ended 29 January 2017.

In the opinion of the directors,

(a) the financial statements as set out on pages 5 to 34 are drawn up so as to give a true and fair view of the financial position of the Company at 29 January 2017 and the financial performance, changes in equity and cash flows of the Company for the financial year covered by the financial statements; and

(b) at the date of this statement, there are reasonable grounds to believe that the Company will be able to pay its debts as and when they fall due.

Directors

The directors in office at the date of this statement are as follows:

Lee Kay Beng
Karen Theresa Burns
Michael John Byron
Rebecca Peters (appointed on 31 May 2016)

Arrangements to enable directors to acquire shares and debentures

Neither at the end of nor at any time during the financial year was the Company a party to any arrangement whose object was to enable the directors of the Company to acquire benefits by means of the acquisition of shares in, or debentures of, the Company or any other body corporate.

Directors' interests in shares and debentures

According to the register of directors' shareholdings, none of the directors holding office at the end of the financial year had any interest in the shares or debentures of the Company or its related corporations, except as follows:

	Holdings registered in name of Director	
	At	At
Ultimate Holding Corporation	**29.01.2017**	01.02.2016
(No. of ordinary shares)		
-Nvidia Corporation		
Michael John Byron	**37,390**	20,575
Rebecca Peters*	**468**	-

* Rebecca Peters was appointed on 31 May 2016. She held 468 ordinary shares in ultimate holding corporation as at that date.

NVIDIA SINGAPORE PTE LTD

DIRECTORS' STATEMENT
For the financial year ended 29 January 2017

Directors' interests in shares and debentures (continued)

	At 29.01.2017	At 01.02.2016
(Unissued ordinary shares under share options)		
-Nvidia Corporation		
Michael John Byron	**11,126**	23,713
Karen Theresa Burns	**47,432**	47,432
Rebecca Peters*	**1,875**	-
(Unvested restricted shares units)		
-Nvidia Corporation		
Michael John Byron	**121,302**	119,432
Karen Theresa Burns	**131,951**	128,838
Rebecca Peters*	**37,175**	-

* Rebecca Peters was appointed on 31 May 2016. She held 3,872 unissued ordinary shares under share options and 33,344 unvested restricted shares units in ultimate holding corporation as at that date.

Share options

No options were granted during the financial year to subscribe for unissued shares of the Company.

No shares were issued during the financial year by virtue of the exercise of options to take up unissued shares of the Company.

There were no unissued shares of the Company under option at the end of the financial year.

Independent auditor

The independent auditor, PricewaterhouseCoopers LLP, has expressed its willingness to accept re-appointment.

On behalf of the directors

MICHAEL JOHN BYRON
Director

27 JUN 2017

KAREN THERESA BURNS
Director

June 27th 2017

INDEPENDENT AUDITOR'S REPORT TO THE SHAREHOLDER OF NVIDIA SINGAPORE PTE LTD

Report on the Financial Statements

Our opinion
In our opinion, the accompanying financial statements of Nvidia Singapore Pte Ltd (the "Company") are properly drawn up in accordance with the provisions of the Companies Act, Chapter 50 (the "Act") and Financial Reporting Standards in Singapore ("FRSs") so as to give a true and fair view of the balance sheet of the Company as at 29 January 2017 and of the financial performance, changes in equity and cash flows of the Company for the year ended on that date.

What we have audited
The financial statements of the Company comprise:
- the balance sheet as at 29 January 2017;
- the statement of comprehensive income for the year then ended;
- the statement of changes in equity for the year then ended;
- the statement of cash flows for the year then ended; and
- the notes to the financial statements, including a summary of significant accounting policies.

Basis for Opinion
We conducted our audit in accordance with Singapore Standards on Auditing ("SSAs"). Our responsibilities under those standards are further described in the *Auditor's Responsibilities for the Audit of the Financial Statements* section of our report.

We believe that the audit evidence we have obtained is sufficient and appropriate to provide a basis for our opinion.

Independence

We are independent of the Company in accordance with the Accounting and Corporate Regulatory Authority ("ACRA") Code of Professional Conduct and Ethics for Public Accountants and Accounting Entities ("ACRA Code") together with the ethical requirements that are relevant to our audit of the financial statements in Singapore, and we have fulfilled our other ethical responsibilities in accordance with these requirements and the ACRA Code.

Other Information
Management is responsible for the other information. The other information comprises the Directors' Statement included in pages 1 to 2 but does not include the financial statements and our auditor's report thereon.

Our opinion on the financial statements does not cover the other information and we do not and will not express any form of assurance conclusion thereon.
In connection with our audit of the financial statements, our responsibility is to read the other information identified above and, in doing so, consider whether the other information is materially inconsistent with the financial statements or our knowledge obtained in the audit, or otherwise appears to be materially misstated.

If, based on the work we have performed on the other information that we obtained prior to the date of this auditor's report, we conclude that there is a material misstatement of this other information, we are required to report that fact. We have nothing to report in this regard.

Responsibilities of Management and Directors for the Financial Statements
Management is responsible for the preparation of financial statements that give a true and fair view in accordance with the provisions of the Act and FRSs, and for devising and maintaining a system of internal accounting controls sufficient to provide a reasonable assurance that assets are safeguarded against loss from unauthorised use or disposition; and transactions are properly authorised and that they are recorded as necessary to permit the preparation of true and fair financial statements and to maintain accountability of assets.
In preparing the financial statements, management is responsible for assessing the Company's ability to continue as a going concern, disclosing, as applicable, matters related to going concern and using the going concern basis of accounting unless management either intends to liquidate the Company or to cease operations, or has no realistic alternative but to do so.

The directors' responsibilities include overseeing the Company's financial reporting process.

INDEPENDENT AUDITOR'S REPORT TO THE SHAREHOLDER OF
NVIDIA SINGAPORE PTE LTD (continued)

Auditor's Responsibilities for the Audit of the Financial Statements
Our objectives are to obtain reasonable assurance about whether the financial statements as a whole are free from material misstatement, whether due to fraud or error, and to issue an auditor's report that includes our opinion. Reasonable assurance is a high level of assurance, but is not a guarantee that an audit conducted in accordance with SSAs will always detect a material misstatement when it exists. Misstatements can arise from fraud or error and are considered material if, individually or in the aggregate, they could reasonably be expected to influence the economic decisions of users taken on the basis of these financial statements.

As part of an audit in accordance with SSAs, we exercise professional judgement and maintain professional scepticism throughout the audit. We also:

- Identify and assess the risks of material misstatement of the financial statements, whether due to fraud or error, design and perform audit procedures responsive to those risks, and obtain audit evidence that is sufficient and appropriate to provide a basis for our opinion. The risk of not detecting a material misstatement resulting from fraud is higher than for one resulting from error, as fraud may involve collusion, forgery, intentional omissions, misrepresentations, or the override of internal control.
- Obtain an understanding of internal control relevant to the audit in order to design audit procedures that are appropriate in the circumstances, but not for the purpose of expressing an opinion on the effectiveness of the Company's internal control.
- Evaluate the appropriateness of accounting policies used and the reasonableness of accounting estimates and related disclosures made by management.
- Conclude on the appropriateness of management's use of the going concern basis of accounting and, based on the audit evidence obtained, whether a material uncertainty exists related to events or conditions that may cast significant doubt on the Company's ability to continue as a going concern. If we conclude that a material uncertainty exists, we are required to draw attention in our auditor's report to the related disclosures in the financial statements or, if such disclosures are inadequate, to modify our opinion. Our conclusions are based on the audit evidence obtained up to the date of our auditor's report. However, future events or conditions may cause the Company to cease to continue as a going concern.
- Evaluate the overall presentation, structure and content of the financial statements, including the disclosures, and whether the financial statements represent the underlying transactions and events in a manner that achieves fair presentation.

We communicate with the directors regarding, among other matters, the planned scope and timing of the audit and significant audit findings, including any significant deficiencies in internal control that we identify during our audit.

Report on Other Legal and Regulatory Requirements
In our opinion, the accounting and other records required by the Act to be kept by the Company have been properly kept in accordance with the provisions of the Act.

PricewaterhouseCoopers LLP
Public Accountants and Chartered Accountants
Singapore, 27 JUN 2017

NVIDIA SINGAPORE PTE LTD

STATEMENT OF COMPREHENSIVE INCOME
For the financial year ended 29 January 2017

	Note	01/02/2016 to 29/01/2017 US$'000	26/01/2015 to 31/01/2016 US$'000
Revenue	4	**5,712,400**	4,077,896
Other income - net	4	**59**	134
Other (loss)/gain - net	5	**(459)**	4
Expenses			
- Raw materials and other consumables consumed		**(5,515,265)**	(3,896,460)
- Employee compensation	7	**(7,606)**	(7,356)
- Travelling expenses		**(172)**	(200)
- Sales and marketing expenses		**(5,422)**	(5,258)
- Depreciation of plant and equipment		**(373)**	(478)
- Intercompany service expenses		**(173,322)**	(159,282)
- Rental on operating leases		**(438)**	(439)
- Other operating expenses	6	**(6,400)**	(4,695)
Total expenses		**(5,708,998)**	(4,074,168)
Profit before income tax		**3,002**	3,866
Income tax expense	8	**(543)**	(843)
Net profit after tax/ Total comprehensive income		**2,459**	3,023

The accompanying notes form an integral part of these financial statements.

NVIDIA SINGAPORE PTE LTD

BALANCE SHEET
As at 29 January 2017

	Note	29/01/2017 US$'000	31/01/2016 US$'000
ASSETS			
Current assets			
Cash and cash equivalents	9	**99,408**	223,210
Trade and other receivables	10	**962,784**	704,405
Inventories	11	**340,657**	208,883
Prepayment and deposits		**4,795**	1,052
		1,407,644	1,137,550
Non-current assets			
Plant and equipment	12	**1,066**	316
Investment in subsidiary	13	**85**	85
Loan to related party	14	**500**	500
		1,651	901
Total assets		**1,409,295**	1,138,451
LIABILITIES			
Current liabilities			
Trade and other payables	15	**1,368,702**	1,101,716
Current income tax liabilities	8	**568**	70
		1,369,270	1,101,786
Total liabilities		**1,369,270**	1,101,786
NET ASSETS		**40,025**	36,665
SHAREHOLDER'S EQUITY			
Share capital	16	**1**	1
Other reserves	17	**10,425**	9,524
Retained earnings		**29,599**	27,140
		40,025	36,665

The accompanying notes form an integral part of these financial statements.

NVIDIA SINGAPORE PTE LTD

STATEMENT OF CHANGES IN EQUITY
For the financial year ended 29 January 2017

	Share capital US$'000	Other reserves US$'000	Retained earnings US$'000	Total equity US$'000
2017				
Beginning of financial year	**1**	**9,524**	**27,140**	**36,665**
Share-based compensation	-	901	-	901
Total comprehensive income	-	-	2,459	2,459
End of financial year	**1**	**10,425**	**29,599**	**40,025**
2016				
Beginning of financial year - restated	**1**	**8,781**	**24,117**	**32,899**
Share-based compensation	-	743	-	743
Total comprehensive income	-	-	3,023	3,023
End of financial year	**1**	**9,524**	**27,140**	**36,665**

The accompanying notes form an integral part of these financial statements.

NVIDIA SINGAPORE PTE LTD

STATEMENT OF CASH FLOWS

For the financial year ended 29 January 2017

	Note	01/02/2016 to 29/01/2017 US$'000	26/01/2015 to 31/01/2016 US$'000
Cash flows from operating activities			
Net profit after tax		**2,459**	3,023
Adjustments for:			
- Tax	8	**543**	843
- Depreciation of plant and equipment	12	**373**	478
- Share-based compensation		**901**	743
- Interest income		**(68)**	(119)
Operating cash flow before working capital changes		**4,208**	4,968
Change in operating assets and liabilities			
- Trade and other receivables		**(258,379)**	6,693
- Inventories		**(131,774)**	11,971
- Trade and other payables		**266,986**	132,849
- Prepayments and deposits		**(3,743)**	1,200
Cash (used in)/generated from operations		**(122,702)**	157,681
Income tax paid		**(45)**	(1,519)
Net cash (used in) /generated from operating activities		**(122,747)**	156,162
Cash flows from investing activities			
Additions of plant and equipment	12	**(1,123)**	(82)
Proceeds from disposal of plant and equipment		**-**	32
Interest received		**68**	119
Net cash (used in)/generated from investing activities		**(1,055)**	69
Net (decrease)/increase in cash and cash equivalents		**(123,802)**	156,231
Cash and cash equivalents at beginning of financial period		**223,210**	66,979
Cash and cash equivalents at end of financial period	9	**99,408**	223,210

The accompanying notes form an integral part of these financial statements.

These notes form an integral part of and should be read in conjunction with the accompanying financial statements.

1. General information

The Company is incorporated in Singapore. The address of its registered office is 112 Robinson Road, #05-01, Singapore 068902.

The principal activity of the Company consists of sales of graphics processors and media and communication devices.

The principal activities of its subsidiary are to provide marketing services to Nvidia Singapore Pte Ltd.

2. Significant accounting policies

(a) Basis of preparation

The financial statements have been prepared in accordance with Singapore Financial Reporting Standards ("FRS") under the historical cost convention, except as disclosed in the accounting policies below.

The preparation of these financial statements in conformity with FRS requires management to exercise its judgement in the process of applying the Company's accounting policies. It also requires the use of certain critical accounting estimates and assumptions. The areas involving a higher degree of judgement or complexity, or areas where assumptions and estimates are significant to the financial statements are disclosed in Note 3.

Interpretations and amendments to published standards effective in 2016

On 01 February 2016, the Company adopted the new or amended FRS and Interpretations to FRS ("INT FRS") that are mandatory for application for the financial year. Changes to the Company's accounting policies have been made as required, in accordance with the relevant transitional provisions in the respective FRS and INT FRS.

The adoption of the above new or revised FRS did not result in any substantial changes to the Company's accounting policies nor any significant impact on these financial statements.

2. Significant accounting policies (continued)

(b) Revenue recognition

Revenue comprises the fair value of the consideration received or receivable for the sales of goods in the ordinary course of the Company's activities. Revenue is presented net of goods and services tax, rebates and discounts. Revenue is recognised as follows:

(1) Sale of goods

Revenue from sale of graphics processors and related media and communication devices is recognised when the Company has delivered the products to the customer, the customer has accepted the products and collectability of the related receivables is reasonably assured.

A sales return allowance for estimated product returns is established at the time revenue is recognised, based primarily on historical return rates.

(2) Interest income

Interest income is recognised using the effective interest method.

(c) Group accounting

These financial statements are the separate financial statements of Nvidia Singapore Pte Ltd. The Company is exempted from the preparation of consolidated financial statements as the Company is a wholly-owned subsidiary of Nvidia Corporation, a US-incorporated company which produces consolidated financial statements available for public use. The basis on which the subsidiary is accounted for is disclosed in Note 2(f). The main office of Nvidia Corporation is as follows: 2701 San Tomas Expressway, Santa Clara, CA 95050, U.S.A..

(d) Employee compensation

(1) Defined contribution plans

Defined contribution plans are post-employment benefit plans under which the Company pays fixed contributions into separate entities such as Mandatory Provident Fund and Superannuation Fund established in Hong Kong, on a mandatory, contractual or voluntary basis. The Company has no further payment obligations once the contributions have been paid. The Company's contribution to defined contribution plans are recognised as employee compensation expense when they are due.

2. Significant accounting policies (continued)

(d) Employee compensation (continued)

(2) Employee leave entitlements

Employee entitlements to annual leave are recognised when they accrue to employees. A provision is made for the estimated liability for annual leave as a result of services rendered by employees up to the balance sheet date.

(3) Share-based compensation

The employees of the Company are eligible for share-based compensation under equity-settled, shared based compensation plans operated by Nvidia Corporation, the ultimate holding corporation of the Company. Awards under these plans include stock options, restricted share units ("RSUs") and employee stock purchase plan ("ESPP").

The Company accounts for share based payments in accordance with FRS 102 – Share based payments. Share based compensation expense is measured at the grant date of the related equity awards, based on the fair value of the awards, and the expense is recognised over the vesting period, with a corresponding increase in the other reserve.

In case of exchange of employees' services, the equity-settled payment is measured at the fair value of equity instruments granted to employees. If exercisable at the time of grant, equity-settled payment is included in relevant cost or expenses at fair value at grant date and capital surplus is increased accordingly; if exercisable after service in waiting period is completed or specified performance conditions are met, the service obtained in current period is included in relevant cost and expenses at fair value based on the best estimation on quantity of exercisable equity instruments made by the Company in accordance with latest changes in the number of exercisable employees and subsequent information such as whether specified performance conditions are met, and capital surplus is increased accordingly.

Stock options

The total amount to be recognised over the vesting period is determined by reference to the fair value of the options on the date of the grant.

The fair value of stock options is determined using option pricing model. The fair value of stock appreciation right is determined using Bimonial option pricing model.

2. Significant accounting policies (continued)

(d) Employee compensation (continued)

(3) Share-based compensation (continued)

Stock options (continued)

At the balance sheet date, subsequent information indicates that the number of equity instruments which are different with previous estimates should be adjusted to the exercisable number of equity instruments on the vesting date. After vesting date, no further adjustment will be raised for relevant cost (expense) or total owner's equity which has been recognised.

Stock options granted to employees, subject to certain exceptions, vest over a four year period, subject to continued service, with 25% vesting on the anniversary of the hire date in the case of new hires or the anniversary of the date of grant in the case of grants to existing employees and 6.25% vesting at the end of each quarterly period thereafter. Stock options granted under the 2007 Plan generally expire ten years from the date of grant.

Restricted Stock Units ("RSU")

RSUs granted to employees vest four years, subject to continued service, with 25% vesting on a pre-determined date that is close to the anniversary of the date of grant and for grants made prior to 18 May 2016, 12.5% vesting semi-annually thereafter until fully vested and for grants made after 18 May 2016, 6.25% vesting quarterly thereafter until fully vested. The total amount to be recognised over the vesting period is determined by reference to the fair value of the ultimate holding corporation stock on the date of the grant.

Employee Stock Purchase Plan ("ESPP")

Employees are also eligible to participate in an offering to have up to 10% of their earnings withheld up to certain limitations and applied on specified dates determined by the Board of Directors of Nvidia Corporation to the purchase of shares of common stock. The Board of Directors of Nvidia Corporation may increase this percentage at its discretion, up to 15%. The price of common stock purchased under our ESPP will be equal to 85% of the lower of the fair market value of the common stock on the commencement date of each offering period and the purchase date of each offering period. Employees may end their participation in the ESPP at any time during the offering period, and participation ends automatically on termination of employment with us. In each case, the employee's contributions are refunded. The employee benefit is recognised in profit or loss in the period in which the employees purchase the shares under common stock under the ESPP.

2. Significant accounting policies (continued)

(e) Plant and equipment

(1) Measurement

Plant and equipment are recognised at cost and subsequently carried at cost less accumulated depreciation and accumulated impairment losses.

The cost of an item of plant and equipment initially recognised includes its purchase price and any cost that is directly attributable to bringing the asset to the location and condition necessary for it to be capable of operating in the manner intended by management. The projected cost of dismantlement, removal or restoration is recognised as part of the cost of plant and equipment if such obligation is incurred either when the item is acquired or as a consequence of using the asset during a particular period for purposes other than to produce inventories during that period.

(2) Depreciation

Depreciation on plant and equipment is calculated using the straight-line method to allocate their depreciable amounts over their estimated useful lives as follows:

	Useful lives
Machinery and equipment	3 - 5 years
Computer equipment	3 - 5 years
Office equipment	5 years
Furniture	5 years
Leasehold improvement	Lesser of estimated useful life or remaining lease term

The residual values, estimated useful lives and depreciation method of plant and equipment are reviewed, and adjusted as appropriate, at each balance sheet date. The effects of any revision are recognised in the statement of comprehensive income for the financial period in which the changes arise.

(3) Subsequent expenditure

Subsequent expenditure relating to plant and equipment that has already been recognised is added to the carrying amount of the asset only when it is probable that future economic benefits associated with the item, will flow to the Company and the cost can be reliably measured. All other repair and maintenance expense is recognised in the statement of comprehensive income when incurred.

2. Significant accounting policies (continued)

(e) Plant and equipment (continued)

(4) Disposal

On disposal of an item of plant and equipment, the difference between the net disposal proceeds and its carrying amount is recognised in the statement of comprehensive income.

(f) Investment in subsidiary

Investment in subsidiary is carried at cost less accumulated impairment losses in the Company's balance sheet. On disposal of investment in subsidiary, the difference between disposal proceeds and the carrying amount of the investment is recognised in the statement of comprehensive income.

(g) Loan to related party

Loan to related party is initially recognised at fair value plus transaction costs and subsequently carried at amortised cost using the effective interest method, less accumulated impairment losses. The Company assesses at each balance sheet date whether there is objective evidence that this financial asset is impaired and recognises an allowance for impairment when such evidence exists.

(h) Impairment of non-financial assets

Plant and equipment and investment in subsidiary are tested for impairment whenever there is any objective evidence or indication that these assets may be impaired.

For the purpose of impairment testing, the recoverable amount (i.e. the higher of the fair value less cost to sell and value in use) is determined on an individual asset basis unless the asset does not generate cash flows that are largely independent of those from other assets. If this is the case, the recoverable amount is determined for the cash-generating unit (CGU) to which the asset belongs.

If the recoverable amount of the asset (or CGU) is estimated to be less than its carrying amount, the carrying amount of the asset (or CGU) is reduced to its recoverable amount. The difference between the carrying amount and recoverable amount is recognised in the statement of comprehensive income.

2. Significant accounting policies (continued)

(h) Impairment of non-financial assets (continued)

An impairment loss for an asset is reversed if, and only if, there has been a change in the estimates used to determine the assets' recoverable amount since the last impairment loss was recognised. The carrying amount of this asset is increased to its revised recoverable amount, provided that this amount does not exceed the carrying amount that would have been determined (net of any accumulated amortisation or depreciation) had no impairment loss been recognised for the asset in prior years. A reversal of impairment loss for an asset is recognised in profit or loss.

(i) Loans and receivables
Bank balances
Trade and other receivables

Bank balances, trade and other receivables are initially recognised at fair value plus transaction cost and subsequently carried at amortised cost using the effective interest method, less allowance for impairment.

The Company assesses at each balance sheet date whether there is objective evidence that these financial assets are impaired and recognises an allowance for impairment when such evidence exists.

The carrying amount of these assets is reduced through the use of an impairment allowance account which is calculated as the difference between the carrying amount and the present value of estimated future cash flows, discounted at the original effective interest rate.

These assets are presented as current assets except for those that are expected to be realised later than 12 months after the balance sheet date, which are presented as non-current assets.

(j) Trade and other payables

Trade and other payables represent liabilities for goods and services provided to the Company prior to the end of financial year which are unpaid. They are classified as current liabilities if payment is due within one year or less (or in the normal operating cycle of the business if longer). Otherwise, they are presented as non-current liabilities.

Trade and other payables are initially recognised at fair value and subsequently carried at amortised cost, using the effective interest method.

2. Significant accounting policies (continued)

(k) Operating leases

Leases of assets where substantially all risks and rewards incidental to ownership are retained by the lessor are classified as operating leases. Payments made under operating leases (net of any incentives received from the lessor) are recognised in profit or loss on a straight-line basis over the period of the lease.

When a lease is terminated before the lease period expires, any payment made by the Company as penalty is recognised as an expense in the period in which the termination takes place.

(l) Inventories

Inventories are carried at the lower of cost and net realisable value. Cost is determined using the first-in, first-out method. Net realisable value is the estimated selling price in the ordinary course of business, less applicable variable selling expenses.

(m) Income taxes

Current income tax for current and prior periods is recognised at the amount expected to be paid to or recovered from the tax authorities, using the tax rates and tax laws that have been enacted or substantively enacted by the balance sheet date.

Deferred income tax is recognised for all temporary differences except when the deferred income tax arises from the initial recognition of an asset or liability that affects neither accounting nor taxable profit or loss at the time of the transaction.

Current and deferred income tax is measured using the tax rates and tax laws that have been enacted or substantively enacted by the balance sheet date, and are recognised as income or expenses in profit or loss, except to the extent that the tax arises from a transaction which is recognised directly in equity.

(n) Provisions for other liabilities and charges

Provisions for other liabilities and charges are recognised when the Company has a present legal or constructive obligation as a result of past events, it is more likely than not that an outflow of resources will be required to settle the obligation and the amount has been reliably estimated.

2. Significant accounting policies (continued)

(o) Currency translation

The financial statements of the Company are presented in United States Dollars, which is the functional currency of the Company.

Transactions in a currency other than United States Dollar ("foreign currency") are translated into United States Dollar using the exchange rates prevailing at the dates of the transactions. Currency translation differences resulting from the settlement of such transactions and from the translation of monetary assets and liabilities denominated in foreign currencies at the closing rates at the balance sheet date are recognised in the statement of comprehensive income.

All other foreign exchange gains and losses impacting profit or loss are presented in the statement of comprehensive income within "other gains/losses".

(p) Cash and cash equivalents

For the purpose of presentation in the cash flow statement, cash and cash equivalents include cash on hand and deposits with financial institutions which are subject to an insignificant risk of change in value.

(q) Share capital

Ordinary shares are classified as equity.

3. Critical accounting estimates, assumptions and judgements

Estimates, assumptions and judgements are continually evaluated and are based on historical experience and other factors, including expectations of future events that are believed to be reasonable under the circumstances

Share-based payments

The Company is required to expense its employees' share-based compensation options and RSUs in accordance with FRS 102 "Share-based Payment". The Company measures share-based compensation cost based on the fair value on the grant date of each option and RSU. This cost is recognised over the period during which an employee is required to provide service in exchange for the award and option or the requisite service period, usually the vesting period, and it is adjusted for actual forfeitures that occur before vesting. In order to assess the fair value of share options, the Company is required to use certain assumptions, including the forfeitures and the service period of each employee. The use of different assumptions and estimates could produce materially different estimated fair values for the share-based compensation options and related expenses.

NVIDIA SINGAPORE PTE LTD

NOTES TO FINANCIAL STATEMENTS
For the financial year ended 29 January 2017

4. Income

	01/02/2016 to 29/01/2017 US$'000	26/01/2015 to 31/01/2016 US$'000
Revenue		
Sale of graphics processors and related media and communication devices	**5,712,400**	4,077,896
Other income – net:		
Interest income	**68**	119
Others	**(9)**	15
	5,712,459	4,078,030

5. Other (loss)/gain – net

	01/02/2016 to 29/01/2017 US$'000	26/01/2015 to 31/01/2016 US$'000
Exchange (loss)/gain - net	**(459)**	4

6. Other operating expenses

	01/02/2016 to 29/01/2017 US$'000	26/01/2015 to 31/01/2016 US$'000
Consulting and professional expense	**2,157**	1,344
Utilities expense	**94**	104
Repair and maintenance expense	**390**	283
Bank charges	**71**	117
Bad debt expense/(reversal)	**541**	157
Insurance for credit	**1,249**	1,180
Engineering and testing expense	**1,500**	713
Restructuring expense	**-**	340
Other expenses	**398**	457
Total	**6,400**	4,695

7. Employee compensation

	01/02/2016 to 29/01/2017 US$'000	26/01/2015 to 31/01/2016 US$'000
Wages and salaries	6,162	6,134
Employer's contribution to defined contribution plans	543	479
Share-based compensation	901	743
	7,606	7,356

The key management is directors of the Company and there is no key management personnel remuneration during the financial year (2016: US$Nil).

8. Income tax

(a) Income tax expense

	01/02/2016 to 29/01/2017 US$'000	26/01/2015 to 31/01/2016 US$'000
Income tax expense attributable to profit is made up of:		
- Current income tax	543	843

The tax expense on profit differs from the amount that would arise using the Singapore standard rate of income tax as explained below:

	01/02/2016 to 29/01/2017 US$'000	26/01/2015 to 31/01/2016 US$'000
Profit before tax	3,002	3,866
Tax calculated at a tax rate of 17% (2016: 17%)	510	657
Effects of:		
- Expenses not deductible for tax purposes	246	207
- Income not subject to tax	(14)	(22)
- Write off of bad debts	-	(156)
- Different tax rates in other country	(15)	(19)
- Others	(184)	176
Tax charge	543	843

8. Income tax (continued)

(b) Movements in current tax liabilities

	29/01/2017 US$'000	31/01/2016 US$'000
Beginning of financial year	**70**	746
Tax expense on profit - current financial year	**543**	843
Income tax paid	**(45)**	(1,519)
End of the financial year	**568**	70

9. Cash and cash equivalents

	29/01/2017 US$'000	31/01/2016 US$'000
Cash at bank	**99,408**	223,210

10. Trade and other receivables

	29/01/2017 US$'000	31/01/2016 US$'000
Trade receivables:		
- Immediate holding company	**53**	48
- Non-related parties	**708,093**	427,874
- Other related companies	**28**	-
Less: Allowance for impairment of receivables		
- Non-related parties	**(2,297)**	(1,756)
Trade receivables - net	**705,877**	426,166
Non-trade receivables:		
- Immediate holding company	**173,486**	185,717
- Other related companies	**82,256**	91,943
	255,742	277,660
Other receivables	**1,165**	579
	962,784	704,405

The non-trade receivables due from immediate holding company and other related companies are unsecured, interest-free and are repayable on demand.

11. Inventories

	29/01/2017 US$'000	31/01/2016 US$'000
Finished goods	**340,657**	208,883

12. Plant and equipment

	Computer equipment US$'000	Machinery and equipment US$'000	Office equipment US$'000	Furniture US$'000	Leasehold improvement US$'000	Total US$'000
2017						
Cost						
Beginning of financial year	521	322	28	26	202	1,099
Additions	88	996	-	-	39	1,123
Disposals	(456)	(282)	(5)	-	-	(743)
End of financial year	153	1,036	23	26	241	1,479
Accumulated depreciation						
Beginning of financial year	360	257	11	18	137	783
Depreciation charge	139	160	4	5	65	373
Disposals	(456)	(282)	(5)	-	-	(743)
End of financial year	43	135	10	23	202	413
Net book value						
End of financial year	**110**	**901**	**13**	**3**	**39**	**1,066**

	Computer equipment US$'000	Machinery and equipment US$'000	Office equipment US$'000	Furniture US$'000	Leasehold improvement US$'000	Total US$'000
2016						
Cost						
Beginning of financial year	866	1,522	28	26	197	2,639
Additions	65	12	-	-	5	82
Disposals	(410)	(1,212)	-	-	-	(1,622)
End of financial year	521	322	28	26	202	1,099
Accumulated depreciation						
Beginning of financial year	499	1,286	5	13	92	1,895
Depreciation charge	239	183	6	5	45	478
Disposals	(378)	(1,212)	-	-	-	(1,590)
End of financial year	360	257	11	18	137	783
Net book value						
End of financial year	**161**	**65**	**17**	**8**	**65**	**316**

13. Investment in subsidiary

	29/01/2017 US$'000	31/01/2016 US$'000
Unquoted equity shares at cost	85	85

Details of investment in subsidiary are as follows:

Name of company	Principal activities	Country of incorporation	Equity holding		Cost of investment	
			29/01/2017 %	31/01/2016 %	29/01/2017 US$'000	31/01/2016 US$'000
NVIDIA GK	Provide marketing services to Nvidia Singapore Pte Ltd	Japan	100	100	85	85

The subsidiary does not require an audit in accordance with the laws in the country of its incorporation.

14. Loan to related party

The loan to Nvidia Brazil is unsecured, denominated in the United States Dollar and is due in full on 7 August 2023. The contractual interest rate on the loan at balance sheet date is 3.0% (2016: 3.0%) per annum. The carrying amount of the loan approximate its fair value.

15. Trade and other payables

	29/01/2017 US$'000	31/01/2016 US$'000
Trade payables		
- Immediate holding company	1,091,874	864,103
	1,091,874	864,103
Non-trade payables		
- Immediate holding company	4,218	4,119
- Other related companies	71,616	57,657
	75,834	61,776
Accrued operating expenses	196,613	169,729
Other payables	4,381	6,108
	200,994	175,837
	1,368,702	1,101,716

The non-trade payables to immediate holding company and other related companies are unsecured, interest-free and are repayable on demand.

16. Share capital

The Company's share capital comprise fully paid-up 2 (2016: 2) ordinary shares with no par value, amounting to a total of US$1,000 (2016: US$1,000).

17. Other reserves

	29/01/2017 US$'000	31/01/2016 US$'000
Beginning of financial year	9,524	8,781
Employee share-based payment scheme		
- Value of employee services	901	743
End of financial year	10,425	9,524

18. Share based payments

The employees of the Company are eligible for share-based compensation under equity-settled, shared based compensation plans operated by Nvidia Corporation, the ultimate holding corporation of the Company. Awards include share options, restricted share units ("RSUs"), and employee share purchase plan ("ESPP"), are granted under the following plans:

Amended and Restated 2007 Equity Incentive Plan
The Nvidia Corporation 2007 Equity Incentive Plan was approved in 2007, and subsequently amended and restated in 2012, 2013 and 2014, or the Restated 2007 Plan. This plan authorizes the issuance of share options and restricted share units.

Share options granted to employees, subject to certain exceptions, vest over a four year period subject to continued service, and generally expire ten years from the date of grant.

Subject to certain exceptions, RSUs granted to employees vest over a four year period, subject to continued service.

1998 and 2012 Employee Share Purchase Plan
The Nvidia Corporation 2012 Employee Share Purchase Plan was approved in 2012, and subsequently amended and restated in 2014, or the Restated 2012 Plan, as the successor to the 1998 Employee Share Purchase Plan.

Employees who participate in an offering may have up to 10% of their earnings withheld to the purchase of shares of common stock. The Board may increase this percentage at its discretion, up to 15%. The price of common stock purchased under our ESPP will be equal to 85% of the lower of the fair market value of the common stock on the commencement date of each offering period and the purchase date of each offering period. Employees may end their participation in the ESPP at any time during the offering period, and participation ends automatically on termination of employment with us. In each case, the employee's contributions are refunded.

The fair value of share option awards on the date of grant is determined using the Binomial model.

Fair value of RSUs is determined using the closing trading price of common stock on the date of grant, minus a dividend yield discount.

18. Share based payments (continued)

The following table summarises the number and weighted average fair value of ESPP granted during the period:

	2017	2016
Number of shares issued under the plan to participating Nvidia Singapore Pte Ltd's employees on 29 January 2017 (31 January 2016)	**24,694**	32,576

The weighted average market price during the year ended 29 January 2017 was US$46.57(2016: US$22.21) and the shares had an average grant date fair value of US$21.66 (2016: US$16.10).

The fair value of shares issued under the ESPP has been estimated at the date of grant with the following assumptions, using the Black-Scholes model:

	2017	2016
Weighted average expected life (years)	**0.5-2.0**	0.5-2.0
Risk free interest rate	**0.5%-0.9%**	0.1%-0.7%
Volatility	**30%-39%**	24%-34%
Dividend yield	**0.7%-1.4%**	1.5%-1.8%

Nvidia Corporation uses the implied volatility as it is expected to be more reflective of market conditions and, therefore, could reasonably be expected to be a better indicator of expected volatility than historical volatility. The risk-free interest rate is based upon observed interest rates on Treasury bills appropriate for the term of employee share options.

Dividend yield is based on history and expectation of dividend pay-outs. RSU awards are not eligible for cash dividends prior to vesting; therefore, the fair value of RSUs is discounted by dividend yield. For awards granted on or subsequent to 8 November 2012, Nvidia Corporation uses a dividend yield at grant date based on the per share dividends declared during the most recent quarter.

18. Share based payments (continued)

A reconciliation of share options and RSU movements for the financial periods ended 29 January 2017 and 31 January 2016 is shown below (average exercise price and fair value reported in US dollars):

Share-based Payment Award Activity

The following table summarises activity for share options:

	Number of share options	Weighted average exercise price
29 January 2017		
Outstanding at beginning of the year	51,013	$14.90
Exercised*	(17,144)	$15.41
Outstanding at end of the year	33,869	$14.64
Options exercisable at year end	31,036	$14.59

*The weighted average share price at the date of exercise of options exercised during the year ended 29 January 2017 was US$47.04. No options were granted during the financial years ended 29 January 2017.

	Number of share options	Weighted average exercise price
31 January 2016		
Outstanding at beginning of the year	79,176	$14.28
Exercised*	(22,402)	$12.90
Forfeited/Cancelled	(5,761)	$14.23
Outstanding at end of the year	51,013	$14.90
Options exercisable at year end	40,245	$15.02

*The weighted average share price at the date of exercise of options exercised during the year ended 31 January 2016 was US$22.50. No options were granted during the financial years ended 31 January 2016.

18. Share based payments (continued)

Range of exercise price details of awards outstanding as at 29 January 2017:

Range of exercise prices – US$	Outstanding options	Weighted average remaining contractual life (years)
US$10.01-US$15.00	22,917	5.52
US$15.01-US$20.00	10,952	5.43
	33,869	

Range of exercise price details of awards outstanding as at 31 January 2016:

Range of exercise prices – US$	Outstanding options	Weighted average remaining contractual life (years)
US$10.01-US$15.00	30,854	6.36
US$15.01-US$20.00	20,159	5.32
	51,013	

Restricted Share Units ("RSUs") Activity

The following table summarises activity for restricted share units:

	Number of RSUs	Weighted average grant date fair value
29 January 2017		
Outstanding at beginning of the year	78,398	$19.96
Granted	17,790	$57.71
Vested	(28,630)	$18.21
Forfeited/Cancelled	(393)	$13.98
Outstanding/non-vested at end of the year	67,165	$30.77

	Number of RSUs	Weighted average grant date fair value
31 January 2016		
Outstanding at beginning of the year	79,388	$15.89
Granted	37,645	$24.12
Vested	(26,852)	$15.27
Forfeited/Cancelled	(11,783)	$16.50
Outstanding/non-vested at end of the year	78,398	$19.96

19. Commitments

Operating lease commitments

The future minimum lease payable under non-cancellable operating leases contracted for at the balance sheet date but not recognised as liabilities are as follows:

	29/01/2017 US$'000	31/01/2016 US$'000
Not later than one year	**144**	433
Later than one year but not later than five years	**-**	145
	144	578

20. Financial risk management

Financial risk factors

The Company's activities expose it to market risk (including currency risk, price risk and interest rate), credit risk and liquidity risk.

The management team is responsible for setting the objectives and underlying principles of financial risk management for the Company, and establishes detailed policies such as risk identification and measurement, exposure limits and hedging strategies. Financial risk management is carried out by finance personnel.

The finance personnel measure actual exposures against the limits set and prepare regular reports for the review of the management team. The information presented below is based on information received by key management.

(a) Market risk

(i) Currency risk

The Company's revenue, cost of operations and majority of the financial assets and liabilities are primarily in United States Dollar ("USD"). The Company is not exposed to any significant currency risk. The Company did not enter into currency forward contracts to hedge this exposure.

(ii) Price risk

The Company has insignificant exposure to equity price risk as it does not hold any equity financial assets.

20. Financial risk management (continued)

(a) Market risk (continued)

(iii) Interest rate risk

The Company is exposed to interest rate risk on its bank deposits. The bank deposits are mainly deposits in USD.

At 29 January 2017, if the USD interest rates had increased/decreased by 0.05% (2016: 0.05%) with all other variables including tax being constant, the profit after tax would have been higher/lower by US$49,000 (2016: US$111,000) as a result of higher/lower interest income on these deposits.

The Company has insignificant financial liabilities that are exposed to interest rate risks.

(b) Credit risk

Credit risk refers to the risk that a counterparty will default on its contractual obligations resulting in financial loss to the Company. The major classes of financial assets of the Company are cash and cash equivalents and trade receivables. Cash and cash equivalents are deposits in banks with sound credit ratings, therefore, the Company does not expect to have high credit risk in this regard.

For trade receivables, the Company adopts the policy of dealing only with customers of appropriate credit history, and obtaining sufficient collateral where appropriate to mitigate credit risk. In addition, the Company monitors its credit risk on an ongoing basis by reviewing the debtors' aging to minimise its exposure to credit risk.

Credit exposure to an individual customer is restricted by the credit limit approved by the credit controller. Customers' payment profile and credit exposure are continuously monitored by the credit controller and reported to the management and Board of Directors. The Company's trade receivables include 5 debtors (2016: 5 debtors) that represented 42.3% and 39.0% of trade receivables at balance sheet date, respectively.

(i) Financial assets that are neither past due nor impaired

Trade receivables that are neither past due nor impaired are substantially companies with a good collection track record with the Company.

20. Financial risk management (continued)

(b) Credit risk (continued)

(ii) Financial assets that are past due and/or impaired

There is no other class of financial assets that is past due and/or impaired except for trade receivables.

The carrying amount of trade receivables individually determined to be impaired and the movement in the related allowance for impairment are as follows:

	29/01/2017 US$'000	31/01/2016 US$'000
Gross amount	**2,297**	1,756
Less: Allowance for impairment	**(2,297)**	(1,756)
	-	-
Beginning of financial year	**1,756**	2,546
Allowance made/(utilised)	**541**	(790)
End of financial year (Note 10)	**2,297**	1,756

(c) Liquidity risk

The Company manage the liquidity risk by maintaining sufficient cash and bank balances to enable it to meet its operational requirements.

(d) Capital risk

The Company's objectives when managing capital are to ensure that the Company is adequately capitalised and to maintain an optimal capital structure.

The Company is not subject to any externally imposed capital requirements.

(e) Financial instruments by category

The carrying amounts of financial assets measured at fair value through profit or loss are disclosed on the face of the balance sheet. The aggregate carrying amounts of loans and receivables and financial liabilities at amortised cost are as follows:

	29/01/2017 US$'000	31/01/2016 US$'000
Loans and receivables	**1,061,617**	927,036
Financial liabilities at amortised cost	**1,368,428**	1,101,449

20. Financial risk management (continued)

(f) Offsetting financial assets and financial liabilities

(i) Financial assets

The Company has the following financial instrument subject to enforceable master netting arrangement or similar agreement as follows:

	Related amounts set off			Related amounts not set off	
	Gross amounts - financial assets (a) US$'000	Gross amounts - financial liabilities (b) US$'000	Net amounts - financial assets presented in the balance sheet (c)=(a)-(b) US$'000	Financial assets (d) US$'000	Net amount (e)=(c)+(d) US$'000
As at 29 January 2017					
Trade and other receivables	**209,731**	**36,245**	**173,486**	**82,337**	**255,823**
Total	**209,731**	**36,245**	**173,486**	**82,337**	**255,823**
As at 31 January 2016					
Trade and other receivables	203,518	17,801	185,717	91,991	277,708
Total	203,518	17,801	185,717	91,991	277,708

(ii) Financial liabilities

The Company has the following financial instruments subject to enforceable master netting arrangements or similar agreement as follows:

	Related amounts set off			Related amounts not set off	
	Gross amounts – financial liabilities (a) US$'000	Gross amounts – financial assets (b) US$'000	Net amounts – financial liabilities presented in the balance sheet (c)=(a)-(b) US$'000	Financial liabilities (d) US$'000	Net amount (e)=(c)+(d) US$'000
As at 29 January 2017					
Trade and other payables	**1,034,886**	**389,421**	**645,465**	**522,243**	**1,167,708**
Total	**1,034,886**	**389,421**	**645,465**	**522,243**	**1,167,708**
As at 31 January 2016					
Trade and other payables	1,206,644	367,730	838,914	86,965	925,879
Total	1,206,644	367,730	838,914	86,965	925,879

21. Immediate and ultimate holding corporations

The Company's immediate holding corporation is Nvidia International Inc., incorporated in the Cayman Islands. The Company's ultimate holding corporation is Nvidia Corporation, incorporated in United States of America.

22. Related party transactions

In addition to the information disclosed elsewhere in the financial statements, the following transactions took place between the Company and related parties at terms agreed between the parties:

	01/02/2016 to 29/01/2017 US$'000	26/01/2015 to 31/01/2016 US$'000
Purchases from immediate holding company	**5,049,758**	3,673,638
Payments made on behalf of a fellow subsidiary	**381**	25
Sales of goods to a fellow subsidiary	**28**	-
Sales of goods to immediate holding company	**5**	-
Intercompany expense on marketing and administrative services	**173,322**	159,282

The Company participates in the centralised cash pooling arrangement with Nvidia Corporation, its ultimate holding company. During the year, certain balances with related companies were settled and netted off through the cash pooling arrangement.

23. New or revised accounting standards and interpretations

Below are the mandatory standards, amendments and interpretations to existing standards that have been published, and are relevant for the Group's accounting periods beginning on or after 30 January 2017 and which the Group has not early adopted:

23. New or revised accounting standards and interpretations (continued)

- FRS 115 Revenue from contracts with customers (effective for annual periods beginning on or after 1 January 2018)

  This is the converged standard on revenue recognition. It replaces FRS 11 Construction contracts, FRS 18 Revenue, and related interpretations. Revenue is recognised when a customer obtains control of a good or service. A customer obtains control when it has the ability to direct the use of and obtain the benefits from the good or service. The core principle of FRS 115 is that an entity recognises revenue to depict the transfer of promised goods or services to customers in an amount that reflects the consideration to which the entity expects to be entitled in exchange for those goods or services. An entity recognises revenue in accordance with that core principle by applying the following steps:

  - Step 1: Identify the contract(s) with a customer
  - Step 2: Identify the performance obligations in the contract
  - Step 3: Determine the transaction price
  - Step 4: Allocate the transaction price to the performance obligations in the contract
  - Step 5: Recognise revenue when (or as) the entity satisfies a performance obligation

  FRS 115 also includes a cohesive set of disclosure requirements that will result in an entity providing users of financial statements with comprehensive information about the nature, amount, timing and uncertainty of revenue and cash flows arising from the entity's contracts with customers.

  Management is currently assessing the effects of applying the new standard on the Company's financial statements and has identified the following areas that are likely to be affected:

  (i) Rights of return – FRS 115 requires separate presentation on the balance sheet of the right to recover the goods from the customer and the refund obligation; and

  (ii) Accounting for certain costs incurred in fulfilling a contract – certain costs which are currently expensed may need to be recognised as an asset under FRS 115.

  At this stage, the Company is not able to estimate the impact of the new rules on the Company's financial statements. The Company will make more detailed assessment of the impact over the next twelve months.

23. New or revised accounting standards and interpretations (continued)

- FRS 116 Leases (effective for annual periods beginning on or after 1 January 2019)

  FRS 116 will result in almost all leases being recognised on the balance sheet, as the distinction between operating and finance leases is removed. Under the new standard, an asset (the right to use the leased item) and a financial liability to pay rentals are recognised. The only exceptions are short-term and low-value leases. The accounting for lessors will not change significantly.

  The standard will affect primarily the accounting for the Company's operating leases. As at the balance sheet date, the Company has non-cancellable operating lease commitments of US$144,000 (Note 19). As at the reporting date, the Company has non-cancellable operating lease commitments of US$144,000. However, the Company has yet to determine to what extent these commitments will result in the recognition of an asset and a liability for future payments and how this will affect the Company's profit and classification of cash flows.

  Some of the commitments may be covered by the exception for short-term and low-value leases and some commitments may relate to arrangements that will not qualify as leases under FRS 116.

24. Authorisation of financial statements

These financial statements were authorised for issue in accordance with a resolution of the Board of Directors of Nvidia Singapore Pte Ltd on 27 JUN 2017

EXHIBIT M

f1900f1b5ae5e37276952a887e08e80a9bc895a99e88d03d250f197c81d664a0

NVIDIA SINGAPORE PTE LTD

(Incorporated in Singapore. Registration Number 200003831M)

FINANCIAL STATEMENT

For the financial year ended 28 January 2018

f1908f1b5ae5e3727695...

NVIDIA SINGAPORE PTE LTD
(Incorporated in Singapore)

FINANCIAL STATEMENT
For the financial year ended 28 January 2018

Contents

NVIDIA SINGAPORE PTE LTD

DIRECTORS' STATEMENT
For the financial year ended 28 January 2018

The directors present their report to the shareholder together with the audited financial statements for the financial year ended 28 January 2018.

In the opinion of the directors,

(a) the financial statements as set out on pages 5 to 35 are drawn up so as to give a true and fair view of the financial position of the Company at 28 January 2018 and the financial performance, changes in equity and cash flows of the Company for the financial year covered by the financial statements; and

(b) at the date of this statement, there are reasonable grounds to believe that the Company will be able to pay its debts as and when they fall due.

Directors

The directors in office at the date of this statement are as follows:

Lee Kay Beng
Karen Theresa Burns
Michael John Byron
Rebecca Peters

Arrangements to enable directors to acquire shares and debentures

Neither at the end of nor at any time during the financial year was the Company a party to any arrangement whose object was to enable the directors of the Company to acquire benefits by means of the acquisition of shares in, or debentures of, the Company or any other body corporate.

Directors' interests in shares and debentures

According to the register of directors' shareholdings, none of the directors holding office at the end of the financial year had any interest in the shares or debentures of the Company or its related corporations, except as follows:

	Holdings registered in name of Director	
	At	At
Ultimate Holding Corporation	**28.01.2018**	29.01.2017
(No. of ordinary shares)		
-Nvidia Corporation		
Michael John Byron	**28,945**	37,390
Rebecca Peters	**1,974**	468

E1904E1b5ae5e3727695Za887e08e80A9bc895a99e88d03d250E197c810664a0

Directors' interests in shares and debentures (continued)

	At 28.01.2018	At 29.01.2017
(Unissued ordinary shares under share options)		
-Nvidia Corporation		
Michael John Byron	**9,018**	11,126
Karen Theresa Burns	**31,275**	47,432
Rebecca Peters	**1,875**	1,875
(Unvested restricted shares units)		
-Nvidia Corporation		
Michael John Byron	**84,858**	121,302
Karen Theresa Burns	**93,151**	131,951
Rebecca Peters	**27,091**	37,175

Share options

No options were granted during the financial year to subscribe for unissued shares of the Company.

No shares were issued during the financial year by virtue of the exercise of options to take up unissued shares of the Company.

There were no unissued shares of the Company under option at the end of the financial year.

Independent auditor

The independent auditor, PricewaterhouseCoopers LLP, has expressed its willingness to accept re-appointment.

On behalf of the directors June 12th 2018

MICHAEL JOHN BYRON
Director

KAREN THERESA BURNS
Director

INDEPENDENT AUDITOR'S REPORT TO THE SHAREHOLDER OF NVIDIA SINGAPORE PTE LTD

Report on the Financial Statements

Our opinion
In our opinion, the accompanying financial statements of Nvidia Singapore Pte Ltd (the "Company") are properly drawn up in accordance with the provisions of the Companies Act, Chapter 50 (the "Act") and Financial Reporting Standards in Singapore ("FRSs") so as to give a true and fair view of the balance sheet of the Company as at 28 January 2018 and of the financial performance, changes in equity and cash flows of the Company for the year ended on that date.

What we have audited
The financial statements of the Company comprise:
- the balance sheet as at 28 January 2018;
- the statement of comprehensive income for the year then ended;
- the statement of changes in equity for the year then ended;
- the statement of cash flows for the year then ended; and
- the notes to the financial statements, including a summary of significant accounting policies.

Basis for Opinion
We conducted our audit in accordance with Singapore Standards on Auditing ("SSAs"). Our responsibilities under those standards are further described in the *Auditor's Responsibilities for the Audit of the Financial Statements* section of our report.

We believe that the audit evidence we have obtained is sufficient and appropriate to provide a basis for our opinion.

Independence

We are independent of the Company in accordance with the Accounting and Corporate Regulatory Authority Code of Professional Conduct and Ethics for Public Accountants and Accounting Entities ("ACRA Code") together with the ethical requirements that are relevant to our audit of the financial statements in Singapore, and we have fulfilled our other ethical responsibilities in accordance with these requirements and the ACRA Code.

Other Information
Management is responsible for the other information. The other information comprises the Directors' Statement but does not include the financial statements and our auditor's report thereon.

Our opinion on the financial statements does not cover the other information and we do not and will not express any form of assurance conclusion thereon.

In connection with our audit of the financial statements, our responsibility is to read the other information and, in doing so, consider whether the other information is materially inconsistent with the financial statements or our knowledge obtained in the audit, or otherwise appears to be materially misstated. If, based on the work we have performed, we conclude that there is a material misstatement of this other information, we are required to report that fact. We have nothing to report in this regard.

Responsibilities of Management and Directors for the Financial Statements
Management is responsible for the preparation of financial statements that give a true and fair view in accordance with the provisions of the Act and FRSs, and for devising and maintaining a system of internal accounting controls sufficient to provide a reasonable assurance that assets are safeguarded against loss from unauthorised use or disposition; and transactions are properly authorised and that they are recorded as necessary to permit the preparation of true and fair financial statements and to maintain accountability of assets.

In preparing the financial statements, management is responsible for assessing the Company's ability to continue as a going concern, disclosing, as applicable, matters related to going concern and using the going concern basis of accounting unless management either intends to liquidate the Company or to cease operations, or has no realistic alternative but to do so.

The directors' responsibilities include overseeing the Company's financial reporting process.

INDEPENDENT AUDITOR'S REPORT TO THE SHAREHOLDER OF NVIDIA SINGAPORE PTE LTD (continued)

Auditor's Responsibilities for the Audit of the Financial Statements

Our objectives are to obtain reasonable assurance about whether the financial statements as a whole are free from material misstatement, whether due to fraud or error, and to issue an auditor's report that includes our opinion. Reasonable assurance is a high level of assurance, but is not a guarantee that an audit conducted in accordance with SSAs will always detect a material misstatement when it exists. Misstatements can arise from fraud or error and are considered material if, individually or in the aggregate, they could reasonably be expected to influence the economic decisions of users taken on the basis of these financial statements.

As part of an audit in accordance with SSAs, we exercise professional judgement and maintain professional scepticism throughout the audit. We also:

- Identify and assess the risks of material misstatement of the financial statements, whether due to fraud or error, design and perform audit procedures responsive to those risks, and obtain audit evidence that is sufficient and appropriate to provide a basis for our opinion. The risk of not detecting a material misstatement resulting from fraud is higher than for one resulting from error, as fraud may involve collusion, forgery, intentional omissions, misrepresentations, or the override of internal control.

- Obtain an understanding of internal control relevant to the audit in order to design audit procedures that are appropriate in the circumstances, but not for the purpose of expressing an opinion on the effectiveness of the Company's internal control.

- Evaluate the appropriateness of accounting policies used and the reasonableness of accounting estimates and related disclosures made by management.

- Conclude on the appropriateness of management's use of the going concern basis of accounting and, based on the audit evidence obtained, whether a material uncertainty exists related to events or conditions that may cast significant doubt on the Company's ability to continue as a going concern. If we conclude that a material uncertainty exists, we are required to draw attention in our auditor's report to the related disclosures in the financial statements or, if such disclosures are inadequate, to modify our opinion. Our conclusions are based on the audit evidence obtained up to the date of our auditor's report. However, future events or conditions may cause the Company to cease to continue as a going concern.

- Evaluate the overall presentation, structure and content of the financial statements, including the disclosures, and whether the financial statements represent the underlying transactions and events in a manner that achieves fair presentation.

We communicate with the directors regarding, among other matters, the planned scope and timing of the audit and significant audit findings, including any significant deficiencies in internal control that we identify during our audit.

Report on Other Legal and Regulatory Requirements

In our opinion, the accounting and other records required by the Act to be kept by the Company have been properly kept in accordance with the provisions of the Act.

PricewaterhouseCoopers LLP
Public Accountants and Chartered Accountants
Singapore, 12 JUN 2018

NVIDIA SINGAPORE PTE LTD

STATEMENT OF COMPREHENSIVE INCOME
For the financial year ended 28 January 2018

	Note	30/01/2017 to 28/01/2018 US$'000	01/02/2016 to 29/01/2017 US$'000
Revenue	4	**8,064,722**	5,712,400
Other income - net	4	**96**	59
Other losses - net	5	**(490)**	(459)
Expenses			
- Raw materials and other consumables consumed		**(7,830,208)**	(5,515,265)
- Employee compensation	7	**(9,043)**	(7,606)
- Travelling expenses		**(237)**	(172)
- Sales and marketing expenses		**(4,063)**	(5,422)
- Depreciation of property, plant and equipment		**(319)**	(373)
- Intercompany service expenses		**(210,571)**	(173,322)
- Rental on operating leases		**(511)**	(438)
- Other operating expenses	6	**(6,789)**	(6,400)
Total expenses		**(8,061,741)**	(5,708,998)
Profit before income tax		**2,587**	3,002
Income tax expense	8	**(790)**	(543)
Net profit after tax/ Total comprehensive income		**1,797**	2,459

The accompanying notes form an integral part of these financial statements.

NVIDIA SINGAPORE PTE LTD

BALANCE SHEET
As at 28 January 2018

	Note	28/01/2018 US$'000	29/01/2017 US$'000
ASSETS			
Current assets			
Cash and cash equivalents	9	**7,509**	99,408
Trade and other receivables	10	**1,130,755**	962,784
Inventories	11	**335,435**	340,657
Prepayment and deposits		**4,619**	4,795
		1,478,318	1,407,644
Non-current assets			
Property, plant and equipment	12	**845**	1,066
Investment in subsidiary	13	**85**	85
Loan to related party	14	**-**	500
		930	1,651
Total assets		**1,479,248**	1,409,295
LIABILITIES			
Current liabilities			
Trade and other payables	15	**1,434,950**	1,368,702
Current income tax liabilities	8	**781**	568
		1,435,731	1,369,270
Total liabilities		**1,435,731**	1,369,270
NET ASSETS		**43,517**	40,025
SHAREHOLDER'S EQUITY			
Share capital	16	**1**	1
Other reserves	17	**12,120**	10,425
Retained earnings		**31,396**	29,599
		43,517	40,025

The accompanying notes form an integral part of these financial statements.

NVIDIA SINGAPORE PTE LTD

STATEMENT OF CHANGES IN EQUITY

For the financial year ended 28 January 2018

	Share capital US$'000	Other reserves US$'000	Retained earnings US$'000	Total equity US$'000
2018				
Beginning of financial year	**1**	**10,425**	**29,599**	**40,025**
Share-based compensation	-	1,695	-	1,695
Total comprehensive income	-	-	1,797	1,797
End of financial year	**1**	**12,120**	**31,396**	**43,517**
2017				
Beginning of financial year	**1**	**9,524**	**27,140**	**36,665**
Share-based compensation	-	901	-	901
Total comprehensive income	-	-	2,459	2,459
End of financial year	**1**	**10,425**	**29,599**	**40,025**

The accompanying notes form an integral part of these financial statements.

NVIDIA SINGAPORE PTE LTD

STATEMENT OF CASH FLOWS

For the financial year ended 28 January 2018

	Note	30/01/2017 to 28/01/2018 US$'000	01/02/2016 to 29/01/2017 US$'000
Cash flows from operating activities			
Net profit after tax		**1,797**	2,459
Adjustments for:			
- Tax	8	**790**	543
- Depreciation of property, plant and equipment	12	**319**	373
- Share-based compensation	7	**1,695**	901
- Interest income	4	**-**	(68)
Operating cash flow before working capital changes		**4,601**	4,208
Change in operating assets and liabilities			
- Trade and other receivables		**(167,971)**	(258,379)
- Inventories		**5,222**	(131,774)
- Trade and other payables		**66,248**	266,986
- Prepayments and deposits		**176**	(3,743)
Cash used in operations		**(91,724)**	(122,702)
Income tax paid	8	**(577)**	(45)
Net cash used in operating activities		**(92,301)**	(122,747)
Cash flows from investing activities			
Additions of property, plant and equipment	12	**(98)**	(1,123)
Repayment of loan to related party	14	**500**	-
Interest received		**-**	68
Net cash generated/(used in) from investing activities		**402**	(1,055)
Net decrease in cash and cash equivalents		**(91,899)**	(123,802)
Cash and cash equivalents at beginning of financial period		**99,408**	223,210
Cash and cash equivalents at end of financial period	9	**7,509**	99,408

The accompanying notes form an integral part of these financial statements.

These notes form an integral part of and should be read in conjunction with the accompanying financial statements.

1. General information

The Company is incorporated in Singapore. The address of its registered office is 112 Robinson Road, #05-01, Singapore 068902.

The principal activity of the Company consists of sales of graphics processors and media and communication devices.

The principal activities of its subsidiary are to provide marketing services to Nvidia Singapore Pte Ltd.

2. Significant accounting policies

(a) Basis of preparation

The financial statements have been prepared in accordance with Singapore Financial Reporting Standards ("FRS") under the historical cost convention, except as disclosed in the accounting policies below.

The preparation of these financial statements in conformity with FRS requires management to exercise its judgement in the process of applying the Company's accounting policies. It also requires the use of certain critical accounting estimates and assumptions. The areas involving a higher degree of judgement or complexity, or areas where assumptions and estimates are significant to the financial statements are disclosed in Note 3.

Interpretations and amendments to published standards effective in 2017

On 30 January 2017, the Company adopted the new or amended FRS and Interpretations to FRS ("INT FRS") that are mandatory for application for the financial year. Changes to the Company's accounting policies have been made as required, in accordance with the relevant transitional provisions in the respective FRS and INT FRS.

The adoption of the above new or revised FRS did not result in any substantial changes to the Company's accounting policies nor any significant impact on these financial statements.

2. Significant accounting policies (continued)

(a) Basis of preparation (continued)

FRS 7 Statement of cash flows

The amendments to FRS 7 Statement of cash flows (Disclosure initiative) sets out required disclosures that enable users of financial statements to evaluate changes in liabilities arising from financing activities, including both changes arising from cash flows and non-cash changes.

The Company does not need to include the additional required disclosures in the Statement of Cash Flows to the Financial Statement.

(b) Revenue recognition

Revenue comprises the fair value of the consideration received or receivable for the sales of goods in the ordinary course of the Company's activities. Revenue is presented net of goods and services tax, rebates and discounts. Revenue is recognised as follows:

(1) Sale of goods

Revenue from sale of graphics processors and related media and communication devices is recognised when the Company has delivered the products to the customer, the customer has accepted the products and collectability of the related receivables is reasonably assured.

A sales return allowance for estimated product returns is established at the time revenue is recognised, based primarily on historical return rates.

(2) Interest income

Interest income is recognised using the effective interest method.

(c) Group accounting

These financial statements are the separate financial statements of Nvidia Singapore Pte Ltd. The Company is exempted from the preparation of consolidated financial statements as the Company is a wholly-owned subsidiary of Nvidia Corporation, a US-incorporated company which produces consolidated financial statements available for public use. The basis on which the subsidiary is accounted for is disclosed in Note 2(f). The main office of Nvidia Corporation is as follows: 2788 San Tomas Expressway, Santa Clara, CA 95051, U.S.A..

2. Significant accounting policies (continued)

(d) Employee compensation

(1) Defined contribution plans

Defined contribution plans are post-employment benefit plans under which the Company pays fixed contributions into separate entities such as Mandatory Provident Fund and Superannuation Fund established in Hong Kong, on a mandatory, contractual or voluntary basis. The Company has no further payment obligations once the contributions have been paid. The Company's contribution to defined contribution plans are recognised as employee compensation expense when they are due.

(2) Employee leave entitlements

Employee entitlements to annual leave are recognised when they accrue to employees. A provision is made for the estimated liability for annual leave as a result of services rendered by employees up to the balance sheet date.

(3) Share-based compensation

The employees of the Company are eligible for share-based compensation under equity-settled, shared based compensation plans operated by Nvidia Corporation, the ultimate holding corporation of the Company. Awards under these plans include stock options, restricted share units ("RSUs") and employee stock purchase plan ("ESPP").

The Company accounts for share based payments in accordance with FRS 102 – Share based payments. Share based compensation expense is measured at the grant date of the related equity awards, based on the fair value of the awards, and the expense is recognised over the vesting period, with a corresponding increase in the other reserve.

In case of exchange of employees' services, the equity-settled payment is measured at the fair value of equity instruments granted to employees. If exercisable at the time of grant, equity-settled payment is included in relevant cost or expenses at fair value at grant date and capital surplus is increased accordingly; if exercisable after service in waiting period is completed or specified performance conditions are met, the service obtained in current period is included in relevant cost and expenses at fair value based on the best estimation on quantity of exercisable equity instruments made by the Company in accordance with latest changes in the number of exercisable employees and subsequent information such as whether specified performance conditions are met, and capital surplus is increased accordingly.

2. Significant accounting policies (continued)

(d) Employee compensation (continued)

(3) Share-based compensation (continued)

Stock options

The total amount to be recognised over the vesting period is determined by reference to the fair value of the options on the date of the grant.

The fair value of stock options is determined using option pricing model. The fair value of stock appreciation right is determined using Binomial option pricing model.

At the balance sheet date, subsequent information indicates that the number of equity instruments which are different with previous estimates should be adjusted to the exercisable number of equity instruments on the vesting date. After vesting date, no further adjustment will be raised for relevant cost (expense) or total owner's equity which has been recognised.

Stock options granted to employees, subject to certain exceptions, vest over a four year period, subject to continued service, with 25% vesting on the anniversary of the hire date in the case of new hires or the anniversary of the date of grant in the case of grants to existing employees and 6.25% vesting at the end of each quarterly period thereafter. Stock options granted under the 2007 Plan generally expire ten years from the date of grant.

Restricted Stock Units ("RSU")

RSUs granted to employees vest four years, subject to continued service, with 25% vesting on a pre-determined date that is close to the anniversary of the date of grant and for grants made prior to 18 May 2016, 12.5% vesting semi-annually thereafter until fully vested and for grants made after 18 May 2016, 6.25% vesting quarterly thereafter until fully vested. The total amount to be recognised over the vesting period is determined by reference to the fair value of the ultimate holding corporation stock on the date of the grant.

2. Significant accounting policies (continued)

(d) Employee compensation (continued)

(3) Share-based compensation (continued)

Stock options (continued)

Employee Stock Purchase Plan ("ESPP")

Employees are also eligible to participate in an offering to have up to 10% of their earnings withheld up to certain limitations and applied on specified dates determined by the Board of Directors of Nvidia Corporation to the purchase of shares of common stock. The Board of Directors of Nvidia Corporation may increase this percentage at its discretion, up to 15%. The price of common stock purchased under our ESPP will be equal to 85% of the lower of the fair market value of the common stock on the commencement date of each offering period and the purchase date of each offering period. Employees may end their participation in the ESPP at any time during the offering period, and participation ends automatically on termination of employment with us. In each case, the employee's contributions are refunded. The employee benefit is recognised in profit or loss in the period in which the employees purchase the shares under common stock under the ESPP.

(e) Property, plant and equipment

(1) Measurement

Property, plant and equipment are recognised at cost and subsequently carried at cost less accumulated depreciation and accumulated impairment losses.

The cost of an item of property, plant and equipment initially recognised includes its purchase price and any cost that is directly attributable to bringing the asset to the location and condition necessary for it to be capable of operating in the manner intended by management. The projected cost of dismantlement, removal or restoration is recognised as part of the cost of property, plant and equipment if such obligation is incurred either when the item is acquired or as a consequence of using the asset during a particular period for purposes other than to produce inventories during that period.

2. Significant accounting policies (continued)

(e) Property, plant and equipment (continued)

(2) Depreciation

Depreciation on property, plant and equipment is calculated using the straight-line method to allocate their depreciable amounts over their estimated useful lives as follows:

	Useful lives
Machinery and equipment	3 - 5 years
Computer equipment	3 - 5 years
Office equipment	5 years
Furniture	5 years
Leasehold improvement	Lesser of estimated useful life or remaining lease term

The residual values, estimated useful lives and depreciation method of property, plant and equipment are reviewed, and adjusted as appropriate, at each balance sheet date. The effects of any revision are recognised in the statement of comprehensive income for the financial period in which the changes arise.

(3) Subsequent expenditure

Subsequent expenditure relating to property, plant and equipment that has already been recognised is added to the carrying amount of the asset only when it is probable that future economic benefits associated with the item, will flow to the Company and the cost can be reliably measured. All other repair and maintenance expense is recognised in the statement of comprehensive income when incurred.

(4) Disposal

On disposal of an item of property, plant and equipment, the difference between the net disposal proceeds and its carrying amount is recognised in the statement of comprehensive income.

(f) Investment in subsidiary

Investment in subsidiary is carried at cost less accumulated impairment losses in the Company's balance sheet. On disposal of investment in subsidiary, the difference between disposal proceeds and the carrying amount of the investment is recognised in the statement of comprehensive income.

2. Significant accounting policies (continued)

(g) Loan to related party

Loan to related party is initially recognised at fair value plus transaction costs and subsequently carried at amortised cost using the effective interest method, less accumulated impairment losses. The Company assesses at each balance sheet date whether there is objective evidence that this financial asset is impaired and recognises an allowance for impairment when such evidence exists.

(h) Impairment of non-financial assets

Property, plant and equipment and investment in subsidiary are tested for impairment whenever there is any objective evidence or indication that these assets may be impaired.

For the purpose of impairment testing, the recoverable amount (i.e. the higher of the fair value less cost to sell and value in use) is determined on an individual asset basis unless the asset does not generate cash flows that are largely independent of those from other assets. If this is the case, the recoverable amount is determined for the cash-generating unit (CGU) to which the asset belongs.

If the recoverable amount of the asset (or CGU) is estimated to be less than its carrying amount, the carrying amount of the asset (or CGU) is reduced to its recoverable amount. The difference between the carrying amount and recoverable amount is recognised in the statement of comprehensive income.

An impairment loss for an asset is reversed if, and only if, there has been a change in the estimates used to determine the assets' recoverable amount since the last impairment loss was recognised. The carrying amount of this asset is increased to its revised recoverable amount, provided that this amount does not exceed the carrying amount that would have been determined (net of any accumulated amortisation or depreciation) had no impairment loss been recognised for the asset in prior years. A reversal of impairment loss for an asset is recognised in profit or loss.

2. Significant accounting policies (continued

(i) Loans and receivables
Bank balances
Trade and other receivables

Bank balances, trade and other receivables are initially recognised at fair value plus transaction cost and subsequently carried at amortised cost using the effective interest method, less accumulated impairment losses.

The Company assesses at each balance sheet date whether there is objective evidence that these financial assets are impaired and recognises an allowance for impairment when such evidence exists. Significant financial difficulties of the debtor, probability that the debtor will enter bankruptcy and default or significant delay in payments are objective evidence that these financial assets are impaired.

The carrying amount of these assets is reduced through the use of an impairment allowance account which is calculated as the difference between the carrying amount and the present value of estimated future cash flows, discounted at the original effective interest rate.

These assets are presented as current assets except for those that are expected to be realised later than 12 months after the balance sheet date, which are presented as non-current assets.

(j) Trade and other payables

Trade and other payables represent liabilities for goods and services provided to the Company prior to the end of financial year which are unpaid. They are classified as current liabilities if payment is due within one year or less (or in the normal operating cycle of the business if longer). Otherwise, they are presented as non-current liabilities.

Trade and other payables are initially recognised at fair value and subsequently carried at amortised cost, using the effective interest method.

(k) Operating leases

Leases of assets where substantially all risks and rewards incidental to ownership are retained by the lessor are classified as operating leases. Payments made under operating leases (net of any incentives received from the lessor) are recognised in profit or loss on a straight-line basis over the period of the lease.

(l) Inventories

Inventories are carried at the lower of cost and net realisable value. Cost is determined using the first-in, first-out method. Net realisable value is the estimated selling price in the ordinary course of business, less applicable variable selling expenses.

2. Significant accounting policies (continued)

(m) Income taxes

Current income tax for current and prior periods is recognised at the amount expected to be paid to or recovered from the tax authorities, using the tax rates and tax laws that have been enacted or substantively enacted by the balance sheet date.

Deferred income tax is recognised for all temporary differences except when the deferred income tax arises from the initial recognition of an asset or liability that affects neither accounting nor taxable profit or loss at the time of the transaction.

Current and deferred income tax is measured using the tax rates and tax laws that have been enacted or substantively enacted by the balance sheet date, and are recognised as income or expenses in profit or loss, except to the extent that the tax arises from a transaction which is recognised directly in equity.

(n) Provisions for other liabilities and charges

Provisions for other liabilities and charges are recognised when the Company has a present legal or constructive obligation as a result of past events, it is more likely than not that an outflow of resources will be required to settle the obligation and the amount has been reliably estimated.

(o) Currency translation

The financial statements of the Company are presented in United States Dollars, which is the functional currency of the Company.

Transactions in a currency other than United States Dollar ("foreign currency") are translated into United States Dollar using the exchange rates prevailing at the dates of the transactions. Currency translation differences resulting from the settlement of such transactions and from the translation of monetary assets and liabilities denominated in foreign currencies at the closing rates at the balance sheet date are recognised in the statement of comprehensive income.

All other foreign exchange gains and losses impacting profit or loss are presented in the statement of comprehensive income within "other losses - net".

(p) Cash and cash equivalents

For the purpose of presentation in the cash flow statement, cash and cash equivalents include cash at bank and on hand which are subject to an insignificant risk of change in value.

2. Significant accounting policies (continued)

(q) Share capital

Ordinary shares are classified as equity.

3. Critical accounting estimates, assumptions and judgements

Estimates, assumptions and judgements are continually evaluated and are based on historical experience and other factors, including expectations of future events that are believed to be reasonable under the circumstances

Share-based payments

The Company is required to expense its employees' share-based compensation options and RSUs in accordance with FRS 102 "Share-based Payment". The Company measures share-based compensation cost based on the fair value on the grant date of each option and RSU. This cost is recognised over the period during which an employee is required to provide service in exchange for the award and option or the requisite service period, usually the vesting period, and it is adjusted for actual forfeitures that occur before vesting. In order to assess the fair value of share options, the Company is required to use certain assumptions, including the forfeitures and the service period of each employee. The use of different assumptions and estimates could produce materially different estimated fair values for the share-based compensation options and related expenses.

4. Income

	30/01/2017 to 28/01/2018 US$'000	01/02/2016 to 29/01/2017 US$'000
Revenue		
Sale of graphics processors and related media and communication devices	**8,064,722**	5,712,400
Other income – net:		
Interest income	**-**	68
Others	**96**	(9)
	8,064,818	5,712,459

5. Other losses – net

	30/01/2017 to 28/01/2018 US$'000	01/02/2016 to 29/01/2017 US$'000
Exchange losses - net	**490**	459

6. Other operating expenses

	30/01/2017 to 28/01/2018 US$'000	01/02/2016 to 29/01/2017 US$'000
Consulting and professional expense	**2,579**	2,157
Utilities expense	**93**	94
Repair and maintenance expense	**441**	390
Bank charges	**128**	71
Bad debt expense	**310**	541
Insurance for credit	**1,417**	1,249
Engineering and testing expense	**1,472**	1,500
Other expenses	**349**	398
Total	**6,789**	6,400

7. Employee compensation

	30/01/2017 to 28/01/2018 US$'000	01/02/2016 to 29/01/2017 US$'000
Wages and salaries	**6,681**	6,162
Employer's contribution to defined contribution plans	**667**	543
Share-based compensation	**1,695**	901
	9,043	7,606

8. Income tax

(a) Income tax expense

	30/01/2017 to 28/01/2018 US$'000	01/02/2016 to 29/01/2017 US$'000
Income tax expense attributable to profit is made up of:		
- Current income tax	**790**	543

The tax expense on profit differs from the amount that would arise using the Singapore standard rate of income tax as explained below:

	30/01/2017 to 28/01/2018 US$'000	01/02/2016 to 29/01/2017 US$'000
Profit before tax	**2,587**	3,002
Tax calculated at a tax rate of 17% (2017: 17%)	**440**	510
Effects of:		
- Expenses not deductible for tax purposes	**402**	246
- Income not subject to tax	**(2)**	(14)
- Different tax rates in other country	**(13)**	(15)
- Others	**(37)**	(184)
Tax charge	**790**	543

(b) Movements in current tax liabilities

	28/01/2018 US$'000	29/01/2017 US$'000
Beginning of financial year	**568**	70
Tax expense on profit - current financial year	**790**	543
Income tax paid	**(577)**	(45)
End of the financial year	**781**	568

9. Cash and cash equivalents

	28/01/2018 US$'000	29/01/2017 US$'000
Cash at bank and on hand	**7,509**	99,408

10. Trade and other receivables

	28/01/2018 US$'000	29/01/2017 US$'000
Trade receivables:		
- Immediate holding company	**54**	53
- Non-related parties	**939,851**	708,093
- Other related companies	**-**	28
Less: Allowance for impairment of receivables		
- Non-related parties	**(2,607)**	(2,297)
Trade receivables - net	**937,298**	705,877
Non-trade receivables:		
- Immediate holding company	**111,571**	173,486
- Other related companies	**81,871**	82,256
	193,442	255,742
Other receivables	**15**	1,165
	1,130,755	962,784

The non-trade receivables due from immediate holding company and other related companies are unsecured, interest-free and are repayable on demand.

11. Inventories

	28/01/2018 US$'000	29/01/2017 US$'000
Finished goods	**335,435**	340,657

12. Property, plant and equipment

	Computer equipment US$'000	Machinery and equipment US$'000	Office equipment US$'000	Furniture US$'000	Leasehold improvement US$'000	Total US$'000
2018						
Cost						
Beginning of financial year	153	1,036	23	26	241	1,479
Additions	18	60	20	-	-	98
Disposals	-	(40)	-	(26)	-	(66)
End of financial year	171	1,056	43	-	241	1,511
Accumulated depreciation						
Beginning of financial year	43	135	10	23	202	413
Depreciation charge	53	217	7	3	39	319
Disposals	-	(40)	-	(26)	-	(66)
End of financial year	96	312	17	-	241	666
Net book value						
End of financial year	**75**	**744**	**26**	**-**	**-**	**845**
2017						
Cost						
Beginning of financial year	521	322	28	26	202	1,099
Additions	88	996	-	-	39	1,123
Disposals	(456)	(282)	(5)	-	-	(743)
End of financial year	153	1,036	23	26	241	1,479
Accumulated depreciation						
Beginning of financial year	360	257	11	18	137	783
Depreciation charge	139	160	4	5	65	373
Disposals	(456)	(282)	(5)	-	-	(743)
End of financial year	43	135	10	23	202	413
Net book value						
End of financial year	**110**	**901**	**13**	**3**	**39**	**1,066**

13. Investment in subsidiary

	28/01/2018 US$'000	29/01/2017 US$'000
Unquoted equity shares at cost	**85**	85

Details of investment in subsidiary are as follows:

Name of company	Principal activities	Country of incorporation	Equity holding 28/01/2018 %	Equity holding 29/01/2017 %	Cost of investment 28/01/2018 US$'000	Cost of investment 29/01/2017 US$'000
NVIDIA GK	Provide marketing services to Nvidia Singapore Pte Ltd	Japan	100	100	85	85

The subsidiary does not require an audit in accordance with the laws in the country of its incorporation.

14. Loan to related party

The loan to Nvidia Brazil is unsecured, denominated in the United States Dollar and is repaid in full with interest (net of tax) of US$62,000 on 30 Aug 2017.

15. Trade and other payables

	28/01/2018 US$'000	29/01/2017 US$'000
Trade payable		
- Immediate holding company	**1,157,464**	1,091,874
	1,157,464	1,091,874
Non-trade payables		
- Immediate holding company	**4,457**	4,218
- Ultimate holding company	**20,197**	-
- Other related companies	**77,065**	71,616
	101,719	75,834
Accrued operating expenses	**170,541**	196,613
Other payables	**5,226**	4,381
	175,767	200,994
	1,434,950	1,368,702

The non-trade payables to immediate holding company, ultimate holding company and other related companies are unsecured, interest-free and are repayable on demand.

16. Share capital

The Company's share capital comprise fully paid-up 2 (2017: 2) ordinary shares with no par value, amounting to a total of US$1,000 (2017: US$1,000).

17. Other reserves

	28/01/2018 US$'000	29/01/2017 US$'000
Beginning of financial year	**10,425**	9,524
Employee share-based payment scheme		
- Value of employee services	**1,695**	901
End of financial year	**12,120**	10,425

18. Share based compensation

The employees of the Company are eligible for share-based compensation under equity-settled, shared based compensation plans operated by Nvidia Corporation, the ultimate holding corporation of the Company. Awards include share options, restricted share units ("RSUs"), and employee share purchase plan ("ESPP"), are granted under the following plans:

Amended and Restated 2007 Equity Incentive Plan
The Nvidia Corporation 2007 Equity Incentive Plan was approved in 2007, and subsequently amended and restated in 2012, 2013 and 2014, or the Restated 2007 Plan. This plan authorizes the issuance of share options and restricted share units.

Share options granted to employees, subject to certain exceptions, vest over a four year period subject to continued service, and generally expire ten years from the date of grant.

Subject to certain exceptions, RSUs granted to employees vest over a four year period, subject to continued service.

Amended and Restated 2012 Employee Stock Purchase Plan
The Nvidia Corporation 2012 Employee Share Purchase Plan was approved in 2012, and subsequently amended and restated in 2014, or the Restated 2012 Plan, as the successor to the 1998 Employee Share Purchase Plan.

Employees who participate in an offering may have up to 10% of their earnings withheld to the purchase of shares of common stock. The Board may increase this percentage at its discretion, up to 15%. The price of common stock purchased under our ESPP will be equal to 85% of the lower of the fair market value of the common stock on the commencement date of each offering period and the purchase date of each offering period. Employees may end their participation in the ESPP at any time during the offering period, and participation ends automatically on termination of employment with us. In each case, the employee's contributions are refunded.

The fair value of share option awards on the date of grant is determined using the Binomial model.

Fair value of RSUs is determined using the closing trading price of common stock on the date of grant, minus a dividend yield discount.

18. Share based compensation (continued)

The following table summarises the number and weighted average fair value of ESPP granted during the period:

	2018	2017
Number of shares issued under the plan to participating Nvidia Singapore Pte Ltd's employees on 28 January 2018 (29 January 2017)	**27,359**	24,694

The weighted average market price during the year ended 28 January 2018 was US$133.89 (2017: US$46.57) and the shares had an average grant date fair value of US$23.23 (2017: US$21.66).

The fair value of shares issued under the ESPP has been estimated at the date of grant with the following assumptions, using the Black-Scholes model:

	2018	2017
Weighted average expected life (years)	**0.5-2.0**	0.5-2.0
Risk free interest rate	**0.8%-1.4%**	0.5%-0.9%
Volatility	**40%-54%**	30%-39%
Dividend yield	**0.3%-0.5%**	0.7%-1.4%

Nvidia Corporation uses the implied volatility as it is expected to be more reflective of market conditions and, therefore, could reasonably be expected to be a better indicator of expected volatility than historical volatility. The risk-free interest rate is based upon observed interest rates on Treasury bills appropriate for the term of employee share options.

Dividend yield is based on history and expectation of dividend pay-outs. RSU awards are not eligible for cash dividends prior to vesting; therefore, the fair value of RSUs is discounted by dividend yield. For awards granted on or subsequent to 8 November 2012, Nvidia Corporation uses a dividend yield at grant date based on the per share dividends declared during the most recent quarter.

18. Share based compensation (continued)

A reconciliation of share options and RSU movements for the financial periods ended 28 January 2018 and 29 January 2017 is shown below (average exercise price and fair value reported in US dollars):

Share-based Payment Award Activity

The following table summarises activity for share options:

	Number of share options	Weighted average exercise price
28 January 2018		
Outstanding at beginning of the year	33,869	$14.64
Exercised*	(2,223)	$14.24
Outstanding at end of the year	31,646	$14.67
Options exercisable at year end	31,646	$14.67

* The weighted average share price at the date of exercise of options exercised during the year ended 28 January 2018 was US$171.86. No options were granted during the financial years ended 28 January 2018.

	Number of share options	Weighted average exercise price
29 January 2017		
Outstanding at beginning of the year	51,013	$14.90
Exercised*	(17,144)	$15.41
Outstanding at end of the year	33,869	$14.64
Options exercisable at year end	31,036	$14.59

* The weighted average share price at the date of exercise of options exercised during the year ended 29 January 2017 was US$47.04. No options were granted during the financial years ended 29 January 2017.

18. **Share based compensation** (continued)

Range of exercise price details of awards outstanding as at 28 January 2018:

Range of exercise prices – US$	Outstanding options	Weighted average remaining contractual life (years)
US$10.01-US$15.00	21,600	4.50
US$15.01-US$20.00	10,046	4.32
	31,646	

Range of exercise price details of awards outstanding as at 29 January 2017:

Range of exercise prices – US$	Outstanding options	Weighted average remaining contractual life (years)
US$10.01-US$15.00	22,917	5.52
US$15.01-US$20.00	10,952	5.43
	33,869	

Restricted Share Units ("RSUs") Activity

The following table summarises activity for restricted share units:

	Number of RSUs	Weighted average grant date fair value
28 January 2018		
Outstanding at beginning of the year	67,165	$30.77
Granted	14,223	$169.15
Vested	(28,000)	$28.67
Forfeited/Cancelled	14,885	$79.53
Outstanding/non-vested at end of the year	68,273	$61.60

	Number of RSUs	Weighted average grant date fair value
29 January 2017		
Outstanding at beginning of the year	78,398	$19.96
Granted	17,790	$57.71
Vested	(28,630)	$18.21
Forfeited/Cancelled	(393)	$13.98
Outstanding/non-vested at end of the year	67,165	$30.77

19. Commitments

Operating lease commitments

The future minimum lease payable under non-cancellable operating leases contracted for at the balance sheet date but not recognised as liabilities are as follows:

	28/01/2018 US$'000	29/01/2017 US$'000
Not later than one year	**544**	144
Later than one year but not later than five years	**765**	-
	1,309	144

20. Financial risk management

Financial risk factors

The Company's activities expose it to market risk (including currency risk, price risk and interest rate), credit risk and liquidity risk.

The management team is responsible for setting the objectives and underlying principles of financial risk management for the Company, and establishes detailed policies such as risk identification and measurement, exposure limits and hedging strategies. Financial risk management is carried out by finance personnel.

The finance personnel measure actual exposures against the limits set and prepare regular reports for the review of the management team. The information presented below is based on information received by key management.

(a) Market risk

(i) Currency risk

The Company's revenue, cost of operations and majority of the financial assets and liabilities are primarily in United States Dollar ("USD"). The Company is not exposed to any significant currency risk. The Company did not enter into currency forward contracts to hedge this exposure.

(ii) Price risk

The Company has insignificant exposure to equity price risk as it does not hold any equity financial assets.

20. Financial risk management (continued)

(b) Credit risk

Credit risk refers to the risk that a counterparty will default on its contractual obligations resulting in financial loss to the Company. The major classes of financial assets of the Company are cash and cash equivalents and trade receivables. Cash and cash equivalents are deposits in banks with sound credit ratings, therefore, the Company does not expect to have high credit risk in this regard.

For trade receivables, the Company adopts the policy of dealing only with customers of appropriate credit history, and obtaining sufficient collateral where appropriate to mitigate credit risk. In addition, the Company monitors its credit risk on an ongoing basis by reviewing the debtors' aging to minimise its exposure to credit risk.

Credit exposure to an individual customer is restricted by the credit limit approved by the credit controller. Customers' payment profile and credit exposure are continuously monitored by the credit controller and reported to the management and Board of Directors. The Company's trade receivables include 5 debtors (2017: 5 debtors) that represented 40.7% and 42.3% of trade receivables at balance sheet date, respectively.

(i) Financial assets that are neither past due nor impaired

Trade receivables that are neither past due nor impaired are substantially companies with a good collection track record with the Company.

(ii) Financial assets that are past due and/or impaired

There is no other class of financial assets that is past due and/or impaired except for trade receivables.

The carrying amount of trade receivables individually determined to be impaired and the movement in the related allowance for impairment are as follows:

	28/01/2018 US$'000	29/01/2017 US$'000
Gross amount	**2,607**	2,297
Less: Allowance for impairment	**(2,607)**	(2,297)
	-	-
Beginning of financial year	**2,297**	1,756
Allowance made/(utilised)	**310**	541
End of financial year (Note 10)	**2,607**	2,297

20. Financial risk management (continued)

(c) Liquidity risk

The Company manage the liquidity risk by maintaining sufficient cash and bank balances to enable it to meet its operational requirements.

(d) Capital risk

The Company's objectives when managing capital are to ensure that the Company is adequately capitalised and to maintain an optimal capital structure.

The Company is not subject to any externally imposed capital requirements.

(e) Financial instruments by category

The carrying amounts of financial assets measured at fair value through profit or loss are disclosed on the face of the balance sheet. The aggregate carrying amounts of loans and receivables and financial liabilities at amortised cost are as follows:

	28/01/2018 US$'000	29/01/2017 US$'000
Loans and receivables	**1,138,248**	1,061,617
Financial liabilities at amortised cost	**1,434,659**	1,368,428

(f) Offsetting financial assets and financial liabilities

(i) Financial assets

The Company has the following financial instrument subject to enforceable master netting arrangement or similar agreement as follows:

	Related amounts set off			Related amounts not set off	
	Gross amounts - financial assets (a) US$'000	Gross amounts - financial liabilities (b) US$'000	Net amounts - financial assets presented in the balance sheet (c)=(a)-(b) US$'000	Financial assets (d) US$'000	Net amount (e)=(c)+(d) US$'000
As at 28 January 2018					
Trade and other receivables	**211,720**	**100,149**	**111,571**	**81,925**	**193,496**
Total	**211,720**	**100,149**	**111,571**	**81,925**	**193,496**
As at 29 January 2017					
Trade and other receivables	209,731	36,245	173,486	82,337	255,823
Total	209,731	36,245	173,486	82,337	255,823

20. Financial risk management (continued)

(f) Offsetting financial assets and financial liabilities (continued)

(ii) Financial liabilities

The Company has the following financial instruments subject to enforceable master netting arrangements or similar agreement as follows:

	Related amounts set off			Related amounts not set off	
	Gross amounts – financial liabilities (a) US$'000	Gross amounts – financial assets (b) US$'000	Net amounts – financial liabilities presented in the balance sheet (c)=(a)-(b) US$'000	Financial liabilities (d) US$'000	Net amount (e)=(c)+(d) US$'000
As at 28 January 2018					
Trade and other payables	**1,564,710**	**407,246**	**1,157,464**	**101,719**	**1,259,183**
Total	**1,564,710**	**407,246**	**1,157,464**	**101,719**	**1,259,183**
As at 29 January 2017					
Trade and other payables	1,034,886	389,421	645,465	522,243	1,167,708
Total	1,034,886	389,421	645,465	522,243	1,167,708

21. Immediate and ultimate holding corporations

The Company's immediate holding corporation is Nvidia International Inc., incorporated in the Cayman Islands. The Company's ultimate holding corporation is Nvidia Corporation, incorporated in United States of America.

22. Related party transactions

In addition to the information disclosed elsewhere in the financial statements, the following transactions took place between the Company and related parties at terms agreed between the parties:

(a) Transactions

	30/01/2017 to 28/01/2018 US$'000	01/02/2016 to 29/01/2017 US$'000
Purchases from immediate holding company	**6,967,070**	5,049,758
Payments made on behalf of a fellow subsidiary	**-**	381
Sales of goods to a fellow subsidiary	**1,470**	28
Sales of goods to immediate holding company	**-**	5
Marketing and administrative services fees charged by related corporations	**210,571**	173,322

22. Related party transactions (continued)

The Company participates in the centralised cash pooling arrangement with Nvidia Corporation, its ultimate holding company. During the year, certain balances with related companies were settled and netted off through the cash pooling arrangement.

(b) Key management personnel compensation

There are no key management remuneration for the financial years ended 28 January 2018 and 29 January 2017 as the key management having authority and responsibility for planning, directing and controlling the activities of the Company are employed by related corporations.

23. New or revised accounting standards and interpretations

Below are the mandatory standards, amendments and interpretations to existing standards that have been published, and are relevant for the Group's accounting periods beginning on or after 29 January 2018 and which the Group has not early adopted:

(a) FRS 115 Revenue from contracts with customers (effective for annual periods beginning on or after 1 January 2018)

This is the converged standard on revenue recognition. It replaces FRS 11 Construction contracts, FRS 18 Revenue, and related interpretations. Revenue is recognised when a customer obtains control of a good or service. A customer obtains control when it has the ability to direct the use of and obtain the benefits from the good or service. The core principle of FRS 115 is that an entity recognises revenue to depict the transfer of promised goods or services to customers in an amount that reflects the consideration to which the entity expects to be entitled in exchange for those goods or services. An entity recognises revenue in accordance with that core principle by applying the following steps:

- Step 1: Identify the contract(s) with a customer
- Step 2: Identify the performance obligations in the contract
- Step 3: Determine the transaction price
- Step 4: Allocate the transaction price to the performance obligations in the contract
- Step 5: Recognise revenue when (or as) the entity satisfies a performance obligation

FRS 115 also includes a cohesive set of disclosure requirements that will result in an entity providing users of financial statements with comprehensive information about the nature, amount, timing and uncertainty of revenue and cash flows arising from the entity's contracts with customers.

23. New or revised accounting standards and interpretations (continued)

(a) FRS 115 Revenue from contracts with customers (effective for annual periods beginning on or after 1 January 2018) (continued)

Management is currently assessing the effects of applying the new standard on the Company's financial statements and has identified the following areas that are likely to be affected:

(i) Rights of return – FRS 115 requires separate presentation on the balance sheet of the right to recover the goods from the customer and the refund obligation; and

(ii) Accounting for certain costs incurred in fulfilling a contract – certain costs which are currently expensed may need to be recognised as an asset under FRS 115.

The Company is expected to adopt this guidance beginning in the first quarter of fiscal year 2019 using the modified retrospective approach. The Company has made progress in, and continue to assess changes in policies, processes, systems and controls necessary to meet the additional requirements of the guidance. While still finalizing the analysis to qualify the adoption impact of the provisions of the new revenue standard, the Company does not expect it to have a material impact on the financial statements. The Company does expect to provide additional disclosure under this guidance, including more information regarding estimates and judgments, practical expedients used, contract balances and performance obligations.

(b) FRS 116 Leases (effective for annual periods beginning on or after 1 January 2019)

FRS 116 will result in almost all leases being recognised on the balance sheet, as the distinction between operating and finance leases is removed.

Under the new standard, an asset (the right to use the leased item) and a financial liability to pay rentals are recognised. The only exceptions are short-term and low-value leases. The accounting for lessors will not change significantly.

Some of the Company's commitments may be covered by the exception for short-term and low-value leases and some commitments may relate to arrangements that will not qualify as leases under FRS 116.

The new standard also introduces expanded disclosure requirements and changes in presentation.

The Company has yet to determine to what extent the commitments as at 28 January 2018 will result in the recognition of an asset and a liability for future payments and how this will affect the Company's profit and classification of cash flows.

23. New or revised accounting standards and interpretations (continued)

(c) FRS 109 Financial instruments (effective for annual periods beginning on or after 1 January 2018)

FRS 109 replaces FRS 39 *Financial instruments: Recognition and Measurement* and its relevant interpretations.

FRS 109 retains the mixed measurement model and establishes three primary measurement categories for financial assets: amortised cost, fair value through Other Comprehensive Income (OCI) and fair value through Profit or Loss. The basis of classification depends on the entity's business model and the contractual cash flow characteristics of the financial asset. Investments in equity instruments are required to be measured at fair value through profit or loss with an irrevocable option at inception to present changes in fair value in OCI (FVOCI). Gains and losses realised on the sale of such financial assets at FVOCI are not transferred to profit or loss on sale but reclassified from the FVOCI reserve to retained earnings.

Under FRS 109, there are no changes to the classification and measurement requirements for financial liabilities except for the recognition of fair value changes arising from changes in own credit risk. For liabilities designated at fair value through profit or loss, such changes are recognised in OCI.

FRS 109 relaxes the requirements for hedge effectiveness by replacing the bright line hedge effectiveness tests. It requires an economic relationship between the hedged item and hedging instrument and for the 'hedged ratio' to be the same as the one management uses for risk management purposes.

There is also now a new expected credit losses impairment model that replaces the incurred loss impairment model used in FRS 39. It applies to financial assets classified at amortised cost, debt instruments measured at fair value through OCI, contract assets under FRS 115 *Revenue from contracts with customers*, lease receivables, loan commitments and certain financial guarantee contracts.

The new standard also introduces expanded disclosure requirements and changes in presentation.

While still finalizing the analysis to qualify the adoption impact of the provisions of the new standard, the Company does not expect any significant impact to the allowance for impairment of receivables and the opening retained earnings to arise from the application of the expected credit losses impairment mode.

f1808f1b5ae5e372759524887e08e80a9bc095a99e88d03d250f197c81d664e0

NVIDIA SINGAPORE PTE LTD

NOTES TO FINANCIAL STATEMENTS
For the financial year ended 28 January 2018

24. Authorisation of financial statements

These financial statements were authorised for issue in accordance with a resolution of the Board of Directors of Nvidia Singapore Pte Ltd on 12 JUN 2018

EXHIBIT N

NVIDIA SINGAPORE PTE LTD

(Incorporated in Singapore. Registration Number 200003831M)

FINANCIAL STATEMENT

For the financial year ended 27 January 2019

pwc

NVIDIA SINGAPORE PTE LTD
(Incorporated in Singapore)

FINANCIAL STATEMENT
For the financial year ended 27 January 2019

Contents

NVIDIA SINGAPORE PTE LTD

DIRECTORS' STATEMENT

For the financial year ended 27 January 2019

The directors present their report to the shareholder together with the audited financial statements for the financial year ended 27 January 2019.

In the opinion of the directors,

(a) the financial statements as set out on pages 5 to 38 are drawn up so as to give a true and fair view of the financial position of the Company at 27 January 2019 and the financial performance, changes in equity and cash flows of the Company for the financial year covered by the financial statements; and

(b) at the date of this statement, there are reasonable grounds to believe that the Company will be able to pay its debts as and when they fall due.

Directors

The directors in office at the date of this statement are as follows:

Lee Kay Beng
Karen Theresa Burns
Michael John Byron
Rebecca Peters

Arrangements to enable directors to acquire shares and debentures

Neither at the end of nor at any time during the financial year was the Company a party to any arrangement whose object was to enable the directors of the Company to acquire benefits by means of the acquisition of shares in, or debentures of, the Company or any other body corporate.

Directors' interests in shares and debentures

According to the register of directors' shareholdings, none of the directors holding office at the end of the financial year had any interest in the shares or debentures of the Company or its related corporations, except as follows:

	Holdings registered in name of Director	
	At	At
Ultimate Holding Corporation	**27.01.2019**	28.01.2018
(No. of ordinary shares)		
-Nvidia Corporation		
Michael John Byron	**33,951**	28,945
Karen Theresa Burns	**3,996**	15,124
Rebecca Peters	**642**	1,974

Directors' interests in shares and debentures (continued)

	At 27.01.2019	At 28.01.2018
(Unissued ordinary shares under share options)		
-Nvidia Corporation		
Michael John Byron	-	9,018
Karen Theresa Burns	31,275	31,275
Rebecca Peters	1,875	1,875
(Unvested restricted shares units)		
-Nvidia Corporation		
Michael John Byron	46,138	84,858
Karen Theresa Burns	50,827	93,151
Rebecca Peters	15,579	27,091

Share options

No options were granted during the financial year to subscribe for unissued shares of the Company.

No shares were issued during the financial year by virtue of the exercise of options to take up unissued shares of the Company.

There were no unissued shares of the Company under option at the end of the financial year.

Independent auditor

The independent auditor, PricewaterhouseCoopers LLP, has expressed its willingness to accept re-appointment.

On behalf of the directors

MICHAEL JOHN BYRON
Director

KAREN THERESA BURNS
Director

27 JUN 2019

INDEPENDENT AUDITOR'S REPORT TO THE SHAREHOLDER OF NVIDIA SINGAPORE PTE LTD

Report on the Financial Statements

Our opinion

In our opinion, the accompanying financial statements of Nvidia Singapore Pte Ltd (the "Company") are properly drawn up in accordance with the provisions of the Companies Act, Chapter 50 (the "Act") and Financial Reporting Standards in Singapore ("FRSs") so as to give a true and fair view of the balance sheet of the Company as at 27 January 2019 and of the financial performance, changes in equity and cash flows of the Company for the year ended on that date.

What we have audited

The financial statements of the Company comprise:

- the balance sheet as at 27 January 2019;
- the statement of comprehensive income for the year then ended;
- the statement of changes in equity for the year then ended;
- the statement of cash flows for the year then ended; and
- the notes to the financial statements, including a summary of significant accounting policies.

Basis for Opinion

We conducted our audit in accordance with Singapore Standards on Auditing ("SSAs"). Our responsibilities under those standards are further described in the *Auditor's Responsibilities for the Audit of the Financial Statements* section of our report.

We believe that the audit evidence we have obtained is sufficient and appropriate to provide a basis for our opinion.

Independence

We are independent of the Company in accordance with the Accounting and Corporate Regulatory Authority Code of Professional Conduct and Ethics for Public Accountants and Accounting Entities ("ACRA Code") together with the ethical requirements that are relevant to our audit of the financial statements in Singapore, and we have fulfilled our other ethical responsibilities in accordance with these requirements and the ACRA Code.

Other Information

Management is responsible for the other information. The other information comprises the Directors' Statement but does not include the financial statements and our auditor's report thereon.

Our opinion on the financial statements does not cover the other information and we do not express any form of assurance conclusion thereon.

In connection with our audit of the financial statements, our responsibility is to read the other information and, in doing so, consider whether the other information is materially inconsistent with the financial statements or our knowledge obtained in the audit, or otherwise appears to be materially misstated. If, based on the work we have performed, we conclude that there is a material misstatement of this other information, we are required to report that fact. We have nothing to report in this regard.

Responsibilities of Management and Directors for the Financial Statements

Management is responsible for the preparation of financial statements that give a true and fair view in accordance with the provisions of the Act and FRSs, and for devising and maintaining a system of internal accounting controls sufficient to provide a reasonable assurance that assets are safeguarded against loss from unauthorised use or disposition; and transactions are properly authorised and that they are recorded as necessary to permit the preparation of true and fair financial statements and to maintain accountability of assets.

In preparing the financial statements, management is responsible for assessing the Company's ability to continue as a going concern, disclosing, as applicable, matters related to going concern and using the going concern basis of accounting unless management either intends to liquidate the Company or to cease operations, or has no realistic alternative but to do so.

The directors' responsibilities include overseeing the Company's financial reporting process.

INDEPENDENT AUDITOR'S REPORT TO THE SHAREHOLDER OF NVIDIA SINGAPORE PTE LTD (continued)

Auditor's Responsibilities for the Audit of the Financial Statements

Our objectives are to obtain reasonable assurance about whether the financial statements as a whole are free from material misstatement, whether due to fraud or error, and to issue an auditor's report that includes our opinion. Reasonable assurance is a high level of assurance, but is not a guarantee that an audit conducted in accordance with SSAs will always detect a material misstatement when it exists. Misstatements can arise from fraud or error and are considered material if, individually or in the aggregate, they could reasonably be expected to influence the economic decisions of users taken on the basis of these financial statements.

As part of an audit in accordance with SSAs, we exercise professional judgement and maintain professional scepticism throughout the audit. We also:

- Identify and assess the risks of material misstatement of the financial statements, whether due to fraud or error, design and perform audit procedures responsive to those risks, and obtain audit evidence that is sufficient and appropriate to provide a basis for our opinion. The risk of not detecting a material misstatement resulting from fraud is higher than for one resulting from error, as fraud may involve collusion, forgery, intentional omissions, misrepresentations, or the override of internal control.

- Obtain an understanding of internal control relevant to the audit in order to design audit procedures that are appropriate in the circumstances, but not for the purpose of expressing an opinion on the effectiveness of the Company's internal control.

- Evaluate the appropriateness of accounting policies used and the reasonableness of accounting estimates and related disclosures made by management.

- Conclude on the appropriateness of management's use of the going concern basis of accounting and, based on the audit evidence obtained, whether a material uncertainty exists related to events or conditions that may cast significant doubt on the Company's ability to continue as a going concern. If we conclude that a material uncertainty exists, we are required to draw attention in our auditor's report to the related disclosures in the financial statements or, if such disclosures are inadequate, to modify our opinion. Our conclusions are based on the audit evidence obtained up to the date of our auditor's report. However, future events or conditions may cause the Company to cease to continue as a going concern.

- Evaluate the overall presentation, structure and content of the financial statements, including the disclosures, and whether the financial statements represent the underlying transactions and events in a manner that achieves fair presentation.

We communicate with the directors regarding, among other matters, the planned scope and timing of the audit and significant audit findings, including any significant deficiencies in internal control that we identify during our audit.

Report on Other Legal and Regulatory Requirements

In our opinion, the accounting and other records required by the Act to be kept by the Company have been properly kept in accordance with the provisions of the Act.

PricewaterhouseCoopers LLP
Public Accountants and Chartered Accountants
Singapore,

27 JUN 2019

NVIDIA SINGAPORE PTE LTD

STATEMENT OF COMPREHENSIVE INCOME
For the financial year ended 27 January 2019

	Note	29/01/2018 to 27/01/2019 US$'000	30/01/2017 to 28/01/2018 US$'000
Revenue	4	**9,727,059**	8,064,722
Other income - net		**1**	96
Other gains/(losses) - net	5	**363**	(490)
Expenses			
- Raw materials and other consumables consumed		**(9,457,611)**	(7,830,208)
- Employee compensation	7	**(13,561)**	(9,043)
- Travelling expenses		**(385)**	(237)
- Sales and marketing expenses		**(2,281)**	(4,063)
- Depreciation of property, plant and equipment		**(335)**	(319)
- Intercompany service expenses		**(246,944)**	(210,571)
- Rental on operating leases		**(547)**	(511)
- Other operating expenses	6	**(4,697)**	(6,789)
Total expenses		**(9,726,361)**	(8,061,741)
Profit before income tax		**1,062**	2,587
Income tax expense	8	**(644)**	(790)
Net profit after tax/ Total comprehensive income		**418**	1,797

The accompanying notes form an integral part of these financial statements.

NVIDIA SINGAPORE PTE LTD

BALANCE SHEET

As at 27 January 2019

	Note	27/01/2019 US$'000	28/01/2018 US$'000
ASSETS			
Current assets			
Cash and cash equivalents	9	**4,699**	7,509
Trade and other receivables	10	**1,348,605**	1,130,755
Inventories	11	**631,270**	335,435
Prepayment and deposits		**863**	4,619
		1,985,437	1,478,318
Non-current assets			
Property, plant and equipment	12	**780**	845
Investment in subsidiary	13	**85**	85
		865	930
Total assets		**1,986,302**	1,479,248
LIABILITIES			
Current liabilities			
Trade and other payables	14	**1,937,843**	1,434,950
Current income tax liabilities	8	**474**	781
		1,938,317	1,435,731
Total liabilities		**1,938,317**	1,435,731
NET ASSETS		**47,985**	43,517
SHAREHOLDER'S EQUITY			
Share capital	15	**1**	1
Other reserves	16	**16,170**	12,120
Retained earnings		**31,814**	31,396
		47,985	43,517

The accompanying notes form an integral part of these financial statements.

NVIDIA SINGAPORE PTE LTD

STATEMENT OF CHANGES IN EQUITY

For the financial year ended 27 January 2019

	Share capital US$'000	Other reserves US$'000	Retained earnings US$'000	Total equity US$'000
2019				
Beginning of financial year	**1**	**12,120**	**31,396**	**43,517**
Share-based compensation	-	4,050	-	4,050
Total comprehensive income	-	-	418	418
End of financial year	**1**	**16,170**	**31,814**	**47,985**
2018				
Beginning of financial year	**1**	**10,425**	**29,599**	**40,025**
Share-based compensation	-	1,695	-	1,695
Total comprehensive income	-	-	1,797	1,797
End of financial year	**1**	**12,120**	**31,396**	**43,517**

The accompanying notes form an integral part of these financial statements.

NVIDIA SINGAPORE PTE LTD

STATEMENT OF CASH FLOWS

For the financial year ended 27 January 2019

	Note	29/01/2018 to 27/01/2019 US$'000	30/01/2017 to 28/01/2018 US$'000
Cash flows from operating activities			
Net profit after tax		**418**	1,797
Adjustments for:			
- Income tax expense	8	**644**	790
- Depreciation of property, plant and equipment	12	**335**	319
- Share-based compensation	7	**4,050**	1,695
Operating cash flow before working capital changes		**5,447**	4,601
Change in operating assets and liabilities			
- Trade and other receivables		**(217,850)**	(167,971)
- Inventories		**(295,835)**	5,222
- Trade and other payables		**502,893**	66,248
- Prepayments and deposits		**3,756**	176
Cash used in operations		**(1,589)**	(91,724)
Income tax paid	8	**(951)**	(577)
Net cash used in operating activities		**(2,540)**	(92,301)
Cash flows from investing activities			
Additions of property, plant and equipment	12	**(270)**	(98)
Repayment of loan to related party		**-**	500
Net cash (used in)/generated from investing activities		**(270)**	402
Net decrease in cash and cash equivalents		**(2,810)**	(91,899)
Cash and cash equivalents at beginning of financial period		**7,509**	99,408
Cash and cash equivalents at end of financial period	9	**4,699**	7,509

The accompanying notes form an integral part of these financial statements.

These notes form an integral part of and should be read in conjunction with the accompanying financial statements.

1. General information

The Company is incorporated in Singapore. The address of its registered office is 112 Robinson Road, #05-01, Singapore 068902.

The principal activity of the Company consists of sales of graphics processors and media and communication devices.

The principal activities of its subsidiary are to provide marketing services to Nvidia Singapore Pte Ltd.

2. Significant accounting policies

2.1 Basis of preparation

The financial statements have been prepared in accordance with Singapore Financial Reporting Standards ("FRS") under the historical cost convention, except as disclosed in the accounting policies below.

The preparation of these financial statements in conformity with FRS requires management to exercise its judgement in the process of applying the Company's accounting policies. It also requires the use of certain critical accounting estimates and assumptions. The areas involving a higher degree of judgement or complexity, or areas where assumptions and estimates are significant to the financial statements are disclosed in Note 3.

Interpretations and amendments to published standards effective in 2018

On 29 January 2018, the Company adopted the new or amended FRS and Interpretations of FRS ("INT FRS") that are mandatory for application for the financial year. Changes to the Company's accounting policies have been made as required, in accordance with the transitional provisions in the respective FRS and INT FRS.

The adoption of these new or amended FRS and INT FRS did not result in substantial changes to the Company's accounting policies and had no material effect on the amounts reported for the current or prior financial years except for the following:

2. Significant accounting policies (continued)

2.1 Basis of preparation (continued)

(a) Adoption of FRS 115 Revenue from Contracts with Customers

The Company has adopted the new standard using the modified retrospective approach with the cumulative impact of the adoption recognised in the opening retained earnings at 29 January 2018. Comparative information for financial year ended 2018 is not restated.

The accounting policies for revenue from contracts with customers under FRS 115 are disclosed in Note 2.2.

The adoption of FRS 115 did not have a material impact on the Company's financial statement as at 28 January 2018 and 27 January 2019.

(b) Adoption of FRS 109 Financial Instruments

The Company has adopted the new standard retrospectively from 29 January 2018, in line with the transition provision permitted under the standards. Comparatives for financial year ended 2018 are not restated and the Company has recognised any difference between the carrying amounts at 28 January 2018 and 29 January 2018 in the opening retained earnings.

The accounting policies for financial instruments under FRS 109 are disclosed in Note 2.8. There are no effects on the adoption of FRS 109 for the current year and opening retained earnings.

(i) Impairment of financial assets

The Company has the following financial assets subject to the expected credit loss impairment model under FRS 109:

- Cash and bank deposits;
- Trade and other receivables; and
- Loan to related corporation.

The impairment methodology for each of these classes of financial assets under FRS 109 are different as disclosed in Note 2.8 and Note 19(b).

2. Significant accounting policies (continued)

2.2 Revenue recognition

Revenue is derived from product sales, including hardware and systems. Revenue recognition is determined through the following steps: (1) identification of the contract with a customer; (2) identification of the performance obligations in the contract; (3) determination of the transaction price; (4) allocation of the transaction price to the performance obligations in the contract; and (5) recognition of revenue when, or as, a performance obligation is satisfied.

Product Sales Revenue

Revenue from product sales (sale of graphics processors and related media and communication devices) is recognized when the Company has delivered the products to the customer, the customer has accepted the products and collectability of the related receivables is reasonably assured. Revenue is recognized net of allowances for returns, customer programs and any taxes collected from customers.

For products sold with a right of return, a reduction to revenue is recorded by establishing a sales return allowance for estimated product returns at the time revenue is recognized, based primarily on historical return rates. However, if product returns for a financial year are anticipated to exceed historical return rates, additional sales return allowances may be determined to be required, to properly reflect the estimated exposure for product returns.

The Company's customer programs involve sales rebates, which are designed to serve as sales incentives to resellers of the products in various target markets, and marketing development funds ("MDFs"), which represent monies paid to the Company's customers that are earmarked for market segment development and are designed to support its activities while also promoting NVIDIA products. These customer programs are accounted as a reduction to revenue and accrued for potential rebates and MDFs based on the amount expected to be claimed by customers.

2.3 Group accounting

These financial statements are the separate financial statements of Nvidia Singapore Pte Ltd. The Company is exempted from the preparation of consolidated financial statements as the Company is a wholly-owned subsidiary of Nvidia Corporation, a US-incorporated company which produces consolidated financial statements available for public use. The basis on which the subsidiary is accounted for is disclosed in Note 2(f). The main office of Nvidia Corporation is as follows: 2788 San Tomas Expressway, Santa Clara, CA 95051, U.S.A..

2. Significant accounting policies (continued)

2.4 Employee compensation

(a) Defined contribution plans

Defined contribution plans are post-employment benefit plans under which the Company pays fixed contributions into separate entities such as Mandatory Provident Fund and Superannuation Fund established in Hong Kong, on a mandatory, contractual or voluntary basis. The Company has no further payment obligations once the contributions have been paid. The Company's contribution to defined contribution plans are recognised as employee compensation expense when they are due.

(b) Employee leave entitlements

Employee entitlements to annual leave are recognised when they accrue to employees. A provision is made for the estimated liability for annual leave as a result of services rendered by employees up to the balance sheet date.

(c) Share-based compensation

The employees of the Company are eligible for share-based compensation under equity-settled, shared based compensation plans operated by Nvidia Corporation, the ultimate holding corporation of the Company. Awards under these plans include stock options, restricted share units ("RSUs") and employee stock purchase plan ("ESPP").

The Company accounts for share based payments in accordance with FRS 102 – Share based payments. Share based compensation expense is measured at the grant date of the related equity awards, based on the fair value of the awards, and the expense is recognised over the vesting period, with a corresponding increase in the other reserve.

In case of exchange of employees' services, the equity-settled payment is measured at the fair value of equity instruments granted to employees. If exercisable at the time of grant, equity-settled payment is included in relevant cost or expenses at fair value at grant date and capital surplus is increased accordingly; if exercisable after service in waiting period is completed or specified performance conditions are met, the service obtained in current period is included in relevant cost and expenses at fair value based on the best estimation on quantity of exercisable equity instruments made by the Company in accordance with latest changes in the number of exercisable employees and subsequent information such as whether specified performance conditions are met, and capital surplus is increased accordingly.

2. Significant accounting policies (continued)

2.4 Employee compensation (continued)

(c) Share-based compensation (continued)

Stock options

The total amount to be recognised over the vesting period is determined by reference to the fair value of the options on the date of the grant.

The fair value of stock options is determined using option pricing model. The fair value of stock appreciation right is determined using Binomial option pricing model.

At the balance sheet date, subsequent information indicates that the number of equity instruments which are different with previous estimates should be adjusted to the exercisable number of equity instruments on the vesting date. After vesting date, no further adjustment will be raised for relevant cost (expense) or total owner's equity which has been recognised.

Stock options granted to employees, subject to certain exceptions, vest over a four year period, subject to continued service, with 25% vesting on the anniversary of the hire date in the case of new hires or the anniversary of the date of grant in the case of grants to existing employees and 6.25% vesting at the end of each quarterly period thereafter. Stock options granted under the 2007 Plan generally expire ten years from the date of grant.

Restricted Stock Units ("RSU")

RSUs granted to employees vest four years, subject to continued service, with 25% vesting on a pre-determined date that is close to the anniversary of the date of grant and for grants made prior to 18 May 2016, 12.5% vesting semi-annually thereafter until fully vested and for grants made after 18 May 2016, 6.25% vesting quarterly thereafter until fully vested. The total amount to be recognised over the vesting period is determined by reference to the fair value of the ultimate holding corporation stock on the date of the grant.

2. Significant accounting policies (continued)

2.4 Employee compensation (continued)

(c) Share-based compensation (continued)

Employee Stock Purchase Plan ("ESPP")

Employees are also eligible to participate in an offering to have up to 10% of their earnings withheld up to certain limitations and applied on specified dates determined by the Board of Directors of Nvidia Corporation to the purchase of shares of common stock. The Board of Directors of Nvidia Corporation may increase this percentage at its discretion, up to 15%. The price of common stock purchased under our ESPP will be equal to 85% of the lower of the fair market value of the common stock on the commencement date of each offering period and the purchase date of each offering period. Employees may end their participation in the ESPP at any time during the offering period, and participation ends automatically on termination of employment with us. In each case, the employee's contributions are refunded. The employee benefit is recognised in profit or loss in the period in which the employees purchase the shares under common stock under the ESPP.

2.5 Property, plant and equipment

Property, plant and equipment are recognised at cost less accumulated depreciation and accumulated impairment losses.

Subsequent expenditure relating to property, plant and equipment that has already been recognised is added to the carrying amount of the asset only when it is probable that future economic benefits associated with the item will flow to the Company and the cost of the item can be measured reliably. All other repair and maintenance expenses are recognised in the statement of comprehensive income when incurred.

Depreciation is calculated using the straight-line method to allocate depreciable amounts over their estimated useful lives. The estimated useful lives are as follows:

	Useful lives
Machinery and equipment	3 - 5 years
Computer equipment	3 - 5 years
Office equipment	5 years
Furniture	5 years
Leasehold improvement	Lesser of estimated useful life or remaining lease term

The residual values, estimated useful lives and depreciation method of property, plant and equipment are reviewed, and adjusted as appropriate, at each balance sheet date. The effects of any revision are recognised in the statement of comprehensive income for the financial period in which the changes arise.

2. Significant accounting policies (continued)

2.6 Investment in subsidiary

Investment in subsidiary is carried at cost less accumulated impairment losses in the Company's balance sheet. On disposal of investment in subsidiary, the difference between disposal proceeds and the carrying amount of the investment is recognised in the statement of comprehensive income.

2.7 Impairment of non-financial assets

Property, plant and equipment and investment in subsidiary are tested for impairment whenever there is any objective evidence or indication that these assets may be impaired.

For the purpose of impairment testing, the recoverable amount (i.e. the higher of the fair value less cost to sell and value in use) is determined on an individual asset basis unless the asset does not generate cash flows that are largely independent of those from other assets. If this is the case, the recoverable amount is determined for the cash-generating unit (CGU) to which the asset belongs.

If the recoverable amount of the asset (or CGU) is estimated to be less than its carrying amount, the carrying amount of the asset (or CGU) is reduced to its recoverable amount. The difference between the carrying amount and recoverable amount is recognised in the statement of comprehensive income.

An impairment loss for an asset is reversed if, and only if, there has been a change in the estimates used to determine the assets' recoverable amount since the last impairment loss was recognised. The carrying amount of this asset is increased to its revised recoverable amount, provided that this amount does not exceed the carrying amount that would have been determined (net of any accumulated amortisation or depreciation) had no impairment loss been recognised for the asset in prior years. A reversal of impairment loss for an asset is recognised in profit or loss.

2. Significant accounting policies (continued)

2.8 Financial assets

(a) The accounting for financial assets before 29 January 2018 under FRS 39 are as follows:

(i) Loan and receivables
Bank balances
Trade and other receivables

Bank balances and trade and other receivables are initially recognised at fair value plus transaction cost and subsequently carried at amortised cost using the effective interest method, less accumulated impairment losses.

The Company assesses at each balance sheet date whether there is objective evidence that these financial assets are impaired and recognises an allowance for impairment when such evidence exists. Significant financial difficulties of the debtor, probability that the debtor will enter bankruptcy and default or significant delay in payments are objective evidence that these financial assets are impaired.

The carrying amount of these assets is reduced through the use of an impairment allowance account which is calculated as the difference between the carrying amount and the present value of estimated future cash flows, discounted at the original effective interest rate.

These assets are presented as current assets except for those that are expected to be realised later than 12 months after the balance sheet date, which are presented as non-current assets.

(b) The accounting for financial assets from 29 January 2018 under FRS 109 are as follows:

The Company classifies its financial assets into the following measurement categories:

- Amortised cost;
- Fair value through other comprehensive income (FVOCI); and
- Fair value through profit or loss (FVPL)

The classification of debt instruments depends on the Company's business model for managing the financial assets as well as the contractual terms of the cash flows of the financial assets.

Financial assets with embedded derivatives are considered in their entirety when determining whether their cash flows are solely payment of principal and interest.

2. Significant accounting policies (continued)

2.8 Financial assets (continued)

(b) The accounting for financial assets from 29 January 2018 under FRS 109 are as follows: (continued)

The Company reclassifies debt instruments when and only when its business model for managing those assets changes.

(i) At initial recognition

At initial recognition, the Company measures a financial asset at its fair value plus, in the case of a financial asset not at fair value through profit or loss, transaction costs that are directly attributable to the acquisition of the financial assets. Transaction costs of financial assets carried at fair value through profit or loss are expensed in profit or loss.

(ii) At subsequent measurement

1. Debt instrument

Debt instruments of the Company mainly comprise of bank balances and trade and other receivables. There are two prescribed subsequent measurement categories, depending on the Company's business model in managing the assets and the cash flow characteristic of the assets. The Company managed these group of financial assets by collecting the contractual cash flow and these cash flows represents solely payment of principal and interest. Accordingly, these group of financial assets are measured at amortised cost subsequent to initial recognition.

A gain or loss on a debt investment that is subsequently measured at amortised cost and is not part of a hedging relationship is recognised in profit or loss when the asset is derecognised or impaired. Interest income from these financial assets are recognised using the effective interest rate method.

The Company assesses on forward looking basis the expected credit losses associated with its debt instruments carried at amortised cost.

For trade receivables, the Company applied the simplified approach permitted by the FRS 109, which requires expected lifetime losses to be recognised from initial recognition of the receivables.

2. Significant accounting policies (continued)

2.8 Financial assets (continued)

(b) The accounting for financial assets from 29 January 2018 under FRS 109 are as follows: (continued)

(ii) At subsequent measurement (continued)

For bank balances and other receivables, the general 3 stage approach is applied. Credit loss allowance is based on 12-month expected credit loss if there is no significant increase in credit risk since initial recognition of the assets. If there is a significant increase in credit risk since initial recognition, lifetime expected credit loss will be calculated and recognised.

2.9 Trade and other payables

Trade and other payables represent liabilities for goods and services provided to the Company prior to the end of financial year which are unpaid. They are classified as current liabilities if payment is due within one year or less (or in the normal operating cycle of the business if longer). Otherwise, they are presented as non-current liabilities.

Trade and other payables are initially recognised at fair value and subsequently carried at amortised cost, using the effective interest method.

2.10 Operating leases

Leases of assets where substantially all risks and rewards incidental to ownership are retained by the lessor are classified as operating leases. Payments made under operating leases (net of any incentives received from the lessor) are recognised in profit or loss on a straight-line basis over the period of the lease.

2.11 Inventories

Inventories are carried at the lower of cost and net realisable value. Cost is determined using the first-in, first-out method. Net realisable value is the estimated selling price in the ordinary course of business, less applicable variable selling expenses.

2. Significant accounting policies (continued)

2.12 Income taxes

Current income tax for current and prior periods is recognised at the amount expected to be paid to or recovered from the tax authorities, using the tax rates and tax laws that have been enacted or substantively enacted by the balance sheet date.

Deferred income tax is recognised for all temporary differences except when the deferred income tax arises from the initial recognition of an asset or liability that affects neither accounting nor taxable profit or loss at the time of the transaction.

Current and deferred income tax is measured using the tax rates and tax laws that have been enacted or substantively enacted by the balance sheet date, and are recognised as income or expenses in profit or loss, except to the extent that the tax arises from a transaction which is recognised directly in equity.

2.13 Provisions for other liabilities and charges

Provisions for other liabilities and charges are recognised when the Company has a present legal or constructive obligation as a result of past events, it is more likely than not that an outflow of resources will be required to settle the obligation and the amount has been reliably estimated.

2.14 Currency translation

The financial statements of the Company are presented in United States Dollars, which is the functional currency of the Company.

Transactions in a currency other than United States Dollar ("foreign currency") are translated into United States Dollar using the exchange rates prevailing at the dates of the transactions. Currency translation differences resulting from the settlement of such transactions and from the translation of monetary assets and liabilities denominated in foreign currencies at the closing rates at the balance sheet date are recognised in the statement of comprehensive income.

All other foreign exchange gains and losses impacting profit or loss are presented in the statement of comprehensive income within "other losses - net".

2.15 Cash and cash equivalents

For the purpose of presentation in the cash flow statement, cash and cash equivalents include cash at bank and on hand which are subject to an insignificant risk of change in value.

2. Significant accounting policies (continued)

2.16 Share capital

Ordinary shares are classified as equity.

3. Critical accounting estimates, assumptions and judgements

Estimates, assumptions and judgements are continually evaluated and are based on historical experience and other factors, including expectations of future events that are believed to be reasonable under the circumstances

Share-based payments

The Company is required to expense its employees' share-based compensation options and RSUs in accordance with FRS 102 "Share-based Payment". The Company measures share-based compensation cost based on the fair value on the grant date of each option and RSU. This cost is recognised over the period during which an employee is required to provide service in exchange for the award and option or the requisite service period, usually the vesting period, and it is adjusted for actual forfeitures that occur before vesting. In order to assess the fair value of share options, the Company is required to use certain assumptions, including the forfeitures and the service period of each employee. The use of different assumptions and estimates could produce materially different estimated fair values for the share-based compensation options and related expenses.

4. Revenue

	29/01/2018 To 27/01/2019 US$'000	30/01/2017 to 28/01/2018 US$'000
Sale of graphics processors and related media and communication devices	**9,727,059**	8,064,722

Revenue from sale of graphics processors and related media and communication devices is recognised at a point in time.

* *The Company has initially applied FRS 115 using the modified retrospective method. Under this method, the comparative information is not restated.*

5. Other losses – net

	29/01/2018 To 27/01/2019 US$'000	30/01/2017 to 28/01/2018 US$'000
Exchange gains/(losses) - net	**363**	(490)

6. Other operating expenses

	29/01/2018 to 27/01/2019 US$'000	30/01/2017 to 28/01/2018 US$'000
Consulting and professional expense	**3,145**	2,579
Utilities expense	**98**	93
Repair and maintenance expense	**407**	441
Bank charges	**148**	128
(Reversal of allowance)/allowance for impairment of receivables	**(663)**	310
Insurance for credit	**(75)**	1,417
Engineering and testing expense	**1,494**	1,472
Other expenses	**143**	349
Total	**4,697**	6,789

7. Employee compensation

	29/01/2018 to 27/01/2019 US$'000	30/01/2017 to 28/01/2018 US$'000
Wages and salaries	**8,730**	6,681
Employer's contribution to defined contribution plans	**781**	667
Share-based compensation	**4,050**	1,695
	13,561	9,043

8. Income tax

(a) Income tax expense

	29/01/2018 to 27/01/2019 US$'000	30/01/2017 to 28/01/2018 US$'000
Income tax expense attributable to profit is made up of:		
- Current income tax	644	790

The tax expense on profit differs from the amount that would arise using the Singapore standard rate of income tax as explained below:

	29/01/2018 to 27/01/2019 US$'000	30/01/2017 to 28/01/2018 US$'000
Profit before tax	1,062	2,587
Tax calculated at a tax rate of 17% (2018: 17%)	181	440
Effects of:		
- Expenses not deductible for tax purposes	530	402
- Income not subject to tax	-	(2)
- Different tax rates in other country	(26)	(13)
- Others	(41)	(37)
Tax charge	644	790

(b) Movements in current tax liabilities

	27/01/2019 US$'000	28/01/2018 US$'000
Beginning of financial year	781	568
Tax expense on profit - current financial year	644	790
Income tax paid	(951)	(577)
End of the financial year	474	781

9. Cash and cash equivalents

	27/01/2019 US$'000	28/01/2018 US$'000
Cash at bank and on hand	**4,699**	7,509

10. Trade and other receivables

	27/01/2019 US$'000	28/01/2018 US$'000
Trade receivables:		
- Immediate holding corporation	**54**	54
- Non-related parties	**1,183,836**	939,851
Less: Allowance for impairment of receivables		
- Non-related parties	**(1,944)**	(2,607)
Trade receivables - net	**1,181,946**	937,298
Non-trade receivables:		
- Immediate holding corporation	**69,761**	111,571
- Related corporations	**96,898**	81,871
	166,659	193,442
Other receivables	**-**	15
	1,348,605	1,130,755

The non-trade receivables due from immediate holding corporations and related corporations are unsecured, interest-free and are repayable on demand.

11. Inventories

	27/01/2019 US$'000	28/01/2018 US$'000
Finished goods	**631,270**	335,435

12. Property, plant and equipment

	Computer equipment US$'000	Machinery and equipment US$'000	Office equipment US$'000	Furniture US$'000	Leasehold improvement US$'000	Total US$'000
2019						
Cost						
Beginning of financial year	171	1,056	43	-	241	1,511
Additions	10	10	44	5	201	270
Disposals	(40)	-	-	-	-	(40)
End of financial year	141	1,066	87	5	442	1,741
Accumulated depreciation						
Beginning of financial year	96	312	17	-	241	666
Depreciation charge	48	232	11	-	44	335
Disposals	(40)					(40)
End of financial year	104	544	28	-	285	961
Net book value						
End of financial year	**37**	**522**	**59**	**5**	**157**	**780**
2018						
Cost						
Beginning of financial year	153	1,036	23	26	241	1,479
Additions	18	60	20	-	-	98
Disposals	-	(40)	-	(26)	-	(66)
End of financial year	171	1,056	43	-	241	1,511
Accumulated depreciation						
Beginning of financial year	43	135	10	23	202	413
Depreciation charge	53	217	7	3	39	319
Disposals	-	(40)	-	(26)	-	(66)
End of financial year	96	312	17	-	241	666
Net book value						
End of financial year	**75**	**744**	**26**	**-**	**-**	**845**

13. Investment in subsidiary

	27/01/2019 US$'000	28/01/2018 US$'000
Unquoted equity shares at cost	**85**	85

Details of investment in subsidiary are as follows:

Name of company	Principal activities	Country of incorporation	Equity holding 27/01/2019 %	Equity holding 28/01/2018 %	Cost of investment 27/01/2019 US$'000	Cost of investment 28/01/2018 US$'000
NVIDIA GK	Provide marketing services to Nvidia Singapore Pte Ltd	Japan	**100**	100	**85**	85

The subsidiary does not require an audit in accordance with the laws in the country of its incorporation.

14. Trade and other payables

	27/01/2019 US$'000	28/01/2018 US$'000
Trade payable		
- Immediate holding corporation	**1,553,630**	1,157,464
	1,553,630	1,157,464
Non-trade payables		
- Immediate holding corporation	**4,745**	4,457
- Ultimate holding corporation	**15**	20,197
- Related corporations	**88,394**	77,065
	93,154	101,719
Accrued operating expenses	**286,785**	170,541
Other payables	**4,274**	5,226
	291,059	175,767
	1,937,843	1,434,950

The non-trade payables to immediate holding corporation, ultimate holding corporation and related corporations are unsecured, interest-free and are repayable on demand.

15. Share capital

The Company's share capital comprise fully paid-up 2 (2018: 2) ordinary shares with no par value, amounting to a total of US$1,000 (2018: US$1,000).

16. Other reserves

	27/01/2019 US$'000	28/01/2018 US$'000
Beginning of financial year	**12,120**	10,425
Employee share-based payment scheme		
- Value of employee services	**4,050**	1,695
End of financial year	**16,170**	12,120

17. Share based compensation

The employees of the Company are eligible for share-based compensation under equity-settled, shared based compensation plans operated by Nvidia Corporation, the ultimate holding corporation of the Company. Awards include share options, restricted share units ("RSUs"), and employee share purchase plan ("ESPP"), are granted under the following plans:

Amended and Restated 2007 Equity Incentive Plan
The Nvidia Corporation 2007 Equity Incentive Plan was approved in 2007, and subsequently amended and restated in 2012, 2013 and 2014, or the Restated 2007 Plan. This plan authorizes the issuance of share options and restricted share units.

Share options granted to employees, subject to certain exceptions, vest over a four year period subject to continued service, and generally expire ten years from the date of grant.

Subject to certain exceptions, RSUs granted to employees vest over a four year period, subject to continued service.

Amended and Restated 2012 Employee Stock Purchase Plan
The Nvidia Corporation 2012 Employee Share Purchase Plan was approved in 2012, and subsequently amended and restated in 2014, or the Restated 2012 Plan, as the successor to the 1998 Employee Share Purchase Plan.

Employees who participate in an offering may have up to 10% of their earnings withheld to the purchase of shares of common stock. The Board may increase this percentage at its discretion, up to 15%. The price of common stock purchased under our ESPP will be equal to 85% of the lower of the fair market value of the common stock on the commencement date of each offering period and the purchase date of each offering period. Employees may end their participation in the ESPP at any time during the offering period, and participation ends automatically on termination of employment with us. In each case, the employee's contributions are refunded.

The fair value of share option awards on the date of grant is determined using the Binomial model.

Fair value of RSUs is determined using the closing trading price of common stock on the date of grant, minus a dividend yield discount.

17. Share based compensation (continued)

The following table summarises the number and weighted average fair value of ESPP granted during the period:

	2019	2018
Number of shares issued under the plan to participating Nvidia Singapore Pte Ltd's employees on 27 January 2019 (28 January 2018)	**5,361**	27,359

The weighted average market price during the year ended 27 January 2019 was US$259.60 (2018: US$133.89) and the shares had an average grant date fair value of US$138.06 (2018: US$23.23).

The fair value of shares issued under the ESPP has been estimated at the date of grant with the following assumptions, using the Black-Scholes model:

	2019	2018
Weighted average expected life (years)	**0.1-2.0**	0.5-2.0
Risk free interest rate	**1.6%-2.8%**	0.8%-1.4%
Volatility	**24%-75%**	40%-54%
Dividend yield	**0.3%-0.4%**	0.3%-0.5%

Nvidia Corporation uses the implied volatility as it is expected to be more reflective of market conditions and, therefore, could reasonably be expected to be a better indicator of expected volatility than historical volatility. The risk-free interest rate is based upon observed interest rates on Treasury bills appropriate for the term of employee share options.

Dividend yield is based on history and expectation of dividend pay-outs. RSU awards are not eligible for cash dividends prior to vesting; therefore, the fair value of RSUs is discounted by dividend yield. For awards granted on or subsequent to 8 November 2012, Nvidia Corporation uses a dividend yield at grant date based on the per share dividends declared during the most recent quarter.

17. Share based compensation (continued)

A reconciliation of share options and RSU movements for the financial periods ended 27 January 2019 and 28 January 2018 is shown below (average exercise price and fair value reported in US dollars):

Share-based Payment Award Activity

The following table summarises activity for share options:

	Number of share options	Weighted average exercise price
27 January 2019		
Outstanding at beginning of the year	31,646	$14.67
Exercised*	(2,266)	$15.49
Forfeited/Cancel/Transfer	22,350	$14.24
Outstanding at end of the year	51,730	$14.45
Options exercisable at year end	51,730	$14.45

* The weighted average share price at the date of exercise of options exercised during the year ended 27 January 2019 was US$261.93 (2018: US$171.86). No options were granted during the financial years ended 27 January 2019.

	Number of share options	Weighted average exercise price
28 January 2018		
Outstanding at beginning of the year	33,869	$14.64
Exercised*	(2,223)	$14.24
Outstanding at end of the year	31,646	$14.67
Options exercisable at year end	31,646	$14.67

* The weighted average share price at the date of exercise of options exercised during the year ended 29 January 2018 was US$47.04. No options were granted during the financial years ended 29 January 2018.

17. **Share based compensation** (continued)

Range of exercise price details of awards outstanding as at 27 January 2019:

Range of exercise prices – US$	Outstanding options	Weighted average remaining contractual life (years)
US$10.01-US$15.00	36,700	3.32
US$15.01-US$20.00	15,030	3.26
	51,730	

Range of exercise price details of awards outstanding as at 28 January 2018:

Range of exercise prices – US$	Outstanding options	Weighted average remaining contractual life (years)
US$10.01-US$15.00	21,600	4.50
US$15.01-US$20.00	10,046	4.32
	31,646	

Restricted Share Units ("RSUs") Activity

The following table summarises activity for restricted share units:

	Number of RSUs	Weighted average grant date fair value
27 January 2019		
Outstanding at beginning of the year	68,273	$61.60
Granted	30,943	$229.05
Vested	(32,098)	$45.89
Forfeited/Cancelled	6,303	$196.92
Outstanding/non-vested at end of the year	73,421	$138.24

	Number of RSUs	Weighted average grant date fair value
28 January 2018		
Outstanding at beginning of the year	67,165	$30.77
Granted	14,223	$169.15
Vested	(28,000)	$28.67
Forfeited/Cancelled	14,885	$79.53
Outstanding/non-vested at end of the year	68,273	$61.60

18. Commitments

Operating lease commitments

The future minimum lease payable under non-cancellable operating leases contracted for at the balance sheet date but not recognised as liabilities are as follows:

	27/01/2019 US$'000	28/01/2018 US$'000
Not later than one year	**571**	544
Later than one year but not later than five years	**194**	765
	765	1,309

19. Financial risk management

Financial risk factors

The Company's activities expose it to market risk (including currency risk, price risk and interest rate), credit risk and liquidity risk.

The management team is responsible for setting the objectives and underlying principles of financial risk management for the Company, and establishes detailed policies such as risk identification and measurement, exposure limits and hedging strategies. Financial risk management is carried out by finance personnel.

The finance personnel measure actual exposures against the limits set and prepare regular reports for the review of the management team. The information presented below is based on information received by key management.

(a) Market risk

(i) Currency risk

The Company's revenue, cost of operations and majority of the financial assets and liabilities are primarily in United States Dollar ("USD"). The Company is not exposed to any significant currency risk. The Company did not enter into currency forward contracts to hedge this exposure.

(ii) Price risk

The Company has insignificant exposure to equity price risk as it does not hold any equity financial assets.

19. Financial risk management (continued)

(b) Credit risk

Credit risk refers to the risk that a counterparty will default on its contractual obligations resulting in financial loss to the Company.

(i) Risk management

The Company adopts the following policy to mitigate the credit risk.

For banks and financial institutions, the Company mitigates its credit risks by transacting only with counterparties who highly-rated by independent rating agencies.

For external customers, the Company adopts the policy of dealing only with customers of appropriate credit history, and obtaining sufficient collateral where appropriate to mitigate credit risk. In addition, the Company monitors its credit risk on an ongoing basis by reviewing the debtors' aging to minimise its exposure to credit risk.

Credit exposure to an individual customer is restricted by the credit limit approved by the credit controller. Customers' payment profile and credit exposure are continuously monitored by the credit controller and reported to the management and Board of Directors. The Company's trade receivables include 5 (2018: 5 debtors) that represented 42.7% and 40.7% of trade receivables at balance sheet date, respectively.

For intercompany receivables, the estimated lifetime credit losses is immaterial to the financial statements.

(ii) Credit rating

The Company uses the following categories of internal credit risk rating for financial assets which are subject to expected credit losses under the 3-stage general approach. These four categories reflect the respective credit risk and how the loss provision is determined for each of those categories.

Category of internal credit rating	Definition of category	Basis for recognition of expected credit losses
Performing	Low risk of default and a strong capacity to meet contractual cash flows	12-month expected credit losses
Underperforming	Significant increase in credit risk since initial recognition	Lifetime expected credit losses
Non-performing	Evidence indicating that the asset is impaired (*)	Lifetime expected credit losses
Write-off	No reasonable expectation of recovery	Asset is written-off

(*) Evidence such as significant difficulty, breach of contract and indicators of potential bankruptcy or other financial reorganisation.

19. Financial risk management (continued)

(b) Credit risk (continued)

(iii) Impairment of financial assets

All of the Company's financial assets are subject to immaterial credit losses where the expected credit loss model has been applied. The Company has applied the simplified approach by using the provision matrix to measure the lifetime expected credit losses for trade receivables from non-related customers.

To measure the expected credit losses, these receivables have been grouped based on shared credit risk characteristics and days past due. In calculating the expected credit loss rates, the Company considers historical loss rates for each category of customers, and adjusts for forward-looking macroeconomic data. The Company has identified the Gross Domestic Product ("GDP") of the countries in which it sell its goods to be the most relevant factor. There are no adjustments to historical loss rates based on expected changes in the factor.

Receivables are written off when there is no reasonable expectation of recovery, such as a debtor failing to engage in a repayment plan with the Company. There has been no historical trend of write off. Where receivables have been written off, the Company continues to engage in enforcement activity to attempt to recover the receivables due. Where recoveries are made, these are recognised in profit or loss.

The Company's credit risk exposure in relation to trade receivables from customers as at 27 January 2019 is minimal with all trade receivables at current.

		Past due			
27 January 2019	Current $'000	Within 30 days $'000	30 to 60 days $'000	More than 60 days $'000	Total $'000
Expected loss rates	-%	-%	-%	-%	
Trade receivables	1,183,836	-	-	-	1,183,836
Loss allowances	(1,944)	-	-	-	(1,944)

		Past due			
28 January 2018	Current $'000	Within 30 days $'000	30 to 60 days $'000	More than 60 days $'000	Total $'000
Expected loss rates	-%	-%	-%	-%	
Trade receivables	939,851	-	-	-	939,851
Loss allowances	(2,607)	-	-	-	(2,607)

19. Financial risk management (continued)

(b) Credit risk (continued)

(iii) Impairment of financial assets (continued)

The Company's credit risk exposure in relation to trade receivables under FRS 39 as at 28 January 2018 are set out as follows:

	Past due			
	Within 30 days $'000	30 to 60 days $'000	More than 60 days $'000	Total $'000
Trade receivables				
Gross carrying amount:				
- Not past due	-	-	-	939,851
				939,851
Less: Allowances for impairment				(2,607)
Net carrying amount				937,298

In 2018, the impairment of the financial assets was assessed based on the incurred loss impairment model. Individual receivables which were known to be uncollectible were written off by reducing the carrying amount directly.

The other receivables were assessed collectively, to determine whether there was objective evidence that an impairment had been incurred but not yet identified.

The Company considered that there was evidence if any of the following indicators were present:

- There is significant difficulty of the debtor.
- Breach of contract, such as default or past due event
- It is becoming probable that the debtor will enter bankruptcy or other financial reorganisation

Financial assets that are neither past due nor impaired

Cash and bank balances that are neither past due nor impaired are mainly deposits with banks which have high credit-ratings as determined by international credit-rating agencies. Trade and other receivables and deposits that are neither past due nor impaired are substantially companies with good collection track records with the Company.

There are no credit loss allowance for other financial asset at amortised cost as at 27 January 2019.

19. Financial risk management (continued)

(c) Liquidity risk

The Company manage the liquidity risk by maintaining sufficient cash and bank balances to enable it to meet its operational requirements.

(d) Capital risk

The Company's objectives when managing capital are to ensure that the Company is adequately capitalised and to maintain an optimal capital structure.

The Company is not subject to any externally imposed capital requirements.

(e) Financial instruments by category

The carrying amounts of financial assets measured at fair value through profit or loss are disclosed on the face of the balance sheet. The aggregate carrying amounts of loans and receivables and financial liabilities at amortised cost are as follows:

	27/01/2019 US$'000
Financial assets, at amortised cost	**1,353,303**
Financial liabilities, at amortised cost	**1,937,459**

	28/01/2018 US$'000
Loans and receivables	1,138,248
Financial liabilities, at amortised cost	1,434,659

19. Financial risk management (continued)

(f) Offsetting financial assets and financial liabilities

(i) Financial assets

The Company has the following financial instrument subject to enforceable master netting arrangement or similar agreement as follows:

	Related amounts set off			Related amounts not set off	
	Gross amounts - financial assets (a) US$'000	Gross amounts - financial liabilities (b) US$'000	Net amounts - financial assets presented in the balance sheet (c)=(a)-(b) US$'000	Financial assets (d) US$'000	Net amount (e)=(c)+(d) US$'000
As at 27 January 2019					
Trade and other receivables	**213,505**	**143,744**	**69,761**	**96,952**	**166,713**
Total	**213,505**	**143,744**	**69,761**	**96,952**	**166,713**
As at 28 January 2018					
Trade and other receivables	211,720	100,149	111,571	81,925	193,496
Total	211,720	100,149	111,571	81,925	193,496

(ii) Financial liabilities

The Company has the following financial instruments subject to enforceable master netting arrangements or similar agreement as follows:

	Related amounts set off			Related amounts not set off	
	Gross amounts – financial liabilities (a) US$'000	Gross amounts – financial assets (b) US$'000	Net amounts – financial liabilities presented in the balance sheet (c)=(a)-(b) US$'000	Financial liabilities (d) US$'000	Net amount (e)=(c)+(d) US$'000
As at 27 January 2019					
Trade and other payables	**1,975,793**	**422,163**	**1,553,630**	**93,154**	**1,646,784**
Total	**1,975,793**	**422,163**	**1,553,630**	**93,154**	**1,646,784**
As at 28 January 2018					
Trade and other payables	1,564,710	407,246	1,157,464	101,719	1,259,183
Total	1,564,710	407,246	1,157,464	101,719	1,259,183

20. Immediate and ultimate holding corporations

The Company's immediate holding corporation is Nvidia International Inc., incorporated in the Cayman Islands. The Company's ultimate holding corporation is Nvidia Corporation, incorporated in United States of America.

21. Related party transactions

In addition to the information disclosed elsewhere in the financial statements, the following transactions took place between the Company and related parties at terms agreed between the parties:

(a) Transactions

	29/01/2018 to 27/01/2019 US$'000	30/01/2017 to 28/01/2018 US$'000
Purchases from immediate holding company	**8,436,643**	6,967,070
Payments made on behalf of fellow subsidiaries	**1,780**	-
Sales of goods to a fellow subsidiary	**-**	1,470
Marketing and administrative services fees charged by related corporations	**246,944**	210,571

The Company participates in the centralised cash pooling arrangement with Nvidia Corporation, its ultimate holding company. During the year, certain balances with related companies were settled and netted off through the cash pooling arrangement.

(b) Key management personnel compensation

There are no key management remuneration for the financial years ended 27 January 2019 and 28 January 2018 as the key management having authority and responsibility for planning, directing and controlling the activities of the Company are employed by related corporations.

22. New or revised accounting standards and interpretations

Below are the mandatory standards, amendments and interpretations to existing standards that have been published, and are relevant for the Group's accounting periods beginning on or after 28 January 2019 and which the Group has not early adopted:

(a) FRS 116 *Leases* (effective for annual periods beginning on or after 1 January 2019)

FRS 116 will result in almost all leases being recognised on the balance sheet, as the distinction between operating and finance leases is removed. Under the new standard, an asset (the right to use the leased item) and a financial liability to pay rentals are recognised.

22. New or revised accounting standards and interpretations (continued)

(a) FRS 116 *Leases* (effective for annual periods beginning on or after 1 January 2019) (continued)

The only exceptions are short-term and low-value leases. The accounting for lessors will not change significantly.

As at the reporting date, the Company has non-cancellable operating lease commitments of $765,200 (Note 18). Of these commitments, none of these operating leases relate to a short-term lease which will be recognised on a straight-line basis as expense in profit or loss.

For the remaining lease commitments the Company expects that the impact on net current assets, net profit after tax and cash flows to be insignificant to the financial statements.

(b) INT FRS 123 Uncertainty Over Income Tax Treatments (effective for annual periods beginning on or after 1 January 2019)

The interpretation explains how to recognise and measure deferred and current income tax assets and liabilities where there is uncertainty over a tax treatment. In particular, it discusses:

i) how to determine the appropriate unit of account, and that each uncertain tax treatment should be considered separately or together as a group, depending on which approach better predicts the resolution of the uncertainty;

ii) that the entity should assume a tax authority will examine the uncertain tax treatments and have full knowledge of all related information, i.e. that detection risk should be ignored

iii) that the entity should reflect the effect of the uncertainty in its income tax accounting when it is not probable that the tax authorities will accept the treatment

iv) that the impact of the uncertainty should be measured using either the most likely amount or the expected value method, depending on which method better predicts the resolution of the uncertainty, and

v) that the judgements and estimates made must be reassessed whenever circumstances have changed or there is new information that affects the judgements.

The Company does not expect additional tax liability to be recognised arising from the uncertain tax positions on the adoption of the interpretation on 28 January 2019.

23. Authorisation of financial statements

These financial statements were authorised for issue in accordance with a resolution of the Board of Directors of Nvidia Singapore Pte Ltd on 27 JUN 2019

EXHIBIT O

ACCOUNTING AND CORPORATE REGULATORY AUTHORITY (ACRA)

INFORMATION RESOURCES

WHILST EVERY ENDEAVOR IS MADE TO ENSURE THAT INFORMATION PROVIDED IS UPDATED AND CORRECT. THE AUTHORITY DISCLAIMS ANY LIABILITY FOR ANY DAMAGE OR LOSS THAT MAY BE CAUSED AS A RESULT OF ANY ERROR OR OMISSION.

Corporate Compliance and Financial Profile of NVIDIA SINGAPORE PTE LTD (200003831M)

Date :18/08/2020

The Following Are The Brief Particulars of :

Registration No.	:	200003831M
Company Name.	:	NVIDIA SINGAPORE PTE LTD
Former Name if any	:	
Incorporation Date.	:	04/05/2000
Company Type	:	PRIVATE COMPANY LIMITED BY SHARES
Status	:	Live Company
Status Date	:	04/05/2000

Principal Activities

Activities (I)	:	WHOLESALE OF ELECTRONIC COMPONENTS(46522)
Description	:	SALE OF GRAPHIC PROCESSORS, AND MEDIA & COMMUNICATION DEVICES
Activities (II)	:	
Description	:	

Capital

Issued Share Capital (AMOUNT)	Number of Shares *	Currency	Share Type
2000	2	SINGAPORE, DOLLARS	ORDINARY

* Number of Shares includes number of Treasury Shares

Paid-Up Capital (AMOUNT)	Number of Shares	Currency	Share Type
2000		SINGAPORE, DOLLARS	ORDINARY

COMPANY HAS THE FOLLOWING ORDINARY SHARES HELD AS TREASURY SHARES

Number Of Shares	Currency

Registered Office Address	:	112 ROBINSON ROAD #05-01 ROBINSON 112 SINGAPORE (068902)

INFORMATION RESOURCES

Corporate Compliance and Financial Profile of NVIDIA SINGAPORE PTE LTD (200003831M)

Date :18/08/2020

Date of Address	:	01/12/2012
Date of Last AGM	:	16/07/2019
Date of Last AR	:	24/07/2019
FYE As At Date of Last AR	:	27/01/2019

Financial Information

Financial Period	29/01/2018 to 27/01/2019	30/01/2017 to 28/01/2018	01/02/2016 to 29/01/2017
Extracted From: #	XBRL FS for period ended 27 JAN 2019	XBRL FS for period ended 27 JAN 2019	XBRL FS for period ended 28 JAN 2018
Are there any changes to figures reported by Company compared to prior year FS?	-	-	-
Company / Group Account	Company	Company	Company
Currency	USD	USD	USD
Balance Sheet			
Total Assets	1,986,302,000.00	1,479,248,000.00	1,409,295,000.00
Total Current Assets	1,985,437,000.00	1,478,318,000.00	1,407,644,000.00
Total Liabilities	1,938,317,000.00	1,435,731,000.00	1,369,270,000.00
Total Current Liabilities	1,938,317,000.00	1,435,731,000.00	1,369,270,000.00
Retained Earnings (Accumulated Loss)	31,814,000.00	31,396,000.00	29,599,000.00
Profit and Loss			
Revenue	9,727,059,000.00	8,064,722,000.00	5,712,400,000.00
Profit(Loss) before tax from continuing operations	1,062,000.00	2,587,000.00	3,002,000.00
Profit(Loss) after tax from continuing operations	418,000.00	1,797,000.00	2,459,000.00
Profit(Loss) after tax from discontinued operations	*	*	*
EBIT	1,062,000.00	2,587,000.00	3,002,000.00

INFORMATION RESOURCES

Corporate Compliance and Financial Profile of NVIDIA SINGAPORE PTE LTD (200003831M)

Date :18/08/2020

Cashflow Statement

Net cashflow from (used in) operating activities	-2,540,000.00	-92,301,000.00	-122,747,000.00
Net cashflow from (used in) investing activities	-270,000.00	402,000.00	-1,055,000.00
Net cashflow from (used in) financing activities	0.00	0.00	0.00

Financial Ratios

Current Ratio (Times)	1.02	1.03	1.03
Operating Profit Margin (%)	0.01	0.03	0.05
Net Profit Margin (%)	0.00	0.02	0.04
Return on Assets (%)	0.05	0.17	0.21
Return on Equity (%)	0.87	4.13	6.14
Total Liabilities to Equity (Times)	40.39	32.99	34.21
Total Asset Turnover (Times)	4.90	5.45	4.05
Interest Cover Ratio (Times)	*	*	*

INFORMATION RESOURCES

Corporate Compliance and Financial Profile of NVIDIA SINGAPORE PTE LTD (200003831M)

Date :18/08/2020

Non-Financial Information

Extracted From: #	XBRL FS for period ended 27 JAN 2019	XBRL FS for period ended 28 JAN 2018	XBRL FS for period ended 29 JAN 2017
Audit Opinion in Auditors' Report:	Unqualified opinion	Unqualified opinion	Unqualified opinion
Reasons for the modified Audit Opinion:	*	*	*
Whether there is any material uncertainty relating to going concern, reported in Auditors' Report:	*	*	*
Audit Firm:	PRICEWATERHOUSECOOPERS LLP	PRICEWATERHOUSECOOPERS LLP	PRICEWATERHOUSECOOPERS LLP
In the Statement by Directors, are the Directors of the opinion that the FS are drawn up to exhibit a true and fair view?	Yes	Yes	Yes

Notes

* : Information is not available. Non-availability may be due to any of the following reasons:

- Non-filing of XBRL FS (e.g. filing of PDF FS only or FS not required to be filed with ACRA); or
- Assets and liabilities presented in order of liquidity, instead of current/non-current classification; or
- Exempted from audit requirements;
- Company falling within the qualification criteria to file a reduced set of elements in XBRL;

: Financial and Non-Financial information may be extracted from different years of XBRL FS.

For detailed definition of the audit opinion, financial ratio formulae and explanation, visit:

https://www.acra.gov.sg/how-to-guides/buying-information/corporate-compliance-and-financial-profile

Audit Firms

NAME
PRICEWATERHOUSECOOPERS LLP

INFORMATION RESOURCES

Corporate Compliance and Financial Profile of NVIDIA SINGAPORE PTE LTD (200003831M)

Date :18/08/2020

Charges

Charge No.	Date Registered	Currency	Amount Secured	Chargee(s)

Officers/Authorised Representative(s)

Name / Address	ID	Nationality/Citizenship / Position Held	Source of Address	Date of Appointment
REBECCA PETERS [redacted]CAPITOLA, CA 95010, USA	481604479	AMERICAN Director	ACRA	31/05/2016
MICHAEL JOHN BYRON [redacted] DUBLIN, CA 94568, UNITED STATES OF AMERICA	506033576	AMERICAN Director	ACRA	31/10/2013
KAREN THERESA BURNS [redacted]PALO ALTO CALIFORNIA 94301, U.S.A.	506005070	AMERICAN Director	ACRA	27/04/2009
LEE KAY BENG [redacted] SINGAPORE (438808)	S2539793B	MALAYSIAN Director	ACRA	18/07/2002
CHAN CHOW PHENG 112 ROBINSON ROAD #05-01 ROBINSON 112 SINGAPORE (068902)	S1298885J	SINGAPORE CITIZEN Secretary	ACRA	13/06/2011
TEO CHIN KEE 112 ROBINSON ROAD #05-01 ROBINSON 112 SINGAPORE (068902)	S2622115C	SINGAPORE CITIZEN Secretary	ACRA	05/01/2005

Shareholder(s)

	Name / Address	ID	Nationality/Citizenship Place of incorporation/ Origin/Registration	Source of Address	Address Changed
1	NVIDIA INTERNATIONAL INC.	T00UF0363E	CAYMAN ISLANDS	ACRA	

ACCOUNTING AND CORPORATE REGULATORY AUTHORITY
(ACRA)

INFORMATION RESOURCES

WHILST EVERY ENDEAVOR IS MADE TO ENSURE THAT INFORMATION PROVIDED IS UPDATED AND CORRECT. THE AUTHORITY DISCLAIMS ANY LIABILITY FOR ANY DAMAGE OR LOSS THAT MAY BE CAUSED AS A RESULT OF ANY ERROR OR OMISSION.

Corporate Compliance and Financial Profile of NVIDIA SINGAPORE PTE LTD (200003831M)

Date :18/08/2020

Shareholder(s)

Name / Address	ID	Nationality/Citizenship / Place of incorporation/ Origin/Registration	Source of Address	Address Changed
P.O. BOX 448, ELGIN COURT, ELGIN AVENUE, GEORGE TOWN, GRAND CAYMAN, KY1-1106, CAYMAN ISLANDS				

Ordinary(Number)	Currency
2	SINGAPORE, DOLLARS

Abbreviation

UL - Local Entity not registered with ACRA

UF - Foreign Entity not registered with ACRA

AR - Annual Return

AGM - Annual General Meeting

FS - Financial Statements

FYE - Financial Year End

OSCARS - One Stop Change of Address Reporting Service by Immigration & Checkpoint Authority.

Note :

- The information contained in this product is collated from lodgements filed with ACRA, and/or information collected by other government sources.

FOR REGISTRAR OF COMPANIES AND BUSINESS NAMES
SINGAPORE

RECEIPT NO. : ACRA200818177017

DATE : 18/08/2020

This is computer generated. Hence no signature required.

EXHIBIT P

UNITED STATES
SECURITIES AND EXCHANGE COMMISSION
Washington, D.C. 20549

FORM 10-K

[x] ANNUAL REPORT PURSUANT TO SECTION 13 OR 15(d) OF THE SECURITIES EXCHANGE ACT OF 1934
For the fiscal year ended January 27, 2019

OR

[_] TRANSITION REPORT PURSUANT TO SECTION 13 OR 15(d) OF THE SECURITIES EXCHANGE ACT OF 1934

Commission file number: 0-23985

NVIDIA CORPORATION

(Exact name of registrant as specified in its charter)

Delaware	**94-3177549**
(State or other jurisdiction of Incorporation or Organization)	**(I.R.S. Employer Identification No.)**

2788 San Tomas Expressway
Santa Clara, California 95051
(408) 486-2000

(Address, including zip code, and telephone number, including area code, of principal executive offices)
Securities registered pursuant to Section 12(b) of the Act:

Title of each class	**Name of each exchange on which registered**
Common Stock, $0.001 par value per share	The Nasdaq Global Select Market

Securities registered pursuant to Section 12(g) of the Act:
None

Indicate by check mark if the registrant is a well-known seasoned issuer, as defined in Rule 405 of the Securities Act. Yes ☒ No ☐

Indicate by check mark if the registrant is not required to file reports pursuant to Section 13 or Section 15(d) of the Act. Yes ☐ No ☒

Indicate by check mark whether the registrant (1) has filed all reports required to be filed by Section 13 or 15(d) of the Securities Exchange Act of 1934 during the preceding 12 months (or for such shorter period that the registrant was required to file such reports), and (2) has been subject to such filing requirements for the past 90 days. Yes ☒ No ☐

Indicate by check mark whether the registrant has submitted electronically every Interactive Data File required to be submitted pursuant to Rule 405 of Regulation S-T (§232.405 of this chapter) during the preceding 12 months (or for such shorter period that the registrant was required to submit such files). Yes ☒ No ☐

Indicate by check mark if disclosure of delinquent filers pursuant to Item 405 of Regulation S-K (§ 229.405 of this chapter) is not contained herein, and will not be contained, to the best of registrant's knowledge, in definitive proxy or information statements incorporated by reference in Part III of this Form 10-K or any amendment to this Form 10-K. ☐

Indicate by check mark whether the registrant is a large accelerated filer, an accelerated filer, a non-accelerated filer, a smaller reporting company, or an emerging growth company. See definitions of "large accelerated filer", "accelerated filer", "smaller reporting company", and "emerging growth company" in Rule 12b-2 of the Exchange Act.

Large accelerated filer ☒ Accelerated filer ☐ Non-accelerated filer ☐ Smaller reporting company ☐ Emerging growth company ☐

If an emerging growth company, indicate by check mark if the registrant has elected not to use the extended transition period for complying with any new or revised financial accounting standards provided pursuant to Section 13(a) of the Exchange Act. ☐

Indicate by check mark whether the registrant is a shell company (as defined in Rule 12b-2 of the Act). Yes ☐ No ☒

The aggregate market value of the voting stock held by non-affiliates of the registrant as of July 27, 2018 was approximately $146.66 billion (based on the closing sales price of the registrant's common stock as reported by the Nasdaq Global Select Market on July 27, 2018). This calculation excludes 26 million shares held by directors and executive officers of the registrant. This calculation does not exclude shares held by such organizations whose ownership exceeds 5% of the registrant's outstanding common stock that have represented to the registrant that they are registered investment advisers or investment companies registered under section 8 of the Investment Company Act of 1940.

The number of shares of common stock outstanding as of February 15, 2019 was 606 million .

DOCUMENTS INCORPORATED BY REFERENCE

Portions of the registrant's Proxy Statement for its 2019 Annual Meeting of Shareholders to be filed with the Securities and Exchange Commission pursuant to Regulation 14A not later than 120 days after the end of the fiscal year covered by this Annual Report on Form 10-K are incorporated by reference into Part III, Items 10-14 of this Annual Report on Form 10-K.

NVIDIA CORPORATION
TABLE OF CONTENTS

WHERE YOU CAN FIND MORE INFORMATION

Investors and others should note that we announce material financial information to our investors using our investor relations website, press releases, SEC filings and public conference calls and webcasts. We also use the following social media channels as a means of disclosing information about the company, our products, our planned financial and other announcements and attendance at upcoming investor and industry conferences, and other matters and for complying with our disclosure obligations under Regulation FD:

NVIDIA Twitter Account (https://twitter.com/nvidia)

NVIDIA Company Blog (http://blogs.nvidia.com)

NVIDIA Facebook Page (https://www.facebook.com/nvidia)

NVIDIA LinkedIn Page (http://www.linkedin.com/company/nvidia)

NVIDIA Instagram Page (https://www.instagram.com/nvidia)

In addition, investors and others can view NVIDIA videos on YouTube.

The information we post through these social media channels may be deemed material. Accordingly, investors should monitor these accounts and the blog, in addition to following our press releases, SEC filings and public conference calls and webcasts. This list may be updated from time to time. The information we post through these channels is not a part of this Annual Report on Form 10-K. These channels may be updated from time to time on NVIDIA's investor relations website.

Forward-Looking Statements

This Annual Report on Form 10-K contains forward-looking statements within the meaning of Section 27A of the Securities Act of 1933, as amended, and Section 21E of the Securities Exchange Act of 1934, as amended, which are subject to the "safe harbor" created by those sections. Forward-looking statements are based on our management's beliefs and assumptions and on information currently available to our management. In some cases, you can identify forward-looking statements by terms such as "may," "will," "should," "could," "goal," "would," "expect," "plan," "anticipate," "believe," "estimate," "project," "predict," "potential" and similar expressions intended to identify forward-looking statements. These statements involve known and unknown risks, uncertainties and other factors, which may cause our actual results, performance, time frames or achievements to be materially different from any future results, performance, time frames or achievements expressed or implied by the forward-looking statements. We discuss many of these risks, uncertainties and other factors in this Annual Report on Form 10-K in greater detail under the heading "Risk Factors." Given these risks, uncertainties and other factors, you should not place undue reliance on these forward-looking statements. Also, these forward-looking statements represent our estimates and assumptions only as of the date of this filing. You should read this Annual Report on Form 10-K completely and with the understanding that our actual future results may be materially different from what we expect. We hereby qualify our forward-looking statements by these cautionary statements. Except as required by law, we assume no obligation to update these forward-looking statements publicly, or to update the reasons actual results could differ materially from those anticipated in these forward-looking statements, even if new information becomes available in the future.

All references to "NVIDIA," "we," "us," "our" or the "Company" mean NVIDIA Corporation and its subsidiaries.

In addition, statements that "we believe" and similar statements reflect our beliefs and opinions on the relevant subject. These statements are based upon information available to us as of the filing date of this Annual Report on Form 10-K , and while we believe such information forms a reasonable basis for such statements, such information may be limited or incomplete, and our statements should not be read to indicate that we have conducted an exhaustive inquiry into, or review of, all potentially available relevant information. These statements are inherently uncertain and investors are cautioned not to unduly rely upon these statements.

PART I

ITEM 1. BUSINESS

Our Company

Starting with a focus on PC graphics, NVIDIA invented the graphics processing unit, or GPU, to solve some of the most complex problems in computer science. We have extended our focus in recent years to the revolutionary field of artificial intelligence, or AI. Fueled by the sustained demand for better 3D graphics and the scale of the gaming market, NVIDIA has evolved the GPU into a computer brain at the intersection of virtual reality, or VR, high performance computing, or HPC, and AI.

The GPU was initially used to simulate human imagination, enabling the virtual worlds of video games and films. Today, it also simulates human intelligence, enabling a deeper understanding of the physical world. Its parallel processing capabilities, supported by up to thousands of computing cores, are essential to running deep learning algorithms. This form of AI, in which software writes itself by learning from data, can serve as the brain of computers, robots and self-driving cars that can perceive and understand the world. GPU-powered deep learning continues to be adopted by thousands of enterprises to deliver services and features that would have been impossible with traditional coding.

NVIDIA has a platform strategy, bringing together hardware, system software, programmable algorithms, libraries, systems, and services to create unique value for the markets we serve. While the requirements of these end markets are diverse, we address them with a unified underlying architecture leveraging our GPUs and Compute Unified Device Architecture, or CUDA, as the fundamental building blocks. The programmable nature of our architecture allows us to support several multi-billion dollar end markets with the same underlying technology by using a variety of software stacks developed either internally or by third party developers and partners. The large and growing number of developers across our platforms strengthens our ecosystem and increases the value of our platform to our customers.

Innovation is at our core. We have invested over $17 billion in research and development since our inception, yielding inventions that are essential to modern computing. Our invention of the GPU in 1999 defined modern computer graphics and established NVIDIA as the leader in visual computing. With our introduction of the CUDA programming model in 2006, we opened the parallel processing capabilities of the GPU for general purpose computing. This approach significantly accelerates the performance of the most demanding applications in HPC in fields such as aerospace, bio-science research, mechanical and fluid simulations, and energy exploration. Today, our GPUs power the fastest supercomputers across the world. In addition, the massively parallel compute architecture of our GPUs and associated software have proven to be well suited for deep learning and are now expanding into machine learning, powering the era of AI. As the laws of physics have begun to slow down Moore’s Law, we continue to deliver GPU performance improvements ahead of Moore’s Law, giving the industry a path forward.

Gamers choose NVIDIA GPUs to enjoy immersive, increasingly cinematic virtual worlds. GPUs also help underpin the world’s fastest growing spectator sport, eSports, which attracts hundreds of millions of viewers to watch top-quality gaming. A rapidly growing new genre of Battle Royale games, such as Fortnite, is also expanding the gaming market.

Researchers use our GPUs to accelerate a wide range of important applications, from simulating viruses to exploring the origins of the universe. With support for more than 550 applications - including the top 15 HPC applications - NVIDIA GPUs enable some of the most promising areas of discovery, from weather prediction to materials science and from wind tunnel simulation to genomics. In 2018, NVIDIA GPUs powered the top two supercomputers in the world, located at Oak Ridge and Lawrence Livermore National Laboratories in the United States, as well as the top supercomputers in Europe and Japan. Five of the six finalists for the Gordon Bell Prize, awarded by the Association for Computing Machinery for outstanding achievement in the field of computing for applications in science, engineering and large-scale data science, did their work on the NVIDIA-powered top-two supercomputers.

The world’s leading cloud service providers use our GPUs to enable, accelerate or enrich the services they deliver to billions of end-users, including search, social networking, online shopping, live video, translation, AI assistants, navigation, and cloud computing.

A rapidly growing number of enterprises and startups use our GPUs to facilitate deep learning that meets, and in several cases surpasses, human perception, in fields ranging from radiology to precision agriculture. For example, the transportation industry is turning to our GPUs and AI to enable autonomous vehicles, or AVs, with several hundred companies and organizations working with NVIDIA’s DRIVE platform.

Professional designers use our GPUs to create visual effects in movies and design products ranging from soft drink bottles to commercial aircraft.

Headquartered in Santa Clara, California, NVIDIA was incorporated in California in April 1993 and reincorporated in Delaware in April 1998.

Our Businesses

Our two reportable segments - GPU and Tegra Processor - are based on a single underlying architecture. Our GPU product brands are aimed at specialized markets including GeForce for gamers; Quadro for designers; Tesla and DGX for AI data scientists and big data researchers; and GRID for cloud-based visual computing users. Our Tegra brand integrates an entire computer onto a single chip, and incorporates GPUs and multi-core CPUs to drive supercomputing for autonomous robots, drones, and cars, as well as for game consoles and mobile gaming and entertainment devices.

GPU	· **GeForce** for PC gaming and mainstream PCs · **GeForce NOW** for cloud-based game-streaming service · **Quadro** for design professionals working in computer-aided design, video editing, special effects, and other creative applications · **Tesla** for AI utilizing deep learning and accelerated computing, leveraging the parallel computing capabilities of GPUs for general purpose computing · **GRID** to provide the power of NVIDIA graphics through the cloud and datacenters · **DGX** for AI scientists, researchers and developers
Tegra Processor	· **Tegra** processors are primarily designed to enable branded platforms - DRIVE and SHIELD · **DRIVE AGX** automotive supercomputers and software stacks that provide self-driving capabilities · **Clara AGX** for intelligent medical instruments · **SHIELD** devices and services designed to harness the power of mobile-cloud to revolutionize home entertainment, AI and gaming · **Jetson AGX** is a power-efficient AI computing platform for robotics and other embedded use

Our Markets

We specialize in markets in which GPU-based visual computing and accelerated computing platforms can provide tremendous throughput for applications. These platforms incorporate processors, systems software, programmable algorithms, systems, and services to deliver value that is unique in the marketplace. From our proprietary processors, we have created platforms that address four large markets where our expertise is critical: Gaming, Professional Visualization, Datacenter, and Automotive.

Gaming

Computer gaming is the largest entertainment industry. Many factors propel computer gaming's growth, including new high production value games and franchises, the rise of competitive online gaming, eSports, and the rise of virtual and augmented reality.

Our GPUs enhance the gaming experience by improving the visual quality of graphics, increasing the frame rate for smoother gameplay and improving realism by incorporating the behavior of light and physical objects. These can be enjoyed independently or together to extend the gaming experience across platforms.

Our gaming platforms utilize sophisticated 3D software and algorithms, including our GameWorks libraries that provide special effects for games. We further enhance gaming with GeForce Experience, our gaming application that optimizes the PC user's settings for each title and enables players to record and share gameplay. It has been downloaded by more than 100 million users.

To enable VR, we provide developers with a suite of software libraries called VRWorks. VRWorks allows developers to create fully immersive experiences by enabling physically realistic visuals, sound, touch interactions, and simulated environments. VR requires advanced high-performance GPUs as the engine to simulate complete immersion.

Our products for the gaming market include GeForce RTX and GeForce GTX GPUs for PC gaming, SHIELD devices for gaming and streaming, GeForce NOW for cloud-based gaming, as well as platforms and development services for specialized console gaming devices.

Professional Visualization

We serve the Professional Visualization market by working closely with independent software vendors to optimize their offerings for NVIDIA GPUs. Our GPU computing solutions enhance productivity and introduce new capabilities for critical parts of the workflow for such major industries as automotive, media and entertainment, architectural engineering, oil and gas, and medical imaging.

Designers who build the products we use every day need the images that they view digitally to mirror reality. This requires simulating the physical behavior of light and materials, or physically-based rendering, an emerging trend in professional design. Our DesignWorks software delivers this to designers and enables an architect designing a building with a computer-aided design package to interact with the model in real time, view it in greater detail, and generate photorealistic renderings for the client. It also allows an automotive designer to create a highly realistic 3D image of a car, which can be viewed from all angles, reducing reliance on costly, time-consuming full-scale clay models.

Just as VR is becoming more important in gaming, it is also being incorporated in a growing number of enterprise applications, including within medicine, architecture, product design, and retail. Virtual car showrooms, surgical training, architectural walkthroughs, and bringing historical scenes to life all deploy this technology, powered by our GPUs.

Visual computing is vital to productivity in many environments, including design and manufacturing and digital content creation. Design and manufacturing includes computer-aided design, architectural design, consumer-products manufacturing, medical instrumentation, and aerospace. Digital content creation includes professional video editing and post production, special effects for films, and broadcast-television graphics.

Our brand for this market is Quadro for workstations. Quadro GPUs enhance the productivity of designers by improving performance and adding functionality, such as photorealistic rendering, high color fidelity, and advanced scalable display capabilities. During fiscal year 2019, we introduced the NVIDIA RTX platform, making it possible to render film-quality, photorealistic objects and environments with physically accurate shadows, reflections and refractions using ray tracing in real-time.

Datacenter

The NVIDIA accelerated computing platform addresses AI and HPC applications. The platform consists of our energy efficient GPUs, our CUDA programming language, specific libraries such as cuDNN and TensorRT, and innovations such as NVLink, which enables application scalability across multiple GPUs.

In the field of AI, NVIDIA's platform accelerates both deep learning and machine learning workloads. Deep learning is a computer science approach where neural networks are trained to recognize patterns from massive amounts of data in the form of images, sounds and text - in some instances better than humans. Machine learning is a related approach that leverages algorithms as well as data to learn how to make determinations or predictions, often used in data science. HPC, also referred to as scientific computing, uses numerical computational approaches to solve large and complex problems. For both AI and HPC applications, the NVIDIA accelerated computing platform greatly increases the performance and power efficiency of high-performance computers and datacenters, as GPUs excel at parallel workloads. For example, an NVIDIA GPU-accelerated machine learning cluster for data science is 1/8 the cost, 1/15 the space, and 1/18 the power of a traditional CPU-based cluster.

We are engaged with thousands of organizations working on AI in a multitude of industries, from automating tasks such as reading medical images, to enabling fraud detection in financial services, to optimizing oil exploration and drilling. These organizations include the world's leading cloud services companies such as Amazon, Baidu, and Facebook, which are infusing AI in applications that enable highly accurate voice recognition and real-time translation; enterprises that are increasingly turning to AI to improve products and services; and startups seeking to implement AI in transformative ways across multiple industries. We have partnered with industry leaders such as IBM, Microsoft, Oracle, and SAP to bring AI to enterprise users. We also have partnerships in healthcare and manufacturing, among others, to accelerate the adoption of AI.

To enable deep learning and machine learning, we provide a family of GPUs designed to speed up training and inferencing of neural networks. They are available in industry standard servers from every major computer maker worldwide, including Cisco, Dell, HP, Inspur, and Lenovo; from every major cloud service provider such as Alicloud, Amazon Web Services, Baidu Cloud, Google Cloud, IBM Cloud, Microsoft Azure, and Oracle Cloud; as well as in our DGX AI supercomputer, a purpose-built system for deep learning and GPU accelerated applications. DGX delivers performance equal to hundreds of conventional servers, comes fully integrated with hardware, software, development tools, support for AI frameworks, and runs popular accelerated applications. We also offer the NVIDIA GPU Cloud, or NGC, a comprehensive catalog of easy-to-use, optimized software stacks across a range of domains including scientific computing, deep learning, and machine learning. With NGC,

AI developers, researchers and data scientists can get started with the development of AI and HPC applications and deploy them on DGX systems, NGC-ready workstations or servers from our systems partners, or with NVIDIA's cloud partners such as Amazon, Google Cloud, Microsoft Azure, or Oracle Cloud.

GPUs also increase the speed of applications used in such fields as aerospace, bio-science research, mechanical and fluid simulations, and energy exploration. They have already had a significant impact on scientific discovery, including improving heart surgery, mapping human genome folds, seismic modeling, and weather simulations.

Accelerated computing is recognized as the path forward for computing amid the slowing of Moore's Law. The proportion of supercomputers utilizing accelerators has grown sharply over the past five years, now accounting for a significant proportion of both the total systems on the TOP500 list, which ranks the 500 most powerful commercially available computer systems, and the list's total floating-point operations per second. Tesla GPU accelerators power many of the world's fastest supercomputers, including the U.S. Department of Energy's new generation of supercomputers, Summit and Sierra, at Oak Ridge and Lawrence Livermore National Laboratories, Europe's fastest supercomputer - Piz Daint - in Switzerland, and Japan's fastest supercomputer, ABCI.

We also serve the datacenter market with GRID for virtualized graphics. GRID makes it possible to run graphics-intensive applications remotely on a server in the datacenter. Applications include accelerating virtual desktop infrastructures and delivering graphics-intensive applications from the cloud for industries such as manufacturing, healthcare, and educational institutions, among others.

Automotive

NVIDIA's Automotive market is comprised of cockpit infotainment solutions, AV platforms, and associated development agreements. Leveraging our technology leadership in AI and building on our long-standing automotive relationships, we are delivering a full solution for the AV market under the DRIVE brand. NVIDIA has demonstrated multiple applications of AI within the car. AI can drive the car itself as a pilot, in either partial or fully autonomous mode. AI can also be a co-pilot, assisting the human driver in creating a safer driving experience.

NVIDIA is working with several hundred partners in the automotive ecosystem including automakers, truck makers, tier-one suppliers, sensor manufacturers, automotive research institutions, HD mapping companies, and startups to develop and deploy AI systems for self-driving vehicles. Our unified AI computing architecture starts with training deep neural networks using our Tesla GPUs, and then running them within the vehicle on the NVIDIA DRIVE computing platform. The platform consists of high-performance, energy efficient hardware - DRIVE AGX, and open, modular software - including DRIVE AV for autonomous driving and DRIVE IX for in-vehicle AI assistance. In addition, we offer a scalable simulation solution, NVIDIA DRIVE Constellation, for testing and validating a self-driving platform before commercial deployment. This end-to-end, software-defined approach allows cars to receive over-the-air updates to add new features and capabilities throughout the life of a vehicle.

NVIDIA DRIVE can perceive and understand in real-time what's happening around the vehicle, precisely locate itself on an HD map, and plan a safe path forward. This advanced self-driving car platform combines deep learning, sensor fusion, and surround vision to change the driving experience. Our DRIVE platform scales from a palm-sized, energy-efficient module for automated highway-driving capabilities to a configuration with multiple systems aimed at enabling driverless cars. Our Xavier SoC, which started shipping in 2018, enables vehicles to use deep neural networks to process data from multiple cameras and sensors. It powers the DRIVE AutoPilot, the first commercially available Level 2+ automated driving system, combining the DRIVE AV self-driving solution with the DRIVE IX cockpit software, including a visualization system for allowing the driver to see what the car sees and plans to do.

Business Strategies

NVIDIA's key strategies that shape our overall business approach include:

Advancing the GPU computing platform. The massive parallel processing capabilities of NVIDIA GPUs can solve complex problems in significantly less time and with lower power consumption than alternative computational approaches. Indeed, GPUs can help solve problems that were previously deemed unsolvable. We work to deliver continued GPU performance leaps that outpace Moore's Law by leveraging innovation across the architecture, chip design, system, and software layers. Our strategy is to target markets where GPUs deliver order-of-magnitude performance advantages relative to legacy approaches. Our target markets so far include gaming, professional visualization, datacenter, and automotive. While the requirements of these end markets are diverse, we address them with a unified underlying architecture leveraging our GPUs and CUDA as the fundamental building blocks. The programmable nature of our architecture allows us to make leveraged investments in R&D: we can support several multi-billion dollar end markets with the same underlying technology

by using a variety of software stacks developed either internally or by third party developers and partners. We utilize this platform approach in each of our target markets.

Extending our technology and platform leadership in AI. We provide a complete, end-to-end GPU computing platform for deep learning and machine learning, addressing both training and inferencing. This includes GPUs, our CUDA programming language, algorithms, libraries, and system software. GPUs are uniquely suited to AI, and we will continue to add AI-specific features to our GPU architecture to further extend our leadership position. Our AI technology leadership is reinforced by our large and expanding ecosystem in a virtuous cycle. Our GPU platforms are available from virtually every major server maker and cloud service provider, as well as on our own AI supercomputer. There are over 1.2 million developers worldwide using CUDA and our other software tools to help deploy our technology in our target markets. We evangelize AI through partnerships with hundreds of universities and more than 3,600 startups through our Inception program. Additionally, our Deep Learning Institute provides instruction on the latest techniques on how to design, train, and deploy neural networks in applications using our accelerated computing platform.

Extending our technology and platform leadership in visual computing. We believe that visual computing is fundamental to the continued expansion and evolution of computing. We apply our research and development resources to extending our leadership in visual computing, enabling us to enhance the user experience for consumer entertainment and professional visualization applications. Our technologies are instrumental in driving gaming forward, as developers leverage our libraries and algorithms to create near-cinematic and VR experiences. Our close collaboration with game developers allows us to deliver an optimized gaming experience on our GeForce platform. Our GeForce Experience gaming application further enhances each gamer’s experience by optimizing their PC’s settings, as well as enabling the recording and sharing of gameplay. We also enable interactive graphics applications - such as games, movie and photo editing and design software - to be accessed by almost any device, almost anywhere, through our cloud platforms such as GRID for enterprise and GeForce NOW for gaming.

Advancing the leading autonomous vehicle platform. We believe the advent of AV will soon revolutionize the transportation industry. In our view, AI is the key technology enabler of this opportunity, as the algorithms required for autonomous driving - such as perception, localization, and planning - are too complex for legacy hand-coded approaches, and will run on multiple trained neural networks instead. Therefore, we have provided a full functionally safe AI-based hardware and software solution for the AV market under the DRIVE brand, which we are bringing to market through our partnerships with automotive original equipment manufacturers, or OEMs, tier-1 suppliers, and start-ups. Our AV solution also includes the GPU-based hardware required to train the neural networks before their in-vehicle deployment, as well as to re-simulate their operation prior to any over-the-air software updates. We believe our comprehensive, top-to-bottom and end-to-end approach will enable the transportation industry to solve the complex problems arising from the shift to autonomous driving.

Leveraging our intellectual property. We believe our intellectual property is a valuable asset that can be accessed by our customers and partners through licenses and development agreements when they desire to build such capabilities directly into their own products, or have us do so through a custom development. Such license and development arrangements can further enhance the reach of our technology.

Sales and Marketing

Our sales strategy involves working with end customers and various industry ecosystems through our partner network. Our worldwide sales and marketing strategy is key to achieving our objective of providing markets with our high-performance and efficient GPU and embedded system-on-a-chip, or SOC, platforms. Our sales and marketing teams, located across our global markets, work closely with end customers in each industry. Our partner network incorporates each industry's respective OEMs, original device manufacturers, or ODMs, system builders, add-in board manufacturers, or AIBs, retailers/distributors, internet and cloud service providers, automotive manufacturers and tier-1 automotive suppliers, mapping companies, start-ups, and other ecosystem participants.

Members of our sales team have technical expertise and product and industry knowledge. We also employ a team of application engineers to assist our partner network in designing, testing, and qualifying system designs that incorporate our platforms. We believe that the depth and quality of our design support are key to improving our partner network’s time-to-market, maintaining a high level of customer satisfaction, and fostering relationships that encourage our end customers and partner network to use the next generation of our products within each platform.

To encourage the development of applications optimized for our GPUs, we seek to establish and maintain strong relationships in the software development community. Engineering and marketing personnel engage with key software developers to promote and discuss our platforms, as well as to ascertain individual product requirements and solve technical problems. Our developer program makes our products available to developers prior to launch in order to encourage the development of AI frameworks, Software Development Kits, and Application Programming Interfaces, or APIs, for software applications and game titles that are optimized for our platforms. Our Deep Learning Institute provides in-person and online training

for developers in industries and organizations around the world to build AI and accelerated computing applications that leverage our GPU and CUDA platforms. We now have over 700 thousand registered developers across our platforms, including accelerated computing, gaming, deep learning, autonomous machines, and others.

As NVIDIA's business has evolved from a focus primarily on gaming products to broader markets, and from chips to platforms and complete systems, so, too, have our avenues to market. Thus, in addition to sales to customers in our partner network, certain of our platforms are also sold through e-tail channels, or direct to cloud service providers and enterprise customers.

Backlog

Our sales are primarily made pursuant to standard purchase orders. The quantity of products purchased by our customers as well as our shipment schedules are subject to revisions that reflect changes in both the customers' requirements and in manufacturing availability. Our industry is characterized by relatively short lead time orders and delivery schedules, thus, we believe that only a small portion of our backlog is non-cancelable and that the dollar amount associated with the non-cancelable portion is not significant.

Seasonality

Our GPU and Tegra processor platforms serve many markets from consumer PC gaming to enterprise workstations to government and cloud service provider datacenters, although a majority of our revenue stems from the consumer industry. Our consumer products have typically seen stronger revenue in the second half of our fiscal year. However, there can be no assurance that this trend will continue; for example, in fiscal year 2019 second half revenue was weaker than the first half.

Manufacturing

We do not directly manufacture semiconductors used for our products. Instead, we utilize a fabless manufacturing strategy, whereby we employ world-class suppliers for all phases of the manufacturing process, including wafer fabrication, assembly, testing, and packaging. This strategy uses the expertise of industry-leading suppliers that are certified by the International Organization for Standardization in such areas as fabrication, assembly, quality control and assurance, reliability, and testing. Additionally, we can avoid many of the significant costs and risks associated with owning and operating manufacturing operations. While we may directly procure certain raw materials used in the production of our products, such as substrates and a variety of components, our suppliers are responsible for procurement of the majority of the raw materials used in the production of our products. As a result, we can focus our resources on product design, additional quality assurance, marketing, and customer support.

We utilize industry-leading suppliers, such as Taiwan Semiconductor Manufacturing Company Limited and Samsung Electronics Co. Ltd, to produce our semiconductor wafers. We then utilize independent subcontractors, such as Advanced Semiconductor Engineering, Inc., Amkor Technology, BYD Auto Co. Ltd., Hon Hai Precision Industry Co., Ltd., JSI Logistics Ltd., King Yuan Electronics Co., Ltd., and Siliconware Precision Industries Company Ltd. to perform assembly, testing, and packaging of most of our products and platforms. We purchase substrates from IbidenCo. Ltd., Kinsus Interconnect Technology Corporation, and Unimicron Technology Corporation, and memory from Micron Technology, Samsung Semiconductor, Inc., and SK Hynix.

We typically receive semiconductor products from our subcontractors, perform incoming quality assurance and configuration, and then ship the semiconductors to contract equipment manufacturers, or CEMs, distributors, motherboard and AIB customers from our third-party warehouse in Hong Kong. Generally, these manufacturers assemble and test the boards based on our design kit and test specifications, and then ship our products to retailers, system builders, or OEMs as motherboard and AIB solutions.

We also utilize industry-leading contract manufacturers, or CMs, such as BYD and Hon Hai Precision Industry Co., and ODMs such as Quanta Computer and Wistron Corporation, to manufacture some of our products for sale directly to end customers. In those cases, key elements such as the GPU, SOC and memory are often consigned by us to the CMs, who are responsible for the procurement of other components used in the production process.

Working Capital

We focus considerable attention on managing our inventories and other working-capital-related items. We manage inventories by communicating with our customers and partners and then using our industry experience to forecast demand on a platform-by-platform basis. We then place manufacturing orders for our products that are based on forecasted demand. We generally maintain substantial inventories of our products because the semiconductor industry is characterized by short lead time orders and quick delivery schedules. A substantial amount of our inventories is maintained as semi-finished products that can be leveraged across a wide range of our processors to balance our customer demands.

Our existing cash, cash equivalents and marketable securities balances increased by 4% to $7.42 billion at the end of fiscal year 2019 compared with the end of fiscal year 2018 .

Competition

The market for our products is intensely competitive and is characterized by rapid technological change and evolving industry standards. We believe that the principal competitive factors in this market are performance, breadth of product offerings, access to customers and partners and distribution channels, software support, conformity to industry standard APIs, manufacturing capabilities, processor pricing, and total system costs. We believe that our ability to remain competitive will depend on how well we are able to anticipate the features and functions that customers and partners will demand and whether we are able to deliver consistent volumes of our products at acceptable levels of quality and at competitive prices. We expect competition to increase from both existing competitors and new market entrants with products that may be less costly than ours, or may provide better performance or additional features not provided by our products. In addition, it is possible that new competitors or alliances among competitors could emerge and acquire significant market share.

A significant source of competition comes from companies that provide or intend to provide GPUs, embedded SOCs, and accelerated and AI computing processor products. Some of our competitors may have greater marketing, financial, distribution and manufacturing resources than we do and may be more able to adapt to customer or technological changes.

Our current competitors include:

- suppliers or licensors of discrete and integrated GPUs and accelerated computing solutions, including chipsets that incorporate 3D graphics, or HPC or accelerated computing functionality as part of their solutions or platforms, such as Advanced Micro Devices, or AMD, Intel Corporation, or Intel, and Xilinx, Inc.; and
- suppliers of SOC products that are embedded into automobiles, autonomous machines, and gaming devices, such as Ambarella, Inc., AMD, Broadcom Inc., Intel, Qualcomm Incorporated, Renesas Electronics Corporation, Samsung, Texas Instruments Incorporated, and Xilinx Inc.

Patents and Proprietary Rights

We rely primarily on a combination of patents, trademarks, trade secrets, employee and third-party nondisclosure agreements, and licensing arrangements to protect our intellectual property in the United States and internationally. Our currently issued patents have expiration dates from February 2019 to February 2038. We have numerous patents issued, allowed, and pending in the United States and in foreign jurisdictions. Our patents and pending patent applications primarily relate to our products and the technology used in connection with our products. We also rely on international treaties, organizations, and foreign laws to protect our intellectual property. The laws of certain foreign countries in which our products are or may be manufactured or sold, including various countries in Asia, may not protect our products or intellectual property rights to the same extent as the laws of the United States. This decreased protection makes the possibility of piracy of our technology and products more likely. We continuously assess whether and where to seek formal protection for particular innovations and technologies based on such factors as:

- the location in which our products are manufactured;
- our strategic technology or product directions in different countries;
- the degree to which intellectual property laws exist and are meaningfully enforced in different jurisdictions; and
- the commercial significance of our operations and our competitors' operations in particular countries and regions.

We have also licensed technology from third parties for incorporation in some of our products and for defensive reasons, and expect to continue to enter into such license agreements.

Employees

As of January 27, 2019 , we had 13,277 employees, 9,486 of whom were engaged in research and development and 3,791 of whom were engaged in sales, marketing, operations, and administrative positions.

Environmental Regulatory Compliance

To date, we have not incurred significant expenses related to environmental regulatory compliance matters.

Executive Officers of the Registrant

The following sets forth certain information regarding our executive officers, their ages and positions as of February 15, 2019 :

Name	Age	Position
Jen-Hsun Huang	55	President and Chief Executive Officer
Colette M. Kress	51	Executive Vice President and Chief Financial Officer
Ajay K. Puri	64	Executive Vice President, Worldwide Field Operations
Debora Shoquist	64	Executive Vice President, Operations
Timothy S. Teter	52	Executive Vice President and General Counsel

Jen-Hsun Huang co-founded NVIDIA in 1993 and has served as our President, Chief Executive Officer and a member of the Board of Directors since our inception. From 1985 to 1993, Mr. Huang was employed at LSI Logic Corporation, a computer chip manufacturer, where he held a variety of positions including as Director of Coreware, the business unit responsible for LSI's SOC. From 1983 to 1985, Mr. Huang was a microprocessor designer for Advanced Micro Devices, Inc., a semiconductor company. Mr. Huang holds a B.S.E.E. degree from Oregon State University and an M.S.E.E. degree from Stanford University.

Colette M. Kress joined NVIDIA in 2013 as Executive Vice President and Chief Financial Officer. Prior to NVIDIA, Ms. Kress most recently served as Senior Vice President and Chief Financial Officer of the Business Technology and Operations Finance organization at Cisco Systems, Inc., a networking equipment company, since 2010. At Cisco, Ms. Kress was responsible for financial strategy, planning, reporting and business development for all business segments, engineering and operations. From 1997 to 2010 Ms. Kress held a variety of positions at Microsoft Corporation, a software company, including, beginning in 2006, Chief Financial Officer of the Server and Tools division, where Ms. Kress was responsible for financial strategy, planning, reporting and business development for the division. Prior to joining Microsoft, Ms. Kress spent eight years at Texas Instruments Incorporated, a semiconductor company, where she held a variety of finance positions. Ms. Kress holds a B.S. degree in Finance from University of Arizona and an M.B.A. degree from Southern Methodist University.

Ajay K. Puri joined NVIDIA in 2005 as Senior Vice President, Worldwide Sales and became Executive Vice President, Worldwide Field Operations in 2009. Prior to NVIDIA, he held positions in sales, marketing, and general management over a 22-year career at Sun Microsystems, Inc., a computing systems company. Mr. Puri previously held marketing, management consulting, and product development positions at Hewlett-Packard Company, an information technology company, Booz Allen Hamilton Inc., a management and technology consulting company, and Texas Instruments Incorporated. Mr. Puri holds a B.S.E.E. degree from the University of Minnesota, an M.S.E.E. degree from the California Institute of Technology and an M.B.A. degree from Harvard Business School.

Debora Shoquist joined NVIDIA in 2007 as Senior Vice President of Operations and in 2009 became Executive Vice President of Operations. Her role has since expanded with responsibility added for Facilities in 2013, and for Information Technology in 2015. Prior to NVIDIA, Ms. Shoquist served from 2004 to 2007 as Executive Vice President of Operations at JDS Uniphase Corp., a provider of communications test and measurement solutions and optical products for the telecommunications industry. She served from 2002 to 2004 as Senior Vice President and General Manager of the Electro-Optics business at Coherent, Inc., a manufacturer of commercial and scientific laser equipment. Previously, she worked at Quantum Corp., a data protection company, as President of the Personal Computer Hard Disk Drive Division, and at Hewlett-Packard Corp. Ms. Shoquist holds a B.S. degree in Electrical Engineering from Kansas State University and a B.S. degree in Biology from Santa Clara University.

Timothy S. Teter joined NVIDIA in 2017 as Senior Vice President, General Counsel and Secretary and became Executive Vice President, General Counsel and Secretary in February 2018. Prior to NVIDIA, Mr. Teter spent more than two decades at the law firm of Cooley LLP. He was most recently a partner at Cooley, where he focused on litigating patent and technology related matters. Prior to attending law school, he worked as an engineer at Lockheed Missiles and Space Company. Mr. Teter holds a B.S. degree in Mechanical Engineering from the University of California at Davis and a J.D. degree from Stanford Law School.

Available Information

Our annual reports on Form 10-K, quarterly reports on Form 10-Q, current reports on Form 8-K and, if applicable, amendments to those reports filed or furnished pursuant to Section 13(a) of the Securities Exchange Act of 1934, as amended, are available free of charge on or through our web site, *http://www.nvidia.com* , as soon as reasonably practicable after we electronically file such material with, or furnish it to, the Securities and Exchange Commission, or the SEC. The SEC's website, *http://www.sec.gov* , contains reports, proxy and information statements, and other information regarding

issuers that file electronically with the SEC. Our web site and the information on it or connected to it are not a part of this Annual Report on Form 10-K.

ITEM 1A. RISK FACTORS

In evaluating NVIDIA and our business, the following factors should be considered in addition to the other information in this Annual Report on Form 10-K. Before you buy our common stock, you should know that making such an investment involves risks including, but not limited to, the risks described below. Any one of the following risks could harm our business, financial condition, results of operations or reputation, which could cause our stock price to decline, and you may lose all or a part of your investment. Additional risks, trends and uncertainties not presently known to us or that we currently believe are immaterial may also harm our business, financial condition, results of operations or reputation.

Risks Related to Our Business, Industry and Partners

If we fail to meet the evolving needs of our markets, or identify new products, services or technologies, our revenue and financial results may be adversely impacted.

We have created GPU-based visual and accelerated computing platforms that address four large markets: Gaming, Professional Visualization, Datacenter, and Automotive. These markets often experience rapid technological change, changes in customer requirements, new product introductions and enhancements, and evolving industry standards. Our success depends on our ability to identify these emerging industry changes and to develop new (or enhance our existing) products, services and technologies that meet the evolving needs of these markets. Such activities may require considerable technical, financial, compliance, sales and marketing investments. We currently devote significant resources to the development of technologies and business offerings in markets where we have a limited operating history, such as the automotive and datacenter markets, which presents additional risks to our business. We must also continue to develop the infrastructure needed to appropriately scale our business in these areas, including customer service and customer support. We also must meet customer safety and compliance standards, which are subject to change. Additionally, we continue to make considerable investments in research and development, which may not produce significant revenue for several years, if at all. If our investments are unsuccessful and we fail to develop new products, services and technologies, or if we focus on technologies that do not become widely adopted, our business, revenue, financial condition and results of operations could be adversely affected. We cannot assure you that our strategic direction will result in innovative products and technologies that provide value to our customers, partners and ultimately, our shareholders. If we fail to anticipate the changing needs of our target markets and emerging technology trends, or if we do not appropriately adapt that strategy as market conditions evolve, in a timely manner to exploit potential market opportunities, our business will be harmed.

Competition in our current and target markets could prevent us from growing our revenue.

Our target markets remain extremely competitive, and we expect competition to intensify as current competitors expand their product and/or service offerings, industry standards continue to evolve, customer needs change and new competitors enter these markets. Our competitors' products, services and technologies may be less costly, or may offer superior functionality or better features, than ours, which may result, among other things, in lower than expected selling prices for our products. In addition, some of our competitors operate and maintain their own fabrication facilities, have longer operating histories, larger customer bases, more comprehensive intellectual property, or IP, portfolios and patent protections, and greater financial, sales, marketing and distribution resources than we do. These competitors may be able to more effectively identify and capitalize upon opportunities in new markets and end user customer trends, quickly transition their products, including semiconductor products, to increasingly smaller line width geometries, and obtain sufficient foundry capacity and packaging materials, which could harm our business. If we are unable to successfully compete in our target markets, respond to changes in our target markets or introduce new offerings to meet the needs of this competitive environment, including in significant international markets such as China, demand for our products, services and technologies could decrease, which would cause our revenue to decline and cause our results of operations to suffer. In addition, the competitive landscape in our target markets has changed and may continue to evolve due to a trend toward consolidation, which could lead to fewer customers, partners, or suppliers, any of which could negatively affect our financial results.

System security and data protection breaches, as well as cyber-attacks, could disrupt our operations, reduce our expected revenue and increase our expenses, which could adversely affect our stock price and damage our reputation.

Security breaches, computer malware and cyber-attacks have become more prevalent and sophisticated in recent years. These threats are constantly evolving, making it increasingly difficult to successfully defend against them or implement adequate preventative measures. These attacks have occurred on our systems in the past and are expected to occur in the future. Experienced computer programmers, hackers and employees may penetrate our security controls and misappropriate or compromise our confidential information, or that of our employees or third parties. These attacks may

create system disruptions or cause shutdowns. These hackers may also develop and deploy viruses, worms and other malicious software programs that attack or otherwise exploit security vulnerabilities in our products, including consumer and automotive products, where we utilize over-the-air updates to improve functionality over time. For portions of our IT infrastructure, including business management and communication software products, we rely on products and services provided by third parties. These providers may also experience breaches and attacks to their products which may impact our systems. Data security breaches may also result from non-technical means, such as actions by an employee with access to our systems. To defend against security threats, both to our internal systems and those of our customers, we must continuously engineer more secure products and enhance security and reliability features, which may result in increased expenses.

Actual or perceived breaches of our security measures or the accidental loss, inadvertent disclosure or unapproved dissemination of proprietary information or sensitive or confidential data about us, our partners, our customers or third parties could expose us and the parties affected to a risk of loss or misuse of this information, resulting in litigation and potential liability, paying damages, regulatory inquiries or actions, damage to our brand and reputation or other harm to our business. Our efforts to prevent and overcome these challenges could increase our expenses and may not be successful. We may experience interruptions, delays, cessation of service and loss of existing or potential customers. Such disruptions could adversely impact our ability to fulfill orders and interrupt other critical functions. Delayed sales, lower margins or lost customers as a result of these disruptions could adversely affect our financial results, stock price and reputation.

If our products contain significant defects, we could incur significant expenses to remediate such defects, our reputation could be damaged, and we could lose market share.

Our products are complex and may contain defects or security vulnerabilities, or experience failures or unsatisfactory performance due to any number of issues in design, fabrication, packaging, materials and/or use within a system. These risks may increase as our products are introduced into new devices, markets, technologies and applications, including into the automotive market, or as new versions are released. Some errors in our products or services may only be discovered after a product or service has been shipped or used by customers or the end users of such product. Undiscovered vulnerabilities in our products or services could expose our customers or end users to hackers or other unscrupulous third parties who develop and deploy viruses, worms and other malicious software programs that could attack our products or services. Failure of our products to perform to specifications, or other product defects, could lead to substantial damage to the products we sell directly to customers, the end product in which our device has been integrated by OEMs, ODMs, AIBs and Tier 1 automotive suppliers, and to the user of such end product. Any such defect may cause us to incur significant warranty, support and repair or replacement costs, write off the value of related inventory, cause us to lose market share, and divert the attention of our engineering personnel from our product development efforts to find and correct the issue. In addition, an error or defect in new products or releases or related software drivers after commencement of commercial shipments could result in failure to achieve market acceptance or loss of design wins, harm our relationships with customers and partners and harm consumers' perceptions of our brand. Also, we may be required to reimburse our customers, partners or consumers, including costs to repair or replace products in the field. A product recall, including automotive recalls or a recall due to a bug in our products, or a significant number of product returns could be expensive, damage our reputation, harm our ability to attract new customers, result in the shifting of business to our competitors and result in litigation against us, such as product liability suits. If a product liability claim is brought against us, the cost of defending the claim could be significant and would divert the efforts of our technical and management personnel, and harm our business. Further, our business liability insurance may be inadequate or future coverage may be unavailable on acceptable terms, which could adversely impact our financial results.

We depend on third parties and their technology to manufacture, assemble, test and/or package our products, which reduces our control over product quantity and quality, manufacturing yields, development, enhancement and product delivery schedule and could harm our business.

We do not manufacture the silicon wafers used for our GPUs and Tegra processors and do not own or operate a wafer fabrication facility. Instead, we are dependent on industry-leading foundries, such as Taiwan Semiconductor Manufacturing Company Limited and Samsung Electronics Co. Ltd., to manufacture our semiconductor wafers using their fabrication equipment and techniques. Similarly, we do not directly assemble, test or package our products, but instead rely on independent subcontractors. We do not have long-term commitment contracts with these foundries or subcontractors. As a result, we face several significant risks which could have an adverse effect on our ability to meet customer demand and/or negatively impact our business operations, gross margin, revenue and/or financial results, including:

- a lack of guaranteed supply of wafers and other components and potential higher wafer and component prices due to supply constraints;
- a failure by our foundries to procure raw materials or to provide or allocate adequate or any manufacturing or test capacity for our products;

- a failure to develop, obtain or successfully implement high quality, leading-edge process technologies, including transitions to smaller geometry process technologies such as advanced process node technologies and memory designs needed to manufacture our products profitably or on a timely basis;
- loss of a supplier and additional expense and/or production delays as a result of qualifying a new foundry or subcontractor and commencing volume production or testing in the event of a loss of or a decision to add or change a supplier;
- a lack of direct control over delivery schedules or product quantity and quality; and
- delays in product shipments, shortages, a decrease in product quality and/or higher expenses in the event our subcontractors or foundries prioritize our competitors' orders over our orders or otherwise.

In addition, low manufacturing yields could have an adverse effect on our ability to meet customer demand, increase manufacturing costs, harm customer or partner relationships, and/or negatively impact our business operations, gross margin, revenue and/or financial results. Manufacturing yields for our products are a function of product design, which is developed largely by us, and process technology, which typically is proprietary to the foundry. Low yields may result from either product design or process technology failure. We do not know whether a yield problem will exist until our design is actually manufactured by the foundry. As a result, yield problems may not be identified until well into the manufacturing process and require us and the foundry to cooperate to resolve the problem.

We also rely on third-party software development tools to assist us in the design, simulation and verification of new products or product enhancements, and to bring such new products and enhancements to market in a timely manner. In the past, we have experienced delays in the introduction of products and enhancements as a result of the inability of then available software development tools to fully simulate the complex features and functionalities of our products. The design requirements necessary to meet consumer demands for more features and greater functionality from our products may exceed the capabilities of available software development tools. If we miss design cycles or lose design wins due to the unavailability of such software development tools, we could lose market share and our revenues could decline. If we fail to achieve design wins for our products, our business will be harmed.

For our products that we do not sell directly to consumers, achieving design wins is an important success factor. Achieving design wins may involve a lengthy process in pursuit of a customer opportunity and depend on our ability to anticipate features and functionality that customers and consumers will demand. Failure to obtain a particular design win may prevent us from obtaining design wins in subsequent generations of a particular product. This could result in lost revenue and could weaken our position in future competitive bid selection processes.

Unanticipated changes in industry standards could render our products incompatible with products developed by major hardware manufacturers and software developers. Further, if our products are not in compliance with prevailing industry standards, including safety standards, our customers may not incorporate our products into their design strategies. Winning a product design does not guarantee sales to a customer or that we will realize as much revenue as anticipated, if any.

Business disruptions could harm our business, lead to a decline in revenues and increase our costs.

Our worldwide operations could be disrupted by earthquakes, telecommunications failures, power or water shortages, outages at cloud service providers, tsunamis, floods, hurricanes, typhoons, fires, extreme weather conditions, cyber-attacks, terrorist attacks, medical epidemics or pandemics and other natural or man-made disasters, catastrophic events or climate change. The occurrence of any of these disruptions could harm our business and result in significant losses, a decline in revenue and an increase in our costs and expenses. Any of these business disruptions could require substantial expenditures and recovery time in order to fully resume operations. Our corporate headquarters, and a portion of our research and development activities, are located in California, and other critical business operations, finished goods inventory, and some of our suppliers are located in Asia, near major earthquake faults known for seismic activity. In addition, a large portion of our current datacenter capacity is located in California, making our operations vulnerable to natural disasters or other business disruptions occurring in these geographical areas. The manufacture of product components, the final assembly of our products and other critical operations are concentrated in certain geographic locations, including Taiwan, China, and Korea. Geopolitical change or changes in government regulations and policies in the United States or abroad also may result in changing regulatory requirements, trade policies, import duties and economic disruptions that could impact our operating strategies, product demand, access to global markets, hiring, and profitability. In particular, revisions to laws or regulations or their interpretation and enforcement could result in increased taxation, trade sanctions, the imposition of import duties or tariffs, restrictions and controls on imports or exports, or other retaliatory actions, which could have an adverse effect on our business plans. For example, regulations to implement the Export Control Reform Act of 2018 could have an adverse effect on our business plans. Catastrophic events can also have an impact on third-party vendors who provide us critical infrastructure services for IT and research and development systems and personnel. Our operations

could be harmed if manufacturing, logistics or other operations in these locations are disrupted for any reason, including natural disasters, high heat events or water shortages, information technology system failures, military actions or economic, business, labor, environmental, public health, regulatory or political issues. The ultimate impact on us, our third-party foundries and other suppliers and our general infrastructure of being located near major earthquake faults and being consolidated in certain geographical areas is unknown. In the event a major earthquake or other disaster or catastrophic event affects us or the third-party systems on which we rely, our business could be harmed as a result of declines in revenue, increases in expenses, substantial expenditures and time spent to fully resume operations.

If we fail to estimate customer demand properly, our financial results could be harmed.

We manufacture our GPUs and Tegra processors based on estimates of customer demand and requirements. We sell many of our products through a channel model, and our channel customers sell to retailers, distributors, and/or end customers. As a result, the decisions made by our channel partners, retailers, and distributors in response to changing market conditions and the changing demand for our products could impact our financial results. In order to have shorter shipment lead times and quicker delivery schedules for our customers, we may build inventories for anticipated periods of growth which do not occur, may build inventory anticipating demand that does not materialize, or may build inventory to serve what we believe is pent-up demand. Such decisions may and have resulted in prolonged channel sell-through, as we experienced with our mid-range gaming GPUs in fiscal year 2019. In estimating demand, we make multiple assumptions, any of which may prove to be incorrect. Situations that may result in excess or obsolete inventory include:

- changes in business and economic conditions, including downturns in our target markets and/or overall economy;
- changes in consumer confidence caused by changes in market conditions, including changes in the credit market;
- a sudden and significant decrease in demand for our products;
- a higher incidence of inventory obsolescence because of rapidly changing technology or customer requirements;
- our introduction of new products resulting in lower demand for older products;
- less demand than expected for newly-introduced products; or
- increased competition, including competitive pricing actions.

The cancellation or deferral of customer purchase orders could result in our holding excess inventory, which could adversely affect our gross margins. In addition, because we often sell a substantial portion of our products in the last month of each quarter, we may not be able to reduce our inventory purchase commitments in a timely manner in response to customer cancellations or deferrals. We could be required to write-down our inventory to the lower of cost or market or write-off excess inventory, and we could experience a reduction in average selling prices if we incorrectly forecast product demand, any of which could harm our financial results.

Conversely, if we underestimate our customers' demand for our products, our foundry partners may not have adequate lead-time or capacity to increase production and we may not be able to obtain sufficient inventory to fill customers' orders on a timely basis. We may also face supply constraints caused by natural disasters or other events. In such cases, even if we are able to increase production levels to meet customer demand, we may not be able to do so in a cost-effective or timely manner. If we fail to fulfill our customers' orders on a timely basis, or at all, our customer relationships could be damaged, we could lose revenue and market share and our reputation could be damaged.

We are subject to risks and uncertainties associated with international operations, which may harm our business.

We conduct our business worldwide and we have offices in various countries outside of the United States. Our semiconductor wafers are manufactured, assembled, tested and packaged by third parties located outside of the United States. We also generate a significant portion of our revenue from sales outside the United States. We allocate revenue to individual countries based on the location to which the products are initially billed even if our customers' revenue is attributable to end customers that are located in a different location. Revenue from sales outside of the United States accounted for 87% of total revenue for each of fiscal years 2019 , 2018 , and 2017 . Revenue from billings to China, including Hong Kong, was 24% of our revenue for fiscal year 2019, even if our customers' revenue is attributable to end customers that are located in a different location. Additionally, as of January 27, 2019 , approximately 46% of our employees were located outside of the United States. The global nature of our business subjects us to a number of risks and uncertainties, which could have a material adverse effect on our business, financial condition and results of operations, including:

- international economic and political conditions, including as a result of the United Kingdom's vote to withdraw from the European Union, and other political tensions between countries in which we do business;
- unexpected changes in, or impositions of, legislative or regulatory requirements, including changes in tax laws;
- differing legal standards with respect to protection of intellectual property and employment practices;

- local business and cultural factors that differ from our normal standards and practices, including business practices that we are prohibited from engaging in by the Foreign Corrupt Practices Act and other anticorruption laws and regulations;
- exporting or importing issues related to export or import restrictions, including deemed export restrictions, tariffs, quotas and other trade barriers and restrictions;
- disruptions of capital and trading markets and currency fluctuations; and
- increased costs due to imposition of climate change regulations, such as carbon taxes, fuel or energy taxes, and pollution limits.

If our sales outside of the United States are delayed or cancelled because of any of the above factors, our revenue may be negatively impacted.

If we are unable to attract, retain and motivate our executives and key employees, we may not be able to execute our business strategy effectively.

To be competitive and execute our business strategy successfully, we must attract, retain and motivate our executives and key employees. The market for highly skilled workers and leaders in our industry is extremely competitive. In particular, hiring qualified executives, scientists, engineers, technical staff and research and development personnel is critical to our business. Additionally, changes in immigration and work permit laws and regulations or the administration or interpretation of such laws or regulations could impair our ability to attract and retain highly qualified employees. If we are less successful in our recruiting efforts, or if we cannot retain key employees, our ability to develop and deliver successful products and services may be adversely affected. Additionally, competition for personnel results in increased costs in the form of cash and stock-based compensation. The interpretation and application of employment related laws to our workforce practices may result in increased operating costs and less flexibility in how we meet our workforce needs. Effective succession planning is also important to our long-term success. Failure to ensure effective transfer of knowledge and smooth transitions involving key employees could hinder our strategic planning and execution.

We may not be able to realize the potential financial or strategic benefits of business acquisitions or strategic investments and we may not be able to successfully integrate acquisition targets, which could hurt our ability to grow our business, develop new products or sell our products.

We have in the past acquired and invested in, and may continue to acquire and invest in, other businesses that offer products, services and technologies that we believe will help expand or enhance our existing products, strategic objectives and business. The risks associated with past or future acquisitions or investments could impair our ability to grow our business, develop new products or sell our products, and ultimately could have a negative impact on our growth or our financial results. Given that our resources are limited, our decision to pursue a transaction has opportunity costs; accordingly, if we pursue a particular transaction, we may need to forgo the prospect of entering into other transactions that could help us achieve our strategic objectives. Additional risks related to acquisitions or strategic investments include, but are not limited to:

- difficulty in combining the technology, products, operations or workforce of the acquired business with our business;
- diversion of capital and other resources, including management's attention;
- assumption of liabilities and incurring amortization expenses, impairment charges to goodwill or write-downs of acquired assets;
- difficulty in realizing a satisfactory return, if at all;
- difficulty in obtaining regulatory, other approvals or financing;
- failure and costs associated with the failure to consummate a proposed acquisition or other strategic investment;
- legal proceedings initiated as a result of an acquisition or investment;
- uncertainties and time needed to realize the benefits of an acquisition or strategic investment, if at all;
- the need to later divest acquired assets if an acquisition does not meet our expectations;
- potential failure of our due diligence processes to identify significant issues with the acquired assets or company; and
- impairment of relationships with, or loss of our or our target's, employees, vendors and customers, as a result of our acquisition or investment.

Risks Related to Regulatory, Legal, Our Common Stock and Other Matters

Actions to adequately protect our IP rights could result in substantial costs to us and our ability to compete could be harmed if we are unsuccessful in doing so or if we are prohibited from making or selling our products.

We have in the past, currently are, and may in the future become involved in lawsuits or other legal proceedings alleging patent infringement or other intellectual property rights violations by us, our employees or parties that we have agreed to indemnify for certain claims of infringement. An unfavorable ruling in any such intellectual property related litigation could include significant damages, invalidation of a patent or family of patents, indemnification of customers, payment of lost profits, or, when it has been sought, injunctive relief. Claims that our products or processes infringe the IP rights of others, regardless of their merit, could cause us to incur significant costs to respond to, defend, and resolve such claims, and they may also divert the efforts and attention of management and technical personnel.

We may commence litigation or other legal proceedings in order to protect our intellectual property rights. Such proceedings may increase our operating expenses, which could negatively impact our operating results. Further, we could be subject to countersuits as a result of our initiation of litigation. If infringement claims are made against us or our products are found to infringe a third party's patent or intellectual property, we or one of our indemnitees may have to seek a license to the third party's patent or other intellectual property rights. However, we may not be able to obtain licenses at all or on terms acceptable to us particularly from our competitors. If we or one of our indemnitees is unable to obtain a license from a third party for technology that we use or that is used in one of our products, we could be subject to substantial liabilities or have to suspend or discontinue the manufacture and sale of one or more of our products. We may also have to make royalty or other payments, or cross license our technology. If these arrangements are not concluded on commercially reasonable terms, our business could be negatively impacted. Furthermore, the indemnification of a customer or other indemnitee may increase our operating expenses which could negatively impact our operating results.

Our success depends in part on protecting our intellectual property. To accomplish this, we rely primarily on a combination of patents, trademarks, trade secrets, employee and third-party nondisclosure agreements, licensing arrangements, and the laws of the countries in which we operate to protect our intellectual property in the United States and internationally. We may be required to spend significant resources to monitor and protect our intellectual property rights, and even with significant expenditures we may not be able to protect our intellectual property rights that are valuable to our business. The laws of certain foreign countries may not protect our products or intellectual property rights to the same extent as the laws of the United States. This makes the possibility of piracy of our technology and products more likely. In addition, the theft or unauthorized use or publication of our trade secrets and other confidential business information could harm our competitive position and reduce acceptance of our products; as a result, the value of our investment in research and development, product development, and marketing could be reduced. We continuously assess whether and where to seek formal protection for existing and new innovations and technologies, but cannot be certain whether our applications for such protections will be approved, and, if approved, whether we will be able to enforce such protections.

Our operating results have in the past fluctuated and may in the future fluctuate, and if our operating results are below the expectations of securities analysts or investors, our stock price could decline.

Our operating results have in the past fluctuated and may in the future continue to fluctuate due to numerous factors. Therefore, investors should not rely on quarterly comparisons of our results of operations as an indication of our future performance.

Factors, other than those described elsewhere in these risk factors, that could affect our results of operations in the future include, but are not limited to:

- our ability to achieve volume production of our next-generation products;
- our inability to adjust spending to offset revenue shortfalls due to the multi-year development cycle for some of our products and services;
- fluctuations in the demand for our products related to cryptocurrencies;
- changes in the timing of product orders due to unexpected delays in the introduction of our partners' products;
- our ability to cover the manufacturing and design costs of our products through competitive pricing;
- our ability to comply and continue to comply with our customers' contractual obligations;
- product rates of return in excess of that forecasted or expected due to quality issues;
- our ability to secure appropriate safety certifications and meet industry safety standards;
- supply constraints for and changes in the cost of the other components incorporated into our products
- inventory write-downs;

- our ability to continue generating revenue from our partner network, including by generating sales within our partner network and ensuring our products are incorporated into our partners product ecosystems, and our partner network's ability to sell products that incorporate our GPUs and Tegra processors;
- the inability of certain of our customers to make required payments to us, and our ability to obtain credit insurance over the purchasing credit extended to these customers;
- customer bad debt write-offs;
- any unanticipated costs associated with environmental liabilities;
- unexpected costs related to our ownership of real property;
- changes in financial accounting standards or interpretations of existing standards; and
- general macroeconomic or industry events and factors affecting the overall market and our target markets.

Any one or more of the factors discussed above could prevent us from achieving our expected future financial results. Any such failure to meet our expectations or the expectations of our investors or security analysts could cause our stock price to decline or experience substantial price volatility.

Privacy concerns relating to our products and services could damage our reputation, deter current and potential users from using our products and services, result in liability, or result in legal or regulatory proceedings.

Our products and services may provide us with access to sensitive, confidential or personal data or information that is subject to privacy and security laws and regulations. Concerns about our practices with regard to the collection, use, retention, security or disclosure of personal information or other privacy-related matters, even if unfounded, could damage our reputation and adversely affect our operating results. The theft, loss, or misuse of personal data collected, used, stored, or transferred by us to run our business or by one of our partners could result in significantly increased security costs, damage to our reputation, regulatory proceedings, disruption of our business activities or increased costs related to defending legal claims.

Worldwide regulatory authorities are considering and have approved various legislative proposals concerning data protection, which continue to evolve and apply to our business. For example, the European Union adopted the General Data Protection Regulation, or GDPR, which requires companies to meet new requirements beginning in May 2018 regarding the handling of personal data, including its use, protection and the ability of persons whose data is stored to correct or delete such data about themselves. Failure to meet GDPR requirements could result in penalties of up to 4% of worldwide revenue. In addition, the interpretation and application of consumer and data protection laws in the United States, Europe and elsewhere are often uncertain and fluid, and may be interpreted and applied in a manner that is inconsistent with our data practices. If so, we may be ordered to change our data practices and/or be fined. Complying with these changing laws has caused, and could continue to cause, us to incur substantial costs, which could have an adverse effect on our business and results of operations. Further, failure to comply with existing or new rules may result in significant penalties or orders to stop the alleged noncompliant activity.

We may have exposure to additional tax liabilities and our operating results may be adversely impacted by higher than expected tax rates.

As a multinational corporation, we are subject to income taxes as well as non-income based taxes, such as payroll, sales, use, value-added, net worth, property and goods and services taxes, in both the United States and various foreign jurisdictions. Our domestic and international tax liabilities are subject to the allocation of revenue and expenses in different jurisdictions. Significant judgment is required in determining our worldwide provision for income taxes and other tax liabilities. Further, changes in United States federal, and state or international tax laws applicable to multinational corporations or other fundamental law changes may materially impact our tax expense and cash flows, as we experienced in fiscal year 2018 with the passage of the Tax Cuts and Jobs Act, or TCJA.

Our future effective tax rate may be affected by such factors as changes in tax laws, changes in our business or statutory rates, changes in jurisdictions in which our profits are determined to be earned and taxed, changes in available tax credits, the resolution of issues arising from tax audits, changes in United States generally accepted accounting principles, adjustments to income taxes upon finalization of tax returns, increases in expenses not deductible for tax purposes, changes in the valuation of our deferred tax assets and liabilities and in deferred tax valuation allowances, changing interpretation of existing laws or regulations, the impact of accounting for stock-based compensation and the recognition of excess tax benefits and tax deficiencies within the income tax provision in the period in which they occur, the impact of accounting for business combinations, shifts in the amount of earnings in the United States compared with other regions in the world and overall levels of income before tax, changes in our international organization, as well as the expiration of statute of limitations and settlements of audits. Any changes in our effective tax rate may reduce our net income.

Our business is exposed to the risks associated with litigation, investigations and regulatory proceedings.

We currently and may in the future face legal, administrative and regulatory proceedings, claims, demands and/or investigations involving shareholder, consumer, competition and/or other issues relating to our business on a global basis. For example, multiple securities litigation claims have recently been filed against us and certain of our officers based on the dissemination of allegedly false and misleading statements related to channel inventory and the impact of cryptocurrency mining on GPU demand. In addition, a stockholder, purporting to act on behalf of the Company, filed a derivative lawsuit seeking to assert claims on behalf of the Company against the members of our board of directors and certain officers based on the dissemination of allegedly false and misleading statements related to channel inventory and the impact of cryptocurrency mining on GPU demand.

Litigation and regulatory proceedings are inherently uncertain, and adverse rulings could occur, including monetary damages, or an injunction stopping us from manufacturing or selling certain products, engaging in certain business practices, or requiring other remedies, such as compulsory licensing of patents. An unfavorable outcome or settlement may result in a material adverse impact on our business, results of operations, financial position, and overall trends. In addition, regardless of the outcome, litigation can be costly, time-consuming, and disruptive to our operations.

In addition, the laws and regulations our business is subject to are complex, and change frequently. We may be required to incur significant expense to comply with, or remedy violations of, these regulations.

Delaware law and provisions in our certificate of incorporation, our bylaws and our agreement with Microsoft Corporation could delay or prevent a change in control.

Our status as a Delaware corporation and the anti-takeover provisions of the Delaware General Corporation Law may discourage, delay, or prevent a change in control by prohibiting us from engaging in a business combination with an interested shareholder for a period of three years after the person becomes an interested shareholder, even if a change of control would be beneficial to our existing shareholders. In addition, our certificate of incorporation and bylaws contain provisions that could make it more difficult for a third party to acquire a majority of our outstanding voting stock. These provisions include the following:

- the ability of our Board of Directors to create and issue preferred stock without prior shareholder approval;
- the prohibition of shareholder action by written consent;
- advance notice requirements for director nominations and shareholder proposals;
- the ability of our Board of Directors to increase or decrease the number of directors without shareholder approval;
- a super-majority voting requirement to amend some provisions in our certificate of incorporation and bylaws;
- the inability of our shareholders to call special meetings of shareholders; and
- the ability of our Board of Directors to make, amend or repeal our bylaws.

On March 5, 2000, we entered into an agreement with Microsoft in which we agreed to develop and sell graphics chips and to license certain technology to Microsoft and its licensees for use in the Xbox. Under the agreement, if an individual or corporation makes an offer to purchase shares equal to or greater than 30% of the outstanding shares of our common stock, Microsoft may have first and last rights of refusal to purchase the stock. The Microsoft provision and the other factors listed above could also delay or prevent a change in control of NVIDIA. These provisions could also discourage proxy contests and make it more difficult for shareholders to elect directors of their choosing and to cause us to take other corporate actions they desire.

ITEM 1B. UNRESOLVED STAFF COMMENTS

Not applicable.

ITEM 2. PROPERTIES

Our headquarters complex is located in Santa Clara, California. It includes ten leased commercial buildings totaling 981,389 square feet, and real property that we own totaling 1,257,346 square feet. Our owned property consists of two commercial buildings on 36 acres of land. In addition, we also lease datacenter space in Santa Clara, California.

Outside of Santa Clara, California, we lease facilities in Austin, Texas and a number of regional facilities in other U.S. locations that are used as research and development centers and/or sales and administrative offices. Outside of the United States, we own a building in Hyderabad, India, that is being used primarily as a research and development center. We also lease facilities in various international locations that are used as research and development centers and/or sales and

administrative offices. These leased facilities are located primarily in Asia and Europe. In addition, we also lease datacenter space in various locations around the world.

We believe that we currently have sufficient facilities to conduct our operations for the next twelve months. For additional information regarding obligations under leases, refer to Note 12 of the Notes to the Consolidated Financial Statements in Part IV, Item 15 of this Annual Report on Form 10-K under the subheading “Lease Obligations,” which information is hereby incorporated by reference.

ITEM 3. LEGAL PROCEEDINGS

Please see Note 12 of the Notes to the Consolidated Financial Statements in Part IV, Item 15 of this Annual Report on Form 10-K for a discussion of our legal proceedings.

ITEM 4. MINE SAFETY DISCLOSURES

Not Applicable.

PART II

ITEM 5. MARKET FOR REGISTRANT’S COMMON EQUITY, RELATED STOCKHOLDER MATTERS AND ISSUER PURCHASES OF EQUITY SECURITIES

Our common stock is traded on the Nasdaq Global Select Market under the symbol NVDA. Public trading of our common stock began on January 22, 1999. Prior to that, there was no public market for our common stock. As of February 15, 2019 , we had approximately 317 registered shareholders, not including those shares held in street or nominee name.

Issuer Purchases of Equity Securities

Beginning August 2004, our Board of Directors authorized us to repurchase our stock.

Since the inception of our share repurchase program, we have repurchased an aggregate of 260 million shares for a total cost of $7.08 billion through January 27, 2019 . All shares delivered from these repurchases have been placed into treasury stock.

In November 2018, the Board authorized an additional $7.00 billion under our share repurchase program and extended it through the end of December 2022. As of January 27, 2019 , we were authorized to repurchase additional shares of our common stock up to $7.24 billion .

We intend to return $3.00 billion to shareholders by the end of fiscal year 2020, including $700 million of share repurchases we made in the fourth quarter of fiscal year 2019.

The repurchases can be made in the open market, in privately negotiated transactions, or in structured share repurchase programs, and can be made in one or more larger repurchases, in compliance with Rule 10b-18 of the Securities Exchange Act of 1934, as amended, subject to market conditions, applicable legal requirements, and other factors. The program does not obligate NVIDIA to acquire any particular amount of common stock and the program may be suspended at any time at our discretion.

The following table presents details of our share repurchase transactions during the fourth quarter of fiscal year 2019:

Period	Total Number of Shares Purchased (In thousands)	Average Price Paid per Share	Total Number of Shares Purchased as Part of Publicly Announced Program (In thousands)	Approximate Dollar Value of Shares that May Yet Be Purchased Under the Program (In billions)
October 29, 2018 - November 25, 2018	123	$ 195.72	123	$ 7.94
November 26, 2018 - December 23, 2018	3,304	$ 142.05	3,304	$ 7.47
December 24, 2018 - January 27, 2019	1,777	$ 129 87	1,777	$ 7 24
Total	**5,204**		**5,204**	

Transactions Related to our 1.00% Convertible Senior Notes Due 2018 and Note Hedges

During fiscal year 2019 , we issued an aggregate of 714 thousand shares of our common stock upon settlement of $16 million in principal amount of 1.00% Convertible Senior Notes Due 2018, or the Convertible Notes, submitted for conversion. In connection with these conversions, we exercised a portion of our Note Hedges to acquire an equal number of shares of our common stock. The counterparty to the Note Hedges may be deemed an “affiliated purchaser” and may have purchased the shares of our common stock deliverable to us upon this exercise of our option. Refer to Note 11 of the Notes to the Consolidated Financial Statements in Part IV, Item 15 of this Annual Report on Form 10-K for further discussion regarding the Convertible Notes and the Note Hedges.

Restricted Stock Unit Share Withholding

We also withhold common stock shares associated with net share settlements to cover tax withholding obligations upon the vesting of restricted stock unit awards under our employee equity incentive program. During fiscal year 2019 , we withheld approximately 4 million shares at a total cost of $1.03 billion through net share settlements. Refer to Note 3 of the Notes to the Consolidated Financial Statements in Part IV, Item 15 of this Annual Report on Form 10-K for further discussion regarding our equity incentive plans.

Stock Performance Graphs

The following graph compares the cumulative total shareholder return for our common stock, the S&P 500 Index, and the Nasdaq 100 Index for the five years ended January 27, 2019 . The graph assumes that $100 was invested on January 26, 2014 in our common stock and in each of the S&P 500 Index and the Nasdaq 100 Index. Our common stock is a component of each of the presented indices. Total return assumes reinvestment of dividends in each of the indices indicated. Total return is based on historical results and is not intended to indicate future performance.

*$100 invested on 1/26/14 in stock and in indices, including reinvestment of dividends.

	1/26/2014	1/25/2015	1/31/2016	1/29/2017	1/28/2018	1/27/2019
NVIDIA Corporation	$ 100.00	$ 135.49	$ 194.78	$ 750.36	$ 1,639.87	$ 1,082.30
S&P 500	$ 100.00	$ 111.92	$ 108.84	$ 127.84	$ 158.41	$ 151.70
Nasdaq 100	$ 100.00	$ 119.26	$ 124.52	$ 150.83	$ 207.18	$ 208.13

ITEM 6. SELECTED FINANCIAL DATA

The following selected financial data should be read in conjunction with our financial statements and the notes thereto, and with Item 7, "Management's Discussion and Analysis of Financial Condition and Results of Operations." The Consolidated Statements of Income data for fiscal years 2019 , 2018 , and 2017 and the Consolidated Balance Sheets data as of January 27, 2019 and January 28, 2018 have been derived from and should be read in conjunction with our audited consolidated financial statements and the notes thereto included in Part IV, Item 15 in this Annual Report on Form 10-K. We operate on a 52- or 53-week year, ending on the last Sunday in January. Fiscal years 2019, 2018, 2017, and 2015 were 52-week years and fiscal year 2016 was a 53-week year.

	Year Ended				
	January 27, 2019	January 28, 2018	January 29, 2017	January 31, 2016 (A)	January 25, 2015
Consolidated Statements of Income Data:	*(In millions, except per share data)*				
Revenue	$ 11,716	$ 9,714	$ 6,910	$ 5,010	$ 4,682
Income from operations	$ 3,804	$ 3,210	$ 1,934	$ 747	$ 759
Net income	$ 4,141	$ 3,047	$ 1,666	$ 614	$ 631
Net income per share:					
Basic	$ 6.81	$ 5.09	$ 3.08	$ 1.13	$ 1.14
Diluted	$ 6.63	$ 4.82	$ 2.57	$ 1.08	$ 1.12
Weighted average shares used in per share computation:					
Basic	608	599	541	543	552
Diluted	625	632	649	569	563

	Year Ended				
	January 27, 2019 (B,C)	January 28, 2018 (B,C)	January 29, 2017 (B,C)	January 31, 2016 (B)	January 25, 2015
Consolidated Balance Sheets Data:	*(In millions, except per share data)*				
Cash, cash equivalents and marketable securities	$ 7,422	$ 7,108	$ 6,798	$ 5,037	$ 4,623
Total assets	$ 13,292	$ 11,241	$ 9,841	$ 7,370	$ 7,201
Debt obligations	$ 1,988	$ 2,000	$ 2,779	$ 1,413	$ 1,384
Convertible debt conversion obligation	$ —	$ —	$ 31	$ 87	$ —
Total shareholders' equity	$ 9,342	$ 7,471	$ 5,762	$ 4,469	$ 4,418
Cash dividends declared and paid per common share (D)	$ 0.610	$ 0.570	$ 0.485	$ 0.395	$ 0.340

(A) In fiscal year 2016, we began the wind down of our Icera modem operations. As a result, our income from operations for fiscal year 2016 included $131 million of restructuring and other charges.

(B) In fiscal year 2014, we issued Convertible Notes in the aggregate principal amount of $1 50 billion . The Convertible Notes first became convertible as of February 1, 2016 and matured on December 1, 2018. Refer to Note 11 of the Notes to the Consolidated Financial Statements in Part IV, Item 15 of this Annual Report on Form 10-K for additional information.

(C) In fiscal year 2017, we issued $1 00 billion of the Notes Due 2021, and $1 00 billion of the Notes Due 2026. Interest on the Notes is payable on March 16 and September 16 of each year, beginning on March 16, 2017. Refer to Note 11 of the Notes to the Consolidated Financial Statements in Part IV, Item 15 of this Annual Report on Form 10-K for additional information.

(D) In November 2012, we initiated a quarterly dividend payment of $0.075 per share, or $0.30 per share on an annual basis. In May 2015, we increased the quarterly cash dividend to $0.0975 per share, or $0.39 per share on an annual basis. In November 2015, we increased the quarterly cash dividend to $0.115 per share, or $0.46 per share on an annual basis. In November 2016, we increased the quarterly cash dividend to $0.14 per share, or $0.56 per share on an annual basis. In November 2017, we increased the quarterly cash dividend to $0.15 per share, or $0.60 per share on an annual basis. In November 2018, we increased the quarterly cash dividend to $0.16 per share, or $0.64 per share on an annual basis.

ITEM 7. MANAGEMENT'S DISCUSSION AND ANALYSIS OF FINANCIAL CONDITION AND RESULTS OF OPERATIONS

The following discussion and analysis of our financial condition and results of operations should be read in conjunction with "Item 1A. Risk Factors", "Item 6. Selected Financial Data", our Consolidated Financial Statements and related Notes thereto, as well as other cautionary statements and risks described elsewhere in this Annual Report on Form 10-K, before deciding to purchase, hold or sell shares of our common stock.

Overview

Our Company and Our Businesses

Starting with a focus on PC graphics, NVIDIA invented the GPU to solve some of the most complex problems in computer science. We have extended our focus in recent years to the revolutionary field of AI. Fueled by the sustained demand for better 3D graphics and the scale of the gaming market, NVIDIA has evolved the GPU into a computer brain at the intersection of VR, HPC, and AI.

Our two reportable segments - GPU and Tegra Processor - are based on a single underlying architecture. From our proprietary processors, we have created platforms that address four large markets where our expertise is critical: Gaming, Professional Visualization, Datacenter, and Automotive.

Our GPU product brands are aimed at specialized markets including GeForce for gamers; Quadro for designers; Tesla and DGX for AI data scientists and big data researchers; and GRID for cloud-based visual computing users. Our Tegra brand integrates an entire computer onto a single chip, and incorporates GPUs and multi-core CPUs to drive supercomputing for autonomous robots, drones, and cars, as well as for game consoles and mobile gaming and entertainment devices.

Headquartered in Santa Clara, California, NVIDIA was incorporated in California in April 1993 and reincorporated in Delaware in April 1998.

Recent Developments, Future Objectives and Challenges

Fiscal Year 2019 Summary

	Year Ended		
	January 27, 2019	January 28, 2018	Change
	($ in millions, except per share data)		
Revenue	$ 11,716	$ 9,714	Up 21%
Gross margin	61.2%	59.9%	Up 130 bps
Operating expenses	$ 3,367	$ 2,612	Up 29%
Income from operations	$ 3,804	$ 3,210	Up 19%
Net income	$ 4,141	$ 3,047	Up 36%
Net income per diluted share	$ 6.63	$ 4.82	Up 38%

Revenue for fiscal year 2019 increased 21% year over year, reflecting growth in each of our market platforms - gaming, professional visualization, datacenter, and automotive. GPU business revenue was $10.17 billion, up 25% from a year earlier. Tegra Processor business revenue - which includes automotive, SOC modules for gaming platforms, and embedded edge AI platforms - was $1.54 billion, up slightly from a year ago.

Gaming revenue was $6.25 billion, up 13% from a year ago driven by growth in gaming GPUs. Gaming GPU growth was fueled by Turing-based GPUs for desktops and by gaming notebooks based on our Max-Q technology. We experienced significant volatility in our Gaming revenue during fiscal year 2019. We believe demand for our desktop gaming GPU products used by end users for cryptocurrency mining and its after-effects have distorted trends in Gaming revenue. We also believe that deteriorating macroeconomic conditions, particularly in China have impacted consumer demand for our GeForce gaming GPU products. In addition, sales of certain high-end GeForce gaming GPUs using our new Turing architecture that we released during fiscal year 2019 were lower than we expected for the launch of a new architecture. As a result, during a portion of fiscal year 2019, we shipped a higher amount of desktop gaming GPU products relative to where end user demand turned out to be and subsequently compensated by shipping a lower amount of desktop gaming GPU products relative to end user demand to allow the channel to work down that inventory. For fiscal year 2020, we expect our Gaming revenue to be slightly down compared to fiscal year 2019, with expected growth from sales of Turing-based GPU products and notebook GPU products partially offsetting decreases that we believe were caused by the previously-noted factors.

Professional visualization revenue was $1.13 billion, up 21% from a year earlier driven by strength across both desktop and mobile workstation products.

Datacenter revenue was $2.93 billion, up 52% from a year ago, led by strong sales of our Volta architecture-based products, including NVIDIA Tesla V100 and DGX systems. Toward the end of fiscal year 2019, we believe that customers across broad-based vertical markets and geographies became increasingly cautious due to economic uncertainty, and a number of

Datacenter deals did not close. While we believe the pause is temporary, our visibility remains relatively low and we do not expect a meaningful recovery in the Datacenter market until later in fiscal year 2020.

Automotive revenue of $641 million was up 15% from a year earlier, driven by infotainment modules, production DRIVE platforms, and development agreements with automotive companies.

OEM and IP revenue was $767 million, down 1% from a year ago, driven by the absence of Intel licensing revenue, which concluded in the first quarter of fiscal year 2018. Revenue from cryptocurrency-specific products in fiscal years 2019 and 2018 was $306 million and $273 million, respectively. We expect revenue from cryptocurrency-specific products to be negligible going forward.

Gross margin for fiscal year 2019 was 61.2%, compared with 59.9% a year earlier, which reflects our continued shift toward higher-value platforms, which more than offset the impact of approximately $128 million in charges for excess DRAM and other components we recorded in the fourth quarter of fiscal year 2019 and a charge of $57 million we recorded during the third quarter of fiscal year 2019 related to prior architecture components and chips.

Operating expenses for fiscal year 2019 were $3.37 billion, up 29% from a year earlier, reflecting primarily employee additions and increases in employee compensation and other related costs, including infrastructure costs.

Income from operations for fiscal year 2019 was $3.80 billion, up 19% from a year earlier. Net income and net income per diluted share for fiscal year 2019 were $4.14 billion and $6.63, respectively, up 36% and 38%, respectively, from a year earlier, fueled primarily by revenue growth and improved gross margin, as well as the impact of the U.S. tax reform benefit.

During fiscal year 2019 , we returned $1.95 billion to shareholders through a combination of $1.58 billion in share repurchases and $371 million in quarterly cash dividends. We intend to return $3.00 billion to shareholders by the end of fiscal year 2020, including $700 million of share repurchases we made in the fourth quarter of fiscal year 2019.

Cash, cash equivalents and marketable securities were $7.42 billion as of January 27, 2019 , compared with $7.11 billion as of January 28, 2018 . The increase was primarily related to the increase in net income, partially offset by changes in working capital and the increases in stock repurchases, dividends and taxes paid related to restricted stock units.

GPU Business

During fiscal year 2019, for gaming, we announced NVIDIA RTX - a computer graphics technology using our Turing architecture that produces movie-quality images in real time using ray tracing and AI. During the year, we released many new GeForce RTX desktop gaming GPU products, including RTX 2080Ti, 2080, 2070 and 2060, as well as many new Max-Q GeForce gaming notebook GPU products - the most recent of which are powered by RTX GPUs.

For our professional visualization platform, we announced the Quadro GV100 GPU with RTX technology, making real-time ray tracing possible on professional design and content creation applications. We also unveiled the Quadro RTX series, which is designed to revolutionize the workflow of designers and artists on the desktop, and announced the NVIDIA CUDA-accelerated REDCODE RAW decode SDK, enabling developers and studios to edit 8K video.

For our datacenter platform, we unveiled many advances to our deep learning computing platform - including NVIDIA Tesla V100 GPUs with 32GB memory, NVIDIA NVSwitch GPU interconnect fabric, the NVIDIA DGX-2 and HGX-2 for AI and HPC, the NVIDIA RTX Server, and TensorRT 4 AI inference accelerator software. In addition, we introduced RAPIDS, an open-source GPU-acceleration platform for data science and machine learning, launched the NVIDIA T4 cloud GPU and NVIDIA TensorRT Hyperscale Inference Platform for advanced acceleration in hyperscale datacenters, announced GPU acceleration for Kubernetes to facilitate enterprise inference deployment on multi-cloud GPU clusters, and announced that five of the world’s seven fastest supercomputers are powered by NVIDIA GPUs.

Tegra Processor Business

During fiscal year 2019, for the automotive market, we introduced the NVIDIA DRIVE AutoPilot Level 2+ automated driving system, announced NVIDIA DRIVE AGX design wins with Toyota, Volvo Cars and Isuzu Motors, and announced that Daimler and Bosch have selected NVIDIA’s DRIVE platform to bring automated and driverless vehicles to city streets. We also began production of our Xavier single-chip autopilot SOC, started shipping the NVIDIA DRIVE AGX Xavier developer kit, and introduced the NVIDIA DRIVE Constellation server with DRIVE Sim software to safely test drive autonomous vehicles over billions of miles in virtual reality by leveraging NVIDIA GPUs and NVIDIA DRIVE Pegasus.

In addition, we launched the NVIDIA Jetson AGX Xavier module to help build the next-generation of autonomous machines and announced that Yamaha Motor Co. will use NVIDIA to power its upcoming lineup of autonomous machines.

Critical Accounting Policies and Estimates

Management's discussion and analysis of financial condition and results of operations are based upon our consolidated financial statements, which have been prepared in accordance with accounting principles generally accepted in the United States, or U.S. GAAP. The preparation of these financial statements requires us to make estimates and judgments that affect the reported amounts of assets, liabilities, revenue, cost of revenue, expenses and related disclosure of contingencies. On an on-going basis, we evaluate our estimates, including those related to revenue recognition, inventories, income taxes, goodwill, cash equivalents and marketable securities, stock-based compensation, and litigation, investigation and settlement costs and other contingencies. We base our estimates on historical experience and on various other assumptions that are believed to be reasonable under the circumstances, the results of which form the basis for making judgments about the carrying values of assets and liabilities.

We believe the following critical accounting policies affect our significant judgments and estimates used in the preparation of our consolidated financial statements. Our management has discussed the development and selection of these critical accounting policies and estimates with the Audit Committee of our Board of Directors. The Audit Committee has reviewed our disclosures relating to our critical accounting policies and estimates in this Annual Report on Form 10-K.

Revenue Recognition

We derive our revenue from product sales, including hardware and systems, license and development arrangements, and software licensing. We determine revenue recognition through the following steps: (1) identification of the contract with a customer; (2) identification of the performance obligations in the contract; (3) determination of the transaction price; (4) allocation of the transaction price to the performance obligations in the contract; and (5) recognition of revenue when, or as, we satisfy a performance obligation.

Product Sales Revenue

Revenue from product sales is recognized upon transfer of control of promised products to customers in an amount that reflects the consideration we expect to receive in exchange for those products. Revenue is recognized net of allowances for returns, customer programs and any taxes collected from customers.

For products sold with a right of return, we record a reduction to revenue by establishing a sales return allowance for estimated product returns at the time revenue is recognized, based primarily on historical return rates. However, if product returns for a fiscal period are anticipated to exceed historical return rates, we may determine that additional sales return allowances are required to properly reflect our estimated exposure for product returns.

Our customer programs involve rebates, which are designed to serve as sales incentives to resellers of our products in various target markets, and marketing development funds, or MDFs, which represent monies paid to our partners that are earmarked for market segment development and are designed to support our partners' activities while also promoting NVIDIA products. We account for customer programs as a reduction to revenue and accrue for potential rebates and MDFs based on the amount we expect to be claimed by customers.

License and Development Arrangements

Our license and development arrangements with customers typically require significant customization of our intellectual property components. As a result, we recognize the revenue from the license and the revenue from the development services as a single performance obligation over the period in which the development services are performed. We measure progress to completion based on actual cost incurred to date as a percentage of the estimated total cost required to complete each project. If a loss on an arrangement becomes probable during a period, we record a provision for such loss in that period.

Software Licensing

Our software licenses provide our customers with a right to use the software when it is made available to the customer. Customers may purchase either perpetual licenses or subscriptions to licenses, which differ mainly in the duration over which the customer benefits from the software. Software licenses are frequently sold along with post-contract customer support, or PCS. For such arrangements, we allocate revenue to the software license and PCS on a relative standalone selling price basis by maximizing the use of observable inputs to determine the standalone selling price for each performance obligation. Revenue from software licenses is recognized up front when the software is made available to the customer. PCS revenue is recognized ratably over the service period, or as services are performed.

Refer to Note 1 of the Notes to the Consolidated Financial Statements in Part IV, Item 15 of this Annual Report on Form 10-K for additional information.

Inventories

Inventory cost is computed on an adjusted standard basis, which approximates actual cost on an average or first-in, first-out basis. We charge cost of sales for inventory provisions to write down our inventory to the lower of cost or net realizable value or to completely write off obsolete or excess inventory. Most of our inventory provisions relate to the write-off of excess quantities of products, based on our inventory levels and future product purchase commitments compared to assumptions about future demand and m arket conditions.

Situations that may result in excess or obsolete inventory include changes in business and economic conditions, changes in market conditions, sudden and significant decreases in demand for our products, inventory obsolescence because of changing technology and customer requirements, failure to estimate customer demand properly, or unexpected competitive pricing actions by our competition. In addition, cancellation or deferral of customer purchase orders could result in our holding excess inventory.

The overall net effect on our gross margin from inventory provisions and sales of items previously written down was an unfavorable impact of 2.0% in fiscal year 2019 and insignificant in fiscal years 2018 and 2017 . The higher amount of charges we took to cost of sales for inventory provisions during fiscal year 2019 were primarily related to excess DRAM, other components, and prior architecture components and chips, whereas the charges we took during fiscal years 2018 and 2017 were primarily related to the write-off of excess quantities of GPU and Tegra products whose inventory levels were higher than our updated forecasts of future demand for those products. As a fabless semiconductor company, we must make commitments to purchase inventory based on forecasts of future customer demand. In doing so, we must account for our third-party manufacturers' lead times and constraints. We also adjust to other market factors, such as product offerings and pricing actions by our competitors, new product transitions, and macroeconomic conditions - all of which may impact demand for our products.

Refer to the Gross Profit and Gross Margin discussion below in this Management's Discussion and Analysis for further discussion.

Income Taxes

We recognize federal, state and foreign current tax liabilities or assets based on our estimate of taxes payable or refundable in the current fiscal year by tax jurisdiction. We recognize federal, state and foreign deferred tax assets or liabilities, as appropriate, for our estimate of future tax effects attributable to temporary differences and carryforwards; and we record a valuation allowance to reduce any deferred tax assets by the amount of any tax benefits that, based on available evidence and judgment, are not expected to be realized.

Our calculation of deferred tax assets and liabilities is based on certain estimates and judgments and involves dealing with uncertainties in the application of complex tax laws. Our estimates of deferred tax assets and liabilities may change based, in part, on added certainty or finality to an anticipated outcome, changes in accounting standards or tax laws in the United States, or foreign jurisdictions where we operate, or changes in other facts or circumstances. In addition, we recognize liabilities for potential United States and foreign income tax contingencies based on our estimate of whether, and the extent to which, additional taxes may be due. If we determine that payment of these amounts is unnecessary or if the recorded tax liability is less than our current assessment, we may be required to recognize an income tax benefit or additional income tax expense in our financial statements accordingly.

As of January 27, 2019 , we had a valuation allowance of $562 million related to state and certain foreign deferred tax assets that management determined are not likely to be realized due to projections of future taxable income and potential utilization limitations of tax attributes acquired as a result of stock ownership changes. To the extent realization of the deferred tax assets becomes more-likely-than-not, we would recognize such deferred tax asset as an income tax benefit during the period.

We recognize the benefit from a tax position only if it is more-likely-than-not that the position would be sustained upon audit based solely on the technical merits of the tax position. Our policy is to include interest and penalties related to unrecognized tax benefits as a component of income tax expense.

The TCJA, which was enacted in December 2017, significantly changed U.S. tax law, including a reduction of the U.S. federal corporate income tax rate from 35% to 21%, a requirement for companies to pay a one-time transition tax on the earnings of certain foreign subsidiaries that were previously tax deferred, and the creation of new taxes (global intangible low-taxed income, or GILTI) on certain foreign-source earnings. As a fiscal year-end taxpayer, certain provisions of the TCJA began to impact us in the fourth quarter of fiscal year 2018, while other provisions impacted us beginning in fiscal year 2019. The SEC had provided guidance in Staff Accounting Bulletin No. 118, Income Tax Accounting Implications of the Tax Cuts and Jobs Act (SAB 118), which allowed companies to record provisional amounts during a measurement period up to one year

from the enactment date. As of January 27, 2019, we completed our accounting for all of the enactment-date income tax effects of the TCJA and elected to account for GILTI in deferred taxes. Refer to Note 13 of the Notes to the Consolidated Financial Statements in Part IV, Item 15 of this Annual Report on Form 10-K for additional information specific to accounting for income taxes and the impacts from the enactment of the TCJA.

Goodwill

Goodwill is subject to our annual impairment test during the fourth quarter of our fiscal year, or earlier, if indicators of potential impairment exist, using either a qualitative or a quantitative assessment. Our impairment review process compares the fair value of the reporting unit in which the goodwill resides to its carrying value. We have identified two reporting units, GPU and Tegra Processor, for the purposes of completing our goodwill analysis. Goodwill assigned to the GPU and Tegra Processor reporting units as of January 27, 2019 was $210 million and $408 million , respectively. Determining the fair value of a reporting unit requires us to make judgments and involves the use of significant estimates and assumptions. We also make judgments and assumptions in allocating assets and liabilities to each of our reporting units. We base our fair value estimates on assumptions we believe to be reasonable but that are unpredictable and inherently uncertain.

During the fourth quarter of fiscal year 2019 , we used the qualitative assessment to test goodwill for impairment for each reporting unit and concluded there was no impairment.

Refe r to Note 5 of the Notes to the Consolidated Financial Statements in Part IV, Item 15 of this Annual Report on Form 10-K for additional information.

Cash Equivalents and Marketable Securities

Cash equivalents consist of financial instruments which are readily convertible into cash and have original maturities of three months or less at the time of acquisition. Marketable securities consist of highly liquid debt investments with maturities greater than three months when purchased. We measure our cash equivalents and marketable securities at fair value. The fair values of our financial assets are determined using quoted market prices of identical assets or quoted market prices of similar assets from active markets. All of our available-for-sale debt investments are subject to a periodic impairment review. We record a charge to earnings when a decline in fair value is significantly below cost basis and judged to be other-than-temporary, or have other indicators of impairments.

We performed an impairment review of our debt investment portfolio as of January 27, 2019 . We concluded that our debt investments were appropriately valued and that no other-than-temporary impairment charges were necessary on our portfolio of available-for-sale debt investments as of January 27, 2019 .

Refe r to Notes 7 and 8 of the Notes to the Consolidated Financial Statements in Part IV, Item 15 of this Annual Report on Form 10-K for additional information.

Stock-based Compensation

Our stock-based compensation expense is associated with restricted stock units, or RSUs, performance stock units that are based on our corporate financial performance targets, or PSUs, performance stock units that are based on market conditions, or market-based PSUs, and our employee stock purchase plan. The number of PSUs and market-based PSUs that will ultimately be awarded is contingent on the Company’s level of achievement compared with the corporate financial performance target established by our Compensation Committee in the beginning of each fiscal year.

Re fer to Notes 1 and 3 of the Notes to the Consolidated Financial Statements in Part IV, Item 15 of this Annual Report on Form 10-K for additional information.

Litigation, Investigation and Settlement Costs

From time to time, we are involved in legal actions and/or investigations by regulatory bodies. We are aggressively defending our current litigation matters. However, there are many uncertainties associated with any litigation or investigations, and we cannot be certain that these actions or other third-party claims against us will be resolved without costly litigation, fines and/or substantial settlement payments. If that occurs, our business, financial condition and results of operations co uld be materially and adversely affected. If information becomes available that causes us to determine that a loss in any of our pending litigation, investigations or settlements is probable, and we can reasonably estimate the loss associated with such events, we will record the loss in accordance with U.S. GAAP. However, the actual liability in any such litigation or investigation may be materially different from our estimates, which could require us to record additional costs.

Results of Operations

The following table sets forth, for the periods indicated, certain items in our Consolidated Statements of Income expressed as a percentage of revenue.

	Year Ended		
	January 27, 2019	January 28, 2018	January 29, 2017
Revenue	100.0 %	100.0 %	100.0 %
Cost of revenue	38.8	40.1	41.2
Gross profit	61.2	59.9	58.8
Operating expenses:			
Research and development	20.3	18.5	21.2
Sales, general and administrative	8.5	8.4	9.6
Total operating expenses	28.7	26.9	30.8
Income from operations	32.5	33.0	28.0
Interest income	1.2	0.7	0.8
Interest expense	(0.5)	(0.6)	(0.8)
Other, net	0.1	(0.2)	(0.4)
Total other income (expense)	0.8	(0.1)	(0.4)
Income before income tax expense	33.3	32.9	27.6
Income tax expense (benefit)	(2.1)	1.5	3.5
Net income	35.3 %	31.4 %	24.1 %

Revenue

Revenue by Reportable Segments

	Year Ended				Year Ended			
	January 27, 2019	January 28, 2018	$ Change	% Change	January 28, 2018	January 29, 2017	$ Change	% Change
	($ in millions)				*($ in millions)*			
GPU	$ 10,175	$ 8,137	$ 2,038	25 %	$ 8,137	$ 5,822	$ 2,315	40 %
Tegra Processor	1,541	1,534	7	— %	1,534	824	710	86 %
All Other	—	43	(43)	(100)%	43	264	(221)	(84)%
Total	$ 11,716	$ 9,714	$ 2,002	21 %	$ 9,714	$ 6,910	$ 2,804	41 %

GPU Business. GPU business revenue increased by 25% in fiscal year 2019 compared to fiscal year 2018 . This increase was due primarily to 18% growth in sales of GeForce GPU products for gaming, driven by initial sales of Turing-based GPUs for desktops and by high-performance notebooks based on our Max-Q technology. Datacenter revenue, including Tesla, GRID and DGX, increased 52%, reflecting strong sales of our Volta architecture products, including NVIDIA Tesla V100 and DGX systems. Revenue from Quadro GPUs for professional visualization increased 21% due primarily to higher sales across desktop and mobile workstation products. Our PC OEM revenue decreased by 1% driven by the absence of Intel licensing revenue in fiscal year 2019. Revenue from cryptocurrency-specific products in fiscal years 2019 and 2018 was $306 million and $273 million, respectively. We expect cryptocurrency-related revenue to be negligible going forward.

GPU business revenue increased by 40% in fiscal year 2018 compared to fiscal year 2017 led by growth in gaming, datacenter and professional visualization. Revenue from sales of GeForce GPU products for gaming increased over 20%, reflecting continued strong demand for our Pascal-based GPU products. Datacenter revenue, including Tesla, GRID and DGX, increased 133%, reflecting strong demand from hyperscale and cloud customers for deep learning training and accelerated GPU computing as well as demand for HPC, DGX AI supercomputing and GRID virtualization platforms. Revenue from Quadro GPUs for professional visualization increased by 12% due primarily to higher sales in both high-end desktop and mobile

workstation products. Revenue from GeForce GPU products for mainstream PC OEMs increased by over 90% due primarily to strong demand for GPU products targeted for cryptocurrency mining.

Tegra Processor Business. Tegra Processor business revenue was up slightly in fiscal year 2019 compared to fiscal year 2018 . This was driven by an increase of over 15% in automotive revenue, primarily from infotainment modules, production DRIVE PX platforms, and development agreements with automotive companies, offset by a decline of approximately 15% in SOC modules for gaming platforms and related development services.

Tegra Processor business revenue increased by 86% in fiscal year 2018 compared to fiscal year 2017 . This was driven by an increase of over 300% in revenue from SOC modules for gaming platforms and development services, and an increase of 15% in automotive revenue, primarily from infotainment modules, DRIVE PX platforms and development agreements for self-driving cars.

All Other. Our patent license agreement with Intel concluded in the first quarter of fiscal year 2018. For fiscal year 2018, we recognized related revenue of $ 43 million , down from $ 264 million for fiscal year 2017.

Concentration of Revenue

Revenue from sales to customers outside of the United States accounted for 87% of total revenue for each of fiscal years 2019 , 2018 , and 2017 . Revenue by geographic region is allocated to individual countries based o n the location to which the products are initially billed even if the revenue is attributable to end customers in a different location.

No single customer represented more than 10% of total revenue for fiscal years 2019 and 2018. In fiscal year 2017, we had one customer that represented 12% of our total revenue.

Gross Profit and Gross Margin

Gross profit consists of total revenue, net of allowances, less cost of revenue. Cost of revenue consists primarily of the cost of semiconductors purchased from subcontractors, including wafer fabrication, assembly, testing and packaging, board and device costs, manufacturing support costs, including labor and overhead associated with such purchases, final test yield fallout, inventory and warranty provisions, memory and component costs, and shipping costs. Cost of revenue also includes development costs for license and service arrangements and stock-based compensation related to personnel associated with manufacturing.

Our overall gross margin was 61.2% , 59.9% , and 58.8% for fiscal years 2019 , 2018 , and 2017 , respectively. The increase in fiscal year 2019 reflects our continued shift toward higher-value platforms, which more than offset the impact of approximately $128 million in charges for excess DRAM and other components we recorded in the fourth quarter of fiscal year 2019 and a charge of $57 million we recorded during the third quarter of fiscal year 2019 related to prior architecture components and chips. The increase in fiscal year 2018 was driven primarily by a favorable shift in mix, the growth of our GeForce gaming GPU revenue, and the growth of our datacenter revenue for cloud, deep learning, AI, and graphics virtualization. The increase in fiscal year 2018 was partially offset by the conclusion of our patent license agreement with Intel in the first quarter of fiscal year 2018.

Inventory provisions totaled $270 million, $48 million, and $62 million for fiscal years 2019 , 2018 , and 2017 , respectively. Sales of inventory that was previously written-off or written-down totaled $41 million, $35 million, and $51 million for fiscal years 2019 , 2018 , and 2017 , respectively. As a result, the overall net effect on our gross margin was an unfavorable impact of 2.0% in fiscal year 2019 and insignificant in fiscal years 2018 and 2017 .

A discussion of our gross margin results for each of our reportable segments is as follows:

GPU Business. The gross margin of our GPU business increased during fiscal year 2019 when compared to fiscal year 2018 , primarily due to strong sales of high-end GeForce gaming GPUs and revenue growth in Datacenter, including Tesla, GRID and DGX, for cloud, deep learning, AI, and graphics virtualization. The gross margin of our GPU business increased during fiscal year 2018 when compared to fiscal year 2017 primarily due to strong sales of our GeForce gaming GPU products and revenue growth in datacenter, including Tesla, GRID and DGX, for cloud, deep learning, AI, and graphics virtualization.

Tegra Processor Business. The gross margin of our Tegra Processor business increased during fiscal year 2019 when compared to fiscal year 2018 , primarily due to a favorable mix shift. The gross margin of our Tegra Processor business increased during fiscal year 2018 when compared to fiscal year 2017 , primarily due to revenue growth in gaming development platforms and automotive.

Operating Expenses

	Year Ended				Year Ended			
	January 27, 2019	January 28, 2018	$ Change	% Change	January 28, 2018	January 29, 2017	$ Change	% Change
	($ in millions)				($ in millions)			
Research and development expenses	$ 2,376	$ 1,797	$ 579	32%	$ 1,797	$ 1,463	$ 334	23 %
% of net revenue	*20.3%*	*18.5%*			*18.5%*	*21.2%*		
Sales, general and administrative expenses	991	815	176	22%	815	663	152	23 %
% of net revenue	*8.5%*	*8.4%*			*8.4%*	*9.6%*		
Restructuring and other charges	—	—	—	—%	—	3	(3)	(100)%
% of net revenue	*—%*	*—%*			*—%*	*—%*		
Total operating expenses	$ 3,367	$ 2,612	$ 755	29%	$ 2,612	$ 2,129	$ 483	23 %

Research and Development

Research and development expenses increased by 32% in fiscal year 2019 compared to fiscal year 2018 and increased by 23% in fiscal year 2018 compared to fiscal year 2017 , driven primarily by employee additions and increases in employee compensation and other related costs, including infrastructure costs and stock-based compensation expense.

Sales, General and Administrative

Sales, general and administrative expenses increased by 22% in fiscal year 2019 compared to fiscal year 2018 and increased by 23% in fiscal year 2018 compared to fiscal year 2017 , driven primarily by employee additions and increases in employee compensation and other related costs, including infrastructure costs and stock-based compensation expense.

Total Other Income (Expense)

Interest Income and Interest Expense

Interest income consists of interest earned on cash, cash equivalents and marketable securities. Interest income was $136 million , $69 million , and $54 million in fiscal years 2019 , 2018 , and 2017 , respectively. The increase in interest income was primarily due to higher average invested balances and higher rates from our floating rate securities and the purchase of new securities.

Interest expense is primarily comprised of coupon interest and debt discount amortization related to the 2.20% Notes Due 2021 and 3.20% Notes Due 2026 issued in September 2016, and the Convertible Notes issued in December 2013. Interest expense was $58 million , $61 million , and $58 million in fiscal years 2019 , 2018 , and 2017 , respectively.

Other, Net

Other, net, consists primarily of realized or unrealized gains and losses from non-affiliated investments, losses on early debt conversions of the Convertible Notes, and the impact of changes in foreign currency rates. Other, net, was $14 million of income during fiscal year 2019, consisting primarily of $12 million unrealized gains from non-affiliated investments. Other, net, was $22 million and $25 million of expense in fiscal years 2018 and 2017, respectively, consisting primarily of $19 million and $21 million of losses recognized from early conversions of the Convertible Notes during fiscal years 2018 and 2017, respectively.

Income Taxes

The TCJA, which was enacted in December 2017, significantly changed U.S. tax law, including a reduction of the U.S. federal corporate income tax rate from 35% to 21%, a requirement for companies to pay a one-time transition tax on the earnings of certain foreign subsidiaries that were previously tax deferred and the creation of new taxes (GILTI) on certain foreign-source earnings. As a fiscal year-end taxpayer, certain provisions of the TCJA began to impact us in the fourth quarter of fiscal year 2018, while other provisions impacted us beginning in fiscal year 2019.

We recognized income tax benefit of $245 million for fiscal year 2019, and income tax expense of $149 million and $239 million for fiscal years 2018, and 2017, respectively. Our annual effective tax rate was (6.3)%, 4.7%, and 12.5% for fiscal years 2019, 2018, and 2017, respectively. The decrease in our effective tax rate in fiscal year 2019 as compared to fiscal years 2018 and 2017 was primarily due to a decrease in the U.S. statutory tax rate from 33.9% to 21%, the finalization of the enactment-date income tax effects of the TCJA, higher U.S federal research tax credits and excess tax benefits related to stock-based compensation in fiscal year 2019.

The decrease in our effective tax rate in fiscal year 2018 as compared to fiscal year 2017 was primarily due to the provisional impact of the tax law changes and recognition of excess tax benefits related to stock-based compensation.

Our effective tax rate for fiscal year 2019 was lower than the U.S. federal statutory rate of 21% due primarily to income earned in jurisdictions, including British Virgin Islands, Hong Kong, China, Taiwan and United Kingdom, where the tax rate was lower than the U.S. federal statutory tax rates, the finalization of the enactment-date income tax effects of the TCJA, favorable recognition of U.S. federal research tax credits, and excess tax benefits related to stock-based compensation.

Our effective tax rate for fiscal years 2018 and 2017 was lower than the blended U.S. federal statutory rate of 33.9% for fiscal year 2018 and 35% for fiscal year 2017 due primarily to income earned in jurisdictions, including British Virgin Islands, Hong Kong, China, Taiwan and United Kingdom, where the tax rate was lower than the U.S. federal statutory tax rates, favorable recognition of U.S. federal research tax credits, the provisional impact of the tax law changes in 2018, and excess tax benefits related to stock-based compensation.

In fiscal year 2018 and the first nine months of fiscal year 2019, we recorded provisional amounts for certain enactment-date effects of the TCJA by applying the SEC guidance in SAB 118 because we had not yet completed our accounting for these effects. Furthermore, under U.S. GAAP, we can make an accounting policy election to either treat taxes due on the GILTI as a current period expense or factor such amounts into our measurement of deferred taxes. Because we were still evaluating the GILTI provisions as of January 28, 2018, we recorded no GILTI-related deferred balances. After further evaluation, we elected to account for GILTI deferred taxes. As of January 27, 2019, we completed our accounting for all of the enactment-date income tax effects of the TCJA and recognized a reduction of $368 million to the provisional amount recorded at January 28, 2018, primarily relating to the effects of electing to account for GILTI in deferred taxes.

Refer to Note 13 of the Notes to the Consolidated Financial Statements in Part IV, Item 15 of this Annual Report on Form 10-K for additional information.

Liquidity and Capital Resources

	January 27, 2019	January 28, 2018
	(In millions)	
Cash and cash equivalents	$ 782	$ 4,002
Marketable securities	6,640	3,106
Cash, cash equivalents, and marketable securities	$ 7,422	$ 7,108

	Year Ended		
	January 27, 2019	January 28, 2018	January 29, 2017
	(In millions)		
Net cash provided by operating activities	$ 3,743	$ 3,502	$ 1,672
Net cash provided by (used in) investing activities	$ (4,097)	$ 1,278	$ (793)
Net cash provided by (used in) financing activities	$ (2,866)	$ (2,544)	$ 291

As of January 27, 2019 , we had $7.42 billion in cash, cash equivalents and marketable securities, an increase of $314 million from the end of fiscal year 2018 . Our investment policy requires the purchase of highly rated fixed income securities, the diversification of investment types and credit exposures, and certain limits on our portfolio duration.

Cash provided by operating activities increased in fiscal year 2019 compared to fiscal year 2018 , primarily due to higher net income, partially offset by changes in working capital. Cash provided by operating activities increased in fiscal year 2018 compared to fiscal year 2017, primarily due to higher net income and changes in working capital.

Cash used in investing activities increased in fiscal year 2019 compared to fiscal year 2018 , due to higher purchases and lower sales of marketable securities, partially offset by higher maturities of marketable securities. Cash provided by investing activities for fiscal year 2018 increased from fiscal year 2017 , primarily due to a reduction in purchases of marketable securities, partially offset by the purchase of our previously-financed Santa Clara campus building.

Cash used in financing activities increased in fiscal year 2019 compared to fiscal year 2018 , due to higher share repurchases and higher tax payments related to employee stock plans, partially offset by lower repayments of Convertible Notes. Cash used in financing activities in fiscal year 2018 increased from fiscal year 2017 , primarily due to cash provided from the issuance of $2.00 billion of Notes in fiscal year 2017 as well as higher repayments of Convertible Notes, tax payments related to employee stock plans, share repurchases and dividend payments in fiscal year 2018.

Liquidity

Our primary sources of liquidity are our cash and cash equivalents, our marketable securities, and the cash generated by our operations. As of January 27, 2019 and January 28, 2018 , we had $ 7.42 billion and $ 7.11 billion , respectively, in cash, cash equivalents and marketable securities. Our marketable securities consist of debt securities issued by the U.S. government and its agencies, highly rated corporations and financial institutions, asset-backed issuers, mortgage-backed securities by government-sponsored enterprises, and foreign government entities. These marketable securities are denominated in United States dollars. Refer to Critical Accounting Policies and Estimates in Part II, Item 7, Quantitative and Qualitative Disclosures About Market Risk in Part II, Item 7A and Note 7 of the Notes to the Consolidated Financial Statements in Part IV, Item 15 of this Annual Report on Form 10-K for additional information.

As a result of the TCJA, substantially all of our cash, cash equivalents and marketable securities held outside of the United States as of January 27, 2019 are available for use in the United States without incurring additional U.S. federal income taxes. Refer to Note 13 of the Notes to the Consolidated Financial Statements in Part IV, Item 15 of this Annual Report on Form 10-K for additional information.

Capital Return to Shareholders

We previously announced our plan to return $1.25 billion to shareholders in fiscal year 2019 and an additional $3.00 billion by the end of fiscal year 2020 - some of which would begin in the fourth quarter of fiscal year 2019. During fiscal year 2019 , we repurchased a total of 9 million shares for $1.58 billion , including $700 million of the $3.00 billion, and paid $371 million in cash dividends.

We intend to return the remaining $2.30 billion of the $3.00 billion to shareholders by the end of fiscal year 2020 through a combination of share repurchases and cash dividends.

In November 2018, the Board authorized an additional $7.00 billion under our share repurchase program and extended it through the end of December 2022. As of January 27, 2019 , we were authorized to repurchase additional shares of our common stock up to $7.24 billion .

In November 2018, we also announced a 7% increase in our quarterly cash dividend to $0.16 per share from $0.15 per share.

Our cash dividend program and the payment of future cash dividends under that program are subject to our Board's continuing determination that the dividend program and the declaration of dividends thereunder are in the best interests of our shareholders. Refer to Note 14 of the Notes to the Consolidated Financial Statements in Part IV, Item 15 of this Annual Report on Form 10-K for further discussion.

Notes Due 2021 and Notes Due 2026

In fiscal year 2017, we issued $1.00 billion of the Notes Due 2021 and $1.00 billion of the Notes Due 2026, collectively, the Notes. The net proceeds from the Notes were $1.98 billion, after deducting debt discounts and issuance costs.

Revolving Credit Facility

We have a Credit Agreement under which we may borrow up to $575 million for general corporate purposes and can obtain revolving loan commitments up to $425 million. As of January 27, 2019 , we had not borrowed any amounts under this agreement.

Commercial Paper

We have a $575 million commercial paper program to support general corporate purposes. As of January 27, 2019 , we had not issued any commercial paper.

Refer to Note 11 of the Notes to the Consolidated Financial Statements in Part IV, Item 15 of this Annual Report on Form 10-K for further discussion.

Operating Capital and Capital Expenditure Requirements

In fiscal year 2019, we began construction on a 750 thousand square foot building on our Santa Clara campus, which is currently targeted for completion in fiscal year 2022. We believe that our existing cash balances and anticipated cash flows from operations will be sufficient to meet our operating requirements for at least the next twelve months.

Off-Balance Sheet Arrangements

As of January 27, 2019, we had no material off-balance sheet arrangements as defined by applicable SEC regulations.

Contractual Obligations

The following table summarizes our contractual obligations as of January 27, 2019 :

	Payment Due By Period					
Contractual Obligations	**Total**	**Less than 1 Year**	**1-3 Years**	**4-5 Years**	**More than 5 Years**	**All Other**
			(In millions)			
Long-term debt (1)	$ 2,302	$ 54	$ 1,100	$ 64	$ 1,084	$ —
Inventory purchase obligations	912	912	—	—	—	—
Transition tax payable (2)	384	33	67	96	188	—
Uncertain tax positions, interest and penalties (3)	163	—	—	—	—	163
Operating leases	683	100	187	131	265	—
Capital purchase obligations	258	192	66	—	—	—
Total contractual obligations	$ 4,702	$ 1,291	$ 1,420	$ 291	$ 1,537	$ 163

(1) Represents the aggregate principal amount of $2.00 billion and anticipated interest payments of $302 million for the Notes. Refer to Note 11 of the Notes to the Consolidated Financial Statements in Part IV, Item 15 of this Annual Report on Form 10-K.

(2) Represents our remaining tax payable of the one-time transition tax that resulted from enactment of the TCJA in fiscal year 2018. As of January 27, 2019, we have paid the first installment of $33 million. The remaining will be payable in seven annual installments. The next installment of $33 million is classified as a current income tax payable. The installment amounts are equal to 8% of the total liability, payable in fiscal years 2019 through 2023, 15% in fiscal year 2024, 20% in fiscal year 2025 and 25% in fiscal year 2026. Refer to Note 13 of the Notes to the Consolidated Financial Statements in Part IV, Item 15 of this Annual Report on Form 10-K, for additional information about the one-time transition tax.

(3) Represents unrecognized tax benefits of $163 million which consists of $142 million and the related interest and penalties of $21 million recorded in non-current income tax payable as of January 27, 2019. We are unable to reasonably estimate the timing of any potential tax liability or interest/penalty payments in individual years due to uncertainties in the underlying income tax positions and the timing of the effective settlement of such tax positions.

Adoption of New and Recently Issued Accounting Pronouncements

Refer to Note 1 of the Notes to the Consolidated Financial Statements in Part IV, Item 15 of this Annual Report on Form 10-K for a discussion of adoption of new and recently issued accounting pronouncements.

ITEM 7A. QUANTITATIVE AND QUALITATIVE DISCLOSURES ABOUT MARKET RISK

Investment and Interest Rate Risk

We are exposed to interest rate risk related to our floating and fixed-rate investment portfolio and outstanding debt. The investment portfolio is managed consistent with our overall liquidity strategy in support of both working capital needs and strategic growth of our businesses.

As of January 27, 2019 , we performed a sensitivity analysis on our floating and fixed rate financial investments. According to our analysis, parallel shifts in the yield curve of plus or minus 0.5% would result in a decrease in fair value for these investments of $8 million, or an increase in fair value for these investments of $7 million , respectively.

In fiscal year 2017, we issued $1.00 billion of the Notes Due 2021 and $1.00 billion of the Notes Due 2026. We carry the Notes at face value less unamortized discount on our Consolidated Balance Sheets. As the Notes bear interest at a fixed rate, we have no financial statement risk associated with changes in interest rates. Refer to Note 11 of the Notes to the Consolidated Financial Statements in Part IV, Item 15 of this Annual Report on Form 10-K for additional information.

Foreign Exchange Rate Risk

We consider our direct exposure to foreign exchange rate fluctuations to be minimal. Gains or losses from foreign currency remeasurement are included in other income or expense and to date have not been significant. The impact of foreign currency transaction gain or loss included in determining net income was not significant for fiscal years 2019, 2018, and 2017.

Sales and arrangements with third-party manufacturers provide for pricing and payment in United States dollars, and, therefore, are not subject to exchange rate fluctuations. Increases in the value of the United States' dollar relative to other currencies would make our products more expensive, which could negatively impact our ability to compete. Conversely, decreases in the value of the United States' dollar relative to other currencies could result in our suppliers raising their prices in order to continue doing business with us. Additionally, we have international operations and incur expenditures in currencies other than U.S. dollars. Our operating expenses benefit from a stronger dollar and are adversely affected by a weaker dollar.

We use foreign currency forward contracts to mitigate the impact of foreign currency exchange rate movements on our operating expenses. We designate these contracts as cash flow hedges and assess the effectiveness of the hedge relationships on a spot to spot basis. Gains or losses on the contracts are recorded in accumulated other comprehensive income or loss, and then reclassified to operating expense when the related operating expenses are recognized in earnings or ineffectiveness should occur.

We also use foreign currency forward contracts to mitigate the impact of foreign currency movements on monetary assets and liabilities that are denominated in currencies other than U.S. dollar. These forward contracts were not designated for hedge accounting treatment. Therefore, the change in fair value of these contracts is recorded in other income or expense and offsets the change in fair value of the hedged foreign currency denominated monetary assets and liabilities, which is also recorded in other income or expense.

Refer to Note 10 of the Notes to the Consolidated Financial Statements in Part IV, Item 15 of this Annual Report on Form 10-K for additional information.

ITEM 8. FINANCIAL STATEMENTS AND SUPPLEMENTARY DATA

The information required by this Item is set forth in our Consolidated Financial Statements and Notes thereto included in this Annual Report on Form 10-K.

ITEM 9. CHANGES IN AND DISAGREEMENTS WITH ACCOUNTANTS ON ACCOUNTING AND FINANCIAL DISCLOSURE

None.

ITEM 9A. CONTROLS AND PROCEDURES

Controls and Procedures

Disclosure Controls and Procedures

Based on their evaluation as of January 27, 2019 , our management, including our Chief Executive Officer and Chief Financial Officer, has concluded that our disclosure controls and procedures (as defined in Rule 13a-15(e) under the Securities Exchange Act of 1934, as amended, or the Exchange Act) were effective to provide reasonable assurance.

Management's Annual Report on Internal Control Over Financial Reporting

Our management is responsible for establishing and maintaining adequate internal control over financial reporting, as such term is defined in Exchange Act Rule 13a-15(f). Under the supervision and with the participation of our management, including our Chief Executive Officer and Chief Financial Officer, we conducted an evaluation of the effectiveness of our internal control over financial reporting as of January 27, 2019 based on the criteria set forth in *Internal Control - Integrated Framework (2013)* issued by the Committee of Sponsoring Organizations of the Treadway Commission. Based on our evaluation under the criteria set forth in *Internal Control — Integrated Framework* , our management concluded that our internal control over financial reporting was effective as of January 27, 2019 .

The effectiveness of our internal control over financial reporting as of January 27, 2019 has been audited by PricewaterhouseCoopers LLP, an independent registered public accounting firm, as stated in its report which is included herein.

Changes in Internal Control Over Financial Reporting

There were no changes in our internal control over financial reporting during our last fiscal quarter that have materially affected, or are reasonably likely to materially affect, our internal control over financial reporting.

Inherent Limitations on Effectiveness of Controls

Our management, including our Chief Executive Officer and Chief Financial Officer, does not expect that our disclosure controls and procedures or our internal controls, will prevent all error and all fraud. A control system, no matter how well conceived and operated, can provide only reasonable, not absolute, assurance that the objectives of the control system are met. Further, the design of a control system must reflect the fact that there are resource constraints, and the benefits of controls must be considered relative to their costs. Because of the inherent limitations in all control systems, no evaluation of controls can provide absolute assurance that all control issues and instances of fraud, if any, within NVIDIA have been detected.

ITEM 9B. OTHER INFORMATION

None.

PART III

Certain information required by Part III is omitted from this report because we will file with the SEC a definitive proxy statement pursuant to Regulation 14A, or the 2019 Proxy Statement, no later than 120 days after the end of fiscal year 2019, and certain information included therein is incorporated herein by reference.

ITEM 10. DIRECTORS, EXECUTIVE OFFICERS AND CORPORATE GOVERNANCE

Identification of Directors

Information regarding directors required by this item will be contained in our 2019 Proxy Statement under the caption “Proposal 1 - Election of Directors,” and is hereby incorporated by reference.

Identification of Executive Officers

Reference is made to the information regarding executive officers appearing under the heading “Executive Officers of the Registrant” in Part I of this Annual Report on Form 10-K, which information is hereby incorporated by reference.

Identification of Audit Committee and Financial Experts

Information regarding our Audit Committee required by this item will be contained in our 2019 Proxy Statement under the captions “Report of the Audit Committee of the Board of Directors” and “Information About the Board of Directors and Corporate Governance,” and is hereby incorporated by reference.

Material Changes to Procedures for Recommending Directors

Information regarding procedures for recommending directors required by this item will be contained in our 2019 Proxy Statement under the caption “Information About the Board of Directors and Corporate Governance,” and is hereby incorporated by reference.

Compliance with Section 16(a) of the Exchange Act

Information regarding compliance with Section 16(a) of the Exchange Act required by this item will be contained in our 2019 Proxy Statement under the caption “Section 16(a) Beneficial Ownership Reporting Compliance,” and is hereby incorporated by reference.

Code of Conduct

Information regarding our Code of Conduct required by this item will be contained in our 2019 Proxy Statement under the caption “Information About the Board of Directors and Corporate Governance - Code of Conduct,” and is hereby incorporated by reference. The full text of our Code of Conduct and Financial Team Code of Conduct are published on the Investor Relations portion of our website, under Corporate Governance, at www.nvidia.com. The contents of our website are not a part of this Annual Report on Form 10-K.

ITEM 11. EXECUTIVE COMPENSATION

Information regarding our executive compensation required by this item will be contained in our 2019 Proxy Statement under the captions “Executive Compensation”, “Compensation Committee Interlocks and Insider Participation”, “Director Compensation” and “Compensation Committee Report,” and is hereby incorporated by reference.

ITEM 12. SECURITY OWNERSHIP OF CERTAIN BENEFICIAL OWNERS AND MANAGEMENT AND RELATED STOCKHOLDER MATTERS

Ownership of NVIDIA Securities

Information regarding ownership of NVIDIA securities required by this item will be contained in our 2019 Proxy Statement under the caption “Security Ownership of Certain Beneficial Owners and Management,” and is hereby incorporated by reference.

Equity Compensation Plan Information

Information regarding our equity compensation plans required by this item will be contained in our 2019 Proxy Statement under the caption "Equity Compensation Plan Information," and is hereby incorporated by reference.

ITEM 13. CERTAIN RELATIONSHIPS AND RELATED TRANSACTIONS, AND DIRECTOR INDEPENDENCE

Information regarding related transactions and director independence required by this item will be contained in our 2019 Proxy Statement under the captions “Review of Transactions with Related Persons” and “Information About the Board of Directors and Corporate Governance - Independence of the Members of the Board of Directors,” and is hereby incorporated by reference.

ITEM 14. PRINCIPAL ACCOUNTING FEES AND SERVICES

Information regarding accounting fees and services required by this item will be contained in our 2019 Proxy Statement under the caption “Fees Billed by the Independent Registered Public Accounting Firm,” and is hereby incorporated by reference.

PART IV

ITEM 15. EXHIBITS, FINANCIAL STATEMENT SCHEDULE

REPORT OF INDEPENDENT REGISTERED PUBLIC ACCOUNTING FIRM

To the Stockholders and Board of Directors of NVIDIA Corporation

Opinions on the Financial Statements and Internal Control over Financial Reporting

We have audited the accompanying consolidated balance sheets of NVIDIA Corporation and its subsidiaries as of January 27, 2019 and January 28, 2018, and the related consolidated statements of income, comprehensive income, shareholders' equity and cash flows for each of the three years in the period ended January 27, 2019, including the related notes and financial statement schedule listed in the index appearing under Item 15(a)(2) (collectively referred to as the "consolidated financial statements"). We also have audited the Company's internal control over financial reporting as of January 27, 2019, based on criteria established in Internal Control - Integrated Framework (2013) issued by the Committee of Sponsoring Organizations of the Treadway Commission (COSO).

In our opinion, the consolidated financial statements referred to above present fairly, in all material respects, the financial position of the Company as of January 27, 2019 and January 28, 2018, and the results of its operations and its cash flows for each of the three years in the period ended January 27, 2019 in conformity with accounting principles generally accepted in the United States of America. Also in our opinion, the Company maintained, in all material respects, effective internal control over financial reporting as of January 27, 2019, based on criteria established in Internal Control - Integrated Framework (2013) issued by the COSO.

Basis for Opinions

The Company's management is responsible for these consolidated financial statements, for maintaining effective internal control over financial reporting, and for its assessment of the effectiveness of internal control over financial reporting, included in Management's Annual Report on Internal Control over Financial Reporting appearing under Item 9A. Our responsibility is to express opinions on the Company's consolidated financial statements and on the Company's internal control over financial reporting based on our audits. We are a public accounting firm registered with the Public Company Accounting Oversight Board (United States) (PCAOB) and are required to be independent with respect to the Company in accordance with the U.S. federal securities laws and the applicable rules and regulations of the Securities and Exchange Commission and the PCAOB.

We conducted our audits in accordance with the standards of the PCAOB. Those standards require that we plan and perform the audits to obtain reasonable assurance about whether the consolidated financial statements are free of material misstatement, whether due to error or fraud, and whether effective internal control over financial reporting was maintained in all material respects.

Our audits of the consolidated financial statements included performing procedures to assess the risks of material misstatement of the consolidated financial statements, whether due to error or fraud, and performing procedures that respond to those risks. Such procedures included examining, on a test basis, evidence regarding the amounts and disclosures in the consolidated financial statements. Our audits also included evaluating the accounting principles used and significant estimates made by management, as well as evaluating the overall presentation of the consolidated financial statements. Our audit of internal control over financial reporting included obtaining an understanding of internal control over financial reporting, assessing the risk that a material weakness exists, and testing and evaluating the design and operating effectiveness of internal control based on the assessed risk. Our audits also included performing such other procedures as we considered necessary in the circumstances. We believe that our audits provide a reasonable basis for our opinions.

Definition and Limitations of Internal Control over Financial Reporting

A company's internal control over financial reporting is a process designed to provide reasonable assurance regarding the reliability of financial reporting and the preparation of financial statements for external purposes in accordance with generally accepted accounting principles. A company's internal control over financial reporting includes those policies and procedures that (i) pertain to the maintenance of records that, in reasonable detail, accurately and fairly reflect the transactions and dispositions of the assets of the company; (ii) provide reasonable assurance that transactions are recorded as necessary to permit preparation of financial statements in accordance with generally accepted accounting principles, and that receipts and expenditures of the company are being made only in accordance with authorizations of management and directors of the company; and (iii) provide reasonable assurance regarding prevention or timely detection of unauthorized acquisition, use, or disposition of the company's assets that could have a material effect on the financial statements.

Because of its inherent limitations, internal control over financial reporting may not prevent or detect misstatements. Also, projections of any evaluation of effectiveness to future periods are subject to the risk that controls may become inadequate because of changes in conditions, or that the degree of compliance with the policies or procedures may deteriorate.

/s/ PricewaterhouseCoopers LLP

San Jose, California

February 21, 2019

We have served as the Company's auditor since 2004.

NVIDIA CORPORATION AND SUBSIDIARIES
CONSOLIDATED STATEMENTS OF INCOME
(In millions, except per share data)

	Year Ended		
	January 27, 2019	January 28, 2018	January 29, 2017
Revenue	$ 11,716	$ 9,714	$ 6,910
Cost of revenue	4,545	3,892	2,847
Gross profit	7,171	5,822	4,063
Operating expenses			
Research and development	2,376	1,797	1,463
Sales, general and administrative	991	815	663
Restructuring and other charges	—	—	3
Total operating expenses	3,367	2,612	2,129
Income from operations	3,804	3,210	1,934
Interest income	136	69	54
Interest expense	(58)	(61)	(58)
Other, net	14	(22)	(25)
Total other income (expense)	92	(14)	(29)
Income before income tax	3,896	3,196	1,905
Income tax expense (benefit)	(245)	149	239
Net income	$ 4,141	$ 3,047	$ 1,666
Net income per share:			
Basic	$ 6.81	$ 5.09	$ 3.08
Diluted	$ 6.63	$ 4.82	$ 2.57
Weighted average shares used in per share computation:			
Basic	608	599	541
Diluted	625	632	649
Cash dividends declared and paid per common share	$ 0.610	$ 0.570	$ 0.485

See accompanying notes to the consolidated financial statements.

NVIDIA CORPORATION AND SUBSIDIARIES
CONSOLIDATED STATEMENTS OF COMPREHENSIVE INCOME
(In millions)

	Year Ended		
	January 27, 2019	January 28, 2018	January 29, 2017
Net income	$ 4,141	$ 3,047	$ 1,666
Other comprehensive income (loss), net of tax			
Available-for-sale debt securities:			
Net unrealized gain (loss)	10	(5)	(17)
Reclassification adjustments for net realized gain included in net income	1	1	1
Net change in unrealized gain (loss)	11	(4)	(16)
Cash flow hedges:			
Net unrealized gain (loss)	6	(1)	2
Reclassification adjustments for net realized gain (loss) included in net income	(11)	3	2
Net change in unrealized gain (loss)	(5)	2	4
Other comprehensive income (loss), net of tax	6	(2)	(12)
Total comprehensive income	$ 4,147	$ 3,045	$ 1,654

See accompanying notes to the consolidated financial statements.

NVIDIA CORPORATION AND SUBSIDIARIES
CONSOLIDATED BALANCE SHEETS
(In millions, except par value)

	January 27, 2019	January 28, 2018
ASSETS		
Current assets:		
Cash and cash equivalents	$ 782	$ 4,002
Marketable securities	6,640	3,106
Accounts receivable, net	1,424	1,265
Inventories	1,575	796
Prepaid expenses and other current assets	136	86
Total current assets	10,557	9,255
Property and equipment, net	1,404	997
Goodwill	618	618
Intangible assets, net	45	52
Other assets	668	319
Total assets	$ 13,292	$ 11,241
LIABILITIES AND SHAREHOLDERS' EQUITY		
Current liabilities:		
Accounts payable	$ 511	$ 596
Accrued and other current liabilities	818	542
Convertible short-term debt	—	15
Total current liabilities	1,329	1,153
Long-term debt	1,988	1,985
Other long-term liabilities	633	632
Total liabilities	3,950	3,770
Commitments and contingencies - see Note 12		
Shareholders' equity:		
Preferred stock, $.001 par value; 2 shares authorized; none issued	—	—
Common stock, $.001 par value; 2,000 shares authorized; 945 shares issued and 606 outstanding as of January 27, 2019; 932 shares issued and 606 outstanding as of January 28, 2018	1	1
Additional paid-in capital	6,051	5,351
Treasury stock, at cost (339 shares in 2019 and 326 shares in 2018)	(9,263)	(6,650)
Accumulated other comprehensive loss	(12)	(18)
Retained earnings	12,565	8,787
Total shareholders' equity	9,342	7,471
Total liabilities and shareholders' equity	$ 13,292	$ 11,241

See accompanying notes to the consolidated financial statements.

NVIDIA CORPORATION AND SUBSIDIARIES
CONSOLIDATED STATEMENTS OF SHAREHOLDERS' EQUITY

(In millions, except per share data)	Common Stock Outstanding Shares	Common Stock Outstanding Amount	Additional Paid-in Capital	Treasury Stock	Accumulated Other Comprehensive Income (Loss)	Retained Earnings	Total Shareholders' Equity
Balances, January 31, 2016	539	$ 1	$ 4,170	$ (4,048)	$ (4)	$ 4,350	$ 4,469
Retained earnings adjustment due to adoption of an accounting standard related to stock-based compensation	—	—	—	—	—	353	353
Other comprehensive loss	—	—	—	—	(12)	—	(12)
Net income	—	—	—	—	—	1,666	1,666
Issuance of common stock in exchange for warrants	44	—	(1)	—	—	—	(1)
Convertible debt conversion	23	—	(6)	—	—	—	(6)
Issuance of common stock from stock plans	20	—	167	—	—	—	167
Tax withholding related to vesting of restricted stock units	(3)	—	—	(177)	—	—	(177)
Share repurchase	(15)	—	—	(739)	—	—	(739)
Exercise of convertible note hedges	(23)	—	75	(75)	—	—	—
Cash dividends declared and paid ($0.485 per common share)	—	—	—	—	—	(261)	(261)
Stock-based compensation	—	—	248	—	—	—	248
Reclassification of convertible debt conversion obligation	—	—	55	—	—	—	55
Balances, January 29, 2017	585	1	4,708	(5,039)	(16)	6,108	5,762
Retained earnings adjustment due to adoption of an accounting standard related to income tax consequences of an intra-entity transfer of an asset	—	—	—	—	—	(27)	(27)
Other comprehensive loss	—	—	—	—	(2)	—	(2)
Net income	—	—	—	—	—	3,047	3,047
Issuance of common stock in exchange for warrants	13	—	—	—	—	—	—
Convertible debt conversion	33	—	(7)	—	—	—	(7)
Issuance of common stock from stock plans	18	—	138	—	—	—	138
Tax withholding related to vesting of restricted stock units	(4)	—	—	(612)	—	—	(612)
Share repurchase	(6)	—	—	(909)	—	—	(909)
Exercise of convertible note hedges	(33)	—	90	(90)	—	—	—
Cash dividends declared and paid ($0.570 per common share)	—	—	—	—	—	(341)	(341)
Stock-based compensation	—	—	391	—	—	—	391
Reclassification of convertible debt conversion obligation	—	—	31	—	—	—	31
Balances, January 28, 2018	606	1	5,351	(6,650)	(18)	8,787	7,471
Retained earnings adjustment due to adoption of new revenue accounting standard	—	—	—	—	—	8	8
Other comprehensive loss	—	—	—	—	6	—	6
Net income	—	—	—	—	—	4,141	4,141
Convertible debt conversion	1	—	—	—	—	—	—
Issuance of common stock from stock plans	13	—	137	—	—	—	137
Tax withholding related to vesting of restricted stock units	(4)	—	—	(1,032)	—	—	(1,032)
Share repurchase	(9)	—	—	(1,579)	—	—	(1,579)
Exercise of convertible note hedges	(1)	—	2	(2)	—	—	—
Cash dividends declared and paid ($0.610 per common share)	—	—	—	—	—	(371)	(371)
Stock-based compensation	—	—	561	—	—	—	561
Balances, January 27, 2019	606	$ 1	$ 6,051	$ (9,263)	$ (12)	$ 12,565	$ 9,342

See accompanying notes to the consolidated financial statements.

NVIDIA CORPORATION AND SUBSIDIARIES
CONSOLIDATED STATEMENTS OF CASH FLOWS
(In millions)

	Year Ended		
	January 27, 2019	January 28, 2018	January 29, 2017
Cash flows from operating activities:			
Net income	$ 4,141	$ 3,047	$ 1,666
Adjustments to reconcile net income to net cash provided by operating activities:			
Stock-based compensation expense	557	391	247
Depreciation and amortization	262	199	187
Deferred income taxes	(315)	(359)	197
Loss on early debt conversions	—	19	21
Other	(45)	20	33
Changes in operating assets and liabilities:			
Accounts receivable	(149)	(440)	(321)
Inventories	(776)	—	(375)
Prepaid expenses and other assets	(55)	21	(18)
Accounts payable	(135)	90	184
Accrued and other current liabilities	256	33	(135)
Other long-term liabilities	2	481	(14)
Net cash provided by operating activities	3,743	3,502	1,672
Cash flows from investing activities:			
Proceeds from maturities of marketable securities	7,232	1,078	969
Proceeds from sales of marketable securities	428	863	1,546
Purchases of marketable securities	(11,148)	(36)	(3,134)
Purchases of property and equipment and intangible assets	(600)	(593)	(176)
Investment in non-affiliates	(9)	(36)	(5)
Proceeds from sale of long-lived assets and investments	—	2	7
Net cash provided by (used in) investing activities	(4,097)	1,278	(793)
Cash flows from financing activities:			
Proceeds from issuance of debt	—	—	1,988
Payments related to repurchases of common stock	(1,579)	(909)	(739)
Repayment of Convertible Notes	(16)	(812)	(673)
Dividends paid	(371)	(341)	(261)
Proceeds related to employee stock plans	137	139	167
Payments related to tax on restricted stock units	(1,032)	(612)	(176)
Other	(5)	(9)	(15)
Net cash provided by (used in) financing activities	(2,866)	(2,544)	291
Change in cash and cash equivalents	(3,220)	2,236	1,170
Cash and cash equivalents at beginning of period	4,002	1,766	596
Cash and cash equivalents at end of period	$ 782	$ 4,002	$ 1,766

	Year Ended		
	January 27, 2019	January 28, 2018	January 29, 2017
Supplemental disclosures of cash flow information:			
Cash paid for income taxes, net	$ 61	$ 22	$ 14
Cash paid for interest	$ 55	$ 55	$ 13
Non-cash investing and financing activity:			
Assets acquired by assuming related liabilities	$ 76	$ 36	$ 16

See accompanying notes to the consolidated financial statements.

NVIDIA CORPORATION AND SUBSIDIARIES

NOTES TO THE CONSOLIDATED FINANCIAL STATEMENTS

Note 1 - Organization and Summary of Significant Accounting Policies

Our Company

Headquartered in Santa Clara, California, NVIDIA was incorporated in California in April 1993 and reincorporated in Delaware in April 1998.

All references to "NVIDIA," "we," "us," "our" or the "Company" mean NVIDIA Corporation and its subsidiaries.

Fiscal Year

We operate on a 52- or 53-week year, ending on the last Sunday in January. Fiscal years 2019, 2018 and 2017 were 52-week years.

Reclassifications

Certain prior fiscal year balances have been reclassified to conform to the current fiscal year presentation.

Principles of Consolidation

Our consolidated financial statements include the accounts of NVIDIA Corporation and our wholly-owned subsidiaries. All intercompany balances and transactions have been eliminated in consolidation.

Use of Estimates

The preparation of financial statements in conformity with U.S. GAAP requires management to make estimates and assumptions that affect the reported amounts of assets and liabilities and disclosures of contingent assets and liabilities at the date of the financial statements and the reported amounts of revenue and expenses during the reporting period. Actual results could differ materially from our estimates. On an on-going basis, we evaluate our estimates, including those related to revenue recognition, cash equivalents and marketable securities, accounts receivable, inventories, income taxes, goodwill, stock-based compensation, litigation, investigation and settlement costs, restructuring and other charges, and other contingencies. These estimates are based on historical facts and various other assumptions that we believe are reasonable.

Revenue Recognition

We derive our revenue from product sales, including hardware and systems, license and development arrangements, and software licensing. We determine revenue recognition through the following steps: (1) identification of the contract with a customer; (2) identification of the performance obligations in the contract; (3) determination of the transaction price; (4) allocation of the transaction price to the performance obligations in the contract; and (5) recognition of revenue when, or as, we satisfy a performance obligation.

Product Sales Revenue

Revenue from product sales is recognized upon transfer of control of promised products to customers in an amount that reflects the consideration we expect to receive in exchange for those products. Revenue is recognized net of allowances for returns, customer programs and any taxes collected from customers.

For products sold with a right of return, we record a reduction to revenue by establishing a sales return allowance for estimated product returns at the time revenue is recognized, based primarily on historical return rates. However, if product returns for a fiscal period are anticipated to exceed historical return rates, we may determine that additional sales return allowances are required to properly reflect our estimated exposure for product returns.

Our customer programs involve rebates, which are designed to serve as sales incentives to resellers of our products in various target markets, and marketing development funds, or MDFs, which represent monies paid to our partners that are earmarked for market segment development and are designed to support our partners' activities while also promoting NVIDIA products. We account for customer programs as a reduction to revenue and accrue for potential rebates and MDFs based on the amount we expect to be claimed by customers.

NVIDIA CORPORATION AND SUBSIDIARIES
NOTES TO THE CONSOLIDATED FINANCIAL STATEMENTS
(Continued)

License and Development Arrangements

Our license and development arrangements with customers typically require significant customization of our intellectual property components. As a result, we recognize the revenue from the license and the revenue from the development services as a single performance obligation over the period in which the development services are performed. We measure progress to completion based on actual cost incurred to date as a percentage of the estimated total cost required to complete each project. If a loss on an arrangement becomes probable during a period, we record a provision for such loss in that period.

Software Licensing

Our software licenses provide our customers with a right to use the software when it is made available to the customer. Customers may purchase either perpetual licenses or subscriptions to licenses, which differ mainly in the duration over which the customer benefits from the software. Software licenses are frequently sold along with post-contract customer support, or PCS. For such arrangements, we allocate revenue to the software license and PCS on a relative standalone selling price basis by maximizing the use of observable inputs to determine the standalone selling price for each performance obligation. Revenue from software licenses is recognized up front when the software is made available to the customer. PCS revenue is recognized ratably over the service period, or as services are performed.

Advertising Expenses

We expense advertising costs in the period in which they are incurred. Advertising expenses for fiscal years 2019 , 2018 , and 2017 were $ 21 million , $25 million , and $17 million , respectively.

Product Warranties

We generally offer a limited warranty to end-users that ranges from one to three years for products in order to repair or replace products for any manufacturing defects or hardware component failures. Cost of revenue includes the estimated cost of product warranties that are calculated at the point of revenue recognition. Under limited circumstances, we may offer an extended limited warranty to customers for certain products. We also accrue for known warranty and indemnification issues if a loss is probable and can be reasonably estimated.

Stock-based Compensation

We use the closing trading price of our common stock on the date of grant, minus a dividend yield discount, as the fair value of awards of restricted stock units, or RSUs, and performance stock units that are based on our corporate financial performance targets, or PSUs. We use a Monte Carlo simulation on the date of grant to estimate the fair value of performance stock units that are based on market conditions, or market-based PSUs. The compensation expense for RSUs and market-based PSUs is recognized using a straight-line attribution method over the requisite employee service period while compensation expense for PSUs is recognized using an accelerated amortization model. We estimate the fair value of shares to be issued under our employee stock purchase plan, or ESPP, using the Black-Scholes model at the commencement of an offering period in March and September of each year. Stock-based compensation for our ESPP is expensed using an accelerated amortization model. Additionally, we estimate forfeitures annually based on historical experience and revise the estimates of forfeiture in subsequent periods if actual forfeitures differ from those estimates.

Litigation, Investigation and Settlement Costs

From time to time, we are involved in legal actions and/or investigations by regulatory bodies. There are many uncertainties associated with any litigation or investigation, and we cannot be certain that these actions or other third-party claims against us will be resolved without litigation, fines and/or substantial settlement payments. If information becomes available that causes us to determine that a loss in any of our pending litigation, investigations or settlements is probable, and we can reasonably estimate the loss associated with such events, we will record the loss in accordance with U.S. GAAP. However, the actual liability in any such litigation or investigation may be materially different from our estimates, which could require us to record additional costs.

Foreign Currency Remeasurement

We use the United States dollar as our functional currency for all of our subsidiaries. Foreign currency monetary assets and liabilities are remeasured into United States dollars at end-of-period exchange rates. Non-monetary assets and liabilities such as property and equipment, and equity are remeasured at historical exchange rates. Revenue and expenses are remeasured at average exchange rates in effect during each period, except for those expenses related to the previously noted balance sheet amounts, which are remeasured at historical exchange rates. Gains or losses from foreign currency

remeasurement are included in other income or expense in our Consolidated Statements of Income and to date have not been significant.

Income Taxes

We recognize federal, state and foreign current tax liabilities or assets based on our estimate of taxes payable or refundable in the current fiscal year by tax jurisdiction. We recognize federal, state and foreign deferred tax assets or liabilities, as appropriate, for our estimate of future tax effects attributable to temporary differences and carryforwards; and we record a valuation allowance to reduce any deferred tax assets by the amount of any tax benefits that, based on available evidence and judgment, are not expected to be realized.

Our calculation of deferred tax assets and liabilities is based on certain estimates and judgments and involves dealing with uncertainties in the application of complex tax laws. Our estimates of deferred tax assets and liabilities may change based, in part, on added certainty or finality to an anticipated outcome, changes in accounting standards or tax laws in the United States, or foreign jurisdictions where we operate, or changes in other facts or circumstances. In addition, we recognize liabilities for potential United States and foreign income tax contingencies based on our estimate of whether, and the extent to which, additional taxes may be due. If we determine that payment of these amounts is unnecessary or if the recorded tax liability is less than our current assessment, we may be required to recognize an income tax benefit or additional income tax expense in our financial statements accordingly.

As of January 27, 2019 , we had a valuation allowance of $562 million related to state and certain foreign deferred tax assets that management determined are not likely to be realized due to projections of future taxable income and potential utilization limitations of tax attributes acquired as a result of stock ownership changes. To the extent realization of the deferred tax assets becomes more-likely-than-not, we would recognize such deferred tax asset as an income tax benefit during the period.

We recognize the benefit from a tax position only if it is more-likely-than-not that the position would be sustained upon audit based solely on the technical merits of the tax position. Our policy is to include interest and penalties related to unrecognized tax benefits as a component of income tax expense.

The Tax Cuts and Jobs Act, or TCJA, which was enacted in December 2017, significantly changes U.S. tax law, including a reduction of the U.S. federal corporate income tax rate from 35% to 21%, a requirement for companies to pay a one-time transition tax on the earnings of certain foreign subsidiaries that were previously tax deferred, and the creation of new taxes (global intangible low-taxed income, or GILTI) on certain foreign-source earnings. As a fiscal year-end taxpayer, certain provisions of the TCJA began to impact us in the fourth quarter of fiscal year 2018, while other provisions impacted us beginning in fiscal year 2019. The Securities and Exchange Commission, or the SEC, had provided guidance in Staff Accounting Bulletin No. 118, Income Tax Accounting Implications of the Tax Cuts and Jobs Act (SAB 118), which allowed companies to record provisional amounts during a measurement period up to one year from the enactment date. As of January 27, 2019, we completed our accounting for all of the enactment-date income tax effects of the TCJA and elected to account for GILTI in deferred taxes. Refer to Note 13 of these Notes to the Consolidated Financial Statements for additional information.

Net Income Per Share

Basic net income per share is computed using the weighted average number of common shares outstanding during the period. Diluted net income per share is computed using the weighted average number of common and potentially dilutive shares outstanding during the period, using the treasury stock method. Under the treasury stock method, the effect of equity awards outstanding is not included in the computation of diluted net income per share for periods when their effect is anti-dilutive. Additionally, we issued convertible notes with a net settlement feature that required us, upon conversion, to settle the principal amount of debt for cash and the conversion premium for cash or shares of our common stock. Our Convertible Notes, Note Hedges, and related Warrants contained various conversion features, which are further described in Note 11 of these Notes to the Consolidated Financial Statements. The potentially dilutive shares resulting from the Convertible Notes and Warrants under the treasury stock method were included in the calculation of diluted income per share when their inclusion was dilutive. However, the Note Hedges were not included in the calculation of diluted net income per share unless actually exercised, as their pre-exercised effect would have been anti-dilutive under the treasury stock method.

NVIDIA CORPORATION AND SUBSIDIARIES
NOTES TO THE CONSOLIDATED FINANCIAL STATEMENTS
(Continued)

Cash and Cash Equivalents

We consider all highly liquid investments that are readily convertible into cash and have an original maturity of three months or less at the time of purchase to be cash equivalents.

Marketable Securities

Marketable securities consist of highly liquid debt investments with maturities of greater than three months when purchased. We generally classify our marketable securities at the date of acquisition as available-for-sale. These debt securities are reported at fair value with the related unrealized gains and losses included in accumulated other comprehensive income or loss, a component of shareholders' equity, net of tax. The fair value of interest-bearing debt securities includes accrued interest. Any unrealized losses which are considered to be other-than-temporary impairments are recorded in the other income or expense, net, section of our Consolidated Statements of Income. Realized gains and losses on the sale of marketable securities are determined using the specific-identification method and recorded in the other income or expense, net, section of our Consolidated Statements of Income.

All of our available-for-sale debt investments are subject to a periodic impairment review. We record a charge to earnings when a decline in fair value is significantly below cost basis and judged to be other-than-temporary or have other indicators of impairments. If the fair value of an available-for-sale debt instrument is less than its amortized cost basis, an other-than-temporary impairment is triggered in circumstances where (1) we intend to sell the instrument, (2) it is more likely than not that we will be required to sell the instrument before recovery of its amortized cost basis, or (3) a credit loss exists where we do not expect to recover the entire amortized cost basis of the instrument. In these situations, we recognize an other-than-temporary impairment in earnings equal to the entire difference between the debt instruments' amortized cost basis and its fair value. For available-for-sale debt instruments that are considered other-than-temporarily impaired due to the existence of a credit loss, if we do not intend to sell and it is not more likely than not that we will not be required to sell the instrument before recovery of its remaining amortized cost basis (amortized cost basis less any current-period credit loss), we separate the amount of the impairment into the amount that is credit related and the amount due to all other factors. The credit loss component is recognized in earnings while loss related to all other factors is recorded in accumulated other comprehensive income or loss.

Fair Value of Financial Instruments

The carrying value of cash equivalents, accounts receivable, accounts payable and accrued liabilities approximate their fair values due to their relatively short maturities as of January 27, 2019 and January 28, 2018 . Marketable securities are comprised of available-for-sale securities that are reported at fair value with the related unrealized gains or losses included in accumulated other comprehensive income or loss, a component of shareholders' equity, net of tax. Fair value of the marketable securities is determined based on quoted market prices. Derivative instruments are recognized as either assets or liabilities and are measured at fair value. The accounting for changes in the fair value of a derivative depends on the intended use of the derivative and the resulting designation. For derivative instruments designated as fair value hedges, the gains or losses are recognized in earnings in the periods of change together with the offsetting losses or gains on the hedged items attributed to the risk being hedged. For derivative instruments designated as cash-flow hedges, the effective portion of the gains or losses on the derivatives is initially reported as a component of other comprehensive income or loss and is subsequently recognized in earnings when the hedged exposure is recognized in earnings. For derivative instruments not designated for hedge accounting, changes in fair value are recognized in earnings.

Concentration of Credit Risk

Financial instruments that potentially subject us to concentrations of credit risk consist primarily of cash equivalents, marketable securities, and accounts receivable. Our investment policy requires the purchase of highly-rated fixed income securities, the diversification of investment type and credit exposures, and includes certain limits on our portfolio duration. Accounts receivable from significant customers, those representing 10% or more of total accounts receivable, aggregated approximately 19% of our accounts receivable balance from one customer as of January 27, 2019 and 28% of our account receivable balance from two customers as of January 28, 2018 . We perform ongoing credit evaluations of our customers' financial condition and maintain an allowance for potential credit losses. This allowance consists of an amount identified for specific customers and an amount based on overall estimated exposure. Our overall estimated exposure excludes amounts covered by credit insurance and letters of credit.

NVIDIA CORPORATION AND SUBSIDIARIES
NOTES TO THE CONSOLIDATED FINANCIAL STATEMENTS
(Continued)

Accounts Receivable

We maintain an allowance for doubtful accounts receivable for estimated losses resulting from the inability of our customers to make required payments. We determine this allowance by identifying amounts for specific customer issues as well as amounts based on overall estimated exposure. Factors impacting the allowance include the level of gross receivables, the financial condition of our customers and the extent to which balances are covered by credit insurance or letters of credit.

Inventories

Inventory cost is computed on an adjusted standard basis, which approximates actual cost on an average or first-in, first-out basis. Inventory costs consist primarily of the cost of semiconductors purchased from subcontractors, including wafer fabrication, assembly, testing and packaging, manufacturing support costs, including labor and overhead associated with such purchases, final test yield fallout, and shipping costs, as well as the cost of purchased memory products and other component parts. We charge cost of sales for inventory provisions to write down our inventory to the lower of cost or net realizable value or to completely write off obsolete or excess inventory. Most of our inventory provisions relate to the write-off of excess quantities of products, based on our inventory levels and future product purchase commitments compared to assumptions about future demand and market conditions. Once inventory has been written-off or written-down, it creates a new cost basis for the inventory that is not subsequently written-up.

Property and Equipment

Property and equipment are stated at cost. Depreciation of property and equipment is computed using the straight-line method based on the estimated useful lives of the assets, generally three to five years. Once an asset is identified for retirement or disposition, the related cost and accumulated depreciation or amortization are removed, and a gain or loss is recorded. The estimated useful lives of our buildings are up to thirty years. Depreciation expense includes the amortization of assets recorded under capital leases. Leasehold improvements and assets recorded under capital leases are amortized over the shorter of the expected lease term or the estimated useful life of the asset.

Goodwill

Goodwill is subject to our annual impairment test during the fourth quarter of our fiscal year, or earlier if indicators of potential impairment exist. For the purposes of completing our impairment test, we perform either a qualitative or a quantitative analysis on a reporting unit basis.

Qualitative factors include industry and market considerations, overall financial performance, and other relevant events and factors affecting the reporting units.

Our quantitative impairment test considers both the income approach and the market approach to estimate a reporting unit's fair value. The income and market valuation approaches consider a number of factors that include, but are not limited to, prospective financial information, growth rates, residual values, discount rates and comparable multiples from publicly traded companies in our industry and require us to make certain assumptions and estimates regarding industry economic factors and the future profitability of our business. Refer to Note 5 of these Notes to the Consolidated Financial Statements for additional information.

Intangible Assets and Other Long-Lived Assets

Intangible assets primarily represent rights acquired under technology licenses, patents, acquired intellectual property, trademarks and customer relationships. We currently amortize our intangible assets with definitive lives over periods ranging from three to ten years using a method that reflects the pattern in which the economic benefits of the intangible asset are consumed or otherwise used up or, if that pattern cannot be reliably determined, using a straight-line amortization method.

Long-lived assets, such as property and equipment and intangible assets subject to amortization are reviewed for impairment whenever events or changes in circumstances indicate that the carrying amount of an asset or asset group may not be recoverable. Recoverability of assets to be held and used is measured by a comparison of the carrying amount of an asset or asset group to estimated undiscounted future cash flows expected to be generated by the asset, or asset group. If the carrying amount of an asset or asset group exceeds its estimated future cash flows, an impairment charge is recognized for the amount by which the carrying amount of the asset or asset group exceeds the estimated fair value of the asset or asset group. Fair value is determined based on the estimated discounted future cash flows expected to be generated by the asset or asset group. Assets and liabilities to be disposed of would be separately presented in the

Consolidated Balance Sheet and the assets would be reported at the lower of the carrying amount or fair value less costs to sell, and would no longer be depreciated.

Adoption of New and Recently Issued Accounting Pronouncements

Recently Adopted Accounting Pronouncements

The Financial Accounting Standards Board, or FASB, issued an accounting standards update that creates a single source of revenue guidance under U.S. GAAP for all companies, in all industries. We adopted this guidance on January 29, 2018 using the modified retrospective approach. Refer to Note 2 of these Notes to the Consolidated Financial Statements for additional information.

In January 2016, the FASB issued an accounting standards update to amend certain aspects of recognition, measurement, presentation, and disclosure of financial instruments. We are now required to recognize changes in the fair value of our equity investments through net income rather than other comprehensive income. We adopted this guidance in the first quarter of fiscal year 2019 and applied it prospectively. The adoption of this guidance did not have a significant impact on our Consolidated Financial Statements.

Recent Accounting Pronouncements Not Yet Adopted

The FASB issued an accounting standards update regarding the accounting for leases under which we will begin recognizing lease assets and liabilities on the balance sheet for lease terms of more than 12 months. We will adopt this guidance using the optional transition method at the beginning of fiscal year 2020 and will not restate comparative prior periods. Additionally, we will elect the package of practical expedients as permitted by the guidance. We are in the process of finalizing changes to our systems and processes in conjunction with our review of lease agreements and currently expect the adoption of this accounting guidance to result in an increase in lease assets and a corresponding increase in lease liabilities on our Consolidated Balance Sheet of approximately $500 million .

In June 2016, the FASB issued a new accounting standard to replace the incurred loss impairment methodology under current GAAP with a methodology that reflects expected credit losses and requires consideration of a broader range of reasonable and supportable information to inform credit loss estimates. We will be required to use a forward-looking expected credit loss model for accounts receivable and other financial instruments, including available-for-sale debt securities. The standard will be effective for us beginning in the first quarter of fiscal year 2021, with early adoption permitted. We are currently evaluating the impact of this standard on our Consolidated Financial Statements.

Note 2 - New Revenue Accounting Standard

Method and Impact of Adoption

On January 29, 2018, we adopted the new revenue accounting standard using the modified retrospective method and applied it to contracts that were not completed as of that date. Upon adoption, we recognized the cumulative effect of the new standard as a $7 million increase to opening retained earnings, net of tax. Comparative information for prior periods has not been adjusted. The impact of the new standard on our consolidated financial statements for fiscal year 2019 was not significant.

Deferred Revenue and Performance Obligations

Deferred revenue is comprised mainly of customer advances and deferrals related to license and development arrangements and PCS related to software licensing. The following table shows the changes in deferred revenue during fiscal year 2019:

NVIDIA CORPORATION AND SUBSIDIARIES
NOTES TO THE CONSOLIDATED FINANCIAL STATEMENTS
(Continued)

	January 27, 2019
	(in millions)
Balance as of January 28, 2018	$ 68
Adjustment to retained earnings upon adoption of new revenue standard	(5)
Balance as of January 29, 2018	63
Deferred revenue added during the period	344
Revenue recognized during the period	(269)
Balance as of January 27, 2019	$ 138

Revenue related to remaining performance obligations represents the amount of contracted license and development arrangements and PCS that has not been recognized. As of January 27, 2019, the amount of our remaining performance obligations that has not been recognized as revenue was $305 million , of which we expect to recognize approximately 50% as revenue over the next twelve months and the remainder thereafter. This amount excludes the value of remaining performance obligations for contracts with an original expected length of one year or less.

Refer to Note 16 of these Notes to the Consolidated Financial Statements for additional information, including disaggregated revenue disclosures.

Note 3 - Stock-Based Compensation

Our stock-based compensation expense is associated with restricted stock units, or RSUs, performance stock units that are based on our corporate financial performance targets, or PSUs, performance stock units that are based on market conditions, or market-based PSUs, and our ESPP.

Our Consolidated Statements of Income include stock-based compensation expense, net of amounts allocated to inventory, as follows:

	Year Ended		
	January 27, 2019	January 28, 2018	January 29, 2017
	(In millions)		
Cost of revenue	$ 27	$ 21	$ 15
Research and development	336	219	134
Sales, general and administrative	194	151	98
Total	$ 557	$ 391	$ 247

Stock-based compensation capitalized in inventories was not significant during fiscal years 2019 , 2018 , and 2017 .

The following is a summary of equity awards granted under our equity incentive plans:

	Year Ended		
	January 27, 2019	January 28, 2018	January 29, 2017
	(In millions, except per share data)		
RSUs, PSUs and Market-based PSUs			
Awards granted	4	6	12
Estimated total grant-date fair value	$ 1,109	$ 929	$ 591
Weighted average grant-date fair value (per share)	$ 258.26	$ 145.91	$ 50.57
ESPP			
Shares purchased	1	5	4
Weighted average price (per share)	$ 107.48	$ 21.24	$ 18.51
Weighted average grant-date fair value (per share)	$ 38.51	$ 7.12	$ 5.80

NVIDIA CORPORATION AND SUBSIDIARIES
NOTES TO THE CONSOLIDATED FINANCIAL STATEMENTS
(Continued)

Beginning fiscal year 2015, we shifted away from granting stock options and toward granting RSUs, PSUs and market-based PSUs to reflect changing market trends for equity incentives at our peer companies. The number of PSUs that will ultimately vest is contingent on the Company's level of achievement versus the corporate financial performance target established by our Compensation Committee in the beginning of each fiscal year.

Of the total fair value of equity awards, we estimated that the stock-based compensation expense related to the equity awards that are not expected to vest for fiscal year 2019 was $88 million .

	January 27, 2019	January 28, 2018
	(In millions)	
Aggregate unearned stock-based compensation expense	$ 1,580	$ 1,091

Estimated weighted average remaining amortization period	*(In years)*	
RSUs, PSUs and market-based PSUs	2.2	2.3
ESPP	0.8	0.7

The fair value of shares issued under our ESPP have been estimated with the following assumptions:

	Year Ended		
	January 27, 2019	January 28, 2018	January 29, 2017
	(Using the Black-Scholes model)		
ESPP			
Weighted average expected life (in years)	0.1-2.0	0.5-2.0	0.5-2.0
Risk-free interest rate	1.6%-2.8%	0.8%-1.4%	0.5%-0.9%
Volatility	24%-75%	40%-54%	30%-39%
Dividend yield	0.3%-0.4%	0.3%-0.5%	0.7%-1.4%

For ESPP shares, the expected term represents the average term from the first day of the offering period to the purchase date. The risk-free interest rate assumption used to value ESPP shares is based upon observed interest rates on Treasury bills appropriate for the expected term. Our expected stock price volatility assumption for ESPP is estimated using historical volatility. For awards granted, we use the dividend yield at grant date. Our RSU, PSU, and market-based PSU awards are not eligible for cash dividends prior to vesting; therefore, the fair values of RSUs, PSUs, and market-based PSUs are discounted for the dividend yield.

Additionally, for RSU, PSU, and market-based PSU awards, we estimate forfeitures annually and revise the estimates of forfeiture in subsequent periods if actual forfeitures differ from those estimates. Forfeitures are estimated based on historical experience.

Equity Incentive Program

We grant or have granted stock options, RSUs, PSUs, market-based PSUs, and stock purchase rights under the following equity incentive plans.

Amended and Restated 2007 Equity Incentive Plan

In 2007, our shareholders approved the NVIDIA Corporation 2007 Equity Incentive Plan, as most recently amended and restated, the 2007 Plan.

The 2007 Plan authorizes the issuance of incentive stock options, non-statutory stock options, restricted stock, restricted stock units, stock appreciation rights, performance stock awards, performance cash awards, and other stock-based awards to employees, directors and consultants. Only our employees may receive incentive stock options. Up to 230 million shares of our common stock may be issued pursuant to stock awards granted under the 2007 Plan. Currently, we grant RSUs, PSUs and market-based PSUs under the 2007 Plan, under which, as of January 27, 2019 , there were 35 million shares available for future issuance.

NVIDIA CORPORATION AND SUBSIDIARIES
NOTES TO THE CONSOLIDATED FINANCIAL STATEMENTS
(Continued)

Stock options previously granted to employees, subject to certain exceptions, vest over a four -year period, subject to continued service, with 25% vesting on the anniversary of the hire date in the case of new hires or the anniversary of the date of grant in the case of grants to existing employees and 6.25% vesting quarterly thereafter. These stock options generally expire ten years from the date of grant.

Subject to certain exceptions, RSUs and PSUs granted to employees vest over a four -year period, subject to continued service, with 25% vesting on a pre-determined date that is close to the anniversary of the date of grant and (i) for grants made prior to May 18, 2016, 12.5% vesting semi-annually thereafter, and (ii) for grants made on or after May 18, 2016, 6.25% vesting quarterly thereafter. Market-based PSUs vest 100% on approximately the three -year anniversary of the date of grant. However, the number of shares subject to both PSUs and market-based PSUs that are eligible to vest is generally determined by the Compensation Committee based on achievement of pre-determined criteria.

Unless terminated sooner, the 2007 Plan is scheduled to terminate on March 21, 2022. Our Board may suspend or terminate the 2007 Plan at any time. No awards may be granted under the 2007 Plan while the 2007 Plan is suspended or after it is terminated. The Board may also amend the 2007 Plan at any time. However, if legal, regulatory or listing requirements require shareholder approval, the amendment will not go into effect until the shareholders have approved the amendment.

Amended and Restated 2012 Employee Stock Purchase Plan

In 2012, our shareholders approved the 2012 Employee Stock Purchase Plan, as most recently amended and restated, the 2012 Plan, as the successor to the 1998 Employee Stock Purchase Plan.

Up to 89 million shares of our common stock may be issued pursuant to purchases under the 2012 Plan. As of January 27, 2019 , we had issued 29 million shares and reserved 60 million shares for future issuance under the 2012 Plan.

The 2012 Plan is intended to qualify as an “employee stock purchase plan” under Section 423 of the Internal Revenue Code. Under the current offerings adopted pursuant to the 2012 Plan, each offering period is approximately 24 months, which is generally divided into four purchase periods of six months.

Employees are eligible to participate if they are employed by us or an affiliate of us as designated by the Board. Employees who participate in an offering may have up to 10% of their earnings withheld up to certain limitations and applied on specified dates determined by the Board to the purchase of shares of common stock. The Board may increase this percentage at its discretion, up to 15% . The price of common stock purchased under our 2012 Plan will be equal to 85% of the lower of the fair market value of the common stock on the commencement date of each offering period and the fair market value on each purchase date within the offering. Employees may end their participation in the 2012 Plan at any time during the offering period, and participation ends automatically on termination of employment with us. In each case, the employee’s contributions are refunded.

The following is a summary of our equity award transactions under our equity incentive plans:

	RSUs, PSUs and Market-based PSUs Outstanding	
	Number of Shares	Weighted Average Grant-Date Fair Value
	(In millions, except years and per share data)	
Balances, January 28, 2018	22	$ 66.72
Granted (1)(2)	4	$ 258.26
Vested restricted stock	(10)	$ 52.56
Canceled and forfeited	—	$ —
Balances, January 27, 2019	16	$ 129.92
Vested and expected to vest after January 27, 2019	13	$ 129.44

(1) Includes PSUs that will be issued and eligible to vest based on the corporate financial performance level achieved for fiscal year 2019 .

(2) Includes market-based PSUs that will be issued and eligible to vest if the maximum target for total shareholder return, or TSR, over the 3 -year measurement period is achieved. Depending on the ranking of our TSR compared to the respective TSRs of the companies comprising the Standard & Poor’s 500 Index during that period, the market-based PSUs issued could be up to 45 thousand shares.

As of January 27, 2019 and January 28, 2018 , there were 35 million and 16 million shares, respectively, of common stock reserved for future issuance under our equity incentive plans.

NVIDIA CORPORATION AND SUBSIDIARIES
NOTES TO THE CONSOLIDATED FINANCIAL STATEMENTS
(Continued)

The total intrinsic value of options exercised was $180 million , $318 million , and $246 million for fiscal years 2019 , 2018 , and 2017 , respectively. Upon exercise of an option, we issue new shares of stock.

Note 4 - Net Income Per Share

The following is a reconciliation of the denominator of the basic and diluted net income per share computations for the periods presented:

	Year Ended		
	January 27, 2019	January 28, 2018	January 29, 2017
	(In millions, except per share data)		
Numerator:			
Net income	$ 4,141	$ 3,047	$ 1,666
Denominator:			
Basic weighted average shares	608	599	541
Dilutive impact of outstanding securities:			
Equity awards	17	24	26
1.00% Convertible Senior Notes	—	5	44
Warrants issued with the 1.00% Convertible Senior Notes	—	4	38
Diluted weighted average shares	625	632	649
Net income per share:			
Basic (1)	$ 6.81	$ 5.09	$ 3.08
Diluted (2)	$ 6.63	$ 4.82	$ 2.57
Equity awards excluded from diluted net income per share because their effect would have been anti-dilutive	5	4	8

(1) Calculated as net income divided by basic weighted average shares.

(2) Calculated as net income divided by diluted weighted average shares.

The 1.00% Convertible Senior Notes Due 2018, or the Convertible Notes, were included in the calculation of diluted net income per share. The Convertible Notes had a dilutive impact on net income per share if our average stock price for the reporting period exceeded the adjusted conversion price of $20.02 per share. The warrants associated with our Convertible Notes, or the Warrants, outstanding were also included in the calculation of diluted net income per share. As of January 27, 2019 , there were no Convertible Notes or Warrants outstanding.

Refer to Note 11 of these Notes to the Consolidated Financial Statements for additional discussion regarding the Convertible Notes, Note Hedges, and Warrants.

Note 5 - Goodwill

The carrying amount of goodwill was $618 million , and the amount of goodwill allocated to our GPU and Tegra Processor reporting units was $210 million and $408 million , respectively, as of both January 27, 2019 and January 28, 2018 . There were no changes to the carrying amount of goodwill during fiscal years 2019 and 2018. During the fourth quarters of fiscal years 2019, 2018, and 2017, we completed our annual impairment tests and concluded that goodwill was no t impaired in any of these years.

Note 6 - Amortizable Intangible Assets

The components of our amortizable intangible assets are as follows:

	January 27, 2019			January 28, 2018		
	Gross Carrying Amount	Accumulated Amortization	Net Carrying Amount	Gross Carrying Amount	Accumulated Amortization	Net Carrying Amount
	(In millions)			(In millions)		
Acquisition-related intangible assets	$ 195	$ (188)	$ 7	$ 195	$ (180)	$ 15
Patents and licensed technology	491	(453)	38	469	(432)	37
Total intangible assets	$ 686	$ (641)	$ 45	$ 664	$ (612)	$ 52

The increase in gross carrying amount of intangible assets is due to purchases of licensed technology during fiscal year 2019. Amortization expense associated with intangible assets for fiscal years 2019 , 2018 , and 2017 was $29 million , $55 million , and $68 million , respectively. Future amortization expense related to the net carrying amount of intangible assets as of January 27, 2019 is estimated to be $21 million in fiscal year 2020 , $12 million in fiscal year 2021 , $5 million in fiscal year 2022 , and $5 million in fiscal year 2023 , and $2 million in fiscal year 2024.

Note 7 - Marketable Securities

Our cash equivalents and marketable securities are classified as “available-for-sale” debt securities.

The following is a summary of cash equivalents and marketable securities as of January 27, 2019 and January 28, 2018 :

	January 27, 2019					
					Reported as	
	Amortized Cost	Unrealized Gain	Unrealized Loss	Estimated Fair Value	Cash Equivalents	Marketable Securities
			(In millions)			
Corporate debt securities	$ 2,626	$ —	$ (6)	$ 2,620	$ 25	$ 2,595
Debt securities of United States government agencies	2,284	—	(4)	2,280	—	2,280
Debt securities issued by the United States Treasury	1,493	—	(1)	1,492	176	1,316
Money market funds	483	—	—	483	483	—
Foreign government bonds	209	—	—	209	—	209
Asset-backed securities	152	—	(1)	151	—	151
Mortgage-backed securities issued by United States government-sponsored enterprises	88	1	—	89	—	89
Total	$ 7,335	$ 1	$ (12)	$ 7,324	$ 684	$ 6,640

NVIDIA CORPORATION AND SUBSIDIARIES
NOTES TO THE CONSOLIDATED FINANCIAL STATEMENTS
(Continued)

	January 28, 2018					
					Reported as	
	Amortized Cost	Unrealized Gain	Unrealized Loss	Estimated Fair Value	Cash Equivalents	Marketable Securities
	(In millions)					
Money market funds	$ 3,789	$ —	$ —	$ 3,789	$ 3,789	$ —
Corporate debt securities	1,304	—	(9)	1,295	—	1,295
Debt securities of United States government agencies	822	—	(7)	815	—	815
Debt securities issued by the United States Treasury	577	—	(4)	573	—	573
Asset-backed securities	254	—	(2)	252	—	252
Mortgage backed securities issued by United States government-sponsored enterprises	128	2	—	130	—	130
Foreign government bonds	42	—	(1)	41	—	41
Total	$ 6,916	$ 2	$ (23)	$ 6,895	$ 3,789	$ 3,106

The following table provides the breakdown of unrealized losses as of January 27, 2019 , aggregated by investment category and length of time that individual securities have been in a continuous loss position:

	Less than 12 Months		12 Months or Greater		Total	
	Estimated Fair Value	Gross Unrealized Losses	Estimated Fair Value	Gross Unrealized Losses	Estimated Fair Value	Gross Unrealized Losses
	(In millions)					
Debt securities issued by United States government agencies	$ 1,674	$ (1)	$ 401	$ (3)	$ 2,075	$ (4)
Corporate debt securities	915	(3)	649	(3)	1,564	(6)
Debt securities issued by the United States Treasury	1,015	—	161	(1)	1,176	(1)
Asset-backed securities	—	—	151	(1)	151	(1)
Total	$ 3,604	$ (4)	$ 1,362	$ (8)	$ 4,966	$ (12)

The gross unrealized losses are related to fixed income securities, temporary in nature, and driven primarily by changes in interest rates. We have the intent and ability to hold our investments until maturity. For fiscal years 2019 , 2018 , and 2017 , there were no other-than-temporary impairment losses and net realized gains were not significant.

The amortized cost and estimated fair value of cash equivalents and marketable securities as of January 27, 2019 and January 28, 2018 are shown below by contractual maturity.

	January 27, 2019		January 28, 2018	
	Amortized Cost	Estimated Fair Value	Amortized Cost	Estimated Fair Value
	(In millions)			
Less than one year	$ 5,042	$ 5,034	$ 5,381	$ 5,375
Due in 1 - 5 years	2,271	2,268	1,500	1,485
Mortgage-backed securities issued by United States government-sponsored enterprises not due at a single maturity date	22	22	35	35
Total	$ 7,335	$ 7,324	$ 6,916	$ 6,895

NVIDIA CORPORATION AND SUBSIDIARIES
NOTES TO THE CONSOLIDATED FINANCIAL STATEMENTS
(Continued)

Note 8 - Fair Value of Financial Assets and Liabilities

The fair values of our financial assets and liabilities are determined using quoted market prices of identical assets or quoted market prices of similar assets from active markets. We review fair value hierarchy classification on a quarterly basis. There were no significant transfers between Levels 1 and 2 financial assets and liabilities for fiscal year 2019 . Level 3 financial assets and liabilities are based on unobservable inputs to the valuation methodology and include our own data about assumptions market participants would use in pricing the asset or liability based on the best information available under the circumstances.

		Fair Value at	
	Pricing Category	January 27, 2019	January 28, 2018
		(In millions)	
Assets			
Cash equivalents and marketable securities:			
Corporate debt securities	Level 2	$ 2,620	$ 1,295
Debt securities of United States government agencies	Level 2	$ 2,280	$ 815
Debt securities issued by the United States Treasury	Level 2	$ 1,492	$ 573
Money market funds	Level 1	$ 483	$ 3,789
Foreign government bonds	Level 2	$ 209	$ 41
Asset-backed securities	Level 2	$ 151	$ 252
Mortgage-backed securities issued by United States government-sponsored enterprises	Level 2	$ 89	$ 130
Liabilities			
Current liability:			
1.00% Convertible Senior Notes (1)	Level 2	$ —	$ 189
Other noncurrent liabilities:			
2.20% Notes Due 2021 (1)	Level 2	$ 978	$ 982
3.20% Notes Due 2026 (1)	Level 2	$ 961	$ 986

(1) These liabilities are carried on our Consolidated Balance Sheets at their original issuance value, net of unamortized debt discount and issuance costs, and are not marked to fair value each period. Refer to Note 11 of these Notes to the Consolidated Financial Statements for additional information.

Note 9 - Balance Sheet Components

Certain balance sheet components are as follows:

	January 27, 2019	January 28, 2018
	(In millions)	
Inventories:		
Raw materials	$ 613	$ 227
Work in-process	238	192
Finished goods	724	377
Total inventories	$ 1,575	$ 796

NVIDIA CORPORATION AND SUBSIDIARIES
NOTES TO THE CONSOLIDATED FINANCIAL STATEMENTS
(Continued)

	January 27, 2019	January 28, 2018	Estimated Useful Life
	(In millions)		*(In years)*
Property and Equipment:			
Land	$ 218	$ 218	(A)
Building	339	348	25-30
Test equipment	516	462	3-5
Computer equipment	522	285	3-5
Leasehold improvements	263	198	(B)
Software and licenses	109	88	3-5
Office furniture and equipment	69	79	5
Capital leases	28	28	(B)
Construction in process	107	31	(C)
Total property and equipment, gross	2,171	1,737	
Accumulated depreciation and amortization	(767)	(740)	
Total property and equipment, net	$ 1,404	$ 997	

(A) Land is a non-depreciable asset.

(B) Leasehold improvements and capital leases are amortized based on the lesser of either the asset's estimated useful life or the remaining expected lease term.

(C) Construction in process represents assets that are not available for their intended use as of the balance sheet date.

Depreciation expense for fiscal years 2019 , 2018 , and 2017 was $233 million , $144 million , and $118 million , respectively.

Accumulated amortization of leasehold improvements and capital leases was $189 million and $178 million as of January 27, 2019 and January 28, 2018 , respectively.

	January 27, 2019	January 28, 2018
	(In millions)	
Accrued and Other Current Liabilities:		
Customer program accruals	$ 302	$ 181
Accrued payroll and related expenses	186	172
Deferred revenue (1)	92	53
Taxes payable	91	33
Accrued legal settlement costs	24	—
Coupon interest on debt obligations	20	20
Warranty accrual (2)	18	15
Professional service fees	14	15
Accrued royalties	10	17
Other	61	36
Total accrued and other current liabilities	$ 818	$ 542

(1) Deferred revenue primarily includes customer advances and deferrals related to license and development arrangements and PCS.

(2) Refer to Note 12 of these Notes to the Consolidated Financial Statements for a discussion regarding warranties.

NVIDIA CORPORATION AND SUBSIDIARIES
NOTES TO THE CONSOLIDATED FINANCIAL STATEMENTS
(Continued)

	January 27, 2019	January 28, 2018
	(In millions)	
Other Long-Term Liabilities:		
Income tax payable (1)	$ 513	$ 559
Deferred revenue (2)	46	15
Deferred rent	21	9
Employee benefits liability	20	12
Deferred income tax liability	19	18
Other	14	19
Total other long-term liabilities	$ 633	$ 632

(1) As of January 27, 2019, represents the long-term portion of the one-time transition tax payable of $350 million , as well as unrecognized tax benefits of $142 million and related interest and penalties of $21 million .

(2) Deferred revenue primarily includes deferrals related to license and development arrangements and PCS.

Note 10 - Derivative Financial Instruments

We enter into foreign currency forward contracts to mitigate the impact of foreign currency exchange rate movements on our operating expenses. We designate these contracts as cash flow hedges and assess the effectiveness of the hedge relationships on a spot to spot basis. Gains or losses on the contracts are recorded in accumulated other comprehensive income or loss and reclassified to operating expense when the related operating expenses are recognized in earnings or ineffectiveness should occur. The fair value of the contracts was not significant as of January 27, 2019 and January 28, 2018 .

We also enter into foreign currency forward contracts to mitigate the impact of foreign currency movements on monetary assets and liabilities that are denominated in currencies other than U.S. dollar. These forward contracts were not designated for hedge accounting treatment. Therefore, the change in fair value of these contracts is recorded in other income or expense and offsets the change in fair value of the hedged foreign currency denominated monetary assets and liabilities, which is also recorded in other income or expense.

The table below presents the notional value of our foreign currency forward contracts outstanding as of January 27, 2019 and January 28, 2018 :

	January 27, 2019	January 28, 2018
	(In millions)	
Designated as cash flow hedges	$ 408	$ 104
Not designated for hedge accounting	$ 241	$ 94

As of January 27, 2019 , all designated foreign currency forward contracts mature within eighteen months. The expected realized gains and losses deferred into accumulated other comprehensive income (loss) related to foreign currency forward contracts within the next twelve months was not significant.

During fiscal years 2019 and 2018 , the impact of derivative financial instruments designated for hedge accounting treatment on other comprehensive income or loss was not significant and all such instruments were determined to be highly effective. Therefore, there were no gains or losses associated with ineffectiveness.

Note 11 - Debt

Long-Term Debt

2.20% Notes Due 2021 and 3.20% Notes Due 2026

In fiscal year 2017, we issued $1.00 billion of the 2.20% Notes Due 2021, and $1.00 billion of the 3.20% Notes Due 2026, or collectively, the Notes. Interest on the Notes is payable on March 16 and September 16 of each year, beginning on March 16, 2017. Upon 30 days' notice to holders of the Notes, we may redeem the Notes for cash prior to maturity, at redemption

NVIDIA CORPORATION AND SUBSIDIARIES
NOTES TO THE CONSOLIDATED FINANCIAL STATEMENTS
(Continued)

prices that include accrued and unpaid interest, if any, and a make-whole premium. However, no make-whole premium will be paid for redemptions of the Notes Due 2021 on or after August 16, 2021, or for redemptions of the Notes Due 2026 on or after June 16, 2026. The net proceeds from the Notes were $1.98 billion , after deducting debt discount and issuance costs.

The Notes are our unsecured senior obligations and rank equally in right of payment with all existing and future unsecured and unsubordinated indebtedness. The Notes are structurally subordinated to the liabilities of our subsidiaries and are effectively subordinated to any secured indebtedness to the extent of the value of the assets securing such indebtedness. All existing and future liabilities of our subsidiaries will be effectively senior to the Notes.

The carrying value of the Notes and the associated interest rates were as follows:

	Expected Remaining Term (years)	Effective Interest Rate	January 27, 2019	January 28, 2018
			(In millions)	
2.20% Notes Due 2021	2.6	2.38%	$ 1,000	$ 1,000
3.20% Notes Due 2026	7.6	3.31%	1,000	1,000
Unamortized debt discount and issuance costs			(12)	(15)
Net carrying amount			$ 1,988	$ 1,985

Convertible Debt

1.00% Convertible Senior Notes Due 2018

In fiscal year 2014, we issued $1.50 billion of Convertible Notes. During fiscal year 2019 , we paid cash to settle an aggregate of $16 million in principal amount of the Convertible Notes and issued 714 thousand shares of our common stock for the excess conversion value. The related loss on early conversions was not significant. As of January 27, 2019 , there were no Convertible Notes outstanding.

Note Hedges

Concurrently with the issuance of the Convertible Notes, we entered into the Note Hedges. Through January 27, 2019 , we had received 57 million shares of our common stock from the exercise of a portion of the Note Hedges related to the settlement of $1.50 billion in principal amount of the Convertible Notes. As of January 27, 2019 , there were no Note Hedges outstanding.

Revolving Credit Facility

We have a Credit Agreement under which we may borrow up to $575 million for general corporate purposes and can obtain revolving loan commitments up to $425 million . As of January 27, 2019 , we had no t borrowed any amounts under this agreement.

Commercial Paper

We have a $575 million commercial paper program to support general corporate purposes. As of January 27, 2019 , we had no t issued any commercial paper.

Note 12 - Commitments and Contingencies

Inventory Purchase Obligations

As of January 27, 2019 , we had outstanding inventory purchase obligations totaling $912 million .

Capital Purchase Obligations

As of January 27, 2019 , we had outstanding capital purchase obligations totaling $258 million .

NVIDIA CORPORATION AND SUBSIDIARIES
NOTES TO THE CONSOLIDATED FINANCIAL STATEMENTS
(Continued)

Lease Obligations

Our headquarters complex is located in Santa Clara, California and includes ten buildings that are leased properties. Future minimum lease payments related to headquarters operating leases total $326 million over the remaining terms of the leases, including predetermined rent escalations, and are included in the future minimum lease payment schedule below.

Additionally, we have other domestic and international office facilities, including datacenter space, under operating leases expiring through fiscal year 2035.

Future minimum lease payments under our non-cancelable operating leases as of January 27, 2019 , are as follows:

	Future Minimum Lease Obligations
	(In millions)
Fiscal Year:	
2020	$ 100
2021	97
2022	90
2023	77
2024	54
2025 and thereafter	265
Total	$ 683

Rent expense for fiscal years 2019 , 2018 , and 2017 was $80 million , $54 million , and $46 million , respectively.

Accrual for Product Warranty Liabilities

The estimated amount of product returns and warranty liabilities was $18 million and $15 million as of January 27, 2019 and January 28, 2018 , respectively.

In connection with certain agreements that we have entered in the past, we have provided indemnities to cover the indemnified party for matters such as tax, product, and employee liabilities. We have included intellectual property indemnification provisions in our technology related agreements with third parties. Maximum potential future payments cannot be estimated because many of these agreements do not have a maximum stated liability. We have not recorded any liability in our Consolidated Financial Statements for such indemnifications.

Litigation

Polaris Innovations Limited

On May 16, 2016, Polaris Innovations Limited, or Polaris, a non-practicing entity and wholly-owned subsidiary of Quarterhill Inc. (formerly WiLAN Inc.), filed a complaint against NVIDIA for patent infringement in the United States District Court for the Western District of Texas. Polaris alleges that NVIDIA has infringed and is continuing to infringe six U.S. patents relating to the control of dynamic random-access memory, or DRAM. The complaint seeks unspecified monetary damages, enhanced damages, interest, fees, expenses, and costs against NVIDIA. On September 14, 2016, NVIDIA answered the Polaris Complaint and asserted various defenses including non-infringement and invalidity of the six Polaris patents.

On December 5, 2016, the Texas Court granted NVIDIA’s motion to transfer and ordered the case transferred to the Northern District of California.

Between December 7, 2016 and July 25, 2017, NVIDIA filed multiple petitions for inter partes review, or IPR, at the United States Patent and Trademark Office, or USPTO, challenging the validity of each of the patents asserted by Polaris in the U.S. litigation. The USPTO instituted IPRs for four U.S. patents and declined to institute IPRs on two U.S. patents. On June 19, 2018, the USPTO issued a Final Written Decision on one IPR, finding claims 1-23 and 28 unpatentable but that claims 24-27 were not proved unpatentable. On November 20, 2018, the USPTO issued Final Written Decisions on two IPRs, finding claims 1, 4, 8-12, 16, 18, 43, 45, and 48-51 unpatentable but that claims 2-3, 5, 14, 17, 19-23, 26-31, and 44 were not proved unpatentable. On December 4, 2018, the USPTO issued a Final Written Decision on one IPR, finding all claims unpatentable. On December 19, 2018, the USPTO issued a Final Written Decision on one IPR, finding claims 1-14 unpatentable.

NVIDIA CORPORATION AND SUBSIDIARIES
NOTES TO THE CONSOLIDATED FINANCIAL STATEMENTS
(Continued)

On June 15, 2017, the California Court granted NVIDIA's motion to stay the district court litigation pending resolution of the petitions for IPR. The California Court has not set a trial date.

On December 30, 2016, Polaris filed a complaint against NVIDIA for patent infringement in the Regional Court of Düsseldorf, Germany. Polaris alleges that NVIDIA has infringed and is continuing to infringe three patents relating to control of DRAM. On July 14, 2017, NVIDIA filed defenses to the infringement allegations including non-infringement with respect to each of the three asserted patents. On September 3, 2018, NVIDIA filed a rejoinder with additional noninfringement arguments. On December 4, 2018, NVIDIA filed a further rejoinder with additional noninfringement, nullity, and FRAND arguments.

An oral hearing is scheduled for February 21, 2019.

Between March 31, 2017 and June 12, 2017, NVIDIA filed nullity actions with the German Patent Court challenging the validity of each of the patents asserted by Polaris in the German litigation.

ZiiLabs 1 Patents Lawsuit

On October 2, 2017, ZiiLabs Inc., Ltd., or ZiiLabs, a non-practicing entity, filed a complaint in the United States District Court for the District of Delaware alleging that NVIDIA has infringed and is continuing to infringe four U.S. patents relating to GPUs, or the ZiiLabs 1 Patents. ZiiLabs is a Bermuda corporation and a wholly-owned subsidiary of Creative Technology Asia Limited, a Hong Kong company which is itself is a wholly-owned subsidiary of Creative Technology Ltd., a publicly traded Singapore company. The complaint seeks unspecified monetary damages, enhanced damages, interest, costs, and fees against NVIDIA and an injunction against further direct or indirect infringement of the ZiiLabs 1 Patents. On November 27, 2017, NVIDIA answered the ZiiLabs complaint and asserted various defenses including non-infringement and invalidity of the ZiiLabs 1 Patents.

On January 10, 2018, ZiiLabs filed a first amended complaint asserting infringement of a fifth U.S. patent.

On February 22, 2018, the Delaware Court stayed the ZiiLabs 1 case pending the resolution of the U.S. International Trade Commission, or USITC, investigation over the ZiiLabs 2 patents.

On February 1, 2019, NVIDIA entered into an immaterial agreement in which it receives a license to the ZiiLabs patents and a dismissal of the ZiiLabs 1 and 2 Patent Lawsuits. The ZiiLabs 1 and 2 district court cases were dismissed pursuant to a stipulation of dismissal filed on February 8, 2019. The Administrative Law Judge issued an Initial Determination on February 12, 2019, granting the motion to terminate the USITC investigation addressing the ZiiLabs 2 patents.

ZiiLabs 2 Patents Lawsuits

On December 27, 2017, ZiiLabs filed a second complaint in the United States District Court for the District of Delaware alleging that NVIDIA has infringed four additional U.S. patents, or the ZiiLabs 2 Patents. The second complaint also seeks unspecified monetary damages, enhanced damages, interest, costs, and fees against NVIDIA and an injunction against further direct or indirect infringement of the ZiiLabs 2 Patents.

On February 22, 2018, the Delaware Court stayed the district court action on the ZiiLabs 2 patents pending the resolution of the USITC Investigation over the ZiiLabs 2 patents.

On December 29, 2017, ZiiLabs filed a request with the USITC to commence an Investigation pursuant to Section 337 of the Tariff Act of 1930 relating to the unlawful importation of certain graphics processors and products containing the same. ZiiLabs alleges that the unlawful importation results from the infringement of the ZiiLabs 2 Patents by products from respondents NVIDIA, ASUSTeK Computer Inc., ASUS Computer International, EVGA Corporation, Gigabyte Technology Co., Ltd., G.B.T. Inc., Micro-Star International Co., Ltd., MSI Computer Corp., Nintendo Co., Ltd., Nintendo of America Inc., PNY Technologies Inc., Zotac International (MCO) Ltd., and Zotac USA Inc.

On February 28, 2018, NVIDIA and the other respondents answered the USITC complaint and asserted various defenses including non-infringement and invalidity of the four asserted ZiiLabs 2 patents.

On May 10, 2018, the Administrative Law Judge then presiding over the investigation issued an Initial Determination terminating the investigation with respect to one of the patents. On July 17, 2018, the USITC affirmed this decision on modified grounds.

NVIDIA CORPORATION AND SUBSIDIARIES
NOTES TO THE CONSOLIDATED FINANCIAL STATEMENTS
(Continued)

On October 18, 2018, the Administrative Law Judge currently presiding over the investigation issued an order construing certain claims of the three remaining patents in the investigation.

The hearing in the investigation is currently scheduled to begin on April 8, 2019. The target date for completion of the investigation is September 9, 2019.

On February 1, 2019, NVIDIA entered into an immaterial agreement in which it receives a license to the ZiiLabs patents and a dismissal of the ZiiLabs 1 and 2 Patent Lawsuits. The ZiiLabs 1 and 2 district court cases were dismissed pursuant to a stipulation of dismissal filed on February 8, 2019. The Administrative Law Judge issued an Initial Determination on February 12, 2019, granting the motion to terminate the USITC investigation addressing the ZiiLabs 2 patents.

Securities Class Action and Derivative Lawsuits

On December 21, 2018, a purported securities class action lawsuit was filed in the United States District Court for the Northern District of California, captioned Iron Workers Joint Funds v. Nvidia Corporation, et al. (Case No. 18-cv-7669), naming as defendants NVIDIA and certain of NVIDIA's officers. The complaint asserts that the defendants violated Section 10(b) of the Securities Exchange Act of 1934, as amended, or the Exchange Act, and SEC Rule 10b-5, by making materially false or misleading statements related to channel inventory and the impact of cryptocurrency mining on GPU demand between August 10, 2017 and November 15, 2018. The plaintiff also alleges that the NVIDIA officers who they named as defendants violated Section 20(a) of the Exchange Act. The plaintiff seeks class certification, an award of unspecified compensatory damages, an award of equitable/injunctive or other further relief as the Court may deem just and proper. On December 28, 2018, a substantially similar purported securities class action was commenced in the Northern District of California, captioned Oto v. Nvidia Corporation, et al. (Case No. 18-cv-07783), naming the same defendants, and seeking substantially similar relief. The two cases have been related and are before the same judge. A stipulation to consolidate the Iron Workers and Oto actions is pending before the Court. On February 19, 2019, a number of shareholders filed motions to consolidate the two cases and to be appointed lead plaintiff and for their respective counsel to be appointed lead counsel.

On January 18, 2019, a shareholder, purporting to act on the behalf of NVIDIA, filed a derivative lawsuit in the Northern District of California, captioned Han v. Huang, et al. (Case No. 19-cv-00341), seeking to assert claims on behalf of NVIDIA against the members of NVIDIA's board of directors and certain officers. The lawsuit asserts claims for breach of fiduciary duty, unjust enrichment, waste of corporate assets, and violations of Sections 14(a), 10(b), and 20(a) of the Exchange Act based on the dissemination of allegedly false and misleading statements related to channel inventory and the impact of cryptocurrency mining on GPU demand. The plaintiff is seeking unspecified damages and other relief, including reforms and improvements to NVIDIA's corporate governance and internal procedures. On February 12, 2019, a substantially similar derivative lawsuit was filed in the Northern District of California captioned Yang v. Huang, et. al. (Case No. 19-cv-00766), naming the same named defendants, and seeking the same relief. On February 19, 2019, a third substantially similar derivative lawsuit was filed in the Northern District of California captioned The Booth Family Trust v. Huang, et. al. (Case No. 3:19-cv-00876), naming the same named defendants, and seeking substantially the same relief.

It is possible that additional suits will be filed, or allegations received from shareholders, with respect to these same or other matters, naming us and/or our officers and directors as defendants.

Accounting for Loss Contingencies

We are engaged in legal actions not described above arising in the ordinary course of business and, while there can be no assurance of favorable outcomes, we believe that the ultimate outcome of these actions will not have a material adverse effect on our operating results, liquidity or financial position. As of January 27, 2019, with the exception of immaterial amounts, we have not recorded any accrual for contingent liabilities associated with the legal proceedings described above based on our belief that liabilities, while possible, are not probable. Further, except as specifically described above, any possible loss or range of loss in these matters cannot be reasonably estimated at this time.

NVIDIA CORPORATION AND SUBSIDIARIES
NOTES TO THE CONSOLIDATED FINANCIAL STATEMENTS
(Continued)

Note 13 - Income Taxes

The income tax expense (benefit) applicable to income before income taxes consists of the following:

	Year Ended		
	January 27, 2019	January 28, 2018	January 29, 2017
	(In millions)		
Current income taxes:			
Federal	$ 1	$ 464	$ 7
State	—	1	1
Foreign	69	43	34
Total current	70	508	42
Deferred taxes:			
Federal	(315)	(376)	199
State	—	—	—
Foreign	—	17	(2)
Total deferred	(315)	(359)	197
Income tax expense (benefit)	$ (245)	$ 149	$ 239

Income before income tax consists of the following:

	Year Ended		
	January 27, 2019	January 28, 2018	January 29, 2017
	(In millions)		
Domestic (1)	$ 1,843	$ 1,600	$ 600
Foreign	2,053	1,596	1,305
Income before income tax	$ 3,896	$ 3,196	$ 1,905

(1) The increase in domestic income is primarily due to jurisdictional allocation of stock-based compensation charges.

The income tax expense (benefit) differs from the amount computed by applying the U.S. federal statutory rate of 21% , 33.9% , and 35% for fiscal years 2019, 2018, and 2017, respectively, to income before income taxes as follows:

	Year Ended		
	January 27, 2019	January 28, 2018	January 29, 2017
	(In millions)		
Tax expense computed at federal statutory rate	$ 818	$ 1,084	$ 667
Expense (benefit) resulting from:			
State income taxes, net of federal tax effect	23	10	4
Foreign tax rate differential	(412)	(545)	(315)
Stock-based compensation	(191)	(181)	(70)
Tax Cuts and Jobs Act of 2017	(368)	(133)	—
U.S. federal R&D tax credit	(141)	(87)	(52)
Other	26	1	5
Income tax expense (benefit)	$ (245)	$ 149	$ 239

NVIDIA CORPORATION AND SUBSIDIARIES
NOTES TO THE CONSOLIDATED FINANCIAL STATEMENTS
(Continued)

The tax effect of temporary differences that gives rise to significant portions of the deferred tax assets and liabilities are presented below:

	January 27, 2019	January 28, 2018
	(In millions)	
Deferred tax assets:		
Net operating loss carryforwards	$ 70	$ 67
Accruals and reserves, not currently deductible for tax purposes	41	24
Property, equipment and intangible assets	2	32
Research and other tax credit carryforwards	626	579
Stock-based compensation	25	24
GILTI deferred tax assets	376	—
Gross deferred tax assets	1,140	726
Less valuation allowance	(562)	(469)
Total deferred tax assets	578	257
Deferred tax liabilities:		
Acquired intangibles	(2)	(4)
Unremitted earnings of foreign subsidiaries	(35)	(26)
Gross deferred tax liabilities	(37)	(30)
Net deferred tax asset (1)	$ 541	$ 227

(1) Net deferred tax asset includes long-term deferred tax assets of $560 million and $245 million and long-term deferred tax liabilities of $19 million and $18 million for fiscal years 2019 and 2018, respectively. Long-term deferred tax assets are included in Other assets and long-term deferred tax liabilities are included in Other long-term liabilities on our Consolidated Balance Sheets.

We recognized an income tax benefit of $245 million for fiscal year 2019, and income tax expense of $149 million and $239 million for fiscal years 2018 , and 2017 , respectively. Our annual effective tax rate was (6.3)% , 4.7% , and 12.5% for fiscal years 2019 , 2018 , and 2017 , respectively.

In December 2017, the TCJA was enacted into law. The TCJA significantly changed U.S. tax law, including a reduction of the U.S. federal corporate income tax rate from 35% to 21% , a requirement for companies to pay a one-time transition tax on the earnings of certain foreign subsidiaries that were previously tax deferred and the creation of new taxes (global intangible low-taxed income, or GILTI) on certain foreign-source earnings. As a fiscal year-end taxpayer, certain provisions of the TCJA began to impact us in the fourth quarter of fiscal year 2018, while other provisions impacted us beginning in fiscal year 2019.

In fiscal year 2018 and the first nine months of fiscal year 2019, we recorded provisional amounts for certain enactment-date effects of the TCJA by applying the SEC guidance in SAB 118 because we had not yet completed our accounting for these effects. As of January 27, 2019, we completed our accounting for all of the enactment-date income tax effects of the TCJA and recognized a reduction of $368 million to the provisional amount recorded at January 28, 2018 as a component of income tax expense (benefit). This adjustment primarily relates to the effects of electing to account for GILTI in deferred taxes, as described below. Our final tax benefit from the TCJA was $501 million .

The one-time transition tax is based on the post-1986 earnings and profits, or E&P, of our foreign subsidiaries. We had previously accrued deferred taxes on a portion of these same earnings. We recorded a provisional one-time transition tax liability of $971 million at January 28, 2018. Upon further analysis of the TCJA and Notices and regulations issued by the US Department of the Treasury and Internal Revenue Service, we finalized our calculations of the transition tax liability during fiscal year 2019. For fiscal year 2019, we increased our transition tax provisional amount by $33 million .

As a result of the reduction of the corporate income tax rate to 21% , companies were required to remeasure their deferred tax assets and liabilities as of the date of enactment. As a result, at January 28, 2018 we had recorded a provisional income tax expense of $43 million on the write-down of our deferred tax balance. Upon further analysis of certain aspects of the TJCA, including immediate expensing of qualified capital expenditures and refinement of our calculations, we reduced our provisional tax expense amount by $20 million .

NVIDIA CORPORATION AND SUBSIDIARIES
NOTES TO THE CONSOLIDATED FINANCIAL STATEMENTS
(Continued)

The TCJA subjects a U.S. corporation to tax on its GILTI. Under U.S. GAAP, we can make an accounting policy election to either treat taxes due on the GILTI as a current period expense or factor such amounts into our measurement of deferred taxes. Because we were still evaluating the GILTI provisions as of January 28, 2018, we recorded no GILTI-related deferred balances. After further evaluation, we elected to account for GILTI deferred taxes. In fiscal year 2019, we recorded additional deferred tax assets as a net $370 million income tax benefit related to GILTI in deferred taxes.

The decrease in the effective tax rate in fiscal year 2019 as compared to fiscal years 2018 and 2017 was primarily due to a decrease in the U.S. statutory tax rate from 33.9% to 21% , the finalization of the enactment-date income tax effects of the TCJA, higher U.S federal research tax credits and excess tax benefits related to stock-based compensation in fiscal year 2019.

The decrease in the effective tax rate in fiscal year 2018 as compared to fiscal year 2017 was primarily due to the provisional impact of the tax law changes and recognition of excess tax benefits related to stock-based compensation.

Our effective tax rate for fiscal year 2019 was lower than the U.S. federal statutory rate of 21% due primarily to income earned in jurisdictions, including British Virgin Islands, Hong Kong, China, Taiwan and United Kingdom, where the tax rate was lower than the U.S. federal statutory tax rates, the finalization of the enactment-date income tax effects of the TCJA, favorable recognition of the U.S. federal research tax credits, and excess tax benefits related to stock-based compensation.

Our effective tax rate for fiscal years 2018 and 2017 was lower than the blended U.S. federal statutory rate of 33.9% for fiscal year 2018 and 35% for fiscal year 2017 due primarily to income earned in jurisdictions, including British Virgin Islands, Hong Kong, China, Taiwan and United Kingdom, where the tax rate was lower than the U.S. federal statutory tax rates, favorable recognition of U.S. federal research tax credits, the provisional impact of the tax law changes in 2018, and excess tax benefits related to stock-based compensation.

As of January 27, 2019 and January 28, 2018, we had a valuation allowance of $562 million and $469 million , respectively, related to state and certain foreign deferred tax assets that management determined not likely to be realized due, in part, to projections of future taxable income. To the extent realization of the deferred tax assets becomes more-likely-than-not, we would recognize such deferred tax asset as an income tax benefit during the period.

As of January 27, 2019, we had federal, state and foreign net operating loss carryforwards of $72 million , $291 million and $290 million , respectively. The federal and state carryforwards will expire beginning in fiscal year 2023 and 2020, respectively. The foreign net operating loss carryforwards of $290 million may be carried forward indefinitely. As of January 27, 2019, we had federal research tax credit carryforwards of $347 million that will begin to expire in fiscal year 2037. We have state research tax credit carryforwards of $718 million , of which $687 million is attributable to the State of California and may be carried over indefinitely, and $31 million is attributable to various other states and will expire beginning in fiscal year 2020. Our tax attributes, net operating loss and tax credit carryforwards, remain subject to audit and may be adjusted for changes or modification in tax laws, other authoritative interpretations thereof, or other facts and circumstances. Utilization of federal, state, and foreign net operating losses and tax credit carryforwards may also be subject to limitations due to ownership changes and other limitations provided by the Internal Revenue Code and similar state and foreign tax provisions. If any such limitations apply, the federal, states, or foreign net operating loss and tax credit carryforwards, as applicable, may expire or be denied before utilization.

As of January 27, 2019, we had $477 million of gross unrecognized tax benefits, of which $432 million would affect our effective tax rate if recognized. However, approximately $82 million of the unrecognized tax benefits were related to state income tax positions taken, that, if recognized, would be in the form of a carryforward deferred tax asset that would likely attract a full valuation allowance. The $432 million of unrecognized tax benefits as of January 27, 2019 consisted of $142 million recorded in non-current income taxes payable and $290 million reflected as a reduction to the related deferred tax assets.

NVIDIA CORPORATION AND SUBSIDIARIES
NOTES TO THE CONSOLIDATED FINANCIAL STATEMENTS
(Continued)

A reconciliation of gross unrecognized tax benefits is as follows:

	January 27, 2019	January 28, 2018	January 29, 2017
		(In millions)	
Balance at beginning of period	$ 447	$ 224	$ 230
Increases in tax positions for prior years	52	7	3
Decreases in tax positions for prior years	(141)	(1)	—
Increases in tax positions for current year	129	222	46
Settlements	—	—	(48)
Lapse in statute of limitations	(10)	(5)	(7)
Balance at end of period	$ 477	$ 447	$ 224

We classify an unrecognized tax benefit as a current liability, or amount refundable, to the extent that we anticipate payment or receipt of cash for income taxes within one year. The amount is classified as a long-term liability, or reduction of long-term deferred tax assets or amount refundable if we anticipate payment or receipt of cash for income taxes during a period beyond a year.

Our policy is to include interest and penalties related to unrecognized tax benefits as a component of income tax expense. As of January 27, 2019 , January 28, 2018 , and January 29, 2017 , we had accrued $21 million , $15 million , and $13 million , respectively, for the payment of interest and penalties related to unrecognized tax benefits, which is not included as a component of our unrecognized tax benefits. As of January 27, 2019 , unrecognized tax benefits of $142 million and the related interest and penalties of $21 million are included in non-current income taxes payable.

While we believe that we have adequately provided for all tax positions, amounts asserted by tax authorities could be greater or less than our accrued position. Accordingly, our provisions on federal, state and foreign tax-related matters to be recorded in the future may change as revised estimates are made or the underlying matters are settled or otherwise resolved. As of January 27, 2019, we do not believe that our estimates, as otherwise provided for, on such tax positions will significantly increase or decrease within the next twelve months.

We are subject to taxation by a number of taxing authorities both in the United States and throughout the world. As of January 27, 2019, the significant tax jurisdictions that may be subject to examination include the United States, Hong Kong, Taiwan, China, United Kingdom, Germany, and India for fiscal years 2003 through 2018. As of January 27, 2019, the significant tax jurisdictions for which we are currently under examination include India, Taiwan, China and UK for fiscal years 2003 through 2018.

Note 14 - Shareholders' Equity

Capital Return Program

Beginning August 2004, our Board of Directors authorized us to repurchase our stock.

During fiscal year 2019 , we repurchased a total of 9 million shares for $1.58 billion and also paid $371 million in cash dividends to our shareholders.

Through January 27, 2019 , we have repurchased an aggregate of 260 million shares under our share repurchase program for a total cost of $7.08 billion . All shares delivered from these repurchases have been placed into treasury stock. In November 2018, our board of directors authorized an additional $7.00 billion under our share repurchase program. As of January 27, 2019 , we were authorized, subject to certain specifications, to repurchase additional shares of our common stock up to $7.24 billion through December 2022.

Preferred Stock

As of January 27, 2019 and January 28, 2018 , there were no shares of preferred stock outstanding.

Common Stock

We are authorized to issue up to 2.00 billion shares of our common stock at $0.001 per share par value.

NVIDIA CORPORATION AND SUBSIDIARIES
NOTES TO THE CONSOLIDATED FINANCIAL STATEMENTS
(Continued)

Note 15 - Employee Retirement Plans

We have a 401(k) retirement plan covering substantially all of our U.S. employees. Under the plan, participating employees may defer up to 80% of their pre-tax earnings, subject to the Internal Revenue Service annual contribution limits and we match a portion of the employee contributions. Our contribution expense for fiscal years 2019 , 2018 , and 2017 was $39 million , $23 million , and $12 million , respectively. We also have defined contribution retirement plans outside of the United States to which we contributed $31 million , $25 million , and $23 million for fiscal years 2019 , 2018 , and 2017 , respectively.

Note 16 - Segment Information

Our Chief Executive Officer, who is considered to be our chief operating decision maker, or CODM, reviews financial information presented on an operating segment basis for purposes of making operating decisions and assessing financial performance. Our operating segments are equivalent to our reportable segments.

We report our business in two primary reportable segments - the GPU business and the Tegra Processor business - based on a single underlying graphics architecture.

Our GPU product brands are aimed at specialized markets including GeForce for gamers; Quadro for designers; Tesla and DGX for AI data scientists and big data researchers; and GRID for cloud-based visual computing users. Our Tegra brand integrates an entire computer onto a single chip, and incorporates GPUs and multi-core CPUs to drive supercomputing for autonomous robots, drones, and cars, as well as for game consoles and mobile gaming and entertainment devices.

Under the single unifying architecture for our GPU and Tegra Processors, we leverage our visual computing expertise by charging the operating expenses of certain core engineering functions to the GPU business, while charging the Tegra Processor business for the incremental cost of the teams working directly for that business. In instances where the operating expenses of certain functions benefit both reportable segments, our CODM assigns 100% of those expenses to the reportable segment that benefits the most.

The “All Other” category presented below represents the revenue and expenses that our CODM does not assign to either the GPU business or the Tegra Processor business for purposes of making operating decisions or assessing financial performance. The revenue includes primarily patent licensing revenue and the expenses include stock-based compensation expense, corporate infrastructure and support costs, acquisition-related costs, legal settlement costs, contributions, restructuring and other charges, product warranty charge, and other non-recurring charges and benefits that our CODM deems to be enterprise in nature.

Our CODM does not review any information regarding total assets on a reportable segment basis. Reportable segments do not record intersegment revenue, and, accordingly, there is none to be reported. The accounting policies for segment reporting are the same as for our consolidated financial statements. The table below presents details of our reportable segments and the “All Other” category.

NVIDIA CORPORATION AND SUBSIDIARIES
NOTES TO THE CONSOLIDATED FINANCIAL STATEMENTS
(Continued)

	GPU	Tegra Processor	All Other	Consolidated
			(In millions)	
Year Ended January 27, 2019:				
Revenue	$ 10,175	$ 1,541	$ —	$ 11,716
Depreciation and amortization expense	$ 197	$ 47	$ 18	$ 262
Operating income (loss)	$ 4,443	$ 241	$ (880)	$ 3,804
Year Ended January 28, 2018:				
Revenue	$ 8,137	$ 1,534	$ 43	$ 9,714
Depreciation and amortization expense	$ 123	$ 37	$ 39	$ 199
Operating income (loss)	$ 3,507	$ 303	$ (600)	$ 3,210
Year Ended January 29, 2017:				
Revenue	$ 5,822	$ 824	$ 264	$ 6,910
Depreciation and amortization expense	$ 116	$ 29	$ 42	$ 187
Operating income (loss)	$ 2,180	$ (9)	$ (237)	$ 1,934

	Year Ended		
	January 27, 2019	January 28, 2018	January 29, 2017
		(In millions)	
Reconciling items included in "All Other" category:			
Unallocated revenue	$ —	$ 43	$ 264
Stock-based compensation expense	(557)	(391)	(247)
Unallocated cost of revenue and operating expenses	(277)	(237)	(215)
Legal settlement costs	(44)	—	(16)
Acquisition-related and other costs	(2)	(15)	(23)
Total	$ (880)	$ (600)	$ (237)

Revenue by geographic region is allocated to individual countries based on the location to which the products are initially billed even if our customers' revenue is attributable to end customers that are located in a different location. The following table summarizes information pertaining to our revenue from customers based on the invoicing address by geographic regions:

	Year Ended		
	January 27, 2019	January 28, 2018	January 29, 2017
Revenue:		*(In millions)*	
Taiwan	$ 3,360	$ 2,991	$ 2,546
China (including Hong Kong)	2,801	1,896	1,305
Other Asia Pacific	2,368	2,066	1,010
United States	1,506	1,274	904
Europe	914	768	659
Other countries	767	719	486
Total revenue	$ 11,716	$ 9,714	$ 6,910

NVIDIA CORPORATION AND SUBSIDIARIES
NOTES TO THE CONSOLIDATED FINANCIAL STATEMENTS
(Continued)

The following table summarizes information pertaining to our revenue by each of the specialized markets we serve:

	Year Ended		
	January 27, 2019	January 28, 2018	January 29, 2017
Revenue:	*(In millions)*		
Gaming	$ 6,246	$ 5,513	$ 4,060
Professional Visualization	1,130	934	835
Datacenter	2,932	1,932	830
Automotive	641	558	487
OEM & IP	767	777	698
Total revenue	$ 11,716	$ 9,714	$ 6,910

The following table presents summarized information for long-lived assets by geographic region. Long-lived assets consist of property and equipment and deposits and other assets, and exclude goodwill and intangible assets.

	January 27, 2019	January 28, 2018
Long-lived assets:	*(In millions)*	
United States	$ 1,266	$ 928
Taiwan	137	58
India	44	40
China (including Hong Kong)	38	33
Europe	26	11
Other Asia Pacific	1	1
Total long-lived assets	$ 1,512	$ 1,071

No customer represented 10% or more of total revenue for fiscal years 2019 and 2018. In fiscal year 2017, we had one customer that represented 12% of our total revenue. The revenue was attributable to the GPU business.

Accounts receivable from significant customers, those representing 10% or more of total accounts receivable, aggregated approximately 19% of our accounts receivable balance from one customer as of January 27, 2019 , and approximately 28% of our accounts receivable balance from two customers as of January 28, 2018 .

NVIDIA CORPORATION AND SUBSIDIARIES
NOTES TO THE CONSOLIDATED FINANCIAL STATEMENTS
(Continued)

Note 17 - Quarterly Summary (Unaudited)

The following table sets forth our unaudited consolidated financial results, for the last eight fiscal quarters:

	Fiscal Year 2019 Quarters Ended			
	January 27, 2019	October 28, 2018	July 29, 2018	April 29, 2018
	(In millions, except per share data)			
Statements of Income Data:				
Revenue	$ 2,205	$ 3,181	$ 3,123	$ 3,207
Cost of revenue	$ 998	$ 1,260	$ 1,148	$ 1,139
Gross profit	$ 1,207	$ 1,921	$ 1,975	$ 2,068
Net income (1)	$ 567	$ 1,230	$ 1,101	$ 1,244
Net income per share (1):				
Basic	$ 0.93	$ 2.02	$ 1.81	$ 2.05
Diluted	$ 0.92	$ 1.97	$ 1.76	$ 1.98

(1) In the third and fourth quarters of fiscal year 2019, we recorded U.S. tax reform benefits of $138 million and $230 million , respectively, associated with the completion of our accounting for the enactment-date income tax effects of the TCJA. Refer to Note 13 of these Notes to the Consolidated Financial Statements for a discussion regarding the U.S. tax reform.

	Fiscal Year 2018 Quarters Ended			
	January 28, 2018	October 28, 2017	July 29, 2017	April 29, 2017
	(In millions, except per share data)			
Statements of Income Data:				
Revenue	$ 2,911	$ 2,636	$ 2,230	$ 1,937
Cost of revenue	$ 1,110	$ 1,067	$ 928	$ 787
Gross profit	$ 1,801	$ 1,569	$ 1,302	$ 1,150
Net income (1)	$ 1,118	$ 838	$ 583	$ 507
Net income per share (1):				
Basic	$ 1.84	$ 1.39	$ 0.98	$ 0.86
Diluted	$ 1.78	$ 1.33	$ 0.92	$ 0.79

(1) In the fourth quarter of fiscal year 2018, we recorded a U.S. tax reform provisional net tax benefit of $133 million associated with the one-time transition tax on our historical foreign earnings and the adjustment of deferred tax balances to the lower corporate tax rate. Refer to Note 13 of these Notes to the Consolidated Financial Statements for a discussion regarding the U.S. tax reform.

NVIDIA CORPORATION AND SUBSIDIARIES
SCHEDULE II – VALUATION AND QUALIFYING ACCOUNTS

Description	Balance at Beginning of Period	Additions	Deductions	Balance at End of Period
		(In millions)		
Fiscal year 2019				
Allowance for doubtful accounts	$ 4	$ — (1)	$ (2) (1)	$ 2
Sales return allowance	$ 9	$ 21 (2)	$ (22) (4)	$ 8
Deferred tax valuation allowance	$ 469	$ 93 (3)	$ —	$ 562
Fiscal year 2018				
Allowance for doubtful accounts	$ 3	$ 1 (1)	$ — (1)	$ 4
Sales return allowance	$ 10	$ 15 (2)	$ (16) (4)	$ 9
Deferred tax valuation allowance	$ 353	$ 116 (3)	$ —	$ 469
Fiscal year 2017				
Allowance for doubtful accounts	$ 2	$ 1 (1)	$ — (1)	$ 3
Sales return allowance	$ 9	$ 9 (2)	$ (8) (4)	$ 10
Deferred tax valuation allowance	$ 272	$ 81 (3)	$ —	$ 353

(1) Additions represent allowance for doubtful accounts charged to expense and deductions represent amounts recorded as reduction to expense upon reassessment of allowance for doubtful accounts at period end.

(2) Represents allowance for sales returns estimated at the time revenue is recognized primarily based on historical return rates and is charged as a reduction to revenue.

(3) Represents change in valuation allowance primarily related to state and certain foreign deferred tax assets that management has determined not likely to be realized due, in part, to projections of future taxable income of the respective jurisdictions. Refer to Note 13 of the Notes to the Consolidated Financial Statements in Part IV, Item 15 of this Annual Report on Form 10-K for additional information.

(4) Represents sales returns.

EXHIBIT INDEX

Exhibit No.	Exhibit Description	Incorporated by Reference			
		Schedule/Form	File Number	Exhibit	Filing Date
3.1	Amended and Restated Certificate of Incorporation	S-8	333-74905	4.1	3/23/1999
3.2	Certificate of Amendment of Amended and Restated Certificate of Incorporation	10-Q	0-23985	3.1	8/21/2008
3.3	Certificate of Amendment of Amended and Restated Certificate of Incorporation	8-K	0-23985	3.1	5/24/2011
3.4	Bylaw of NVIDIA Corporation, Amended and Re tated as of November 29, 2016	8 K	0 23985	3 1	12/1/2016
4.1	Reference is made to Exhibits 3.1, 3.2, 3.3 and 3.4				
4.2	Specimen Stock Certificate	S-1/A	333-47495	4.2	4/24/1998
4.3	Indenture (including the form of Notes) dated December 2, 2013 between NVIDIA Corporation and Wells Fargo Bank, National Association	8-K	0-23985	4.1	12/2/2013
4.4	Form of 1.00% Convertible Senior Note due 2018	8-K	0-23985	Exhibit A to Exhibit 4.1	12/2/2013
4.5	Indenture, dated as of September 16, 2016, by and between the Company and Wells Fargo Bank, National Association, as Trustee	8-K	0-23985	4.1	9/16/2016
4.6	Officers' Certificate, dated as of September 16, 2016	8-K	0-23985	4.2	9/16/2016
4.7	Form of 2021 Note	8-K	0-23985	Annex A to Exhibit 4.2	9/16/2016
4.8	Form of 2026 Note	8-K	0-23985	Annex B to Exhibit 4.2	9/16/2016
10.1	Form of Indemnity Agreement between NVIDIA Corporation and each of its directors and officers	8-K	0-23985	10.1	3/7/2006
10.2+	Amended and Restated 2007 Equity Incentive Plan	8-K	0-23985	10.1	5/21/2018
10.3+	2007 Equity Incentive Plan - Non-Statutory Stock Option (Annual Grant - Board Service (2011))	10-Q	0-23985	10.41	5/27/2011
10.4+	2007 Equity Incentive Plan - Non-Statutory Stock Option (Initial Grant - Board Service (2011))	8-K	0-23985	10.1	12/14/2011
10.5+	Amended and Restated 2007 Equity Incentive Plan - Non-Employee Director Stock Option Grant (2012 Annual Board Retainer)	10-Q	0-23985	10.4	5/23/2012
10.6+	2007 Equity Incentive Plan - Non Statutory Stock Option	8-K	0-23985	10.2	9/13/2010
10.7+	2007 Equity Incentive Plan - Incentive Stock Option	8-K	0-23985	10.21	9/13/2010
10.8+	Amended and Restated 2007 Equity Incentive Plan - Non Statutory Stock Option	10-Q	0-23985	10.1	8/22/2012
10.9+	Amended and Restated 2007 Equity Incentive Plan - Incentive Stock Option	10-Q	0-23985	10.2	8/22/2012

10.10+	Amended and Restated 2007 Equity Incentive Plan - Restricted Stock Unit Grant Notice and Restricted Stock Unit Purchase Agreement	10-Q	0-23985	10.3	8/22/2012
10.11+	Amended and Restated 2007 Equity Incentive Plan - Non-Employee Director Restricted Stock Unit (with deferral option)	10-Q	0-23985	10.3	5/23/2012
10.12+	Amended and Restated 2007 Equity Incentive Plan - Non Statutory Stock Option (Initial Grant - Board Service)	8-K	0-23985	10.1	7/23/2013
10.13+	Amended and Restated 2007 Equity Incentive Plan - Non-Employee Director Deferred Restricted Stock Unit Grant Notice and Deferred Restricted Stock Unit Agreement (2015)	10-K	0-23985	10.25	3/12/2015
10.14+	Amended and Restated 2007 Equity Incentive Plan - Non-Employee Director Deferred Restricted Stock Unit Grant Notice and Deferred Restricted Stock Unit Agreement (2016)	10-K	0-23985	10.26	3/12/2015
10.15+	Amended and Restated 2007 Equity Incentive Plan - Non-Employee Director Restricted Stock Unit Grant Notice and Restricted Stock Unit Agreement (2016)	10-K	0-23985	10.27	3/12/2015
10.16+	Amended and Restated 2007 Equity Incentive Plan - Non-Employee Director Restricted Stock Unit (Initial Grant - with deferral options)	10-Q	0-23985	10.1	5/20/2015
10.17+	Amended and Restated 2007 Equity Incentive Plan - Restricted Stock Unit Grant Notice and Restricted Stock Unit Agreement & Performance-Based Restricted Stock Unit Grant Notice and Performance-Based Restricted Stock Unit Agreement (2015)	10-Q	0-23985	10.2	5/20/2015
10.18+	Amended and Restated 2007 Equity Incentive Plan - Restricted Stock Unit Grant Notice and Restricted Stock Unit Agreement & Performance-Based Restricted Stock Unit Grant Notice and Performance-Based Restricted Stock Unit Agreement (2018)	10-Q	0-23985	10.2	5/22/2018
10.19+*	Amended and Restated 2007 Equity Incentive Plan - Global Restricted Stock Unit Grant Notice and Global Restricted Stock Unit Agreement (2019)				
10.20+	Amended and Restated 2012 Employee Stock Purchase Plan	10-Q	0-23985	10.2	5/21/2018
10.21+	Fiscal Year 2018 Variable Compensation Plan	8-K	0-23985	10.1	3/13/2017
10.22+	Fiscal Year 2019 Variable Compensation Plan	8-K	0-23985	10.1	3/13/2018
10.23+	Offer Letter between NVIDIA Corporation and Colette Kress, dated September 13, 2013	8-K	0-23985	10.1	9/16/2013
10.24+	Offer Letter between NVIDIA Corporation and Tim Teter, dated December 16, 2016	8-K	0-23985	10.1	1/19/2017
10.25	Base Convertible Note Hedge Transaction Confirmation	8-K	0-23985	99.1	12/2/2013
10.26	Additional Convertible Note Hedge Transaction Confirmation	8-K	0-23985	99.3	12/2/2013

10.27	Credit Agreement, dated as of October 7, 2016 by and among NVIDIA Corporation, Well Fargo Bank, National Association, as administrative agent, and the lenders party thereto	8-K	0-23985	1.1	10/13/2016
10.28	Form of Commercial Paper Dealer Agreement between NVIDIA Corporation, as Issuer, and the Dealer party thereto	8-K	0-23985	10.1	12/15/2017
21.1*	List of Registrant's Subsidiaries				
23.1*	Consent of PricewaterhouseCoopers LLP				
24.1*	Power of Attorney (included in signature page)				
31.1*	Certification of Chief Executive Officer as required by Rule 13a-14(a) of the Securities Exchange Act of 1934				
31.2*	Certification of Chief Financial Officer as required by Rule 13a-14(a) of the Securities Exchange Act of 1934				
32.1#*	Certification of Chief Executive Officer as required by Rule 13a-14(b) of the Securities Exchange Act of 1934				
32.2#*	Certification of Chief Financial Officer as required by Rule 13a-14(b) of the Securities Exchange Act of 1934				
101.INS*	XBRL Instance Document				
101.SCH*	XBRL Taxonomy Extension Schema Document				
101.CAL*	XBRL Taxonomy Extension Calculation Linkbase Document				
101.DEF*	XBRL Taxonomy Extension Definition Linkbase Document				
101.LAB*	XBRL Taxonomy Extension Labels Linkbase Document				
101.PRE*	XBRL Taxonomy Extension Presentation Linkbase Document				

* Filed herewith.

\+ Management contract or compensatory plan or arrangement.

\# In accordance with Item 601(b)(32)(ii) of Regulation S-K and SEC Release Nos. 33-8238 and 34-47986, Final Rule: Management's Reports on Internal Control Over Financial Reporting and Certification of Disclosure in Exchange Act Periodic Reports, the certifications furnished in Exhibits 32.1 and 32.2 hereto are deemed to accompany this Annual Report on Form 10-K and will not be deemed “filed” for purpose of Section 18 of the Exchange Act. Such certifications will not be deemed to be incorporated by reference into any filing under the Securities Act or the Exchange Act, except to the extent that the registrant specifically incorporates it by reference.

Copies of above exhibits not contained herein are available to any shareholder upon written request to:
Investor Relations: NVIDIA Corporation, 2788 San Tomas Expressway, Santa Clara, CA 95051

SIGNATURES

Pursuant to the requirements of Section 13 or 15(d) of the Securities Exchange Act of 1934, the Registrant has duly caused this report to be signed on its behalf by the undersigned, thereunto duly authorized, on February 21, 2019 .

NVIDIA Corporation

By: /s/ Jen-Hsun Huang

Jen-Hsun Huang

President and Chief Executive Officer

POWER OF ATTORNEY

KNOW ALL PERSONS BY THESE PRESENTS, that each person whose signature appears below constitutes and appoints Jen-Hsun Huang and Colette M. Kress, and each or any one of them, his true and lawful attorney-in-fact and agent, with full power of substitution and resubstitution, for him and in his name, place and stead, in any and all capacities, to sign any and all amendments to this report, and to file the same, with all exhibits thereto, and other documents in connection therewith, with the Securities and Exchange Commission, granting unto said attorneys-in-facts and agents, and each of them, full power and authority to do and perform each and every act and thing requisite and necessary to be done in connection therewith, as fully to all intents and purposes as he might or could do in person, hereby ratifying and confirming all that said attorneys-in-fact and agents, or any of them, or their or his substitutes or substitutes, may lawfully do or cause to be done by virtue hereof.

Pursuant to the requirements of the Securities Exchange Act of 1934, this report has been signed below by the following persons on behalf of the registrant and in the capacities and on the dates indicated.

Signature	Title	Date
/s/ JEN-HSUN HUANG Jen-Hsun Huang	President, Chief Executive Officer and Director (Principal Executive Officer)	**February 21, 2019**
/s/ COLETTE M. KRESS Colette M. Kress	Executive Vice President and Chief Financial Officer (Principal Financial Officer)	**February 21, 2019**
/s/ MICHAEL J. BYRON Michael J. Byron	Vice President and Chief Accounting Officer (Principal Accounting Officer)	**February 21, 2019**
Robert Burgess	Director	**February 21, 2019**
/s/ TENCH COXE Tench Coxe	Director	**February 21, 2019**
/s/ PERSIS DRELL Persis Drell	Director	**February 21, 2019**
/s/ JAMES C. GAITHER James C. Gaither	Director	**February 21, 2019**
/s/ DAWN HUDSON Dawn Hudson	Director	**February 21, 2019**
/s/ HARVEY C. JONES Harvey C. Jones	Director	**February 21, 2019**
/s/ MICHAEL MCCAFFERY Michael McCaffery	Director	**February 21, 2019**
/s/ MARK L. PERRY Mark L. Perry	Director	**February 21, 2019**
/s/ A. BROOKE SEAWELL A. Brooke Seawell	Director	**February 21, 2019**
Mark Stevens	Director	**February 21, 2019**

NVIDIA CORPORATION
GLOBAL RESTRICTED STOCK UNIT GRANT NOTICE
AMENDED & RESTATED 2007 EQUITY INCENTIVE PLAN

NVIDIA Corporation (the " ***Company*** "), pursuant to its Amended & Restated 2007 Equity Incentive Plan (the " ***Plan*** "), hereby awards to Participant a Restricted Stock Unit Award for the number of restricted stock units (the ***"Restricted Stock Units"***) set forth below (the " ***Award*** "). The Award is subject to all of the terms and conditions as set forth in this Grant Notice, in the attached Global Restricted Stock Unit Agreement, including any special terms and conditions for Participant's country set forth in any appendix thereto (the " ***Appendix*** "), and in the Plan, the latter two being incorporated by reference herein. Capitalized terms not otherwise defined in this Grant Notice or the Global Restricted Stock Unit Agreement (including the Appendix) (collectively, the " ***Agreement*** ") will have the meanings set forth in the Plan. In the event of any conflict between the terms in this Agreement and the Plan, the terms of the Plan will control.

Participant: __
Date of Grant: __
Vesting Commencement Date: __
Number of Restricted Stock Units/Shares Subject to Award: __

Vesting Schedule : This Award will vest as to _____, subject to Participant's Continuous Service through such vesting date. However, this Award will become fully vested prior to such date on the date of Participant's "separation from service" (as defined under Treasury Regulation Section 1.409A-1(h), without regard to any alternative definitions therein, a " ***Separation from Service*** ") by reason of death. If the Award is not vested as of Participant's Separation from Service for any other reason, it will immediately expire. Each installment of Restricted Stock Units that vests hereunder is a "separate payment" for purposes of Treasury Regulations Section 1.409A-2(b)(2).

Issuance Schedule: Except as provided in Section 6 of the Agreement, the Company will issue and deliver one (1) share of Common Stock for each Restricted Stock Unit that has vested under this Award on the date of vesting, but in all cases within the period necessary for compliance with Treasury Regulation Section 1.409A-1(b)(4).

Additional Terms/Acknowledgements: Participant acknowledges receipt of, and understands and agrees to, all of the terms and conditions set forth in the Agreement and the Plan. Participant acknowledges and agrees that the Agreement may not be modified, amended or revised except as provided in the Plan or the Agreement. Participant further acknowledges that as of the Date of Grant, the Agreement sets forth the entire understanding between Participant and the Company regarding this Award, and supersedes all prior oral and written agreements on that subject with the exception, if applicable, of: (i) the current written employment agreement entered into between the Employer (as defined in Section 9 of the Global Restricted Stock Unit Agreement) and Participant expressly specifying the terms that should govern this Award; (ii) the Company's insider trading policy; and (iii) any compensation recovery policy that is adopted by the Company or one of its Affiliates or is otherwise required by applicable law. By accepting this Award, Participant consents to receive Plan documents by electronic delivery and to participate in the Plan through an on-line or electronic system established and maintained by the Company or a third party designated by the Company.

NVIDIA CORPORATION **PARTICIPANT:**

By: ___ __
Signature Signature

Title: ___ Date: __

Date: __

ATTACHMENT I

NVIDIA CORPORATION
AMENDED & RESTATED 2007 EQUITY INCENTIVE PLAN
GLOBAL RESTRICTED STOCK UNIT AGREEMENT

Pursuant to the Global Restricted Stock Unit Grant Notice (" ***Grant Notice*** ") and this Global Restricted Stock Unit Agreement (including any special terms and conditions for your country set forth in the appendix attached hereto (the " ***Appendix*** ")) (collectively, the " ***Agreement*** "), NVIDIA Corporation (the " ***Company*** ") has awarded you a Restricted Stock Unit Award (the " ***Award*** ") under its Amended & Restated 2007 Equity Incentive Plan (the " ***Plan*** "). This Award is granted to you effective as of the date of grant set forth in the Grant Notice (the " ***Date of Grant*** "). Capitalized terms not explicitly defined in this Agreement will have the same meanings given to them in the Plan.

1. GRANT OF THE AWARD. The Award represents the right to be issued on a future date one share of Common Stock for each Restricted Stock Unit that vests under this Award, subject to the terms and conditions provided in this Agreement and in the Plan. As of the Date of Grant, the Company will credit to a bookkeeping account maintained by the Company for your benefit (the " ***Account*** ") the number of Restricted Stock Units subject to the Award. Except as otherwise provided in this Agreement, you will not be required to make any payment to the Company with respect to your receipt of the Award, the vesting of the Restricted Stock Units or the delivery of the underlying Common Stock.

2. VESTING. Subject to the limitations contained in this Agreement, your Award will vest, if at all, in accordance with the vesting schedule provided in the Grant Notice. Vesting will cease upon the termination of your Continuous Service (subject to any acceleration provided for in the Agreement or the Plan). On the termination of your Continuous Service, the Restricted Stock Units credited to the Account that were not vested on the date of such termination will be forfeited and returned to the Company at no cost to the Company and you will have no further right, title or interest in or to such Restricted Stock Units or the underlying shares of Common Stock.

3. NUMBER OF RESTRICTED STOCK UNITS AND SHARES OF COMMON STOCK.

(a) The number of Restricted Stock Units (and the related shares of Common Stock) subject to your Award will be adjusted from time to time for Capitalization Adjustments, as provided in the Plan.

(b) Any Restricted Stock Units, shares, cash or other property that become subject to the Award as a result of a Capitalization Adjustment, if any, will be subject to the same forfeiture restrictions, restrictions on transferability, and time and manner of delivery as applicable to the other shares covered by your Award.

(c) No fractional shares or rights for fractional shares of Common Stock will be created by this Section 3. The Board will round down, to the nearest whole share or whole unit of rights, any fractional shares or rights for fractional shares.

4. COMPLIANCE WITH LAW. You will not be issued any shares under your Award unless either (a) the shares are registered under the Securities Act; or (b) the Company has determined that such issuance would be exempt from the registration requirements of the Securities Act. Your Award also must comply with other applicable laws and regulations governing the Award, including any U.S. and non-U.S. state, federal and local laws, and you will not receive such shares if the Company determines that such receipt would not be in material compliance with such laws and regulations.

5. LIMITATIONS ON TRANSFER. Your Award is not transferable, except by will or by the laws of descent and distribution. In addition to any other limitation on transfer created by applicable securities laws, you agree not to assign, hypothecate, donate, encumber or otherwise dispose of any interest in any of the shares of Common Stock subject to the Award until the shares are issued to you. After the shares have been issued to you, you are free to assign, hypothecate, donate, encumber or otherwise dispose of any interest in such shares provided that any such actions are in compliance with the provisions in this Agreement and applicable securities laws. If permitted by the Board and valid under applicable law, you may, by delivering written notice to the Company, in a form satisfactory to the Company, designate a third party who, in the event of your death, will thereafter be entitled to receive any distribution of Common Stock to which you were entitled at the time of your death pursuant to this Agreement.

6. DATE OF ISSUANCE.

(a) The issuance of shares of Common Stock in respect of the Restricted Stock Units is intended to comply with Treasury Regulations Section 1.409A-1(b)(4) and will be construed and administered in such a manner to the extent applicable.

(b) Subject to the satisfaction of any withholding obligation for Tax-Related Items (as defined in Section 10 of this Agreement), in the event one or more Restricted Stock Units vests, the Company will issue to you, on the applicable vesting date, one share of Common Stock for each Restricted Stock Unit that vests and such issuance date is referred to as the " ***Original Issuance Date*** ." If the Original Issuance Date falls on a date that is not a business day, delivery will instead occur on the next following business day.

(c) However, if (i) the Original Issuance Date does not occur (1) during an "open window period" applicable to you, as determined by the Company in accordance with the Company's then-effective policy on trading in Company securities, or (2) on a date when you are otherwise permitted to sell shares of Common Stock on an established stock exchange or stock market (including but not limited to under a previously established Company-approved 10b5-1 trading plan), *and* (ii) the Company elects, prior to the Original Issuance Date, (1) not to satisfy any withholding obligations for Tax-Related Items (as defined in Section 10 below) by withholding shares of Common Stock from the shares otherwise due, on the Original

Issuance Date, to you under this Award, (2) not to permit you to enter into a "same day sale" commitment with a broker-dealer pursuant to this Agreement (including but not limited to a commitment under a previously established Company-approved 10b5-1 trading plan) and (3) not to permit you to cover any withholding obligations for Tax-Related Items (as defined in Section 10 below) in cash, then the shares that would otherwise be issued to you on the Original Issuance Date will not be delivered on such Original Issuance Date and will instead be delivered on the first business day when you are not prohibited from selling shares of the Company's Common Stock in the open public market, but in no event later than December 31 of the calendar year in which the Original Issuance Date occurs (that is, the last day of your taxable year in which the Original Issuance Date occurs), or, if and only if permitted in a manner that complies with Treasury Regulations Section 1.409A-1(b)(4), no later than the date that is the 15th day of the third calendar month of the year following the year in which the shares of Common Stock under this Award are no longer subject to a "substantial risk of forfeiture" within the meaning of Treasury Regulations Section 1.409A-1(d). The form of such delivery of the shares (*e.g.* , a stock certificate or electronic entry evidencing such shares) shall be determined by the Company.

7. DIVIDENDS. You will receive no benefit or adjustment to your Award and any unissued shares thereunder with respect to any cash dividend, stock dividend or other distribution that does not result from a Capitalization Adjustment. Following the date of vesting, in the event of any cash dividend, stock dividend or other distribution that does not result from a Capitalization Adjustment, no cash, stock or other property related to such dividend or distribution will be issuable in respect of your vested Restricted Stock Units.

8. RESTRICTIVE LEGENDS. The shares of Common Stock issued under your Award will be endorsed with appropriate legends if determined by the Company that legends are required under applicable law or otherwise.

9. AWARD NOT A SERVICE CONTRACT.

(a) Your Continuous Service with the Company or, if different, the Affiliate that employs you or for which you otherwise render services (the " ***Employer*** ") is not for any specified term and may be terminated by you or by the Employer at any time, for any reason, with or without cause and, if permitted under applicable law, with or without notice. Nothing in this Agreement (including, but not limited to, the vesting of your Award pursuant to the schedule set forth in the Grant Notice or the issuance of the shares subject to your Award), the Plan or any covenant of good faith and fair dealing that may be found implicit in this Agreement or the Plan will: (i) confer upon you any right to continue in the employ of, or affiliation with the Employer; (ii) constitute any promise or commitment by the Company, the Employer or any other Affiliate regarding the fact or nature of future positions, future work assignments, future compensation or any other term or condition of employment or affiliation; (iii) confer any right or benefit under this Agreement or the Plan unless such right or benefit has specifically accrued under the terms of this Agreement or Plan; or (iv) deprive the Employer of the right to terminate you at any time and without regard to any future vesting opportunity

that you may have. The grant of the Award shall not be interpreted as forming or amending an employment or service contract with the Company or the Employer.

(b) By accepting this Award, you acknowledge and agree that the right to continue vesting in the Award is earned only through Continuous Service (not through the act of being hired, being granted this Award or any other award or benefit) and that the Company has the right to reorganize, sell, spin-out or otherwise restructure one or more of its businesses or Affiliates at any time or from time to time, as it deems appropriate (a " ***reorganization*** "). You further acknowledge and agree that such a reorganization could result in the termination of your Continuous Service, or the termination of Affiliate status of the Employer and the loss of benefits available to you under this Agreement, including but not limited to, the termination of the right to continue vesting in the Award. You further acknowledge and agree that this Agreement, the Plan, the transactions contemplated hereunder and the vesting schedule set forth in this Agreement or any covenant of good faith and fair dealing that may be found implicit in any of them do not constitute an express or implied promise of continued engagement as an Employee or Consultant for the term of this Agreement, for any period, or at all, and will not interfere in any way with your right or the right of the Employer to terminate your Continuous Service at any time, with or without cause and, if permitted under applicable law, with or without notice, and will not interfere in any way with the Company's right to conduct a reorganization.

10. RESPONSIBILITY FOR TAXES.

(a) You acknowledge that, regardless of any action the Company or the Employer takes with respect to any or all income tax, social insurance, payroll tax, fringe benefit tax, payment on account or other tax related items related to your participation in the Plan and legally applicable to you (" ***Tax-Related Items*** "), the ultimate liability for all Tax-Related Items is and remains your responsibility and may exceed the amount actually withheld by the Company or the Employer, if any. You further acknowledge that the Company and the Employer (i) make no representations or undertakings regarding the treatment of any Tax-Related Items in connection with any aspect of your Restricted Stock Units, including, but not limited to, the grant of the Restricted Stock Units, the vesting and settlement of the Restricted Stock Units, the delivery or sale of any shares of Common Stock and the issuance of any dividends, and (ii) do not commit to and are under no obligation to structure the terms of the grant or any aspect of your Award to reduce or eliminate your liability for Tax-Related Items or achieve any particular tax result. You acknowledge and agree that you will not make any claim against the Company, or any of its Officers, Directors, Employees or Affiliates for Tax-Related Items arising from your Award. Further, if you are subject to Tax-Related Items in more than one jurisdiction, you acknowledge that the Company and/or the Employer may be required to withhold or account for Tax-Related Items in more than one jurisdiction.

(b) Prior to the relevant taxable or tax withholding event, as applicable, you agree to make adequate arrangements satisfactorily to the Company and/or the Employer to satisfy all Tax-Related Items. In this regard, you authorize the Company and/or the Employer, or their respective agents, at their discretion, to satisfy their withholding obligations with

regard to all Tax-Related Items by one or a combination of the following: (i) withholding from your wages or any other cash compensation otherwise payable to you by the Company and/or the Employer; (ii) causing you to tender a cash payment; (iii) permitting or requiring you to enter into a "same day sale" commitment with a broker-dealer that is a member of the Financial Industry Regulatory Authority (a " ***FINRA Dealer*** ") (if required, pursuant to this authorization and without further consent) whereby you irrevocably elect to sell a portion of the shares to be delivered in connection with your Restricted Stock Units to satisfy the Tax-Related Items and whereby the FINRA Dealer irrevocably commits to forward the proceeds necessary to satisfy the Tax-Related Items directly to the Company and/or the Employer, including a commitment pursuant to a previously established Company-approved 10b5-1 plan, and/or (iv) withholding shares of Common Stock from the shares of Common Stock issued or otherwise issuable to you in connection with the Award with a value equal to the amount of such Tax-Related Items or such other amount as may be permitted while still avoiding classification of the Award as a liability for financial accounting purposes; *provided, however* that if you are an Officer, then the Company will withhold a number of shares of Common Stock upon the relevant taxable or tax withholding event, as applicable, unless the use of such withholding method is not feasible under applicable tax or securities law or has materially adverse accounting consequences, as determined by the Board, in its sole discretion, in which case, the obligation for Tax-Related Items may be satisfied by one or a combination of methods (i)-(iii) above.

(c) The Company or the Employer may withhold or account for Tax-Related Items by considering applicable minimum statutory withholding amounts or other applicable withholding rates, including maximum applicable rates in your jurisdiction(s), in which case you may receive a refund of any over-withheld amount in cash and will have no entitlement to the Common Stock equivalent. Maximum tax rates are based on the applicable rates in your country, including your share of payroll or similar taxes, as provided in tax law, regulations, or the tax authority's administrative practices, not to exceed the highest rate in that jurisdiction, even if that rate exceeds the highest rate that may be applicable to you. If the obligation for Tax-Related Items is satisfied by withholding a number of shares of Common Stock, for tax purposes, you will be deemed to have been issued the full number of shares of Common Stock subject to the vested Restricted Stock Units, notwithstanding that a number of the shares of Common Stock is held back solely for the purpose of paying the Tax-Related Items.

(d) Unless any withholding obligation for Tax-Related Items is satisfied, the Company will have no obligation to deliver to you any shares of Common Stock or other consideration pursuant to this Award.

(e) In the event the obligation to withhold arises prior to the delivery to you of shares of Common Stock or it is determined after the delivery of shares of Common Stock to you that the amount of the withholding obligation was greater than the amount withheld, you agree to indemnify and hold the Company and the Employer harmless from any failure by the Company or the Employer to withhold the proper amount.

11. NATURE OF GRANT. In accepting your Award, you acknowledge, understand and agree that:

(a) the Plan is established voluntarily by the Company, it is discretionary in nature and it may be modified, amended, suspended or terminated by the Company at any time, to the extent permitted under the Plan;

(b) the Award is exceptional, voluntary and occasional and does not create any contractual or other right to receive future Awards (whether on the same or different terms), or benefits in lieu of an Award, even if an Award has been granted in the past;

(c) all decisions with respect to future Awards, if any, will be at the sole discretion of the Company;

(d) you are voluntarily participating in the Plan;

(e) the Award and the shares of Common Stock subject to the Award, and the income from and value of same, are an extraordinary item which, if the Employer is not the Company, does not constitute compensation of any kind for services of any kind rendered to the Employer, and is outside the scope of your employment contract, if any;

(f) the Award and the shares of Common Stock subject to the Award, and the income from and value of same, are not intended to replace any pension rights or compensation;

(g) the Award and the shares of Common Stock subject to the Award, and the income from and value of same, are not part of normal or expected compensation for purposes of, without limitation, calculating any severance, resignation, termination, redundancy, dismissal, end-of-service payments, bonuses, long-service awards, pension or retirement or welfare benefits or similar payments under any employee benefit plan sponsored by the Company or any Affiliate, except as such plan otherwise expressly provides (and the Company expressly reserves its rights to amend, modify, or terminate any of the Company's or any Affiliate's employee benefit plans);

(h) the future value of the underlying shares of Common Stock is unknown and cannot be predicted with certainty;

(i) no claim or entitlement to compensation or damages shall arise from forfeiture of the Award resulting from the termination of your Continuous Service (for any reason whatsoever and whether or not later found to be invalid or in breach of employment laws in the jurisdiction where you are employed or rendering services or the terms of your employment agreement, if any);

(j) unless otherwise provided herein, in the Plan or by the Company in its discretion, the Award and the benefits evidenced by this Agreement do not create any entitlement to have the Award or any such benefits transferred to, or assumed by, another

company nor to be exchanged, cashed out or substituted for, in connection with any corporate transaction affecting the shares of Common Stock;

(k) unless otherwise agreed with the Company, the Award and the shares of Common Stock subject to the Award, and the income from and value of same, are not granted as consideration for, or in connection with, the service you may provide as a director of an Affiliate; and

(l) if you are in Continuous Service outside the United States:

i. the Award and the shares of Common Stock subject to the Award, and the income from and value of same, are not part of normal or expected compensation for any purpose; and

ii. neither the Company, the Employer nor any other Affiliate shall be liable for any foreign exchange rate fluctuation between your local currency and the United States Dollar that may affect the value of the Award or of any amounts due to you pursuant to the vesting of the Award or the subsequent sale of any shares of Common Stock acquired upon vesting.

12. NO ADVICE REGARDING GRANT. The Company is not providing any tax, legal or financial advice, nor is the Company making any recommendations regarding your participation in the Plan, or your acquisition or sale of the underlying shares of Common Stock. You should consult with your own personal tax, financial and/or legal advisors regarding your participation in the Plan, and by accepting this Award, you have agreed that you have done so or knowingly and voluntarily declined to do so.

13. UNSECURED OBLIGATION. Your Award is unfunded, and as a holder of a vested Award, you will be considered an unsecured creditor of the Company with respect to the Company's obligation, if any, to issue shares pursuant to this Agreement. You will not have voting or any other rights as a stockholder of the Company with respect to the shares to be issued pursuant to this Agreement until such shares are issued to you. Upon such issuance, you will obtain full voting and other rights as a stockholder of the Company. Nothing contained in this Agreement, and no action taken pursuant to its provisions, will create or be construed to create a trust of any kind or a fiduciary relationship between you and the Company or any other person.

14. OTHER DOCUMENTS . You hereby acknowledge receipt or the right to receive a document providing the information required by Rule 428(b)(1) promulgated under the Securities Act, which includes the Plan prospectus. In addition, you acknowledge receipt of the Company's policy permitting certain individuals to sell shares only during certain "window" periods and the Company's insider trading policy, in effect from time to time and understand that this policy applies to shares received under this Award.

15. NOTICES; ELECTRONIC DELIVERY/ACCEPTANCE. Any notices provided for in your Award or the Plan will be given in writing and will be deemed effectively given upon

receipt or, in the case of notices delivered by the Company to you, five (5) days after deposit in the United States mail, postage prepaid, addressed to you at the last address you provided to the Company. Notwithstanding the foregoing, the Company may, in its sole discretion, decide to deliver any documents and transmit or require you to transmit notices related to participation in the Plan and this Award by electronic means. You hereby consent to receive such documents and notices, and to give such notices, by electronic delivery and to participate in the Plan through the on-line or electronic system established and maintained by the Company or a third party designated by the Company from time to time.

16. GOVERNING PLAN DOCUMENT/RECOUPMENT. Your Award is subject to all the provisions of the Plan, the provisions of which are hereby made a part of your Award, and is further subject to all interpretations, amendments, rules and regulations which may from time to time be promulgated and adopted pursuant to the Plan. In addition, this Award (and any shares issued under this Award) is subject to recoupment in accordance with the Dodd–Frank Wall Street Reform and Consumer Protection Act and any implementing regulations thereunder, any clawback policy adopted by the Company and any compensation recovery policy otherwise required by applicable law.

17. LANGUAGE. You acknowledge that you are sufficiently proficient in the English language, or have consulted with an advisor who is sufficiently proficient in English, so as to allow you to understand the terms and conditions of this Agreement. Further, if you have received this Agreement or any other document related to the Plan translated into a language other than English and if the meaning of the translated version is different than the English version, the English version will control.

18. INSIDER TRADING RESTRICTIONS/MARKET ABUSE LAWS. You may be subject to insider trading restrictions and/or market abuse laws based on the exchange on which the shares of Common Stock are listed and in applicable jurisdictions, including the United States and your country or your broker's country, if different, which may affect your ability to accept, acquire, sell or otherwise dispose of shares of Common Stock, rights to shares of Common Stock (e.g., Restricted Stock Units) or rights linked to the value of shares of Common Stock during such times as you are considered to have "inside information" regarding the Company (as defined by the laws in applicable jurisdictions). Local insider trading laws and regulations may prohibit the cancellation or amendment of orders you placed before you possessed inside information. Furthermore, you could be prohibited from (i) disclosing the inside information to any third party, which may include fellow employees and (ii) "tipping" third parties or causing them otherwise to buy or sell securities. Any restrictions under these laws or regulations are separate from and in addition to any restrictions that may be imposed under any applicable insider trading policy of the Company. You acknowledge that it is your responsibility to comply with any applicable restrictions and you should speak with your personal legal advisor on this matter.

19. FOREIGN ASSETS/ACCOUNT AND TAX REPORTING, EXCHANGE CONTROLS. Your country may have certain foreign asset, account and/or tax reporting requirements and exchange controls which may affect your ability to acquire or hold shares of Common Stock

under the Plan or cash received from participating in the Plan (including from any dividends received or sale proceeds arising from the sale of shares of Common Stock) in a brokerage or bank account outside your country. You understand that you may be required to report such accounts, assets or transactions to the tax or other authorities in your country. You also may be required to repatriate sale proceeds or other funds received as a result of participation in the Plan to your country through a designated bank or broker and/or within a certain time after receipt. In addition, you may be subject to tax payment and/or reporting obligations in connection with any income realized under the Plan and/or from the sale of shares of Common Stock. You acknowledge that you are responsible for complying with all such requirements, and that you should consult personal legal and tax advisors, as applicable, to ensure compliance.

20. APPENDIX. Notwithstanding any provisions in this Agreement, your Award shall be subject to the special terms and conditions for your country set forth in the Appendix attached hereto as Attachment II. Moreover, if you relocate to one of the countries included therein, the terms and conditions for such country will apply to you to the extent the Company determines that the application of such terms and conditions is necessary or advisable for legal or administrative reasons. The Appendix constitutes part of this Agreement.

21. IMPOSITION OF OTHER REQUIREMENTS. The Company reserves the right to impose other requirements on your participation in the Plan, on the Award and on any shares of Common Stock acquired under the Plan, to the extent the Company determines it is necessary or advisable in order to comply with local law or facilitate the administration of the Plan.

22. SEVERABILITY. If all or any part of this Agreement or the Plan is declared by any court or governmental authority to be unlawful or invalid, such unlawfulness or invalidity will not invalidate any portion of this Agreement or the Plan not declared to be unlawful or invalid. Any Section of this Agreement (or part of such a Section) so declared to be unlawful or invalid will, if possible, be construed in a manner which will give effect to the terms of such Section or part of a Section to the fullest extent possible while remaining lawful and valid.

23. GOVERNING LAW/VENUE. The interpretation, performance and enforcement of this Agreement will be governed by the law of the state of Delaware without regard to such state's conflicts of laws rules. For purposes of litigating any dispute that arises directly or indirectly from the relationship of the parties evidenced by this grant or the Agreement, the parties hereby submit to and consent to the exclusive jurisdiction of the State of California and agree that such litigation shall be conducted only in the courts of Santa Clara County, California, or the federal courts for the United States for the Northern District of California, and no other courts, where this grant is made and/or to be performed.

24. MISCELLANEOUS.

(a) The rights and obligations of the Company under your Award will be transferable to any one or more persons or entities, and all covenants and agreements hereunder will inure to the benefit of, and be enforceable by the Company's successors and

assigns. Your rights and obligations under your Award may only be assigned with the prior written consent of the Company.

(b) You agree upon request to execute any further documents or instruments necessary or desirable in the sole determination of the Company to carry out the purposes or intent of your Award.

(c) You acknowledge and agree that you have reviewed your Award in its entirety, have had an opportunity to obtain the advice of counsel prior to executing and accepting your Award, and fully understand all provisions of your Award.

(d) All obligations of the Company under the Plan and this Agreement will be binding on any successor to the Company, whether the existence of such successor is the result of a direct or indirect purchase, merger, consolidation, or otherwise, of all or substantially all of the business and/or assets of the Company.

25. AMENDMENT. This Agreement may not be modified, amended or terminated except by an instrument in writing, signed by you and by a duly authorized representative of the Company. Notwithstanding the foregoing, this Agreement may be amended solely by the Board by a writing which specifically states that it is amending this Agreement, so long as a copy of such amendment is delivered to you, and provided that no such amendment adversely affecting your rights hereunder may be made without your written consent. Without limiting the foregoing, the Board reserves the right to change, by written notice to you, the provisions of this Agreement in any way it may deem necessary or advisable to carry out the purpose of the grant as a result of any change in applicable laws or regulations or any future law, regulation, ruling, or judicial decision, provided that any such change will be applicable only to rights relating to that portion of the Award which is then subject to restrictions as provided in this Agreement.

26. COMPLIANCE WITH SECTION 409A OF THE CODE . This Award is intended to comply with U.S. Treasury Regulation Section 1.409A-1(b)(4) and thus to not be treated as “deferred compensation”, and will be construed and administered in such a manner, and any ambiguous or missing terms that may otherwise be supplied from and/or defined under Code Section 409A in a manner that fulfills such intention hereby incorporated by reference. Each installment of Restricted Stock Units that vests hereunder is intended to constitute a “separate payment” for purposes of Treasury Regulation Section 1.409A-2(b)(2). Notwithstanding the foregoing, if it is determined that the Award fails to satisfy the requirements of the short-term deferral rule and is otherwise not exempt from, and determined to be deferred compensation subject to Code Section 409A, this Award shall comply with Code Section 409A to the extent necessary to avoid adverse personal tax consequences and any ambiguities herein shall be interpreted accordingly. If it is determined that the Award is deferred compensation subject to Code Section 409A and you are a “specified employee” (as determined under Code Section 409A) on your Separation from Service, then the issuance of any shares, cash or other property that would otherwise be made on the date of your Separation from Service (or within the first six months thereafter as a result of your Separation from Service) will not be made on the originally scheduled date(s) and will instead be issued in a

lump sum on the date that is six months and one day after the date of the Separation from Service, but if and only if such delay in the issuance is necessary to avoid the imposition of taxation on you in respect of the shares, cash or property under Code Section 409A.

ATTACHMENT II

APPENDIX TO
NVIDIA CORPORATION
GLOBAL RESTRICTED STOCK UNIT AGREEMENT

ADDITIONAL TERMS AND CONDITIONS FOR NON-U.S. PARTICIPANTS

Capitalized terms used but not defined in this Appendix have the meanings set forth in the Plan and/or in the Global Restricted Stock Unit Agreement.

Terms and Conditions

This Appendix includes additional terms and conditions that govern the Restricted Stock Units granted to you under the Plan if you reside and/or work in one of the countries listed below. If you are a citizen or resident (or are considered as such for local law purposes) of a country other than the country in which you are currently residing and/or working, or if you relocate to another country after the grant of the Restricted Stock Units, the Company shall, in its discretion, determine to what extent the additional terms and conditions contained herein shall be applicable to you.

Notifications

This Appendix may also include information regarding exchange controls and certain other issues of which you should be aware with respect to your participation in the Plan. The information is based on the securities, exchange control and other laws in effect in the respective countries as of January 2019. Such laws are often complex and change frequently. As a result, the Company strongly recommends that you not rely on the information in this Appendix as the only source of information relating to the consequences of your participation in the Plan because the information may be out of date at the time the Restricted Stock Units vest, dividends are paid on shares of Common Stock acquired under the Plan or you sell shares of Common Stock acquired under the Plan.

In addition, the information contained herein is general in nature and may not apply to your particular situation, and the Company is not in a position to assure you of a particular result. Accordingly, you are advised to seek appropriate professional advice as to how the relevant laws in your country may apply to your situation.

Finally, if you are a citizen or resident (or are considered as such for local law purposes) of a country other than the country in which you are currently residing and/or working, or if you relocate to another country after the grant of the Restricted Stock Units, the notifications contained herein may not be applicable to you in the same manner.

DATA PRIVACY PROVISIONS FOR ALL NON-U.S. PARTICIPANTS

TERMS AND CONDITIONS

Data Privacy Consent For Participants Working and/or Residing Outside the European Union/European Economic Area .

(a) ***Data Collection and Usage . The Company and the Employer collect, process and use certain personal information about you, including, but not limited to, your name, home address and telephone number, email address, date of birth, social insurance, passport or other identification number, salary, nationality, job title, any shares of Common Stock or directorships held in the Company, details of all Restricted Stock Units or any other entitlement to shares of Common Stock or equivalent benefits awarded, canceled, exercised, vested, unvested or outstanding in your favor ("Data"), for the purposes of implementing, administering and managing the Plan. The legal basis, where required, for the processing of Data is your consent.***

(b) ***Stock Plan Administration Service Providers . The Company will transfer Data to Charles Schwab & Co., Inc. (including its affiliated companies) (collectively, "Schwab"), which is assisting the Company with the implementation, administration and management of the Plan. The Company may select different or additional service providers in the future and share Data with such other provider(s) serving in a similar manner. You may be asked to agree on separate terms and data processing practices with Schwab, with such agreement being a condition to the ability to participate in the Plan.***

(c) ***International Data Transfers . The Company and Schwab are based in the United States. Your country or jurisdiction may have different data privacy laws and protections than the United States. For example, the European Commission has issued a limited adequacy finding with respect to the United States that applies only to the extent companies register for the EU-U.S. Privacy Shield program. The Company's legal basis, where required, for the transfer of Data is your consent.***

(d) ***Data Retention . The Company will hold and use Data only as long as is necessary to implement, administer and manage your participation in the Plan, or as required to comply with legal or regulatory obligations, including under tax, exchange control, labor and securities laws.***

(e) ***Voluntariness and Consequences of Consent Denial or Withdrawal . Participation in the Plan is voluntary, and you are providing the consents herein on a purely voluntary basis. If you do not consent, or if you later seek to revoke your consent, your salary from or employment and career with the Employer will not be affected; the only consequence of refusing or withdrawing consent is that the Company would not be able to grant the Restricted Stock Units or other equity awards to you or administer or maintain such awards.***

(f) ***Data Subject Rights . You may have a number of rights under data privacy laws in your jurisdiction. Depending on where you are based, such rights may include the right to (i) request access or copies of Data the Company processes, (ii) rectification of incorrect Data, (iii)***

deletion of Data, (iv) restrictions on processing of Data, (v) portability of Data, (vi) lodge complaints with competent authorities in your jurisdiction, and/or (vii) receive a list with the names and addresses of any potential recipients of Data. To receive clarification regarding these rights or to exercise these rights, you can contact your local HR representative.

(g) ***Additional Acknowledgment/Consent . You understand that the Company may rely on a different basis for the processing or transfer of Data in the future and/or request that you provide another data privacy consent. If applicable, you agree that upon request of the Company or the Employer, you will provide an executed acknowledgement or data privacy consent form (or any other agreements or consents) that the Company and/or the Employer may deem necessary to obtain from you for the purpose of administering your participation in the Plan in compliance with the data privacy laws in your country, either now or in the future. You understand and agree that you will not be able to participate in the Plan if you fail to provide any such consent or agreement requested by the Company and/or the Employer.***

Data Privacy Notification For Participants Working and/or Residing In the European Union/European Economic Area ***. The Company collects, processes, uses and transfers certain personally-identifiable information about you for the exclusive legitimate purpose of granting Restricted Stock Units and implementing, administering and managing your participation in the Plan. Specifics of the data processing are described below.***

(a) Purposes and Legal Bases of Processing . The Company processes the Personal Data (as defined below) for the purpose of performing its contractual obligations under this Agreement, granting Restricted Stock Units, implementing, administering and managing your participation in the Plan and facilitating compliance with applicable tax, exchange control, securities and labor law. The legal basis for the processing of the Personal Data (as defined below) by the Company and the third-party service providers described below is the necessity of the data processing for the Company to perform its contractual obligations under this Agreement and for the Company's legitimate business interests of managing the Plan and generally administering employee equity awards.

(b) Personal Data Subject to Processing . The Company collects, processes and uses the following types of personal data about you: your name, home address, email address, date of birth, social insurance, passport number or other identification number, any shares of Common Stock or directorships held in the Company, details of all Restricted Stock Units or any other entitlement to shares of Common Stock awarded, canceled, settled, vested, unvested or outstanding in your favor, which the Company receives from you or the Employer ("Personal Data").

(c) Stock Plan Administration Service Providers . The Company transfers Personal Data to Charles Schwab & Co., Inc. and its affiliated companies ("Schwab"), an independent stock plan administrator with operations, relevant to the Company, in the United States, which assists the Company with the implementation, administration and management of the Plan. In the future, the Company may select different service providers and may share Personal Data with such service providers. Schwab will open an account for you to receive and trade shares of Common Stock. You will be asked to agree on separate terms and data processing practices with Schwab, which is a condition of your ability to participate in the Plan. Your Personal Data will only be accessible

by those individuals requiring access to it for purposes of implementing, administering and operating your participation in the Plan. You understand that you may request a list with the names and addresses of any potential recipients of Personal Data by contacting your local human resources manager.

(d) Other Recipients . The Company may further transfer Personal Data to other third party service providers, if necessary to ensure compliance with applicable tax, exchange control, securities and labor law. Such third party service providers may include the Company's outside legal counsel as well as the Company's auditor. Wherever possible, the Company will anonymize data, but you understand that your Personal Data may need to be transferred to such providers to ensure compliance with applicable law and/or tax requirements.

(e) International Data Transfers . The Company and its service providers, including, without limitation, Schwab, operate, relevant to the Company, in the United States, which means that it will be necessary for Personal Data to be transferred to, and processed in, the United States. You understand and acknowledge that the United States is not subject to an unlimited adequacy finding by the European Commission and that your Personal Data may not have an equivalent level of protection as compared to your country of residence.

To provide appropriate protection of your Personal Data, the Company complies with the EU-U.S. Privacy Shield Framework and Swiss-U.S. Privacy Shield Framework as set forth by the U.S. Department of Commerce regarding the collection, use, and retention of Personal Data transferred from the European Union and Switzerland to the United States. The Company has certified to the Department of Commerce that it adheres to the Privacy Shield Principles.

If there is any conflict between the terms in this Agreement and the Privacy Shield Principles, the Privacy Shield Principles shall govern. To learn more about the Privacy Shield program, and to view the Company's certification, please visit www.privacyshield.gov.

(f) Data Retention . The Company will use the Personal Data only as long as necessary to implement, administer and manage your participation in the Plan, or as required to comply with legal or regulatory obligations, including tax, exchange control, labor and securities laws.

(g) Data Subject Rights . To the extent provided by law, you have the right to:

i. Request access to and obtain a copy of your Personal Data;

ii. Request rectification (or correction) of Personal Data that is inaccurate;

iii. Request erasure (or deletion) of Personal Data that is no longer necessary to fulfill the purposes for which it was collected, or does not need to be retained by the Company for other legitimate purposes;

iv. Restrict or object to the processing of your Personal Data; and

v. If applicable, request your Personal Data be ported (transferred) to another company.

Subject to the applicable data protection laws, application of the above rights may vary depending on the type of data involved, and the Company's particular basis for processing the Personal Data.

To make a request to exercise one of the above rights, you can contact your local HR representative. The Company will consider and act upon any requests in accordance with applicable data protection laws. The Company may request specific information from you to enable it to confirm your identity and right to access, as well as to search for and provide you with the Personal Data that it holds about you.

(h) Contractual Requirement . Your provision of Personal Data and its processing as described above is a contractual requirement and a condition to your ability to participate in the Plan. You understand that, as a consequence of you refusing to provide Personal Data, the Company may not be able to allow you to participate in the Plan, grant Restricted Stock Units to you or administer or maintain such Restricted Stock Units. However, your participation in the Plan is purely voluntary. While you will not receive Restricted Stock Units if he or she decides against participating in the Plan or providing Personal Data as described above, your career and salary will not be affected in any way. For more information on the consequences of the refusal to provide Personal Data, you may contact your local HR representative.

(i) How to Contact Us . For copies of additional privacy documents mentioned in this Agreement, or if you have privacy concerns or questions related to this Agreement, you may contact your local HR representative.

AUSTRALIA

Notifications

Securities Law Information . This offer of Restricted Stock Units is intended to comply with the provisions of the Corporations Act 2001, Australian Securities and Investments Commission (" ***ASIC*** ") Regulatory Guide 49 and ASIC Class Order 14/1000. Additional details are set forth in the Offer Document for the Offer of Restricted Stock Units to Australian-Resident Participants, which is being provided to you along with this Agreement.

Exchange Control Information . Exchange control reporting is required for cash transactions exceeding AUD 10,000 and international fund transfers. The Australian bank assisting with the transaction will file the report. If there is no Australian bank involved in the transfer, you will be required to file the report.

Tax Information . The Plan is a plan to which Subdivision 83A-C of the Income Tax Assessment Act 1997 (Cth) (the " **Act** ") applies, subject to the conditions in the Act.

BRAZIL

Terms and Conditions

Nature of Grant . This provision supplements Section 11 of the Global Restricted Stock Unit Agreement:

You acknowledge and agree that (i) by accepting this Award, you are making an investment decision, and (ii) the value of the underlying shares of Common Stock is not fixed and may increase or decrease over the vesting period, without compensation to you.

Further, you acknowledge and agree that, for all legal purposes, (i) any benefits provided to you under the Plan are unrelated to your employment or service; (ii) the Plan is not a part of the terms and conditions of your employment or service; and (iii) the income from your participation in the Plan, if any, is not part of your remuneration from employment or service.

Compliance with Law . By accepting this Award, you agree to comply with all applicable Brazilian laws and pay any and all applicable Tax-Related Items associated with the vesting or settlement of the Award, the sale of shares of Common Stock acquired under the Plan and the receipt of any dividends paid on such shares of Common stock.

Notifications

Exchange Control Reporting . Brazilian residents and persons domiciled in Brazil are required to submit an annual declaration of assets and rights held outside of Brazil to the Central Bank of Brazil if the aggregate value of such assets and rights is equal to or greater than US$100,000. Quarterly reporting is required if such value exceeds US$100,000,000. The assets and rights that must be reported include shares of Common Stock acquired under the Plan. The thresholds are subject to change annually.

Tax on Financial Transaction (IOF) . Repatriation of funds into Brazil and the conversion between Brazilian Real and United States Dollars associated with such fund transfers may be subject to the Tax on Financial Transactions. It is your responsibility to comply with any applicable Tax on Financial Transactions arising from your participation in the Plan. You should consult with your personal tax advisor for additional details.

CANADA

Terms and Conditions

Form of Settlement . Restricted Stock Units granted to individuals resident in Canada shall be paid in shares of Common Stock only. In no event shall any Restricted Stock Units be paid in cash, notwithstanding any discretion contained in the Plan to the contrary.

The following provisions apply if you are a resident of Quebec:

Language Consent . The parties acknowledge that it is their express wish that this Agreement, as well as all documents, notices and legal proceedings entered into, given or instituted pursuant hereto or relating directly or indirectly hereto, be drawn up in English.

Les parties reconnaissent avoir exigé la rédaction en anglais de cette convention ("Agreement"), ainsi que de tous documents exécutés, avis donnés et procédures judiciaires intentées, directement ou indirectement, relativement à la présente convention.

Data Privacy . This provision supplements the Data Privacy Consent For Participants Working and/or Residing Outside the European Union/European Economic Area provision of this Appendix:

You hereby authorize the Company and the Company's representatives to discuss with and obtain all relevant information from all personnel, professional or not, involved in the administration and operation of the Plan. You further authorize the Company, the Employer and/or any other Affiliate to disclose and discuss such information with their advisors. You also authorize the Company, the Employer and/or any other Affiliate to record such information and to keep such information in your employee file.

Notifications

Securities Law Information . You are permitted to sell shares of Common Stock acquired through the Plan through the designated broker appointed by the Company provided the resale of such shares takes place outside of Canada and through the facilities of a stock exchange, which should be the case because the Common Stock is currently listed on the Nasdaq Global Select Market.

Foreign Asset/Account Reporting Information . Specified Foreign property, including Restricted Stock Units, shares of Common Stock acquired under the Plan and other rights to receive shares (e.g., options) of a non-Canadian company, held by a Canadian resident must generally be reported annually on a Form T1135 (Foreign Income Verification Statement) if the total cost of the specified foreign property exceeds C$100,000 at any time during the year. Restricted Stock Units must be reported – generally at a nil cost – if the C$100,000 cost threshold is exceeded because other specified foreign property is held by you. When shares of Common Stock are acquired, their cost generally is the adjusted cost base ("ACB") of the shares. The ACB would ordinarily equal the fair market value of the shares at the time of acquisition, but if you own other shares of the same company, this ACB may have to be averaged with the ACB of the other shares. You should consult with your personal tax advisor to determine your reporting requirements.

CHINA

Terms and Conditions

*The following provisions apply to you if you are subject to exchange control regulations in the People's Republic of China (" **China** "), including the requirements imposed by the State Administration of Foreign Exchange (" **SAFE** "), as determined by the Company in its sole discretion:*

Settlement of Award and Sale of Shares . This provision supplements Section 6 of the Global Restricted Stock Unit Agreement.

You will not be permitted to vest in any shares of Common Stock unless and until the necessary approvals for the Plan have been obtained from SAFE and remain in place, as determined by the Company in its sole discretion. Further, the Company is under no obligation to issue shares of Common Stock if the Company has not or does not obtain SAFE approval or if any such SAFE approval subsequently becomes invalid or ceases to be in effect by the time you vest in the Restricted Stock Units.

To facilitate compliance with regulatory requirements in China, you understand and agree that any shares of Common Stock you acquire upon vesting of your Restricted Stock Units may be immediately sold at vesting or, at the Company's discretion, at a later time. You agree that the Company is authorized to instruct its designated broker to assist with the sale of such shares of Common Stock (on your behalf pursuant to this authorization and without further consent) and you expressly authorize the designated broker to complete the sale of such shares. You acknowledge that the designated broker is under no obligation to arrange for the sale of the shares of Common Stock at any particular price. Upon the sale of the shares of Common Stock, the Company agrees to pay the cash proceeds from the sale, less any brokerage fees or commissions, to you in accordance with applicable exchange control laws and regulations and provided any liability for Tax-Related Items resulting from your participation in the Plan has been satisfied.

If the Company, in its discretion, does not exercise its right to require the sale of shares of Common Stock upon vesting, as described in the preceding paragraph, and your employment with the Employer terminates, you understand and agree to sell these shares within 90 days after your termination date. You further agree that if you do not sell these shares within 90 days after your termination date, the Company is authorized to instruct its designated broker to assist with the mandatory sale of such shares (on your behalf pursuant to this authorization) and you expressly authorize the Company's designated broker to complete the sale of such shares. You acknowledge that the Company's designated broker is under no obligation to arrange for the sale of the shares at any particular price. Upon the sale of the shares, the Company agrees to pay you the cash proceeds from the sale of the shares, less any brokerage fees or commissions and subject to any withholding obligation for Tax-Related Items. Such payment will have to be effectuated through a special exchange control account established by the Company or an Affiliate in China. If the funds are converted into local currency, the Company will not bear the exchange rate risk and does not undertake to convert the funds at any particular time.

Exchange Control Requirements . You understand and agree that you will not be permitted to transfer any shares of Common Stock acquired under the Plan out of the account established for you with the Company's designated broker and that you will be required to immediately repatriate to China any cash proceeds from the sale of the shares of Common Stock acquired under the Plan or from dividends paid on such shares. You further understand that such repatriation of cash proceeds will need to be effectuated through a special exchange control account established by the Company or an Affiliate in China, and you hereby consent and agree that any proceeds from the sale of shares of Common Stock or dividends paid on such shares may be transferred to such special account prior to being delivered to you.

The proceeds may be paid to you in U.S. dollars or local currency at the Company's discretion. In the event the proceeds are paid in U.S. dollars, you understand that you will be required to set up a U.S. dollar bank account in China and provide the bank account details to the Employer and/or the Company, so that the proceeds may be deposited into this account. If the proceeds are paid in local currency, you agree to bear any currency fluctuation risk between the time the shares of Common Stock are sold or dividends on such shares are paid and the time the proceeds are distributed to you through any such special account.

You agree to comply with any other requirements that may be imposed by the Company in the future in order to facilitate compliance with exchange control requirements in China.

Notifications

Exchange Control Information . Chinese residents must report to SAFE all details of foreign financial assets and liabilities, as well as details of any economic transactions conducted with non-Chinese residents (including the Company), either directly or through financial institutions.

CZECH REPUBLIC

Notifications

Exchange Control Information . The Czech National Bank (" ***CNB*** ") may require you to fulfill certain notification duties in relation to the shares of Common Stock acquired under the Plan or any dividends paid on such shares, and the opening and maintenance of a foreign account. However, because exchange control regulations change frequently and without notice, you should consult your personal legal advisor prior to vesting to ensure compliance with current regulations. It is your responsibility to comply with applicable Czech exchange control laws.

FINLAND

There are no country specific provisions.

FRANCE

Terms and Conditions

Restricted Stock Units Not French-qualified . The Restricted Stock Units granted under this Agreement are not intended to qualify for special tax and social security treatment pursuant to Sections L. 225-197-1 to L. 225-197-6 of the French Commercial Code, as amended

Language Consent . By accepting the Award, you confirm having read and understood the Plan and the Agreement, which were provided in the English language. You accept the terms of those documents accordingly.

Consentement Relatif à la Langue Utilisée . En acceptant cette Attribution, vous confirmez avoir lu et comprendre le Plan et ce Contrat qui ont été transmis en langue anglaise. Vous acceptez les dispositions de ces documents en connaissance de cause.

Notifications

Foreign Asset/Account Reporting Information . French residents holding cash or securities (including shares of Common Stock acquired under the Plan) outside of France or maintaining foreign bank or brokerage account (including accounts opened or closed during the tax year) must declare such assets and accounts to the French tax authorities when filing an annual tax return. Failure to comply could trigger significant penalties.

GERMANY

Notifications

Exchange Control Information . Cross-border payments in excess of €12,500 in connection with the sale of securities (including shares of Common Stock acquired under the Plan) and/or the receipt of dividends paid on securities must be reported on a monthly basis to the German Federal Bank (Bundesbank). If you receive a payment in excess of this amount, you must report the payment to Bundesbank electronically by the fifth day of the month following the month in which the payment was received. The form of the report (" *Allgemeine Meldeportal Statistik* ") can be accessed via the Bundesbank's website (www.bundesbank.de) and is available in both German and English.

Foreign Asset/Account Reporting Information . German residents holding shares of Common Stock must notify their local tax office of the acquisition of shares of Common Stock when they file their tax returns for the relevant year if the value of the shares of Common Stock acquired exceeds €150,000 or in the unlikely event that the resident holds Common Stock exceeding 10% of the Company's total Common Stock outstanding.

HONG KONG

Terms and Conditions

Form of Settlement . Restricted Stock Units granted to individuals resident in Hong Kong shall be paid in shares of Common Stock only. In no event shall any Restricted Stock Units be paid in cash, notwithstanding any discretion contained in the Plan to the contrary.

Issuance of Shares and Sale of Shares . This provision supplements Sections 2 and 3 of the Global Restricted Stock Unit Agreement:

Any shares of Common Stock issued in settlement of the Award are accepted as a personal investment. In the event shares of Common Stock subject to the Award are issued to you within six months of the Date of Grant, you agree that you will not offer the shares of Common Stock to the public in Hong Kong or otherwise dispose of any such shares prior to the six-month anniversary of the Date of Grant.

Notifications

Securities Law Information : *WARNING: The contents of this document have not been reviewed by any regulatory authority in Hong Kong. You should exercise caution in relation to the offer. If you are in any doubt about any of the contents of the Agreement, the Plan or any other incidental communication materials, you should obtain independent professional advice. The Award and any shares of Common Stock issued upon settlement do not constitute a public offering of securities under Hong Kong law and are available only to service providers of the Company and its Affiliates. The Agreement, the Plan and other incidental communication materials have not been prepared in accordance with and are not intended to constitute a "prospectus" for a public offering of securities under the applicable securities legislation in Hong Kong. The Awards and any related documents are intended only for the personal use of each eligible service provider of the Employer, the Company or any other Affiliate and may not be distributed to any other person.*

INDIA

Notifications

Exchange Control Information . You understand that you must repatriate any proceeds from the sale of shares of Common Stock acquired under the Plan to India and convert the proceeds into local currency within such time as prescribed under applicable Indian exchange control laws, which may be amended from time to time. You must obtain a foreign inward remittance certificate (" ***FIRC*** ") from the bank where you deposit the foreign currency and must maintain the FIRC as evidence of the repatriation of funds in the event the Reserve Bank of India or the Employer requests proof of repatriation. You also understand that it is your responsibility to comply with all exchange control laws in India and that you should consult with your own legal advisor about the applicable requirements.

Foreign Asset/Account Reporting Information . Indian residents must declare the following items in their annual tax returns: (i) any foreign assets held (including shares of Common Stock acquired under the Plan), and (ii) any foreign bank accounts for which the resident has signing authority. It is your responsibility to comply with applicable tax laws in India. You should consult with your personal tax advisor to ensure that you are properly reporting your foreign assets and bank accounts.

ISRAEL

Terms and Conditions

The following provision applies to you if you are in Israel on the Date of Grant.

Israeli Sub-plan . You acknowledge and agree that the Awards are granted under the Israeli sub-plan to the Plan which contains additional terms and conditions that govern your Award. In addition, your Award is subject to Section 102 capital gains route of the Income Tax Ordinance (New Version) – 1961, the rules and regulations promulgated in connection therewith (the " ***Ordinance*** "), any tax ruling to be obtained by the Company (collectively, the " ***CGR*** "), and the Trust Agreement, copies of which have been provided to you or made available for your review. You agree that the Awards will be issued to and controlled by a trustee appointed by the

Company (the " ***Trustee*** ") for your benefit, pursuant to the terms of the CGR and the Trust Agreement. You also confirm that you are familiar with the terms and provisions of Section 102 of the Ordinance and the CGR and understand that the Awards will be subject to the lockup period and you undertake not to sell or require the Trustee to release the Awards or the underlying shares of Common Stock, prior to the expiration of the lockup period, unless you pay all taxes which may arise in connection with such sale and/or transfer.

The classification of the Restricted Stock Units as Trustee 102 Awards is conditioned upon the approval of the Plan, the Sub-Plan and the Trustee by the Israeli Tax Authorities (" ***ITA*** "). In the event that such approval is not granted, regardless of reason, then the Restricted Stock Unit shall be deemed to be Non-Trustee 102 Award, unless otherwise determined by the ITA. In addition, the Company does not undertake to maintain the tax-qualified status and you acknowledge that you will not be entitled to damages of any nature whatsoever if the Award becomes disqualified and no longer qualifies under the capital gains tax route.

The Restricted Stock Unit will be issued to the Trustee. The Trustee will hold the units and the shares of Common Stock to be issued and all other shares of Common Stock received following any realization of rights, including bonus shares, dividends (whether in cash or in kind), or other rights issued or distributed in connection with the Restricted Stock Unit or the shares of Common Stock, in trust, until the later of: (i) the expiration of the minimum Lockup Period as required under Section 102, or (ii) the full payment of all requisite taxes by you, as shall be determined by the Company and the Trustee, in their sole discretion. You agree to comply with any additional requirements that may be imposed by a designated trustee for the Plan.

The Company and/or its Affiliate and/or the Trustee shall be entitled to withhold Taxes according to requirement of any applicable laws, rules and regulations and the CGR. The Company and/or the Trustee shall not be required to release any Restricted Stock Units and/or shares of Common Stock to you or to any third party until all required tax payments have been fully made or will be made to the full satisfaction of the Company and the Trustee.

The following provision applies if you transfer into Israel after the Date of Grant.

Settlement **.** The following provision supplements Sections 2 and 3 of the Global Restricted Stock Unit Agreement.

At the discretion of the Company, you may be subject to an immediate forced sale restriction, pursuant to which all shares of Common Stock acquired at vesting will be immediately sold and you will receive the sale proceeds less any Tax-Related Items and applicable broker fees and commissions. In this case, you will not be entitled to hold any shares of Common Stock acquired at vesting.

ITALY

Terms and Conditions

Plan Document Acknowledgment. In participating in the Plan, you acknowledge that you have received a copy of the Plan and the Agreement and have reviewed the Plan and the Agreement in their entirety and fully understand and accept all provisions of the Plan and the Agreement. You further acknowledge that you have read and specifically and expressly approve the Sections of the Agreement addressing (i) Compliance with Law (Section 4 of the Global Restricted Stock Unit Agreement), (ii) Limitations on Transfer (Section 5 of the Global Restricted Stock Unit Agreement), (iii) Responsibility for Taxes (Section 10 of the Global Restricted Stock Unit Agreement), (iv) Nature of Grant (Section 11 of the Global Restricted Stock Unit Agreement), , (v) Imposition of Other Requirements (Section 21 of the Global Restricted Stock Unit Agreement), (vi) Governing Law/Venue (Section 23 of the Global Restricted Stock Unit Agreement) and (vii) the Data Privacy Notification For Participants Working and/or Residing In the European Union/European Economic Area provision of this Appendix.

Notifications

Foreign Asset/Account Reporting Information. Italian residents who, during any fiscal year, hold investments or financial assets outside of Italy (*e.g.* , cash, shares of Common Stock) which may generate income taxable in Italy (or who are the beneficial owners of such an investment or asset even if not directly holding the investment or asset), are required to report such investments or assets on the annual tax return for such fiscal year (on UNICO Form, RW Schedule, or on a special form if not required to file a tax return).

Foreign Financial Asset Tax Notification**.** The value of any shares of Common Stock (and certain other foreign assets) an Italian resident holds outside Italy may be subject to a foreign financial assets tax. You should consult your personal tax advisor for additional information.

JAPAN

Notifications

Foreign Asset/Account Reporting Information. Japanese residents are required to report details of any assets held outside Japan (e.g., shares of Common Stock acquired under the Plan) to the extent such assets have a total net fair market value exceeding ¥50,000,000 as of December 31 each year. You should consult with your personal tax advisor to ensure that you are properly complying with applicable reporting requirements in Japan.

KOREA

Notifications

Exchange Control Information. If you realize US$500,000 or more from the sale of shares of Common Stock or the receipt of dividends paid on such shares in a single transaction, Korean exchange control laws require you to repatriate the proceeds to Korea within three years of receipt if the transaction occurred before July 18, 2017. You should consult a personal tax advisor to determine whether this repatriation requirement applies to a particular transaction.

Foreign Asset/Account Reporting Information . Korean residents are required to declare foreign accounts (*i.e.* , non-Korean bank accounts, brokerage accounts, etc.) to the Korean tax authorities if the monthly balance of such accounts exceeds a certain limit (currently KRW

500 million or an equivalent amount in foreign currency) on any month-end date during a calendar year.

MEXICO

Terms and Conditions

Acknowledgement of the Agreement . By accepting the Award, you acknowledge that you have received a copy of the Plan and the Agreement, which you have reviewed. You further acknowledge that you accept all the provisions of the Plan and the Agreement. You also acknowledge that you have read and specifically and expressly approve the terms and conditions set forth in Section 11 of the Agreement, which clearly provide as follows:

(1) Your participation in the Plan does not constitute an acquired right;

(2) The Plan and your participation in it are offered by the Company on a wholly discretionary basis;

(3) You shall not be considered to have any claim or entitlement to compensation or damages from the grant of the Award or from the forfeiture of this Award;

(4) Your participation in the Plan is voluntary; and

(5) The Company and its Affiliates are not responsible for any decrease in the value of any shares of Common Stock acquired at vesting of the Restricted Stock Units.

Labor Law Acknowledgement and Policy Statement . By accepting the Award, you acknowledge that the Company, with registered offices at 2788 San Tomas Expressway, Santa Clara, California 95051, U.S.A, is solely responsible for the administration of the Plan. You further acknowledge that your participation in the Plan, the grant of Restricted Stock Units and any acquisition of shares of Common Stock under the Plan do not constitute an employment relationship between you and the Company because you are participating in the Plan on a wholly commercial basis and your sole employer is NV Computing Mexico, S. de R.L. de C.V. (" ***NVIDIA Mexico*** "). Based on the foregoing, you expressly acknowledge that the Plan and the benefits that you may derive from participation in the Plan do not establish any rights between you and your employer, NVIDIA Mexico, and do not form part of the employment conditions and/or benefits provided by NVIDIA Mexico, and any modification of the Plan or its termination shall not constitute a change or impairment of the terms and conditions of your employment.

You further understand that your participation in the Plan is the result of a unilateral and discretionary decision of the Company; therefore, the Company reserves the absolute right to amend and/or discontinue your participation in the Plan at any time, without any liability to you.

Finally, you hereby declare that you do not reserve to yourself any action or right to bring any claim against the Company for any compensation or damages regarding any provision of the Plan or the benefits derived under the Plan, and that you therefore grant a full and broad release to the Company, its Affiliates, branches, representation offices, shareholders, officers, agents and legal representatives, with respect to any claim that may arise.

Spanish Translation

Reconocimiento del Contrato . Al aceptar el Premio, usted reconoce que ha recibido una copia del Plan y del Contrato, los cuales que ha revisado. Además, usted reconoce que acepta todas las disposiciones del Plan y del Contrato. También, usted reconoce que ha leído y que específica y expresamente aprueba de los términos y condiciones de la Sección 11 del Contrato, que claramente dispone lo siguiente:

(1) Su participación en el Plan no constituye un derecho adquirido;

(2) El Plan y su participación en el Plan se ofrecen por la Compañía de una manera totalmente discrecional;

(3) No tendrá ningún derecho o reclamación por compensación o daño derivado de la concesión del Premio o derivado de la pérdida de este Premio;

(4) SU participación en el Plan es voluntaria; y

(5) La Compañía y sus Afiliadas no son responsables por ninguna disminución del valor de las Acciones adquiridas cuando las Unidades de Acciones Restringidas se maduren.

*Reconocimiento Ley Laboral y Declaración de la Política . Al aceptar el Premio, usted reconoce que la Compañía, con oficinas registradas en 2788 San Tomas Expressway, Santa Clara, California 95051, EE.UU., es únicamente responsable por la administración del Plan. Además, usted reconoce que su participación en el Plan, la concesión de las Unidades de Acciones Restringidas y cualquier adquisición de Acciones de conformidad con el Plan no constituyen una relación laboral entre usted y la Compañía, ya que usted está participando en el Plan sobre una base totalmente comercial y su único patrón es NV Computing Mexico, S. de R.L. de C.V. (" **NVIDIA Mexico** "). Derivado de lo anterior, usted expresamente reconoce que el Plan y los beneficios que se podrían derivar al participar en el Plan no establecen ningún derecho entre usted y su patrón, NVIDIA Mexico, y que no forman parte de las condiciones del empleo y/o las prestaciones otorgadas por NVIDIA Mexico, y cualquier modificación del Plan o su terminación no constituirán un cambio o deterioro de los términos y condiciones de su empleo.*

Además, usted entiende que su participación en el Plan se resulta de una decisión unilateral y discrecional de la Compañía; por lo tanto, la Compañía se reserva el derecho absoluto de modificar y/o discontinuar su participación en el Plan en cualquier momento, sin responsabilidad alguna hacia usted.

Finalmente, en este acto usted manifiesta que no se reserva acción o derecho alguno para interponer una reclamación o demanda en contra de la Compañía, por cualquier compensación o daño en relación con cualquier disposición del Plan o de los beneficios derivados del Plan, y, por lo tanto, otorga un amplio y total finiquito a la Compañía, sus Afiliadas, sucursales, oficinas de representación, sus accionistas, funcionarios, agentes y representantes legales con respecto a cualquier reclamación o demanda que pudiera surgir.

NETHERLANDS

There are no country specific provisions.

POLAND

Notifications

Exchange Control Information . Polish residents holding foreign securities (including shares of Common Stock) and maintaining accounts abroad must report information to the National Bank of Poland on transactions and balances regarding such securities and cash deposited into such accounts if the value of any transactions or balances exceeds PLN 7,000,000. If required, the reports must be filed on a quarterly basis on special forms available on the website of the National Bank of Poland. In addition, transfers of funds into and out of Poland in excess of €15,000 (or PLN 15,000 if such transfer of funds is connected with business activity of an entrepreneur) must be made via a bank account held at a bank in Poland.

Lastly, you are required to store all documents connected with any foreign exchange transactions that you engaged in for a period of five years, as measured from the end of the year in which such transaction occurred.

RUSSIA

Terms and Conditions

U.S. Transaction . You understand that the acceptance of the Restricted Stock Units results in an agreement between you and the Company that is completed in the U.S. and that this Agreement is governed by the laws of the State of Delaware, without giving effect to the conflict of law principles thereof.

Notifications

Securities Law Information . You acknowledge that the Restricted Stock Units, this Agreement, the Plan and all other materials that you may receive regarding participation in the Plan do not constitute advertising or an offering of securities in Russia. The shares of Common Stock acquired pursuant to the Plan have not and will not be registered in Russia and therefore, neither the Restricted Stock Units nor the shares of Common Stock may be offered or publicly circulated in Russia. You acknowledge that you may hold shares of Common Stock acquired upon settlement of the Restricted Stock Units in an account with the designated broker in the U.S. However, in no event will shares of Common Stock issued to you under the Plan be

delivered to you in Russia. Further, you are not permitted to sell shares of Common Stock directly to other Russian individuals.

Foreign Asset/Account Reporting Information . Russian residents may be required to notify the Russian tax authorities within one month of opening or closing a foreign bank account, or of changing any account details. Russian residents also may be required to file with the Russian tax authorities reports of the transactions in their foreign bank accounts.

Anti-Corruption Notification . Anti-corruption laws prohibit certain public servants, their spouses and their dependent children from owning any foreign source financial instruments (e.g., shares of foreign companies such as the Company). Accordingly, you should inform the Company if you are covered by these laws as you should not hold shares of Common Stock acquired under the Plan.

Labor Law Information . If you continue to hold shares of Common Stock acquired at settlement of the Restricted Stock Units after an involuntary termination of Continuous Service, you will not be eligible to receive unemployment benefits in Russia.

SINGAPORE

Terms and Conditions

Settlement of Awards and Sale of Shares . This provision supplements Sections 2 and 3 of the Global Restricted Stock Unit Agreement:

The Restricted Stock Units are subject to section 257 of the Securities and Futures Act (Chap. 289, 2006 Ed.) (" ***SFA*** ") and you hereby agree that you will not make (i) any subsequent sale of the shares of Common Stock in Singapore, or (ii) any offer of such subsequent sale of the shares of Common Stock in Singapore, unless such sale or offer is made: (a) more than six months after the Date of Grant or (b) pursuant to the exemptions under Part XIII Division (1) Subdivision (4) (other than section 280) of the SFA, or pursuant to, and in accordance with the conditions of, any applicable provisions of the SFA.

Notifications

Securities Law Information . The Award is being made to you in reliance on the "Qualifying Person" exemption under section 273(1)(f) of the SFA, is exempt from the prospectus and registration requirements under the SFA and is not made to you with a view to the Award or underlying shares of Common Stock being subsequently offered for sale to any other party. The Plan has not been and will not be lodged or registered as a prospectus with the Monetary Authority of Singapore.

Chief Executive Officer and Director Notification Obligation . The Chief Executive Officer ("CEO") and directors (including alternate, substitute, associate and shadow directors) of a Singapore Affiliate are subject to certain notification requirements under the Singapore Companies Act. Among these requirements is an obligation to notify such entity in writing within two business

days of any of the following events: (i) the acquisition or disposal of an interest (*e.g.* , Awards granted under the Plan or shares of Common Stock) in the Company or any Affiliate, (ii) any change in previously-disclosed interests (*e.g.* , sale of shares of Common Stock), of (iii) becoming the CEO, a director, an associate director or a shadow director of an Affiliate in Singapore, if the individual holds such an interest at that time. These notification requirements apply regardless of whether the CEO or directors are residents of or employed in Singapore.

SPAIN

Terms and Conditions

Nature of Grant . The following provision supplements Section 11 of the Global Restricted Stock Unit Agreement:

By accepting the Award, you consent to participation in the Plan and acknowledge that you have received a copy of the Plan.

You understand that the Company has unilaterally, gratuitously and in its sole discretion decided to grant an Award under the Plan to individuals who may be Consultants, Directors, or Employees of the Employer, the Company, or one of its other Affiliates throughout the world. The decision is limited and entered into based upon the express assumption and condition that any Award will not economically or otherwise bind the Company or any Affiliate, including the Employer, on an ongoing basis, other than as expressly set forth in the Agreement. Consequently, you understand that the Award is given on the assumption and condition that the Award shall not become part of any employment contract (whether with the Company or any Affiliate, including the Employer) and shall not be considered a mandatory benefit, salary for any purpose (including severance compensation) or any other right whatsoever. Furthermore, you understand and freely accept that there is no guarantee that any benefit whatsoever shall arise from the Award, which is gratuitous and discretionary, since the future value of the Award, and the underlying shares of Common Stock, is unknown and unpredictable.

Further, your participation in the Plan is expressly conditioned on your continued and active rendering of service, such that, unless otherwise set forth in the Plan, if your Continuous Service terminates for any reason whatsoever, your participation in the Plan will cease immediately. This will be the case, for example, even if (1) you are considered to be unfairly dismissed without good cause (*i.e.* , subject to a " *despido improcedente* "); (2) you are dismissed for disciplinary or objective reasons or due to a collective dismissal; (3) your Continuous Service ceases due to a change of work location, duties or any other employment or contractual condition; (4) your Continuous Service ceases due to a unilateral breach of contract by the Company or any of its Affiliates; or (5) your Continuous Service terminates for any other reason whatsoever. Consequently, upon termination of your Continuous Service for any of the above reasons, you automatically lose any right to participate in the Plan on the date of your termination of Continuous Service, as described in the Plan and the Agreement.

Notifications

Securities Law Information . The Award and shares of Common Stock described in the Agreement do not qualify under Spanish regulations as securities. No "offer of securities to the public", as defined under Spanish law, has taken place or will take place in the Spanish territory. The Agreement has not been nor will it be registered with the *Comisión Nacional del Mercado de Valores* , and does not constitute a public offering prospectus.

Exchange Control Information . The acquisition of shares of Common Stock and subsequent sales of shares of Common Stock must be declared for statistical purposes to the *Dirección General de Comercio e Inversiones* (the " ***DGCI*** "). Because you will not purchase or sell the shares of Common Stock through the use of a Spanish financial institution, you will need to make the declaration yourself by filing a D-6 form with the DGCI. Generally, the D-6 form must be filed each January while the shares of Common Stock are owned. However, if the value of the shares of Common Stock acquired under the Plan or the amount of the sale proceeds exceeds €1,502,530, the declaration must be filed within one month of the acquisition or sale, as applicable.

In addition, any securities accounts (including brokerage accounts held abroad), as well as the securities (including shares of Common Stock) held in such accounts, may need to be declared electronically to the Bank of Spain, depending on the value of the transactions during the prior tax year or the balances in such accounts as of December 31 of the prior tax year.

Foreign Asset/Account Reporting Information . Rights or assets (*e.g.* , shares of Common Stock or cash held in a bank or brokerage account) held outside of Spain with a value in excess of €50,000 per type of right or asset (*e.g.* , shares of Common Stock, cash, etc.) as of December 31, must be reported on your annual tax return. After such rights and/or assets are initially reported, the reporting obligation will only apply for subsequent years if the value of any previously-reported rights or assets increases by more than €20,000. Shares of Common Stock acquired under the Plan or other equity programs offered by the Company constitute assets for purposes of this requirement, but unvested rights (e.g., Restricted Stock Units, etc.) are not considered assets or rights for purposes of this requirement.

SWEDEN

There are no country specific provisions.

SWITZERLAND

Notifications

Securities Law Information . The Award is not intended to be publicly offered in or from Switzerland. Because the offer of the Award is considered a private offering in Switzerland; it is not subject to registration in Switzerland. Neither this document nor any other materials relating to the Award constitute a prospectus as such term is understood pursuant to article 652a of the Swiss Code of Obligations, and neither this document nor any other materials relating to the Award may be publicly distributed nor otherwise made publicly available in Switzerland. Further, neither this document nor any other offering or marketing material

relating to the Award have been or will be filed with, approved or supervised by any Swiss regulatory authority (in particular, the Swiss Financial Market Supervisory Authority (FINMA)).

TAIWAN

Notifications

Securities Law Information **.** The offer of participation in the Plan is available only for Consultants, Directors and Employees of the Company and its Affiliates. The offer of participation in the Plan is not a public offer of securities by a Taiwanese company.

Exchange Control Information . Taiwanese residents may acquire and remit foreign currency (including proceeds from the sale of shares of Common Stock and the receipt of any dividends paid on such shares of Common Stock) into Taiwan up to US$5,000,000 per year without justification. If the transaction amount is TWD 500,000 or more in a single transaction, a Foreign Exchange Transaction Form must be submitted, along with supporting documentation, to the satisfaction of the remitting bank. You should consult your personal legal advisor to ensure compliance with applicable exchange control laws in Taiwan.

UNITED ARAB EMIRATES (DUBAI)

Notifications

Securities Law Information . Participation in the Plan is being offered only to eligible service providers and is in the nature of providing equity incentives to employees in the United Arab Emirates. The Plan and the Agreement are intended for distribution only to such service providers and must not be delivered to, or relied on by, any other person. Prospective purchasers of the securities offered should conduct their own due diligence on the securities. If you do not understand the contents of the Plan or the Agreement, you should consult an authorized financial adviser. The Emirates Securities and Commodities Authority has no responsibility for reviewing or verifying any documents in connection with the Plan. Neither the Ministry of Economy nor the Dubai Department of Economic Development have approved the Plan or the Agreement nor taken steps to verify the information set out therein, and have no responsibility for such documents.

UNITED KINGDOM

Terms and Conditions

Form of Settlement . Restricted Stock Units granted to individuals resident in the United Kingdom shall be paid in shares of Common Stock only. In no event shall any Restricted Stock Units be paid in cash, notwithstanding any discretion contained in the Plan to the contrary.

Responsibility for Taxes . The following provisions supplement Section 10 of the Global Restricted Stock Unit Agreement:

Without limitation to Section 10 of the Global Restricted Stock Unit Agreement, you agree that you are liable for all Tax-Related Items and hereby covenant to pay all such Tax-Related Items, as and when requested by the Company or the Employer or by Her Majesty's Revenue and Customs (" ***HMRC*** ") (or any other tax authority or any other relevant authority). You also agree to indemnify and keep indemnified the Company and the Employer against any Tax-Related Items that they are required to pay or withhold or have paid or will pay to HMRC on your behalf (or any other tax authority or any other relevant authority).

Notwithstanding the foregoing, if you are a director or an executive officer of the Company (within the meaning of such terms for purposes of Section 13(k) of the Exchange Act), you acknowledge that may not be able to indemnify the Company or the Employer for the amount of any income tax not collected from or paid by you, as it may be considered a loan. In this case, the amount of any income tax not collected within 90 days of the end of the U.K. tax year in which the event giving rise to the Tax-Related Item(s) occurs may constitute an additional benefit to you on which additional income tax and National Insurance Contributions may be payable. You will be responsible for reporting and paying any income tax due on this additional benefit directly to HMRC under the self-assessment regime and for reimbursing the Company or the Employer (as appropriate) for the value of any National Insurance Contributions due on this additional benefit, which the Company or the Employer may recover from you by any of the means referred to in the Plan or Section 10 of the Global Restricted Stock Unit Agreement.

Joint Election . As a condition of participation in the Plan, you agree to accept any liability for secondary Class 1 National Insurance contributions that may be payable by the Company or the Employer (or any successor to the Company or the Employer) in connection with the Restricted Stock Units and any event giving rise to Tax-Related Items (the " ***Employer NICs*** "). The Employer NICs may be collected by the Company or the Employer using any of the methods described in the Plan or in Section 10 of the Global Restricted Stock Unit Agreement.

Without prejudice to the foregoing, you agree to execute a joint election with the Company and/or the Employer (a " ***Joint Election*** "), the form of such Joint Election being formally approved by HMRC, and any other consent or elections required by the Company or the Employer in respect of the Employer NICs liability. You further agree to execute such other elections as may be required by any successor to the Company and/or the Employer for the purpose of continuing the effectiveness of your Joint Election.

ATTACHMENT 6

NVIDIA CORPORATION
AMENDED AND RESTATED 2007 EQUITY INCENTIVE PLAN

EXHIBIT 21.1

LIST OF REGISTRANT'S SUBSIDIARIES

Subsidiaries of Registrant (All 100% owned)	Country of Organization
Icera LLC	United States
Icera Semiconductor LLC	United States
JAH Venture Holdings, Inc.	United States
LPN Facilitator LLC	United States
NVIDIA (BVI) Holdings Limited	Virgin Islands, British
NVIDIA ARC GmbH	Germany
NVIDIA Brasil Computação Visual Limitada	Brazil
NVIDIA Development France SAS	France
NVIDIA Development UK Limited	England and Wales
NVIDIA Development, Inc.	Canada
NVIDIA Dutch B.V.	Netherlands
NVIDIA Entertainment Devices (Shanghai) Co., Ltd	China
NVIDIA FZ-LLC	United Arab Emirates
NVIDIA GK	Japan
NVIDIA Global Ltd	Virgin Islands, British
NVIDIA GmbH	Germany
NVIDIA Graphics Holding Company	Mauritius
NVIDIA Graphics Private Limited	India
NVIDIA Hong Kong Development Limited	Hong Kong
NVIDIA Hong Kong Holdings Limited	Hong Kong
NVIDIA International Holdings Inc.	United States
NVIDIA International, Inc.	Cayman Islands
NVIDIA Israel Ltd.	Israel
NVIDIA Italy S.r.l.	Italy
NVIDIA Ltd.	England and Wales
NVIDIA Poland sp.z o.o	Poland
NVIDIA Pty Limited	Australia
NVIDIA Semiconductor (Shenzhen) Co., Ltd.	China
NVIDIA Semiconductor Holding Company	Mauritius
NVIDIA Semiconductor R&D (Tianjin) Co., Ltd.	China
NVIDIA Semiconductor Shenzhen Holding Company	Mauritius
NVIDIA Semiconductor Technical Service (Shanghai) Co., Ltd.	China
NVIDIA Semiconductor Technology (Shanghai) Co., Ltd.	China
NVIDIA Singapore Development Pte. Ltd.	Singapore
NVIDIA Singapore Pte Ltd	Singapore
NVIDIA Switzerland AG	Switzerland
NVIDIA Technical Service (Beijing) Co., Ltd.	China
NVIDIA Technology UK Limited	England and Wales
NVIDIA, Helsinki Oy	Finland
VC Worldwide Ltd.	Virgin Islands, British

EXHIBIT 23.1

CONSENT OF INDEPENDENT REGISTERED PUBLIC ACCOUNTING FIRM

We hereby consent to the incorporation by reference in the Registration Statements on Form S-3 (No. 333-213516) and S-8 (Nos. 333-74905, 333-51520, 333-74868, 333-100010, 333-106191, 333-114375, 333-123933, 333-132493, 333-140021, 333-143953, 333-181625, 333-185036, 333-188775, 333-196259 and 333-211615) of NVIDIA Corporation of our report dated February 21, 2019 relating to the financial statements, financial statement schedule and the effectiveness of internal control over financial reporting, which appears in this Form 10-K.

/s/ PricewaterhouseCoopers LLP

San Jose, California
February 21, 2019

EXHIBIT 31.1

CERTIFICATION

I, Jen-Hsun Huang, certify that:

1. I have reviewed this Annual Report on Form 10-K of NVIDIA Corporation;

2. Based on my knowledge, this report does not contain any untrue statement of a material fact or omit to state a material fact necessary to make the statements made, in light of the circumstances under which such statements were made, not misleading with respect to the period covered by this report;

3. Based on my knowledge, the financial statements, and other financial information included in this report, fairly present in all material respects the financial condition, results of operations and cash flows of the registrant as of, and for, the periods presented in this report;

4. The registrant's other certifying officer(s) and I are responsible for establishing and maintaining disclosure controls and procedures (as defined in Exchange Act Rules 13a-15(e) and 15d-15(e)) and internal control over financial reporting (as defined in Exchange Act Rules 13a-15(f) and 15d-15(f)) for the registrant and have:

(a) designed such disclosure controls and procedures, or caused such disclosure controls and procedures to be designed under our supervision, to ensure that material information relating to the registrant, including its consolidated subsidiaries, is made known to us by others within those entities, particularly during the period in which this report is being prepared;

(b) designed such internal control over financial reporting, or caused such internal control over financial reporting to be designed under our supervision, to provide reasonable assurance regarding the reliability of financial reporting and the preparation of financial statements for external purposes in accordance with generally accepted accounting principles;

(c) evaluated the effectiveness of the registrant's disclosure controls and procedures and presented in this report our conclusions about the effectiveness of the disclosure controls and procedures, as of the end of the period covered by this report based on such evaluation; and

(d) disclosed in this report any change in the registrant's internal control over financial reporting that occurred during the registrant's most recent fiscal quarter (the registrant's fourth fiscal quarter in the case of an annual report) that has materially affected, or is reasonably likely to materially affect, the registrant's internal control over financial reporting; and

5. The registrant's other certifying officer(s) and I have disclosed, based on our most recent evaluation of internal control over financial reporting, to the registrant's auditors and the audit committee of the registrant's board of directors (or persons performing the equivalent functions):

(a) all significant deficiencies and material weaknesses in the design or operation of internal control over financial reporting which are reasonably likely to adversely affect the registrant's ability to record, process, summarize and report financial information; and

(b) any fraud, whether or not material, that involves management or other employees who have a significant role in the registrant's internal control over financial reporting.

Date: February 21, 2019

/s/ JEN-HSUN HUANG
Jen-Hsun Huang
President and Chief Executive Officer

EXHIBIT 31.2

CERTIFICATION

I, Colette M. Kress, certify that:

1. I have reviewed this Annual Report on Form 10-K of NVIDIA Corporation;

2. Based on my knowledge, this report does not contain any untrue statement of a material fact or omit to state a material fact necessary to make the statements made, in light of the circumstances under which such statements were made, not misleading with respect to the period covered by this report;

3. Based on my knowledge, the financial statements, and other financial information included in this report, fairly present in all material respects the financial condition, results of operations and cash flows of the registrant as of, and for, the periods presented in this report;

4. The registrant's other certifying officer(s) and I are responsible for establishing and maintaining disclosure controls and procedures (as defined in Exchange Act Rules 13a-15(e) and 15d-15(e)) and internal control over financial reporting (as defined in Exchange Act Rules 13a-15(f) and 15d-15(f)) for the registrant and have:

(a) designed such disclosure controls and procedures, or caused such disclosure controls and procedures to be designed under our supervision, to ensure that material information relating to the registrant, including its consolidated subsidiaries, is made known to us by others within those entities, particularly during the period in which this report is being prepared;

(b) designed such internal control over financial reporting, or caused such internal control over financial reporting to be designed under our supervision, to provide reasonable assurance regarding the reliability of financial reporting and the preparation of financial statements for external purposes in accordance with generally accepted accounting principles;

(c) evaluated the effectiveness of the registrant's disclosure controls and procedures and presented in this report our conclusions about the effectiveness of the disclosure controls and procedures, as of the end of the period covered by this report based on such evaluation; and

(d) disclosed in this report any change in the registrant's internal control over financial reporting that occurred during the registrant's most recent fiscal quarter (the registrant's fourth fiscal quarter in the case of an annual report) that has materially affected, or is reasonably likely to materially affect, the registrant's internal control over financial reporting; and

5. The registrant's other certifying officer(s) and I have disclosed, based on our most recent evaluation of internal control over financial reporting, to the registrant's auditors and the audit committee of the registrant's board of directors (or persons performing the equivalent functions):

(a) all significant deficiencies and material weaknesses in the design or operation of internal control over financial reporting which are reasonably likely to adversely affect the registrant's ability to record, process, summarize and report financial information; and

(b) any fraud, whether or not material, that involves management or other employees who have a significant role in the registrant's internal control over financial reporting.

Date: February 21, 2019

/s/ COLETTE M. KRESS
Colette M. Kress
Executive Vice President and Chief Financial Officer

EXHIBIT 32.1

CERTIFICATION

Pursuant to the requirement set forth in Rule 13a-14(b) of the Securities Exchange Act of 1934, as amended (the "Exchange Act"), and Section 1350 of Chapter 63 of Title 18 of the United States Code (18 U.S.C. § 1350), Jen-Hsun Huang, the President and Chief Executive Officer of NVIDIA Corporation (the "Company"), hereby certifies that, to the best of his knowledge:

1. The Company's Annual Report on Form 10-K for the year ended January 27, 2019 , to which this Certification is attached as Exhibit 32.1 (the "Periodic Report"), fully complies with the requirements of Section 13(a) or Section 15(d) of the Exchange Act; and

2. The information contained in the Periodic Report fairly presents, in all material respects, the financial condition of the Company at the end of the period covered by the Periodic Report and results of operations of the Company for the period covered by the Periodic Report.

Date: February 21, 2019

/s/ JEN-HSUN HUANG
Jen-Hsun Huang
President and Chief Executive Officer

A signed original of this written statement required by Section 906 of 18 U.S.C. § 1350 has been provided to NVIDIA Corporation and will be retained by NVIDIA Corporation and furnished to the Securities and Exchange Commission or its staff upon request.

This certification accompanies the Form 10-K to which it relates, is not deemed filed with the Securities and Exchange Commission and is not to be incorporated by reference into any filing of the Company under the Securities Act of 1933, as amended, or the Exchange Act (whether made before or after the date of the Form 10-K), irrespective of any general incorporation language contained in such filing.

EXHIBIT 32.2

CERTIFICATION

Pursuant to the requirement set forth in Rule 13a-14(b) of the Securities Exchange Act of 1934, as amended (the "Exchange Act"), and Section 1350 of Chapter 63 of Title 18 of the United States Code (18 U.S.C. § 1350), Colette M. Kress, Executive Vice President and Chief Financial Officer of NVIDIA Corporation (the "Company"), hereby certifies that, to the best of her knowledge:

1. The Company's Annual Report on Form 10-K for the year ended January 27, 2019 , to which this Certification is attached as Exhibit 32.2 (the "Periodic Report"), fully complies with the requirements of Section 13(a) or Section 15(d) of the Exchange Act; and

2. The information contained in the Periodic Report fairly presents, in all material respects, the financial condition of the Company at the end of the period covered by the Periodic Report and results of operations of the Company for the period covered by the Periodic Report.

Date: February 21, 2019

/s/ COLETTE M. KRESS
Colette M. Kress
Executive Vice President and Chief Financial Officer

A signed original of this written statement required by Section 906 of 18 U.S.C. § 1350 has been provided to NVIDIA Corporation and will be retained by NVIDIA Corporation and furnished to the Securities and Exchange Commission or its staff upon request.

This certification accompanies the Form 10-K to which it relates, is not deemed filed with the Securities and Exchange Commission and is not to be incorporated by reference into any filing of the Company under the Securities Act of 1933, as amended, or the Exchange Act (whether made before or after the date of the Form 10-K), irrespective of any general incorporation language contained in such filing.