Douglas G. Muehlhauser (State Bar No. 179,495)
doug.muehlhauser@knobbe.com
Alan G. Laquer (State Bar No. 259,257)
alan.laquer@knobbe.com
KNOBBE, MARTENS, OLSON & BEAR, LLP
2040 Main Street, Fourteenth Floor
Irvine, CA 92614
Telephone: 949-760-0404
Facsimile: 949-760-9502

Attorneys for Plaintiff Pepperdata, Inc.

IN THE UNITED STATES DISTRICT COURT

FOR THE NORTHERN DISTRICT OF CALIFORNIA

PEPPERDATA, INC., Plaintiff, v. YAHOO! INC., Defendant.	Civil Action No. 5:16-CV-7211-EJD **FIRST AMENDED COMPLAINT FOR INFRINGEMENT OF U.S. PATENT NOS. 8,849,891 AND 9,325,593** **and** **DEMAND FOR JURY TRIAL**

Plaintiff Pepperdata, Inc. ("Pepperdata") brings this complaint for patent infringement against Yahoo! Inc. ("Yahoo") and alleges as follows:

I. NATURE OF THE ACTION

1. This is an action for patent infringement arising under the patent laws of the United States, 35 U.S.C. § 100 *et seq*.

II. PARTIES

2. Pepperdata is a Delaware corporation with its principal place of business at 19409 Stevens Creek Boulevard, Suite 260, Cupertino, California 95014.

3. On information and belief, Yahoo is a Delaware corporation with its principal place of business at 701 First Avenue, Sunnyvale, California 94089.

III. JURISDICTION

4. This Court has subject-matter jurisdiction over this patent infringement action under 28 U.S.C. §§ 1331 and 1338(a).

5. On information and belief, Yahoo has its principal place of business in this judicial district and is subject to general personal jurisdiction here.

6. Venue is proper in this judicial district pursuant to 28 U.S.C. § 1391(b) and (c) and § 1400(b).

IV. FACTS

7. Pepperdata was founded in 2012 and provides its customers with its innovative product that improves the performance of Hadoop distributed computing clusters, for example by adding functionality to a Hadoop cluster to dynamically overcommit resources in that cluster.

8. Pepperdata has obtained patents to protect its business investments in researching and developing its innovative solutions for improving the performance of Hadoop environments, including dynamically overcommitting resources in a Hadoop cluster.

9. On September 30, 2014, the United States Patent and Trademark Office ("PTO") issued U.S. Patent No. 8,849,891 ("the '891 patent"), titled "Systems, Methods, and Devices for Dynamic Resource Monitoring and Allocation in a Cluster System." A copy of

the ’891 patent is provide as Exhibit A to this Complaint.

10. On April 26, 2016, the PTO issued U.S. Patent No. 9,325,593 (“the ’593 patent”) titled “Systems, Methods, and Devices for Dynamic Resource Monitoring and Allocation in a Cluster System.” A copy of the ’593 patent is provided as Exhibit B to this Complaint.

11. Pepperdata owns the ’891 and ’593 patents.

12. Pepperdata has marked the products that it has manufactured and sold under the ’891 and ’593 patents in a manner that complies with 35 U.S.C. § 287(a).

13. Yahoo uses Hadoop clusters within the United States.

14. Yahoo made the following statements in a November 18, 2016 Hadoop group tumblr post at http://yahoohadoop.tumblr.com/post/153336735536/10-years-of-hadoop-and-its-israeli-pioneering

- “The Apache Hadoop technology suite is the engine behind the Big Data revolution that has been transforming multiple industries over the last decade.”
- “These days, Yahoo is the largest Hadoop deployment in the industry. We run tens of thousands of Hadoop machines in our datacenters and manage more than 600 petabytes of data. Our products use Hadoop in a variety of ways that reflect a wealth of data processing patterns.”
- “Yahoo’s commitment to Hadoop goes far beyond operating the technology at Web scale. The company’s engineers and scientists make contributions to both entrenched and incubating Hadoop projects.”
- “Our team launched approximately three years ago. Collectively we add many years of experience to Yahoo and the Hadoop community in distributed computing research and development.
- “Our researchers regularly present their innovations at leading

industrial conferences (Hadoop Summit and HBaseCon), as well as at top academic venues."

15. Yahoo made a software patch identified as YARN-5202, titled "Dynamic Overcommit of Node Resources."

16. Nathan Roberts, a Yahoo software architect, is the assignee of the JIRA for YARN-5202, accessible at https://issues.apache.org/jira/browse/YARN-5202.

17. Yahoo made the YARN-5202 software patch available for download on the YARN-5202 JIRA. For example, YARN-5202 was available for download from its JIRA on August 4, 2016.

18. Upon information and belief, Yahoo made the YARN-5202 software patch in the United States in 2016.

19. Yahoo's YARN-5202 software patch adds to a Hadoop cluster functionality for dynamic overcommitting node resources.

20. The YARN-5202 JIRA explains that "[t]his Jira is to present a proof-of-concept implementation (collaboration between Jason Lowe and myself) of a dynamic over-commit implementation in YARN."

21. Jason Lowe has the title "Senior Principal Engineer, Hadoop" at Yahoo.

22. On June 28, 2016, Mr. Lowe gave a presentation, on behalf of Yahoo, titled "Investigating the Effects of Overcommitting YARN Resources" at the 2016 Hadoop Summit in San Jose, California ("Yahoo's YARN-5202 Presentation").

23. Over 3,000 people attended the 2016 Hadoop Summit in San Jose, California.

24. As of December 15, 2016, a video of Yahoo's YARN-5202 Presentation is available at https://www.youtube.com/watch?v=hILD2g9putc, and this video of Yahoo's YARN-5202 Presentation was published and made available for public viewing at least by June 29, 2016.

25. A transcript of Yahoo's YARN-5202 Presentation is provided as Exhibit C to this Complaint.

26. The YARN-5202 JIRA shows that by June 14, 2016, Yahoo had uploaded the

YARN-5202 patch to make it available to the public for free download, installation and use. A copy of the YARN-5202 JIRA, printed on July 14, 2016, is attached hereto as Exhibit D.

27. The YARN-5202 JIRA describes the YARN-5202 patch as an "improvement" having a "major" priority.

28. Upon information and belief, Yahoo has used the YARN-5202 software patch on Hadoop clusters that it uses in the United States.

29. Mr. Lowe explained in Yahoo's YARN-5202 Presentation that Yahoo has used the YARN-5202 software patch on Yahoo's main research cluster, which runs Hadoop YARN.

30. Mr. Lowe explained in Yahoo's YARN-5202 Presentation that Yahoo has enjoyed a "significant improvement" in the performance of its Hadoop cluster as a result of the dynamic overcommit functionality in the YARN-5202 software patch. For example, Mr. Lowe explained that:

> before [adding dynamic overcommit] utilization was hovering like around 40/50% now it's hovering – especially during the peg thing – it's hovering closer to 70/80% so that's a significant improvement for us – it's like close to 40/50% CPU utilization improvement. And so that's been a big deal for us. This like down at the bottom you can see that in terms of the YARN size of the cluster reported it's like 50% bigger which is like taking your cluster and putting half again as many nodes on the cluster without actually buying those nodes. So it's a really big deal for us.

31. On August 4, 2016, Pepperdata provided to Yahoo written notice that Yahoo infringes the '891 and '593 patents. A copy of Pepperdata's August 4, 2016 letter to Yahoo is attached hereto as Exhibit E.

32. Pepperdata's August 4, 2016 letter to Yahoo included claim charts showing, by detailed and exemplary explanation, how Yahoo's use of its YARN-5202 patch in a

Hadoop computer cluster falls within the scope of representative claims of the ’891 and ’593 patents.

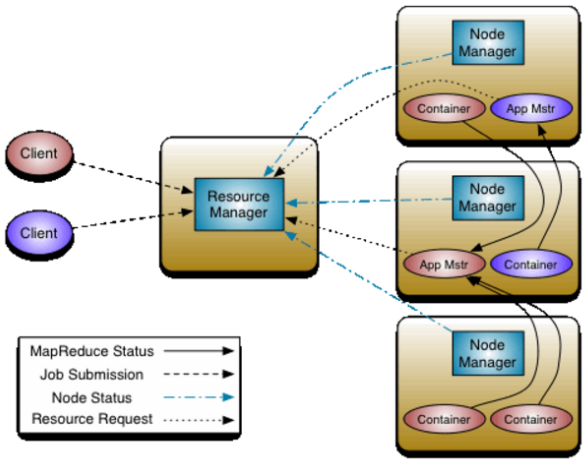
33. For example, the claim chart provided to Yahoo on August 4, 2016, includes the following detailed exemplary explanation of how Yahoo’s use of its YARN-5202 patch in a Hadoop computer cluster falls within the scope of claim 1 of the ’891 patent:

U.S. Patent No. 8,849,891 Claim 1	Yahoo’s Hadoop Clusters with Yahoo’s Dynamic Overcommit of Node Resources
1. A computer cluster comprising:	
a management computing device comprising a supervisor controller configured to coordinate processing of a plurality of sub-jobs for a plurality of overall jobs;	Each of Yahoo’s Hadoop computer clusters comprises a management computing device running a ResourceManager, which comprises a supervisor controller configured to coordinate processing of a plurality of sub-jobs for a plurality of overall jobs. For example, the Hadoop MapReduce Tutorial1 explains “The MapReduce framework consists of a single master ResourceManager, one slave NodeManager per cluster-node, and MRAppMaster per application.” As another example, Yahoo’s jira2 for its modified Hadoop software identifies its components as “nodemanager, resourcemanager.”
a plurality of computer system nodes configured to communicate with the management computing device, and to perform processing of received sub-jobs, the computing system nodes each comprising:	Each of Yahoo’s Hadoop computer clusters comprises a plurality of computer system nodes configured to communicate with the management computing device, and to perform processing of received sub-jobs.

one or more processors configured to perform computing processes on received sub-jobs;	Each of the computing system nodes comprises one or more processors configured to perform computing processes on received sub-jobs. For example, the Hadoop NodeManager Overview explains3 that "The NodeManager is responsible for launching and managing containers on a node. Containers execute tasks as specified by the AppMaster."
an agent controller comprising:	Each of the computing systems nodes comprises a NodeManager which comprises an agent controller.
a monitoring interface configured to monitor utilization by sub-jobs of system resources of a first computing system node; and	The agent controller of the NodeManager comprises a monitoring interface configured to monitor utilization by sub-jobs of system resources of a first computing system node. For example, the Hadoop 2.7.2 YARN Architecture4 describes: "The NodeManager is the per-machine framework agent who is responsible for containers, monitoring their resource usage (cpu, memory, disk, network) and reporting the same to the ResourceManager/Scheduler."
a reporting controller configured to transmit the monitored system resources utilization to the supervisor controller in substantially real-time;	The agent controller of the NodeManager comprises a reporting controller configured to transmit the monitored system resources utilization to the supervisor controller in substantially real-time. For example, the Hadoop 2.7.2 YARN Architecture describes: "The NodeManager is the per-machine framework agent who is responsible for containers,

	monitoring their resource usage (cpu, memory, disk, network) and reporting the same to the ResourceManager/Scheduler." Yahoo explained in its dynamic overcommit presentation5 that its NodeManager reports this information in its heartbeat. (See at 10 minute mark.)
wherein the supervisor controller is configured to assign an additional sub-job to the first computing system node based on determining that the utilization of at least one system resource of the first computing system node is below a threshold level, the determining based on the monitored system resources utilization transmitted from the reporting controller to the supervisor controller;	The supervisor controller of the ResourceManager is configured to assign an additional sub-job to the first computing system node based on determining that the utilization of at least one system resource of the first computing system node is below a threshold level, the determining based on the monitored system resource utilization transmitted from the reporting controller of the NodeManager to the supervisor controller of the ResourceManager. For example, the ResourceManager determines that the memory utilization reported by NodeManager is below the corresponding low water mark, as reflected in the "conf" variable, and therefore assigns one or more additional sub-jobs to the first computing system node by incrementing the containers allocated on that node, such as through the memIncrement = getConfInt(conf,…) function call at line 1249 of Yahoo's dynamic overcommit patch. As another example, the ResourceManager

	determines that the virtual core utilization reported by NodeManager is below the corresponding low water mark, as reflected in the "conf" variable, and therefore assigns one or more additional sub-jobs to the first computing system node by incrementing the containers allocated on that node, such as through the vcoreIncrement = getConfInt(conf,…) function call at line 1272 of Yahoo's dynamic overcommit patch. Yahoo explained in its dynamic overcommit presentation: "How does the ResourceManager do that scaling? We have a high-level watermark and a low watermark… As long as the node keeps reporting lower than the low watermark utilization we'll increment over time the size of that node inside the ResourceManager so that it'll keep allocating more resources inside that node." (See at 12 minute mark.) As another example, in Yahoo's dynamic overcommit patch, the updateTotalResource function is described as "Adjust overcommit metrics... The amount to overcommit will be re-calculated on next node heartbeat."
wherein the at least one system resource of the first computing system node is a first electronic random access memory capacity,	As described above in the context of the memIncrement = getConfInt(conf,…) function call, at least one system resource of the first computing system node is a first electronic random access

	memory capacity. As another example, in Yahoo's dynamic overcommit patch, Yahoo's source code comments explain that RM_OVERCOMMIT_MEM_LWM corresponds to: "Low water memory utilization mark for overcommit. If the node's memory utilization is below this value then the scheduler will try to maximize the memory overcommit."
wherein the supervisor controller is configured to monitor a second electronic random access memory capacity of a second computing system node,	The supervisor controller of the ResourceManager is configured to monitor a second electronic random access memory capacity of a second computing system node. For example, the ResourceManager receives memory utilization information from the node's NodeManager. For example, the Hadoop 2.7.2 YARN Architecture6 illustrates that the architecture includes multiple nodes: 

wherein the assigning by the supervisor controller of the additional sub-job comprises assigning the additional sub-job to the first computing system node based on determining that utilization of the first electronic random access memory capacity is below the threshold level,	The supervisor controller of the ResourceManager assigns the additional sub-job to the first computing system node based on determining that utilization of the first electronic random access memory capacity is below the corresponding low watermark threshold level. For example, the ResourceManager determines that the memory utilization reported by NodeManager is below the corresponding low water mark, as reflected in the "conf" variable, and therefore assigns one or more additional sub-jobs to the first computing system node by incrementing the containers allocated on that node, such as through the memIncrement = getConfInt(conf,…) function call at line 1249 of Yahoo's dynamic overcommit patch. Yahoo explained in its dynamic overcommit presentation: "How does the ResourceManager do that scaling? We have a high-level watermark and a low watermark… As long as the node keeps reporting lower than the low watermark utilization we'll increment over time the size of that node inside the ResourceManager so that it'll keep allocating more resources inside that node." (See at 12 minute mark.)
wherein the supervisor controller is configured to prevent assignment of	The supervisor controller of the ResourceManager is configured to prevent assignment of additional

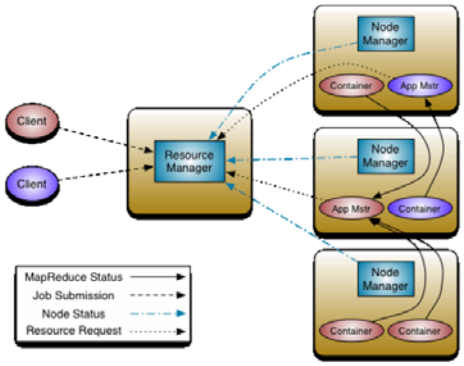
additional sub-jobs to the second computing system node based on determining that utilization of the second electronic random access memory capacity is at or above a threshold value,	sub-jobs to the second computing system node based on determining that utilization of the second electronic random access memory capacity is at or above a high watermark threshold value. For example, at line 453 of Yahoo's dynamic overcommit patch, Yahoo's code checks whether the value of the variable "pmemUsedPercent" is greater than or equal to the highWaterMark and, if it is, sets needToTrim = true. Because needToTrim is set to true, the function does not return at line 486 and instead goes on to perform trim functions.
wherein the additional sub-job requires utilization of the first electronic random access memory capacity that is unused on the first computing system node.	The additional sub-jobs assigned to the require utilization of the first electronic random access memory capacity that is unused on the first computing resource code. For example, Yahoo explained in its dynamic overcommit presentation that the ResourceManager scales overcommit of resources so that the node utilizes the resources (particularly RAM) to perform the sub-jobs corresponding to its allocated containers. (See at 12 minute mark.)

34. Also, for example, the claim chart provided to Yahoo on August 4, 2016, includes the following detailed exemplary explanation of how Yahoo's use of its YARN-5202 patch in a Hadoop computer cluster falls within the scope of claim 1 of the '593 patent:

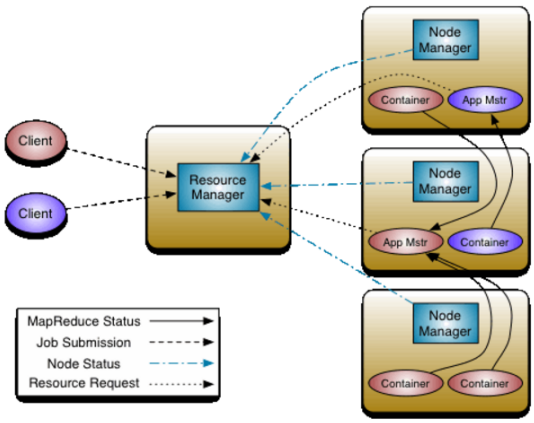
U.S. Patent No. 9,325,593 Claim 1	Yahoo's Hadoop Clusters with Yahoo's Dynamic Overcommit of Node Resources
1. A hadoop computer cluster comprising:	
one or more processors of a master node, wherein the master node comprises a supervisor controller;	Each of Yahoo's Hadoop computer clusters comprises one or more processors of a master node, wherein the master node comprises a ResourceManager, which comprises a supervisor controller. For example, the Hadoop MapReduce Tutorial1 explains "The MapReduce framework consists of a single master ResourceManager, one slave NodeManager per cluster-node, and MRAppMaster per application."
one or more processors of a plurality of computing system nodes, the one or more processors of the plurality of computing system nodes configured to perform computing processes on received sub-jobs, wherein each computing system node comprises an agent controller;	Each of Yahoo's Hadoop computer clusters comprises one or more processors of a plurality of computing system nodes, the one or more processors of the plurality of computing system nodes configured to perform computing processes on received sub-jobs, wherein each computing system node comprises a NodeManager which comprises an agent controller. For example, the Hadoop MapReduce Tutorial explains "The MapReduce framework consists of a

	single master ResourceManager, one slave NodeManager per cluster-node, and MRAppMaster per application." For example, the Hadoop NodeManager Overview explains2 that "The NodeManager is responsible for launching and managing containers on a node. Containers execute tasks as specified by the AppMaster."
wherein each agent controller comprises:	Each agent controller comprises…
a monitoring interface configured to monitor system resources utilization by sub-jobs of its respective computing system node; and	Each agent controller of each NodeManger comprises a monitoring interface configured to monitor system resources utilization by sub-jobs of its respective computing system node. For example, the Hadoop 2.7.2 YARN Architecture3 describes: "The NodeManager is the per-machine framework agent who is responsible for containers, monitoring their resource usage (cpu, memory, disk, network) and reporting the same to the ResourceManager/Scheduler."
a reporting controller configured to transmit the monitored system resources utilization to the supervisor controller in substantially realtime;	Each agent controller of each NodeManager comprises a reporting controller configured to transmit the monitored system resource utilization to the supervisor controller of the ResourceManager in substantially real-time. For example, the Hadoop 2.7.2 YARN Architecture describes: "The NodeManager is the per-machine

	framework agent who is responsible for containers, monitoring their resource usage (cpu, memory, disk, network) and reporting the same to the ResourceManager/Scheduler." Yahoo explained in its dynamic overcommit presentation4 that its NodeManager reports this information in its heartbeat. (See at 10 minute mark.)
wherein the supervisor controller is configured to assign an additional sub-job to a first computing system node based on determining that the utilization of a first electronic random access memory capacity of the first computing system node is below a threshold level, the determining based on the monitored system resources utilization transmitted from the reporting controller of the first computing system node to the supervisor controller,	The supervisor controller of the ResourceManager is configured to assign an additional sub-job to a first computing system node based on determining that the utilization of a first electronic random access memory capacity of the first computing system node is below a low watermark threshold level, the determination based on the monitored system resources utilization transmitted from the reporting controller of the NodeManager of the first computing system node to the supervisor controller of the ResourceManager. For example, the ResourceManager determines that the memory utilization reported by NodeManager is below the corresponding low water mark, as reflected in the "conf" variable, and therefore assigns one or more additional sub-jobs to the first computing system node by incrementing the containers allocated on that node, such as through the memIncrement = getConfInt(conf,…) function

	call at line 1249 of Yahoo's dynamic overcommit patch. Yahoo explained in its dynamic overcommit presentation: "How does the ResourceManager do that scaling? We have a high-level watermark and a low watermark… As long as the node keeps reporting lower than the low watermark utilization we'll increment over time the size of that node inside the ResourceManager so that it'll keep allocating more resources inside that node." (See at 12 minute mark.) As another example, in Yahoo's dynamic overcommit patch, the updateTotalResource function is described as "Adjust overcommit metrics... The amount to overcommit will be re-calculated on next node heartbeat."
wherein the supervisor controller is configured to monitor a second electronic random access memory capacity of a second computing system node,	The supervisor controller is configured to monitor a second electronic random access memory capacity of a second computer system node. For example, the Hadoop 2.7.2 YARN Architecture5 illustrates that the architecture includes multiple nodes: 

wherein the supervisor controller is configured to prevent assignment of additional sub-jobs to a second computing system node based on determining that utilization of the second electronic random access memory capacity is at or above a threshold value, and	The supervisor controller is configured to prevent assignment of additional sub-jobs to a second computing system node based on determining that utilization of the second electronic random access memory capacity is at or above a threshold value. For example, in Yahoo's dynamic overcommit patch, Yahoo's source code comments explain that RM_OVERCOMMIT_MEM_HWM corresponds to: "High water memory utilization mark for overcommit. If the node's memory utilization is greater than or equal to this value then the node's memory will no longer be increased to allow further overcommit. A value <= 0 will disable memory overcommit." Yahoo explained in its dynamic overcommit presentation: "If it's above the high watermark, then we're aggressively scaling back, meaning we don't want to allocate any more containers on that node at this point." (See at 12 minute mark.)
wherein the master node and the plurality of computing system nodes include a computer processor and an electronic storage medium.	The master node and the plurality of computing system nodes include a computer processor and electronic storage medium. For example, the Hadoop 2.7.2 YARN Architecture illustrates:

35. However, despite being aware at least as early as August 4, 2016, of the ’891 and ’593 patents and also aware of how its YARN-5202 patch falls within the scope of representative claims of both of those patents, Yahoo continued to use the YARN-5202 patch on Hadoop clusters in the United States at least until December 16, 2016, when Pepperdata commenced this action.

36. Upon information and belief, Yahoo continues to use its YARN-5202 patch.

37. Moreover, by presenting and explaining to numerous attendees at the 2016 Hadoop Summit that using Yahoo’s YARN-5202 patch provides significant improvements including increasing computing capacity in a Hadoop cluster by 50%, by publishing a video of that same presentation on the popular website youtube.com where it can be viewed for free by any member of the public with Internet access, and by making the YARN-5202 patch available for free Internet download to any member of the public by way of its JIRA, Yahoo actively encouraged Hadoop Summit attendees as well as other members of the public—particularly those operating Hadoop computer clusters—to download Yahoo’s YARN-5202 patch and use it in combination with a Hadoop computer cluster—and Yahoo did so while having knowledge of Pepperdata’s ’891 and ’593 patents and while knowing that the use of its YARN-5202 patch in a Hadoop computer cluster falls within the scope of claims of Pepperdata’s ’891 and ’593 patents.

38. Upon and information and belief, as a result of Yahoo’s active encouragement of Hadoop Summit attendees and others to download Yahoo’s YARN-5202 patch and use it

in combination with a Hadoop computer cluster, Hadoop Summit attendees and others have in fact done so.

39. Yahoo knows that the YARN-5202 patch was created specifically to be used in combination with a Hadoop computer cluster and for the specific purpose of monitoring and allocating resources to facilitate overcommit functionality in a Hadoop computer cluster, and Yahoo knows that using the YARN-5202 patch in a Hadoop computer cluster to monitor and allocate resources to facilitate overcommit functionality falls within the scope of claims in Pepperdata's '891 and '593 patents and constitutes a material part of those claimed inventions. While having that knowledge, Yahoo offered the YARN-5202 patch for free use by others, making it available for free Internet download to any member of the public, and intending that it be combined with and used in a Hadoop computer cluster. Moreover, and again while having that same knowledge, Yahoo offered the YARN-5202 patch for free use by others outside of the United States, making it available for free Internet download to persons outside of the United States, and intending that it be combined with and used in a Hadoop computer cluster outside of the United States.

40. YARN-5202 is not a staple article or commodity of commerce, nor is it one that is suitable for substantial noninfringing use.

41. On information and belief, Yahoo has made, used, and made publicly available the YARN-5202 patch within and outside of the United States.

42. Weeks after Pepperdata's written notice to Yahoo of infringement of the '891 and '593 patents, Yahoo informed Pepperdata that it had stopped making the YARN-5202 patch available for download.

FIRST CLAIM FOR RELIEF:

INFRINGEMENT OF U.S. PATENT NO. 8,849,891

43. Pepperdata incorporates paragraphs 1-42 of this Complaint.

44. This is a claim for patent infringement arising under the patent laws of the United States, Title 35 of the United States Code.

45. Without authority, Yahoo, through its agents, employees, and servants, has manufactured, used, promoted, offered for sale, and/or sold within the United States, and/or supplied in or from the United States, products and/or components covered by one or more claims of the '891 patent, and has, with knowledge of the '891 patent, actively induced others to do the same while knowing that the induced acts constituted infringement of the '891 patent. Moreover, with knowledge of the '891 patent, Yahoo has provided products and components knowing that they, alone or as material components in combination with other components, infringe the '891 patent and has thereby contributed to others' infringement of the '891 patent. Yahoo has thereby infringed, actively induced others to infringe, and/or contributed to others' infringement of one or more claims of the '891 patent, including, for example and without limitation, claims 1-4 of the '891 patent, in violation of 35 U.S.C. § 271, including 35 U.S.C. §§ 271(a), (b), (c), and/or (f). This infringement is currently ongoing. The devices relating to Yahoo's infringement include, without limitation, Yahoo's Hadoop clusters using dynamic overcommit functionality, including those Hadoop clusters using the YARN-5202 software patch.

46. By no later than August 4, 2016, Pepperdata had given Yahoo written notice of its infringement of the '891 patent.

47. Yahoo's infringement of the '891 patent has been and continues to be deliberate and willful.

48. Yahoo's infringement of the '891 patent will continue unless enjoined by this Court.

49. Yahoo has derived and received, and will continue to derive and receive, gains, profits, and advantages from the aforesaid acts of infringement in an amount that is not presently known to Pepperdata.

50. Pepperdata has lost profits from the aforesaid acts of infringement in an amount that is not presently known to Pepperdata.

51. Due to Yahoo's infringement of the '891 patent, Pepperdata has been damaged and is entitled to monetary relief in an amount to be determined at trial.

52. Unless Yahoo is enjoined from infringing the ’891 patent, Pepperdata will continue to suffer irreparable injury for which it has no adequate remedy at law.

SECOND CLAIM FOR RELIEF:

INFRINGEMENT OF U.S. PATENT NO. 9,325,593

53. Pepperdata incorporates paragraphs 1-42 of this Complaint.

54. This is a claim for patent infringement arising under the patent laws of the United States, Title 35 of the United States Code.

55. Without authority, Yahoo, through its agents, employees, and servants, has manufactured, used, promoted, offered for sale, and/or sold within the United States, and/or supplied in or from the United States, products and/or components covered by one or more claims of the ’593 patent, and has, with knowledge of the ’593 patent, actively induced others to do the same while knowing that the induced acts constituted infringement of the ’593 patent. Moreover, with knowledge of the ’593 patent, Yahoo has provided products and components knowing that they, alone or as material components in combination with other components, infringe the ’593 patent and has thereby contributed to others’ infringement of the ’593 patent. Yahoo has thereby infringed, actively induced others to infringe, and/or contributed to others’ infringement of one or more claims of the ’593 patent, including, for example and without limitation, claims 1-3 of the ’593 patent, in violation of 35 U.S.C. § 271, including 35 U.S.C. §§ 271(a), (b), (c), and/or (f). This infringement is currently ongoing. The devices relating to Yahoo’s infringement include, without limitation, Yahoo’s Hadoop clusters using dynamic overcommit functionality, including those Hadoop clusters using the YARN-5202 software patch.

56. By no later than August 4, 2016, Pepperdata had given Yahoo written notice of its infringement of the ’593 patent.

57. Yahoo’s infringement of the ’593 patent has been and continues to be deliberate and willful.

58. Yahoo’s infringement of the ’593 patent will continue unless enjoined by this Court.

59. Yahoo has derived and received, and will continue to derive and receive, gains, profits, and advantages from the aforesaid acts of infringement in an amount that is not presently known to Pepperdata.

60. Pepperdata has lost profits from the aforesaid acts of infringement in an amount that is not presently known to Pepperdata.

61. Due to Yahoo's infringement of the '593 patent, Pepperdata has been damaged and is entitled to monetary relief in an amount to be determined at trial.

62. Unless Yahoo is enjoined from infringing the '593 patent, Pepperdata will continue to suffer irreparable injury for which it has no adequate remedy at law.

DEMAND FOR JUDGMENT

Pepperdata respectfully prays for the following relief:

A. an order adjudging Yahoo to have infringed each of the '891 and '593 patents;

B. a permanent injunction enjoining Yahoo, as well as its officers, agents, servants, employees, and attorneys and those persons in active concert or participation with Yahoo, from infringing either or both of the '891 and '593 patents;

C. an accounting of all gains, profits, and advantages derived by Yahoo's infringement of the '891 and '593 patents and an award of damages adequate to compensate Pepperdata for that infringement;

D. an order trebling damages and/or for exemplary damages due to Yahoo's intentional and willful conduct;

E. an award of prejudgment and post judgment interest and costs to this action against Yahoo;

F. an award to Pepperdata of its attorneys' fees incurred in connection with this action; and

G. such other and further relief that the Court deems just and proper.

/ / /

/ / /

/ / /

Respectfully submitted,

KNOBBE, MARTENS, OLSON & BEAR, LLP

Dated: February 10, 2017

By: */s/ Douglas G. Muehlhauser*

Douglas G. Muehlhauser (State Bar No. 179,495)
doug.muehlhauser@knobbe.com
Alan G. Laquer (State Bar No. 259,257)
alan.laquer@knobbe.com
KNOBBE, MARTENS, OLSON & BEAR, LLP
2040 Main Street, Fourteenth Floor
Irvine, CA 92614
Telephone: 949-760-0404
Facsimile: 949-760-9502

Attorneys for Plaintiff Pepperdata, Inc.

JURY DEMAND

Pursuant to Fed. R. Civ. P. 38(b), Plaintiff Pepperdata, Inc. demands a trial by jury of all issues raised by this Complaint that are triable by jury.

Respectfully submitted,

KNOBBE, MARTENS, OLSON & BEAR, LLP

Dated: February 10, 2017

By: */s/ Douglas G. Muehlhauser*

Douglas G. Muehlhauser (State Bar No. 179,495)
doug.muehlhauser@knobbe.com
Alan G. Laquer (State Bar No. 259,257)
alan.laquer@knobbe.com
KNOBBE, MARTENS, OLSON & BEAR, LLP
2040 Main Street, Fourteenth Floor
Irvine, CA 92614
Telephone: 949-760-0404
Facsimile: 949-760-9502

Attorneys for Plaintiff Pepperdata, Inc.

25206186