US 20110295813A1

(19) United States

(12) Patent Application Publication

(10) Pub. No.: US 2011/0295813 A1

(43) Pub. Date: Dec. 1, 2011

Pickard et al.

(54) **MECHANISM FOR MANAGING AND ARCHIVING SYSTEM AND APPLICATION LOG FILES**

(76) Inventors: **Jonathan J. Pickard**, Cave Creek, AZ (US); **William W. Foster, JR.**, Raleigh, NC (US)

(21) Appl. No.: **12/787,664**

(22) Filed: **May 26, 2010**

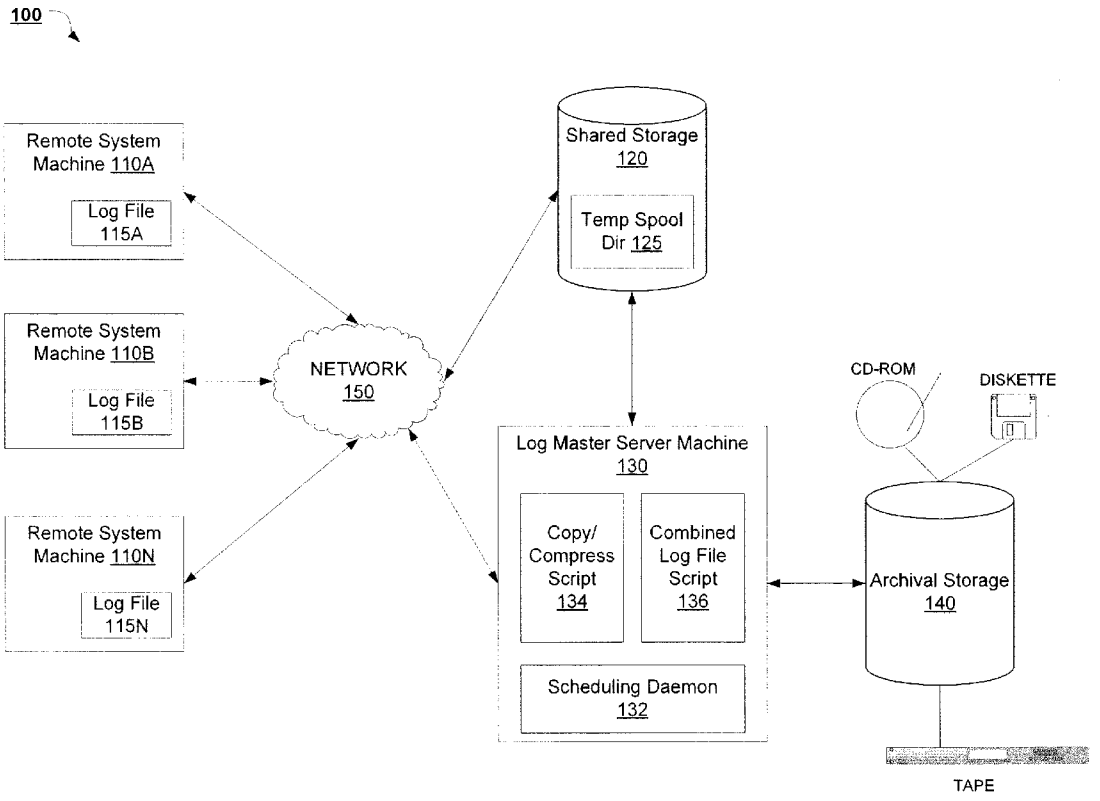
Publication Classification

(51) **Int. Cl.**
G06F 17/30 (2006.01)

(52) **U.S. Cl.** **707/672**; 707/E17.005; 707/687

(57) **ABSTRACT**

A mechanism for managing and archiving system and application log files is disclosed. A method of the invention includes accessing log files on shared storage that satisfy grouping requirements, combining the accessed log files that satisfy the grouping requirements into a single combined log file, compressing the single combined log file, and storing the single combined log file to an archival storage location.



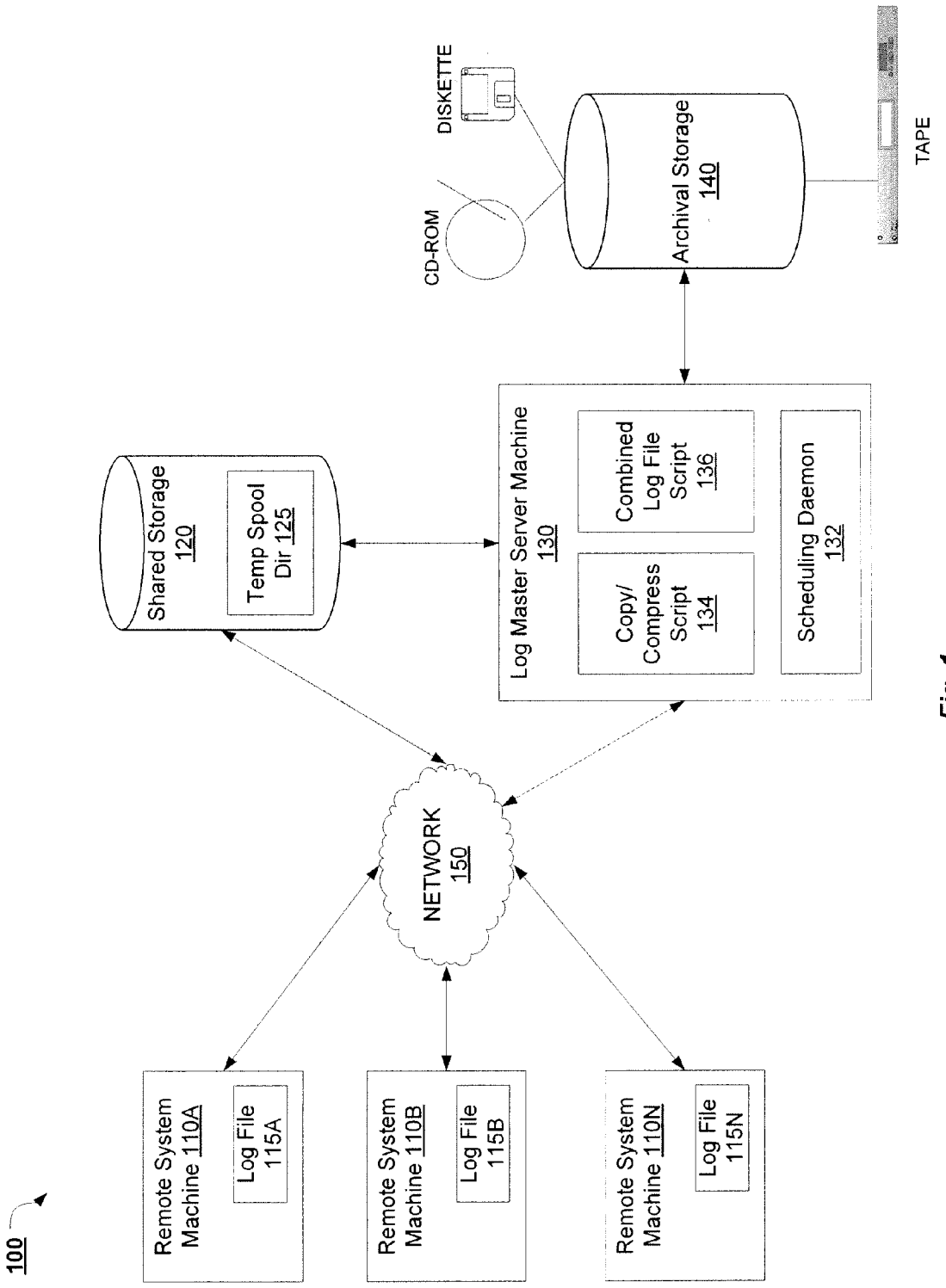

100
Remote System Machine 110A
Log File 115A
Remote System Machine 110B
Log File 115B
Remote System Machine 110N
Log File 115N
NETWORK 150
Shared Storage 120
Temp Spool Dir 125
Log Master Server Machine 130
Copy/ Compress Script 134
Combined Log File Script 136
Scheduling Daemon 132
Archival Storage 140
CD-ROM
DISKETTE
TAPE

Fig. 1

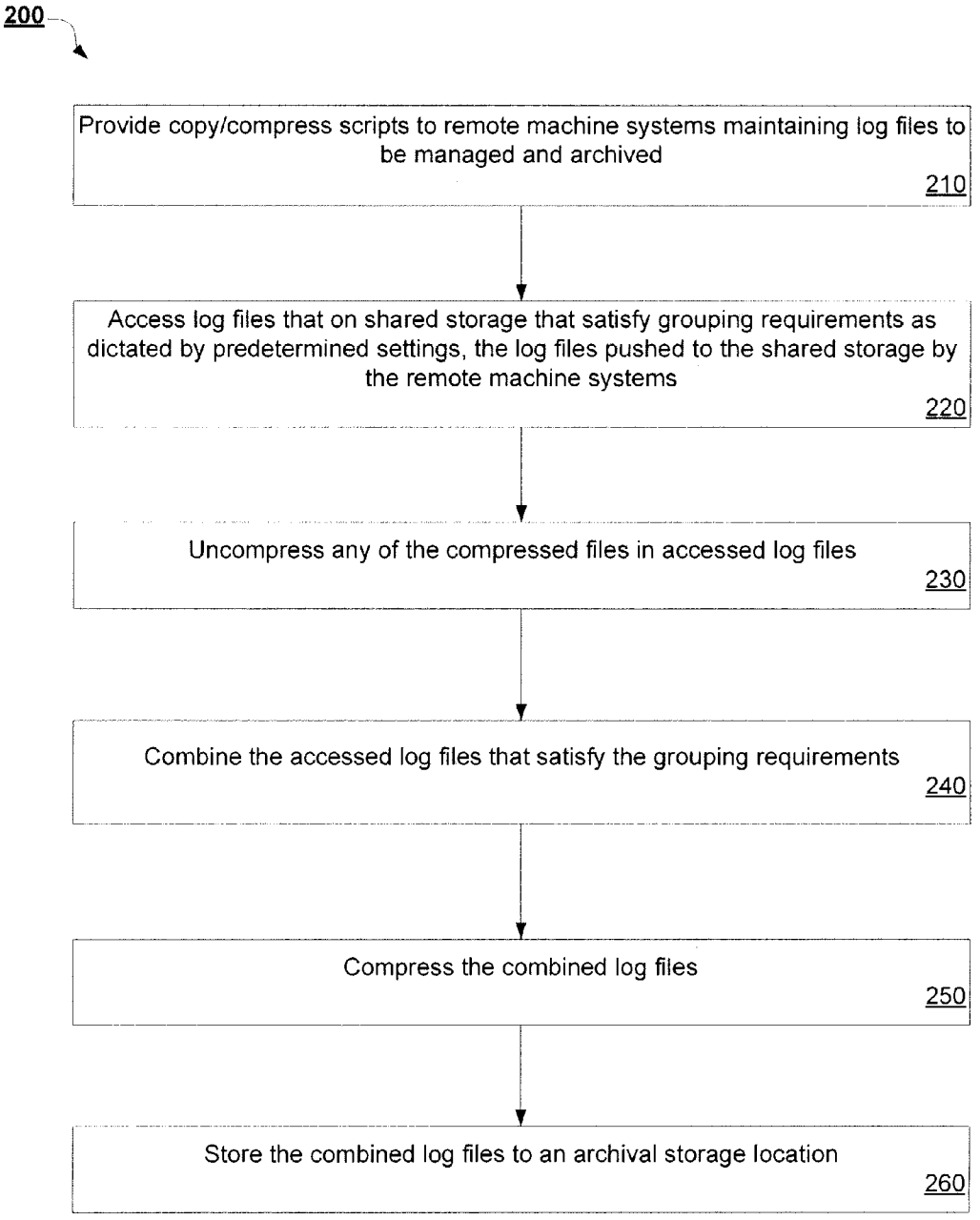

200
Provide copy/compress scripts to remote machine systems maintaining log files to be managed and archived
210
Access log files that on shared storage that satisfy grouping requirements as dictated by predetermined settings, the log files pushed to the shared storage by the remote machine systems
220
Uncompress any of the compressed files in accessed log files
230
Combine the accessed log files that satisfy the grouping requirements
240
Compress the combined log files
250
Store the combined log files to an archival storage location
260

Fig. 2

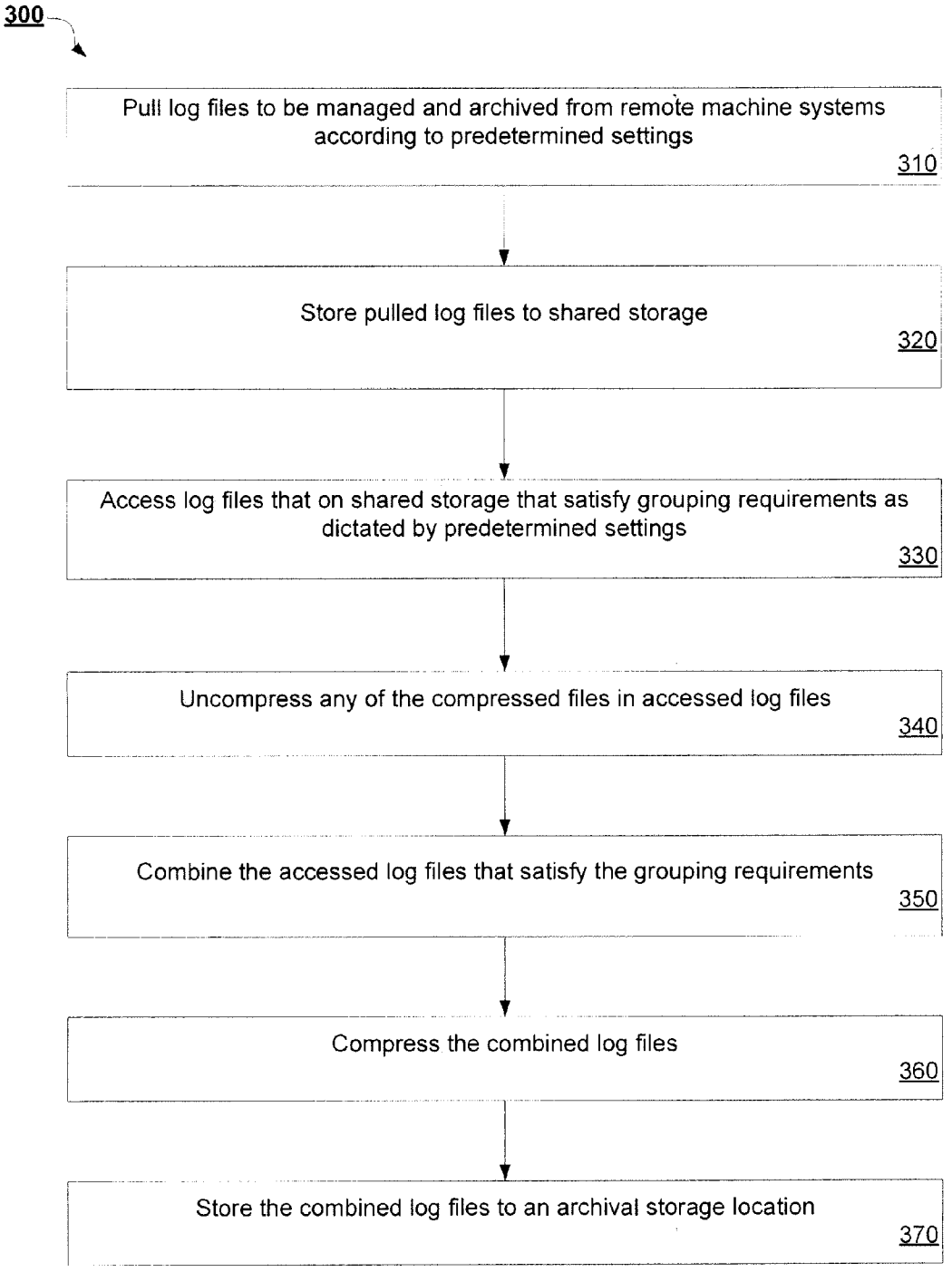

300
Pull log files to be managed and archived from remote machine systems according to predetermined settings
310
Store pulled log files to shared storage
320
Access log files that on shared storage that satisfy grouping requirements as dictated by predetermined settings
330
Uncompress any of the compressed files in accessed log files
340
Combine the accessed log files that satisfy the grouping requirements
350
Compress the combined log files
360
Store the combined log files to an archival storage location
370

Fig. 3

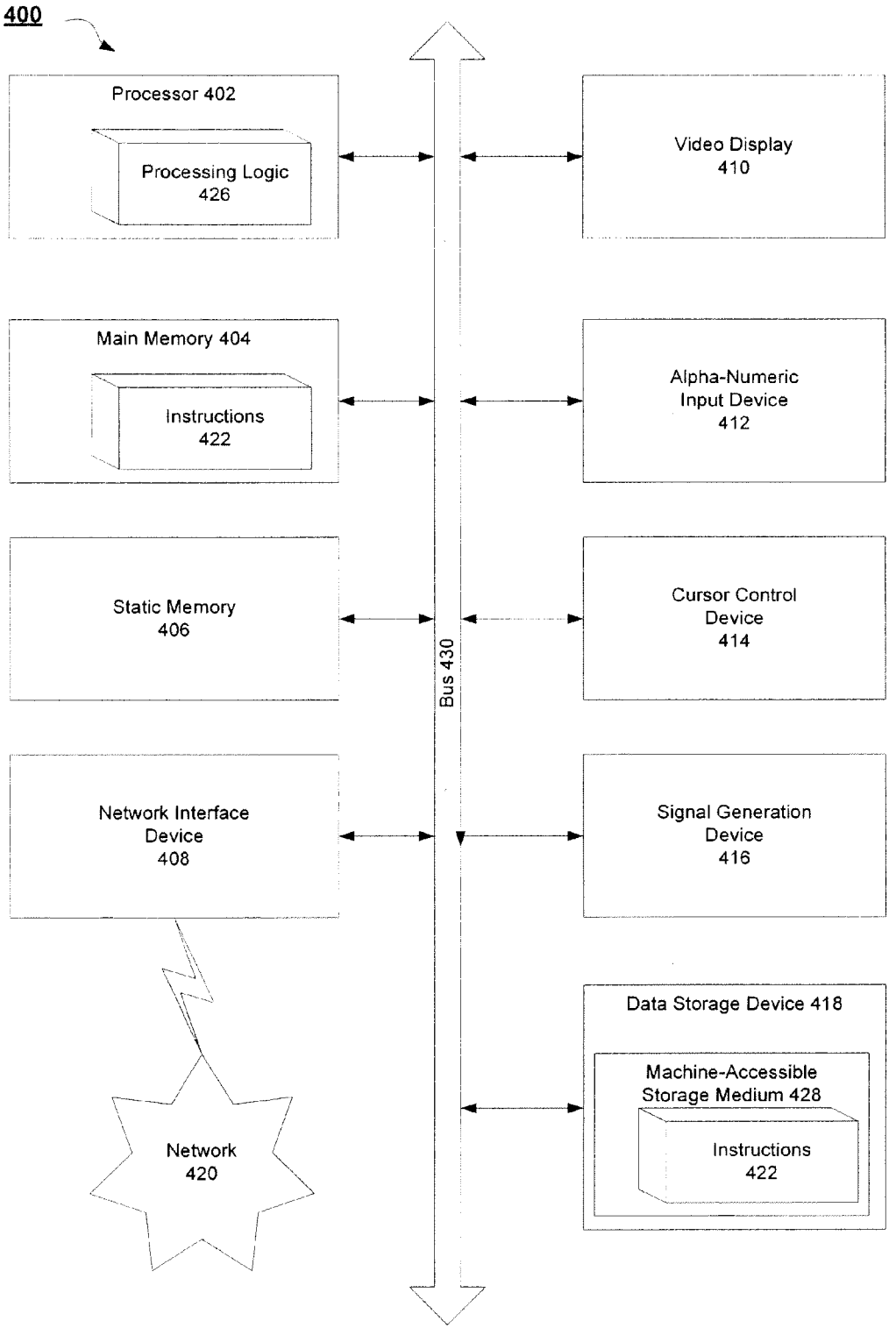

400
Processor 402
Processing Logic
426
Main Memory 404
Instructions
422
Static Memory
406
Network Interface
Device
408
Network
420
Bus 430
Video Display
410
Alpha-Numeric
Input Device
412
Cursor Control
Device
414
Signal Generation
Device
416
Data Storage Device 418
Machine-Accessible
Storage Medium 428
Instructions
422

Fig. 4

MECHANISM FOR MANAGING AND ARCHIVING SYSTEM AND APPLICATION LOG FILES

TECHNICAL FIELD

[0001] The embodiments of the invention relate generally to system and application log files and, more specifically, relate to a mechanism for managing and archiving system and application log files.

BACKGROUND

[0002] Due to various reasons, log file management and archival is a common requirement in Information Technology (IT). These reasons may include legal requirements, fraud prevention and detection, statistic collection and analysis, incident and problem detection.

[0003] The process of managing and archiving these logs is very manual and time-consuming without an automated process to manage and store the files. An organization may have many physical servers that each capture log data whenever any action occurs. For example, when someone visits the organization's website, a log file may be created on one of the servers that serves the organization's website. However, there may be multiple servers each tasked with serving the organization's website and each may serve the same data. As a result, different users may hit different servers when they access the website. Because each server has its own log file specific to the server and separate from the other servers, log files for the same task may be kept on many different servers. Log files are generally not logged to a unified log file. This results in the problem of multiple, spread-out log files without any process to bring them together in a concise format for ease of management and archiving.

BRIEF DESCRIPTION OF THE DRAWINGS

[0004] The invention will be understood more fully from the detailed description given below and from the accompanying drawings of various embodiments of the invention. The drawings, however, should not be taken to limit the invention to the specific embodiments, but are for explanation and understanding only.

[0005] FIG. **1** is a block diagram of a log file system according to an embodiment of the invention;

[0006] FIG. **2** is a flow diagram illustrating a method for managing and archiving system and application log files according to an embodiment of the invention;

[0007] FIG. **3** is a flow diagram illustrating a method for an alternative e embodiment for managing and archiving system and application log files according to an embodiment of the invention; and

[0008] FIG. **4** illustrates a block diagram of one embodiment of a computer system.

DETAILED DESCRIPTION

[0009] Embodiments of the invention provide for managing and archiving system and application log files. A method of embodiments of the invention includes accessing log files on shared storage that satisfy grouping requirements, combining the accessed log files that satisfy the grouping requirements into a single combined log file, compressing the single combined log file, and storing the single combined log file to an archival storage location.

[0010] In the following description, numerous details are set forth. It will be apparent, however, to one skilled in the art, that the present invention may be practiced without these specific details. In some instances, well-known structures and devices are shown in block diagram form, rather than in detail, in order to avoid obscuring the present invention.

[0011] Some portions of the detailed descriptions which follow are presented in terms of algorithms and symbolic representations of operations on data bits within a computer memory. These algorithmic descriptions and representations are the means used by those skilled in the data processing arts to most effectively convey the substance of their work to others skilled in the art. An algorithm is here, and generally, conceived to be a self-consistent sequence of steps leading to a desired result. The steps are those requiring physical manipulations of physical quantities. Usually, though not necessarily, these quantities take the form of electrical or magnetic signals capable of being stored, transferred, combined, compared, and otherwise manipulated. It has proven convenient at times, principally for reasons of common usage, to refer to these signals as bits, values, elements, symbols, characters, terms, numbers, or the like.

[0012] It should be borne in mind, however, that all of these and similar terms are to be associated with the appropriate physical quantities and are merely convenient labels applied to these quantities. Unless specifically stated otherwise, as apparent from the following discussion, it is appreciated that throughout the description, discussions utilizing terms such as "sending", "receiving", "attaching", "forwarding", "caching", "accessing", "combining", "compressing", "storing", or the like, refer to the action and processes of a computer system, or similar electronic computing device, that manipulates and transforms data represented as physical (electronic) quantities within the computer system's registers and memories into other data similarly represented as physical quantities within the computer system memories or registers or other such information storage, transmission or display devices.

[0013] The present invention also relates to an apparatus for performing the operations herein. This apparatus may be specially constructed for the required purposes, or it may comprise a general purpose computer selectively activated or reconfigured by a computer program stored in the computer. Such a computer program may be stored in a machine readable storage medium, such as, but not limited to, any type of disk including floppy disks, optical disks, CD-ROMs, and magnetic-optical disks, read-only memories (ROMs), random access memories (RAMs), EPROMs, EEPROMs, magnetic or optical cards, or any type of media suitable for storing electronic instructions, each coupled to a computer system bus.

[0014] The algorithms and displays presented herein are not inherently related to any particular computer or other apparatus. Various general purpose systems may be used with programs in accordance with the teachings herein, or it may prove convenient to construct more specialized apparatus to perform the required method steps. The required structure for a variety of these systems will appear as set forth in the description below. In addition, the present invention is not described with reference to any particular programming language. It will be appreciated that a variety of programming languages may be used to implement the teachings of the invention as described herein.

[0015] The present invention may be provided as a computer program product, or software, that may include a machine-readable medium having stored thereon instructions, which may be used to program a computer system (or other electronic devices) to perform a process according to the present invention. A machine-readable medium includes any mechanism for storing or transmitting information in a form readable by a machine (e.g., a computer). For example, a machine-readable (e.g., computer-readable) medium includes a machine (e.g., a computer) readable storage medium (e.g., read only memory ("ROM"), random access memory ("RAM"), magnetic disk storage media, optical storage media, flash memory devices, etc.), a machine (e.g., computer) readable transmission medium (non-propagating electrical, optical, or acoustical signals), etc.

[0016] Embodiments of the invention provide a mechanism for managing and archiving system and application log files. The mechanism for managing and archiving log files provides improvements around the scripts and steps that run on a master log file system server that relate to condition and error checking and reporting, and the option for "pulling" the log files from the remote hosts, rather than the remote hosts pushing them to the shared storage location that the master log file system server can access. Embodiments of the invention automate each step of the log file management and archival process and require little to no oversight, allowing an IT department to work on more time-consuming tasks.

[0017] FIG. **1** is a block diagram of a log file system **100** according to an embodiment of the invention. In one embodiment, log file system **100** includes one or more remote system machines **110**A-**110**N, a shared storage location **120**, a log master server machine **130**, and an archival storage location **140**. The shared storage locations **120** may be a separate database machine or may be co-located with any of the remote system machines **110**A-**110**N. Similarly, the archival storage location **140** may be a separate database machine or may be co-located with the log master server machine **130**. The remote system machines **110**A-**110**N are communicably coupled to the shared storage source **120** via network **150**. Network **150** may be a private network (e.g., a local area network (LAN), wide area network (WAN), intranet, etc.) or a public network (e.g., the Internet). In some embodiments, the remote system machines **110**A-**110**N may be directly connected (not shown) to the shared storage source **120**.

[0018] In some embodiments, remote system machines **110**A-**110**N may be operating as part of a larger organization to provide data and other service for the organization. Each remote system machine **110**A-**110**N, upon performance of any action at the remote system machine **110**A-**110**N, actively writes to a log file **115**A-**115**N stored on the remote system machine **110**A-**110**N. In some embodiments, examples of data that could be logged include, but are not limited to, web logs storing access request for specific web sites hosted on servers, login attempts to a server, and automated monitoring messages from the operating system. These log files could be stored in one or more different formats. The remote system machine **110**A-**110**N is configured to stop writing to its log file **115**A-**115**N after a certain amount of time or a certain amount of log entries. In some embodiments, the remote system machine **110**A-**110**N writes to its specific log file **115**A-**115**N until that log file is rotated out by another application (such as logrotate).

[0019] Once the log file **115**A-**115**N is rotated out and is no longer being written to by the system or the application, it is moved or copied to a temporary spool directory **125** on a shared storage source **120**. In some embodiments, the log file **115**A-**115**N may be compressed at this time to reduce the amount of space it uses. The copy/move and compression of this file is initiated by a regularly scheduled script **134** that is started by a scheduling daemon **132**, such as 'crond', of the log master server machine **130**. In some embodiments, the copy/compress script **134** is placed on each remote system machine **110**A-**110**N and runs according to predetermined time settings.

[0020] The shared storage source **120** is accessible by all of the systems **110**A-**110**N whose log files **115**A-**115**N are being managed and archived by the log master server machine **130**. In addition, the shared storage source **120** is also accessible by the log master server machine **130**. The log master server machine **130** is a master server that performs additional steps to combine and archive the log files in embodiments of the invention.

[0021] On a regular basis, and through the scheduling daemon **132**, a script **136** runs on the log master server machine **130** that uncompresses any compressed log files on the shared storage source **120**, combines log files from grouped systems, and then stores the combined log file to an archival storage location **140**. A system may be grouped by the type of data it serves, the department it is associated with, the services it runs, and so on.

[0022] In one embodiment, an administrator may logically group logs together when developing a profile that gets passed to the combined log file script **136**. This profile basically provides grouping requirement that may be provided as arguments to the combined log file script **136**. In embodiments of the invention, multiple profiles may be created to accommodate the variety of types of log files. For example, three separate profiles could be created for log files falling into the three example groups previously discussed of web logs, login attempts, and automated monitoring messages. As a result, each group of log files would be archived separately. This allows the compressed archived file to be easily analyzed if ever needed.

[0023] In one embodiment, the combined log file script **136** accesses the temporary spool directory **125** of the shared storage **120**, uncompresses any compressed files, and merges them to one large log file. This one large log file may be stored in the temporary spool directory **125** or on another spool directory on the log master server machine **130**. The combined log file script **136** identifies which log files should be combined based on the servers the files originated from and the time the files were stored. In one embodiment, the script **136** is given certain arguments that help identify these log files that should be combined.

[0024] The archival location **140** stores the combined log files for later retrieval and analysis as needed, or archives these files to a backup medium, such as CD-ROM, tape, or diskette, if desired. The archived and combined log can then be stored permanently in-place. Once the logs are combined by the log master server machine **130**, the individual logs **115**A-**115**N stored in the temporary spool directory **125** of the shared storage **120** are deleted. In one embodiment, once the log files are converged/merged onto an archival location **140**, the archive log files are compressed again.

[0025] In one alternative embodiment, an option exists to temporarily store the log files **115**A-**115**N locally on the remote systems **110**A-**110**N, rather than copy or move them to the shared storage source **120**. If this method is employed,

the scheduled script **134** on the logrunner system pulls the script from the remote systems **110**A-**110**N storing the log files **115**A-**115**N using a protocol such as HyperText Transport Protocol (HTTP), File Transfer Protocol (FTP), Secure Copy (SCP), or RSYNC. An application/daemon should be configured on the remote systems **110**A-**110**N storing the log files **115**A-**115**N to allow the log master server machine **130** to access the remote systems **110**A-**110**N via that method.

[0026] For instance, if HTTP is to be used, the remote system **110**A-**110**N should run a web server application, such as Apache, that is configured to allow the log master server machine **130** to retrieve a specific log file **115**A-**115**N in its temporary location on the remote system **110**A-**110**N and store it to the temporary spool directory **125** on the shared storage source **120** in a pull copy fashion. This avoids a need to place a script on each remote system **110**A-**110**N that performs the copying/compression function. The log master server machine **130** accomplishes this job instead, thereby saving resources on the remote system machines **110**A-**110**N.

[0027] Embodiments of the invention are also able to automatically troubleshoot the management and archival process of log files. Each script **134**, **136** that is produced by the log master server machine **130** includes error detection code with conditions that indicate any problems that may occur in the process. For example, the error code for scripts **134** running on each individual remote system machine **110**A-**110**N may check that the environment is properly set up, directories exist, that log files **115**A-**115**N that it thinks should be there are there, and that the file system that it is writing to (shared storage source) has enough space available in advance before it starts copying data over. The error code for scripts running on the log master server machine **130** may check to make sure files are there from individual remote system machines **110**A-**110**N, check to see if it is okay to ignore any missing log files, confirm there is available space in the archival storage location **140**, and so on. In addition, this error checking code provides explanations about why any log files were not processed so that an administrator can determine what went wrong.

[0028] FIG. **2** is a flow diagram illustrating a method **200** for managing and archiving system and application log files according to an embodiment of the invention. Method **200** may be performed by processing logic that may comprise hardware (e.g., circuitry, dedicated logic, programmable logic, microcode, etc.), software (such as instructions run on a processing device), firmware, or a combination thereof. In one embodiment, method **200** is performed by log master server machine **130** of FIG. **1**.

[0029] Method **200** begins at block **210** where copy/compress scripts are provided to remote system machines that maintain log files to be managed and archived by the log master server machine. In one embodiment, these copy/compress scripts are provided from a scheduling daemon on the log master server machine that configures the timing during which the copy/compress scripts should be run. The copy/compress scripts cause the remote server machines to copy their log files to a shared storage source. The copy/compress script may also cause the log files to be compressed at this point. At block **220**, any log files that are on the shared storage location that satisfy certain grouping requirements are accessed. In one embodiment, these grouping requirements dictate how log files should be combined. For instance, the grouping requirement may include arguments that specify accessing all log files from a particular server (e.g., a web page server) that were created on a certain day. These log files have been pushed to the shared storage location per the copy/compress script provided to the remote system machines at block **210**.

[0030] At block **230**, any of the accessed log files that are compressed are uncompressed. Then, at block **240**, these accessed log files are combined into a single file. The combined log files are then compressed at block **250**. Finally, the single combined log file is stored to an archival storage location at block **260**. The archival location provides for later retrieval and analysis of the log files as needed, or for further archival to a backup medium, such as CD-ROM, tape, or diskette, if desired.

[0031] FIG. **3** is a flow diagram illustrating a method **300** for an alternative embodiment for managing and archiving system and application log files according to an embodiment of the invention. Method **300** may be performed by processing logic that may comprise hardware (e.g., circuitry, dedicated logic, programmable logic, microcode, etc.), software (such as instructions run on a processing device), firmware, or a combination thereof. In one embodiment, method **300** is performed by log master server machine **130** of FIG. **1**.

[0032] Method **300** begins at block **310** where log files to be managed and archived by the log master server machine are pulled from remote machine systems according to predetermined settings. In one embodiment, a scheduling daemon on the log master server machine may cause a script to run at preconfigured time intervals that pulls the log files from particular remote system machines. At block **320**, these pulled log files are saved to a shared storage location. In one embodiment, the log files may be saved to a temporary spool directory on the shared storage location.

[0033] At block **330**, any log files that are on the shared storage location that satisfy certain grouping requirements are accessed. In one embodiment, these grouping requirements dictate how log files should be combined. For instance, the grouping requirement may include arguments that specify accessing all log files from a particular server (e.g., a web page server) that were created on a certain day.

[0034] At block **340**, any of the accessed log files that are compressed are uncompressed. Then, at block **350**, these accessed log files are combined into a single file. The combined log files in the single file are then compressed at block **360**. Finally, the single combined log file is stored to an archival storage location at block **370**. The archival location provides for later retrieval and analysis of the log files as needed, or for further archival to a backup medium, such as CD-ROM, tape, or diskette, if desired.

[0035] FIG. **4** illustrates a diagrammatic representation of a machine in the exemplary form of a computer system **400** within which a set of instructions, for causing the machine to perform any one or more of the methodologies discussed herein, may be executed. In alternative embodiments, the machine may be connected (e.g., networked) to other machines in a LAN, an intranet, an extranet, or the Internet. The machine may operate in the capacity of a server or a client machine in a client-server network environment, or as a peer machine in a peer-to-peer (or distributed) network environment. The machine may be a personal computer (PC), a tablet PC, a set-top box (STB), a Personal Digital Assistant (PDA), a cellular telephone, a web appliance, a server, a network router, switch or bridge, or any machine capable of executing a set of instructions (sequential or otherwise) that specify actions to be taken by that machine. Further, while only a

single machine is illustrated, the term "machine" shall also be taken to include any collection of machines that individually or jointly execute a set (or multiple sets) of instructions to perform any one or more of the methodologies discussed herein.

[0036] The exemplary computer system **400** includes a processing device **402**, a main memory **404** (e.g., read-only memory (ROM), flash memory, dynamic random access memory (DRAM) (such as synchronous DRAM (SDRAM) or Rambus DRAM (RDRAM), etc.), a static memory **406** (e.g., flash memory, static random access memory (SRAM), etc.), and a data storage device **418**, which communicate with each other via a bus **430**.

[0037] Processing device **402** represents one or more general-purpose processing devices such as a microprocessor, central processing unit, or the like. More particularly, the processing device may be complex instruction set computing (CISC) microprocessor, reduced instruction set computer (RISC) microprocessor, very long instruction word (VLIW) microprocessor, or processor implementing other instruction sets, or processors implementing a combination of instruction sets. Processing device **402** may also be one or more special-purpose processing devices such as an application specific integrated circuit (ASIC), a field programmable gate array (FPGA), a digital signal processor (DSP), network processor, or the like. The processing device **402** is configured to execute the processing logic **426** for performing the operations and steps discussed herein.

[0038] The computer system **400** may further include a network interface device **408**. The computer system **400** also may include a video display unit **410** (e.g., a liquid crystal display (LCD) or a cathode ray tube (CRT)), an alphanumeric input device **412** (e.g., a keyboard), a cursor control device **414** (e.g., a mouse), and a signal generation device **416** (e.g., a speaker).

[0039] The data storage device **418** may include a machine-accessible storage medium **428** on which is stored one or more set of instructions (e.g., software **422**) embodying any one or more of the methodologies of functions described herein. For example, software **422** may store instructions to perform managing and archiving system and application log files by log master server machine **130** described with respect to FIG. **1**. The software **422** may also reside, completely or at least partially, within the main memory **404** and/or within the processing device **402** during execution thereof by the computer system **400**; the main memory **404** and the processing device **402** also constituting machine-accessible storage media. The software **422** may further be transmitted or received over a network **420** via the network interface device **408**.

[0040] The machine-readable storage medium **428** may also be used to store instructions to perform methods **200** and **300** for managing and archiving system and application log files described with respect to FIGS. **2** and **3**, and/or a software library containing methods that call the above applications. While the machine-accessible storage medium **428** is shown in an exemplary embodiment to be a single medium, the term "machine-accessible storage medium" should be taken to include a single medium or multiple media (e.g., a centralized or distributed database, and/or associated caches and servers) that store the one or more sets of instructions. The term "machine-accessible storage medium" shall also be taken to include any medium that is capable of storing, encoding or carrying a set of instruction for execution by the machine and that cause the machine to perform any one or more of the methodologies of the present invention. The term "machine-accessible storage medium" shall accordingly be taken to include, but not be limited to, solid-state memories, and optical and magnetic media.

[0041] Whereas many alterations and modifications of the present invention will no doubt become apparent to a person of ordinary skill in the art after having read the foregoing description, it is to be understood that any particular embodiment shown and described by way of illustration is in no way intended to be considered limiting. Therefore, references to details of various embodiments are not intended to limit the scope of the claims, which in themselves recite only those features regarded as the invention.

What is claimed is:

1. A computer-implemented method, comprising:

accessing, by a log master server machine, log files on shared storage that satisfy grouping requirements, wherein the log files are created on a plurality of disparate remote system machines and placed on the shared storage;

combining, by the log master server machine, the accessed log files that satisfy the grouping requirements into a single combined log file;

compressing, by the log master server machine, the single combined log file; and

storing, by the log master server machine, the single combined log file to an archival storage location.

2. The method of claim **1**, wherein the log files are pushed to the shared storage by one or more of the plurality of disparate remote system machines that created the log files, the pushing caused by a copy script placed on each of the one or more disparate remote system machines by the log master server machine.

3. The method of claim **1**, wherein the log files are pulled to the shared storage from one or more of the plurality of disparate remote system machines that created the log files, the pulling caused by a pull script run by the log master server machine.

4. The method of claim **1**, wherein the grouping requirements include a name of a remote system machine the created the log files and a time interval that the log files were created.

5. The method of claim **4**, wherein the grouping requirements are provided as arguments into a combine log files script rung by the log master server machine.

6. The method of claim **1**, further comprising running error checking code on the log master server to at least one of check that all accessed log files originate from one or more of the disparate remote system machines authorized by the log master server, check that it is okay to ignore any missing log files, and confirm that there is available space in the archival storage location.

7. The method of claim **6**, further comprising running error checking code on each of the one or more disparate remote system machines to at least one of check that the environment of the disparate remote system machine is properly set up, check that log files directories exist on the disparate remote system machine, check that the log files are actually there, and check that the shared storage source has enough space available in advance before it starts copying data over to the shared storage source.

8. The method of claim **7**, wherein the error checking code provides explanations to an administrator of the log master server machine about any errors it encounters.

9. A system, comprising:
a shared storage device;
an archival storage device; and
a log master server device communicably coupled to the shared storage device and the archival storage device, the log master server machine configured to:
access log files on the shared storage device that satisfy grouping requirements, wherein the log files are created on a plurality of disparate remote system machines and placed on the shared storage device;
combine the accessed log files that satisfy the grouping requirements into a single combined log file;
compress the single combined log file; and
store the single combined log file to the archival storage device.

10. The system of claim **9**, wherein the log files are pushed to the shared storage device by one or more of the disparate remote system machines that create the log files, the pushing caused by a copy script placed on each of the one or more disparate remote system machines by the log master server device.

11. The system of claim **9**, wherein the log files are pulled to the shared storage from one or more of the plurality of disparate remote system machines that create the log files, the pulling caused by a pull script run by the log master server device.

12. The system of claim **9**, wherein the grouping requirements include a name of a remote system machine the created the log files and a time interval that the log files were created.

13. The system of claim **12**, wherein the grouping requirements are provided as arguments into a combine log files script rung by the log master server device.

14. The system of claim **9**, further comprising running error checking code on the log master server device to at least one of check that all accessed log files originate from one or more of the plurality of disparate remote system machines authorized by the log master server device, check that it is okay to ignore any missing log files, and confirm that there is available space in the archival storage location.

15. The system of claim **14**, further comprising running error checking code on each of the one or more disparate remote system machines to at least one of check that the environment of the disparate remote system machine is properly set up, check that log files directories exist on the disparate remote system machine, check that the log files are actually there, and check that the shared storage source has enough space available in advance before it starts copying data over to the shared storage source.

16. An article of manufacture comprising a machine-readable storage medium including data that, when accessed by a machine, cause the machine to perform operations comprising:
accessing, by a log master server machine, log files on shared storage that satisfy grouping requirements, wherein the log files are created on a plurality of disparate remote system machines and placed on the shared storage;
combining, by the log master server machine, the accessed log files that satisfy the grouping requirements into a single combined log file;
compressing, by the log master server machine, the single combined log file; and
storing, by the log master server machine, the single combined log file to an archival storage location.

17. The article of manufacture of claim **16**, wherein the log files are pushed to the shared storage by one or more of the plurality of disparate remote system machines that created the log files, the pushing caused by a copy script placed on each of the one or more disparate remote system machines by the log master server machine.

18. The article of manufacture of claim **16**, wherein the log files are pulled to the shared storage from one or more of the plurality of disparate remote system machines that created the log files, the pulling caused by a pull script run by the log master server machine.

19. The article of manufacture of claim **16**, wherein the grouping requirements include a name of a remote system machine the created the log files and a time interval that the log files were created.

20. The article of manufacture of claim **19**, wherein the grouping requirements are provided as arguments into a combine log files script rung by the log master server machine.

* * * * *