

TITLE

DYNAMIC STORAGE MANAGEMENT USING PREDICTIVE ANALYTICS

TECHNICAL FIELD

[0001] The following disclosure relates generally to storage management and, more specifically, to systems and methods for optimizing data retention policies.

SUMMARY

[0002] In one aspect, the present disclosure provides a computer-implemented method that includes collecting, by a data storage management system, file data of log files associated with a distributed file system, the file data comprising size values and temporal values associated with the log files; partitioning, by the data storage management system, the log files based on the temporal values to create temporal-based partitions; identifying, by the data storage management system, historic patterns in the file data; and optimizing, by the data storage management system, a data retention policy of the distributed file system based on the historic patterns.

[0003] In one aspect, the present disclosure provides a data storage management system for optimizing a data retention policy of a distributed file system. The data storage management system includes a data processing module to: retrieve file data of log files associated with the distributed file system, the file data comprising size values and temporal values associated with the log files; and partition the log files based on the temporal values to create temporal-based partitions. The data storage management system may further include a predictive module to communicate with the data processing module, wherein the predictive module is to identify historic patterns in the file data. The data storage management system may further include an optimizer module to communicate with the predictive module, wherein the optimizer module is to optimize the data retention policy of the distributed file system based on the historic patterns.

BRIEF DESCRIPTION OF THE DRAWINGS

[0004] In the description, for purposes of explanation and not limitation, specific details are set forth, such as particular aspects, procedures, techniques, etc., to provide a thorough understanding of the present technology. However, it will be apparent to one skilled in the art that the present technology may be practiced in other aspects that depart from these specific details.

[0005] The accompanying drawings, where like reference numerals refer to identical or functionally similar elements throughout the separate views, together with the detailed

description below, are incorporated in and form part of the specification, and they serve to further illustrate aspects of concepts that include the claimed disclosure and explain various principles and advantages of those aspects.

[0006] The systems and methods disclosed herein have been represented, where appropriate, by conventional symbols in the drawings, showing only those specific details that are pertinent to understanding the various aspects of the present disclosure so as not to obscure the disclosure with details that will be readily apparent to those of ordinary skill in the art, having the benefit of the description herein.

[0007] FIG. 1 illustrates a system for managing storage in a distributed file system, according to at least one aspect of the present disclosure.

[0008] FIG. 2 illustrates a method of managing storage in a distributed file system, according to at least one aspect of the present disclosure.

[0009] FIG. 3 illustrates a method of identifying historic patterns in file data of a distributed file system, according to at least one aspect of the present disclosure.

[0010] FIG. 4 illustrates a method of predicting a data retention policy for a distributed file system, according to at least one aspect of the present disclosure.

[0011] FIG. 5 is a method to implement an optimization of a data retention policy associated, in accordance with at least one aspect of the present disclosure.

[0012] FIG. 6 is a block diagram of a computer apparatus with data processing subsystems or components, according to at least one aspect of the present disclosure.

[0013] FIG. 7 is a diagrammatic representation of an example system that includes a host machine within which a set of instructions to perform any one or more of the methodologies discussed herein may be executed, according to at least one aspect of the present disclosure.

DESCRIPTION

[0014] The following disclosure may provide exemplary systems, devices, and methods for conducting a financial transaction and related activities. Although reference may be made to such financial transactions in the examples provided below, aspects are not so limited. That is, the systems, methods, and apparatuses may be utilized for any suitable purpose.

[0015] Before discussing specific embodiments, aspects, or examples, some

descriptions of terms used herein are provided below.

[0016] As used herein, the term “system” may refer to one or more computing devices or combinations of computing devices (e.g., processors, servers, client devices, software applications, components of such, and/or the like).

[0017] The proliferation of data has given rise to many challenges in data storage and data management. Large data volumes necessitated powerful hardware, efficient algorithms, and scalable systems to process and analyze big data effectively. Consequently, distributed systems like Hadoop Distributed File System (HDFS) and cloud-based solutions have become crucial. These systems distribute data across multiple nodes, ensuring fault tolerance and high availability. However, managing and maintaining distributed file systems can be complex and require expertise. Moreover, increasing storage capacity is associated with an increase in cost, which gives rise to data retention policies that determine appropriate data retention periods.

[0018] Organizations that address data retention manually, have static data retention policies, or update data retention policies manually expend large resources to define what data should be retained, for how long, and under what circumstances it should be deleted—even though the manual analysis of data usage and access patterns can be time-consuming and error-prone. Various aspects of the present disclosure provide methods and systems for dynamic data storage that employ predictive analytics to address many of these concerns.

[0019] The present disclosure provides a method 100 (FIG. 2) and system 50 (FIG. 1) for dynamically optimizing data retention policies in data storage systems. For brevity, the following description of the method 100 and system 50 focuses on HDFS storing transaction files, but it will be readily understood that the description associated with the method 100 and system 50 can be readily applied to any suitable data storage system and any suitable data files associated with any file type storable on a file storage system.

[0020] The system 50 includes a data processing module 51, a machine-learning module 52, and an optimizer module 54. The components of system 50 can be implemented in software, hardware, or a combination of hardware and software. FIGS. 5 and 6 depict a computer apparatus and a system including a host machine, respectively, which can be used separately, or in combination, to implement one or more components of the system 50, for example. While the system 50 comprises three components, it is readily understood that it can include more or less than the three described components. For example, the system 50 may only include the data processing module 51 and the machine-learning module 52,

but not the optimizer module 54, for example. Moreover, the configurations attributed to one module can be implemented, or partially implemented, by one of the other two components or additional components.

[0021] The system 50 interacts with one or more distributed file systems for optimizing data retention policies associated with such systems. FIG. 1 depicts a communication link 56 between the data processing module 51 and an HDFS 55. Additional communication links may connect the data processing module 51 to other distributed file systems. The data processing module 51 collects and pre-processes file data of log files stored in HDFS 55 for the purpose of dynamically optimizing its data retention policies. The machine-learning module 52 analyzes the file data associated with the log files on HDFS 55 to dynamically and autonomously make decisions about retention of the log files in HDFS 55.

[0022] The optimizer module 54 can also be connected to the one or more distributed file systems through a dedicated communication link or a common communication link with the data processing module 51. FIG. 1 depicts a communication link 57 interconnecting the optimizer module 54 with HDFS 55 to implement, or at least communicate, the optimization decisions. The optimizer module 54 monitors and adjusts the retention policies based on changing data patterns of the collected data files, as analyzed by the machine-learning module 52, ensuring efficient storage management over time.

[0023] The system 50 may further include a user interface 58. The user interface 58 may include a dashboard to visualize the historical data patterns and the retention policies associated with distributed file systems managed by the system 50.

[0024] Turning to FIG. 2, the method 100 is a computer-implemented method for optimizing data retention policies in distributed file systems. The method 100 can be implemented in whole, or in part, by the system 50. The method 100 includes collecting 101 file data of log files associated with a distributed file system, such as, for example, HDFS 55. The data processing module 51 may access and read file data stored on HDFS 55. The file data can include, for example, metadata associated with log files stored on HDFS 55. The file data may include, for example, file sizes and/or file creation data, such as, for example, creation month, creation day, creation hour, and/or any suitable creation time. In some aspects, the data processing module 51 utilizes the communication link 56 to access the data files stored on HDFS 55 and retrieves the data files for further processing by the system 50.

[0025] The method 100 further includes partitioning 102 the log files based on temporal

values in the file data to create temporal-based partitions. The data processing module 51 partitions the log files by creation year, creation month, and/or creation day, based on the retrieved file data. The data processing module 51 then determines the total size of the log files grouped into each partition. Accordingly, the data processing module 51 outputs a number of partitions characterized by temporal values associated with creation of the log files in HDFS 55.

[0026] The method 100 further includes identifying 103 historic patterns in the file data associated with the distributed file system. The output of the data processing module 51 constitutes an input into the machine-learning module 52. The log files are grouped by the machine-learning module 52 in various manners to identify historic patterns relevant to data retention in HDFS 55. The machine-learning module 52 applies a clustering algorithm, such as, for example, a K-means clustering algorithm to identify the historic patterns associated with the file data.

[0027] The historic patterns can indicate periods of high usage, low usage, and/or normal usage. In some aspects, the historic patterns are indicative of transitions such as, for example, transitions from high usage to low usage or from low usage to high usage. The machine-learning module 52 can also analyze historic retention policies and their impact on HDFS 55. Such information can be assigned different weights depending on the impact of such policies on the performance of HDFS 55, for example. Other factors can also be considered, such as, for example, access frequencies of the log files. In one example, the machine-learning module 52 analyzes historical (e.g., past four to five years) log generation rates to identify historic patterns indicative of past performance of HDFS 55.

[0028] FIG. 3 illustrates a method 200 for the identification of historic patterns in the file data, which can be performed by the machine-learning module 52. The method 200 includes grouping 201 the file data into groups characterized by temporal parameters representative of file-creation-time values, such as, for example, by file-creation month. In addition, the method 200 includes determining 202 average sizes associated with the groups, based on the collected file sizes of the log files in the groups, and it may further include identifying 203 the historic patterns based on the determined average sizes associated with the groups, and based on the temporal parameters associated with the groups.

[0029] In one example, identifying 103 the historic patterns comprises employing the K-means clustering algorithm to identify a plurality of clusters based on the determined average sizes associated with the groups, and based on the temporal parameters associated with the groups. Each cluster is assigned a retention time based on the identified

103 historic patterns. For example, a regular operation cluster can be associated with a log generation parameter (e.g., rate such as log files per day or total file size per day) L1, and it can be assigned T1 retention time. Additionally, a minor-system-update cluster can be associated with a log generation parameter (e.g., rate such as log files per day or total file size per day) L2, greater than L1, and it can be assigned T2 retention time, lesser than T1. Additionally, a major-system-update cluster can be associated with a log generation parameter (e.g., rate such as log files per day or total file size per day) L3, greater than L2, and it can be assigned T3 retention time, lesser than T2. Additionally, a peak-user-activity cluster can be associated with a log generation parameter (e.g., rate such as log files per day or total file size per day) L4, greater than L3, and it can be assigned T4 retention time, lesser than T3. It will be readily understood that the listed clusters are only for demonstrative purposes and that different, more, or less clusters can be identified.

[0030] Referring again to FIG. 2, the method 100 further includes optimizing 104 a data retention policy of the log files based on the historic patterns. In one example, optimizing 104 the data retention policy may include predicting a data retention policy for log files in a current temporal unit, such as, for example, a current month. As illustrated in FIG. 4, a method 300 predicts the data retention policy by grouping 301 log files in groups based on a common temporal parameter, for example, grouping the log files by day, and determining 302 a log generation parameter (e.g., log files per day or total size per day) for each of the groups. The method 300 further includes matching 303 each group to the cluster based on the log generation parameter to that of the group, for example, by matching each group to the cluster with the closest value of the log generation parameter to that of the group. The groups are then assigned 304 data retention periods based on their matching clusters.

[0031] In one example, the log generation parameters L1, L2, L3, L4 represent log generation rates (e.g., total log file size per day). Consequently, the new/current log files are grouped by day, and a total log file size per day is calculated for each group. Then, each group is matched to one of the plurality of clusters based on the log generation rate of the group and the log generation rates of the clusters. Each group is then assigned a retention policy based on the cluster assignment. For example, a first group with a log generation rate closest to L4 is matched to the peak-user-activity cluster and, accordingly, is assigned a T4 retention time.

[0032] Additionally, or alternatively, optimizing 104 the data retention policy, in accordance with the method 100, may include modifying previously determined data retention periods associated with older log files. For example, a log file group previously matched to one cluster can be switched to another cluster by the machine-learning module

52, resulting in a change in the group's retention period assignment. Accordingly, the system 50 dynamically manages the data retention policy associated with HDFS 55. Data retention policies associated with current/new log files are dynamically determined based on past behavior. Furthermore, data retention policies previously set for older log files can be dynamically modified to ensure an optimal operation of HDFS 55.

[0033] Depending on selected settings, the system 50 may automatically implement an optimization 104 of the data retention policy or may request a user input. FIG. 5 is a method 400 executable by the system 50 to implement an optimization of the data retention policy associated with HDFS 55. The method 400 includes predicting 401 a data retention policy for HDFS 55. If automatic adjustment is selected 402, the method 400 automatically adjusts 403 the data retention policy based on a matched cluster, for example. If, however, automatic adjustment is not selected 402, the method 400 issues 404 an alert to inform the administrators about the suggested change in the retention policy and awaits approval to implement.

[0034] The method 400 can be executed by the system 50. In one example, the optimizer module 54 issues 403 and alerts to the administrators via the user interface 58 about the suggested change in the retention policy. Additional, or alternative, alerts can be triggered by the optimizer module 54, such as, for example, alerts based on the total storage capacity of HDFS 55. As illustrated in FIG. 5, the method 400 may further include detecting 405 whether a storage threshold is triggered.

[0035] In one example, the storage threshold is triggered if the storage threshold is reached or exceeded, or will be reached or exceeded, based on the implementation of the suggested adjustment of the data retention policy. The storage threshold can be a predefined percentage of the total storage capacity of HDFS 55, for example.

[0036] If the storage threshold is triggered, and automatic adjustment is selected 406, the storage level of HDFS 55 is adjusted 407. The adjustment may include deleting older and/or non-essential data, for example. If, however, the automatic adjustment is not selected 406, an alert is issued 404 to inform the administrators about reaching or exceeding the storage threshold.

[0037] The user interface 58 may include a dashboard that displays the daily log file creation rate, the data policy prediction, current storage usage, and/or the applied retention policy (whether automatically adjusted or manually approved). This provides a visual representation of the storage situation and the implemented policies.

[0038] The optimizer module 54 may additionally solicit user feedback regarding a determined optimization of the data retention policy. The user input can be received through the user interface 58. The optimizer module 54 may transmit the user feedback to the machine-learning module 52 to update the historical data and retention policy, for example.

[0039] The aforementioned systems and methods, as described above with respect to each of FIGS. 1-4, may include, or make use of, a number of computer apparatuses, computer systems, or the like. In other words, in order to utilize the systems and methods disclosed herein, at least one of a computer apparatus, computer system, or the like may be implemented. Each of these computer apparatuses, computer systems, or the like are described in greater detail below with respect to the computer apparatus 3000 shown in FIG. 6 and the example system 4000 shown in FIG. 7, which provide a connection between the solution disclosed herein and how such a solution may be implemented within a business entity, such as a payment network, a processing network, a payment processing network, or the like.

[0040] FIG. 6 is a block diagram of a computer apparatus 3000 with data processing subsystems or components, according to at least one aspect of the present disclosure. The subsystems shown in FIG. 6 are interconnected via a system bus 3010. Additional subsystems such as a printer 3018, keyboard 3026, fixed disk 3028 (or other memory comprising computer-readable media), monitor 3022 (which is coupled to a display adapter 3020), and others are shown. Peripherals and input/output (I/O) devices, which couple to an I/O controller 3012 (which can be a processor or other suitable controller), can be connected to the computer system by any number of means known in the art, such as a serial port 3024. For example, the serial port 3024 or external interface 3030 can be used to connect the computer apparatus to a wide area network such as the Internet, a mouse input device, or a scanner. The interconnection via system bus 3010 allows the central processor 3016 to communicate with each subsystem and to control the execution of instructions from system memory 3014 or the fixed disk 3028, as well as the exchange of information between subsystems. The system memory 3014 and/or the fixed disk 3028 may embody a computer-readable medium.

[0041] FIG. 7 is a diagrammatic representation of an example system 4000 that includes a host machine 4002 within which a set of instructions to perform any one or more of the methodologies discussed herein may be executed, according to at least one aspect of the present disclosure. In various aspects, the host machine 4002 operates as a stand-alone device or may be connected (e.g., networked) to other machines. In a networked deployment, the host machine 4002 may operate in the capacity of a server or a client

machine in a server-client network environment, or as a peer machine in a peer-to-peer (or distributed) network environment. The host machine 4002 may be a computer or computing device, a personal computer (PC); a tablet PC; a set-top box; a personal digital assistant; a cellular telephone; a portable music player (e.g., a portable hard drive audio device, such as an Moving Picture Experts Group Audio Layer 3 (MP3) player); a web appliance; 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.

[0042] The example system 4000 includes the host machine 4002, running a host operating system (OS) 4004 on a processor or multiple processor(s)/processor core(s) 4006 (e.g., a central processing unit (CPU), a graphics processing unit, or both), and various memory nodes 4008. The host OS 4004 may include a hypervisor 4010, which is able to control the functions and/or communicate with a virtual machine (VM) 4012 running on machine-readable media. The VM 4012 also may include a virtual CPU or vCPU 4014. The memory nodes 4008 may be linked or pinned to virtual memory nodes or vNodes 4016. When the memory node 4008 is linked or pinned to a corresponding vNode 4016, then data may be mapped directly from the memory nodes 4008 to their corresponding vNodes 4016.

[0043] All the various components shown in host machine 4002 may be connected with and to each other or communicate to each other via a bus (not shown) or via other coupling or communication channels or mechanisms. The host machine 4002 may further include a video display, audio device, or other peripherals 4018 (e.g., a liquid crystal display; alpha-numeric input device(s) including, e.g., a keyboard; a cursor control device, e.g., a mouse; a voice recognition or biometric verification unit; an external drive; a signal generation device, e.g., a speaker); a persistent storage device 4020 (also referred to as disk drive unit); and a network interface device 4022. The host machine 4002 may further include a data encryption module (not shown) to encrypt data. The components provided in the host machine 4002 are those typically found in computer systems that may be suitable for use with aspects of the present disclosure and are intended to represent a broad category of such computer components that are known in the art. Thus, the system 4000 can be a server, minicomputer, mainframe computer, or any other computer system. The computer may also include different bus configurations, networked platforms, multi-processor platforms, and the like. Various OSs may be used, including UNIX, LINUX, WINDOWS, QNX ANDROID, IOS, CHROME, TIZEN, and other suitable OSs.

[0044] The disk drive unit 4024 also may be a solid-state drive, a hard disk drive, or

other drive that includes a computer or machine-readable medium on which is stored one or more sets of instructions and data structures (e.g., data/instructions 4026) embodying or utilizing any one or more of the methodologies or functions described herein. The data/instructions 4026 also may reside, completely or at least partially, within the main memory node 4008 and/or within the processor(s) 4006 during execution thereof by the host machine 4002. The data/instructions 4026 may further be transmitted or received over a network 4028 via the network interface device 4022 utilizing any one of several well-known transfer protocols (e.g., Hyper Text Transfer Protocol (HTTP)).

[0045] The processor(s) 4006 and memory nodes 4008 also may comprise machine-readable media. The term “computer-readable medium” or “machine-readable medium” should be taken to include a single medium or multiple medium (e.g., a centralized or distributed database and/or associated caches and servers) that store the one or more sets of instructions. The term “computer-readable medium” shall also be taken to include any medium that is capable of storing, encoding, or carrying a set of instructions for execution by the host machine 4002 and that causes the host machine 4002 to perform any one or more of the methodologies of the present application or that is capable of storing, encoding, or carrying data structures utilized by or associated with such a set of instructions. The term “computer-readable medium” shall accordingly be taken to include, but not be limited to, solid-state memories, optical and magnetic media, and carrier wave signals. Such media may also include, without limitation, hard disks, floppy disks, flash memory cards, digital video disks, random access memory (RAM), read-only memory (ROM), and the like. The example aspects described herein may be implemented in an operating environment comprising software installed on a computer, in hardware, or in a combination of software and hardware.

[0046] One skilled in the art will recognize that Internet service may be configured to provide Internet access to one or more computing devices that are coupled to the Internet service and that the computing devices may include one or more processors, buses, memory devices, display devices, I/O devices, and the like. Furthermore, those skilled in the art may appreciate that the Internet service may be coupled to one or more databases, repositories, servers, and the like, which may be utilized to implement any of the various aspects of the disclosure as described herein.

[0047] The computer program instructions also may be loaded onto a computer, a server, other programmable data processing apparatus, or other devices to cause a series of operational steps to be performed on the computer, other programmable apparatus, or other devices to produce a computer-implemented process such that the instructions that execute on the computer or other programmable apparatus provide processes for implementing the

functions/acts specified in the flowchart and/or block diagram block or blocks.

[0048] Suitable networks may include or interface with any one or more of, for instance, a local intranet; a personal area network (PAN); a local area network (LAN); a wide area network (WAN); a metropolitan area network (MAN); a virtual private network (VPN); a storage area network (SAN); a frame relay connection; an advanced intelligent network (AIN) connection; a synchronous optical network (SONET) connection; a digital T1, T3, E1, or E3 line; a digital data service (DDS) connection; a digital subscriber line (DSL) connection; an Ethernet connection; an integrated services digital network (ISDN) line; a dial-up port, such as a V.90, V.34, or V.34bis analog modem connection; a cable modem; an Asynchronous Transfer Mode (ATM) connection; or an Fiber Distributed Data Interface (FDDI) or Copper Distributed Data Interface (CDDI) connection. Furthermore, communications may also include links to any of a variety of wireless networks, including Wireless Application Protocol (WAP), General Packet Radio Service (GPRS), Global System for Mobile Communication (GSM), Code Division Multiple Access (CDMA) or Time Division Multiple Access (TDMA), cellular phone networks, global positioning system (GPS), cellular digital packet data (CDPD), Research in Motion, Limited (RIM) duplex paging network, Bluetooth radio, or an Institute of Electrical and Electronics Engineers (IEEE) 802.11-based radio frequency (RF) network. The network 4028 can further include or interface with any one or more of an RS-232 serial connection, an IEEE-1394 (Firewire) connection, a Fiber Channel connection, an IrDA (infrared (IR)) port, a Small Computer Systems Interface (SCSI) connection, a Universal Serial Bus(USB) connection or other wired or wireless, digital, or analog interface or connection, mesh, or Digi® networking.

[0049] In general, a cloud-based computing environment is a resource that typically combines the computational power of a large grouping of processors (such as within web servers) and/or that combines the storage capacity of a large grouping of computer memories or storage devices. Systems that provide cloud-based resources may be utilized exclusively by their owners or such systems may be accessible to outside users who deploy applications within the computing infrastructure to obtain the benefit of large computational or storage resources.

[0050] The cloud is formed, for example, by a network of web servers that comprise a plurality of computing devices, such as the host machine 4002, with each server 4030 (or at least a plurality thereof) providing processor and/or storage resources. These servers manage workloads provided by multiple users (e.g., cloud resource customers or other users). Typically, each user places workload demands upon the cloud that vary in real-time, sometimes dramatically. The nature and extent of these variations typically depends on the

type of business associated with the user.

[0051] It is noteworthy that any hardware platform suitable for performing the processing described herein is suitable for use with the technology. The terms “computer-readable storage medium” and “computer-readable storage media” as used herein refer to any medium or media that participate in providing instructions to a CPU for execution. Such media can take many forms, including, but not limited to, non-volatile media, volatile media, and transmission media. Non-volatile media include, for example, optical or magnetic disks, such as a fixed disk. Volatile media include dynamic memory, such as system RAM. Transmission media include coaxial cables, copper wire and fiber optics, among others, including the wires that comprise one aspect of a bus. Transmission media can also take the form of acoustic or light waves, such as those generated during RF and IR data communications. Common forms of computer-readable media include, for example, a flexible disk, a hard disk, magnetic tape, any other magnetic medium, a compact disc ROM (CD-ROM) disk, digital video disc, any other optical medium, any other physical medium with patterns of marks or holes, a RAM, a programmable ROM, an erasable programmable ROM (EPROM), an electrically erasable programmable ROM (EEPROM), a FLASH EPROM, any other memory chip or data exchange adapter, a carrier wave, or any other medium from which a computer can read.

[0052] Various forms of computer-readable media may be involved in carrying one or more sequences of one or more instructions to a CPU for execution. A bus carries the data to system RAM, from which a CPU retrieves and executes the instructions. The instructions received by system RAM can optionally be stored on a fixed disk either before or after execution by a CPU.

[0053] Computer program code for carrying out operations for aspects of the present technology may be written in any combination of one or more programming languages, including an object-oriented programming language such as Java, Smalltalk, C++, or the like and conventional procedural programming languages, such as the “C” programming language, Go, Python, or other programming languages, including assembly languages. The program code may execute entirely on the user’s computer, partly on the user’s computer, as a stand-alone software package, partly on the user’s computer and partly on a remote computer, or entirely on the remote computer or server. In the latter scenario, the remote computer may be connected to the user’s computer through any type of network, including a LAN or a WAN, or the connection may be made to an external computer (for example, through the Internet using an Internet service provider).

[0054] Examples of the systems and methods according to various aspects of the

present disclosure are provided below in the following numbered clauses. Any aspect of a system or method may include any one or more than one, and any combination of, the numbered clauses described below.

[0055] Clause 1. A computer-implemented method, comprising: collecting, by a data storage management system, file data of log files associated with a distributed file system, the file data comprising size values and temporal values associated with the log files; partitioning, by the data storage management system, the log files based on the temporal values to create temporal-based partitions; identifying, by the data storage management system, historic patterns in the file data; and optimizing, by the data storage management system, a data retention policy of the distributed file system based on the historic patterns.

[0056] Clause 2. The computer-implemented method of Clause 1, wherein the optimizing of the data retention policy comprises adjusting the data retention policy of a subset of the log files.

[0057] Clause 3. The computer-implemented method of anyone of Clauses 2-3, wherein collecting the file data of the log files comprises reading metadata of the log files on the distributed file system.

[0058] Clause 4. The computer-implemented method of anyone of Clauses 2-4, wherein the temporal values comprise file-generation-time values of the log files.

[0059] Clause 5. The computer-implemented method of Clause 4, wherein identifying the historic patterns comprises: grouping the file data into groups characterized by temporal parameters, wherein the grouping of the file data is based on the file-generation-time values; and determining average sizes associated with the groups, wherein the average sizes are based on the size values.

[0060] Clause 6. The computer-implemented method of Clause 5, wherein identifying the historic patterns are based on the average sizes and the temporal parameters.

[0061] Clause 7. The computer-implemented method of Clause 6, wherein identifying the historic patterns is further based on file-access frequencies of the log files.

[0062] Clause 8. The computer-implemented method of anyone of Clauses 2-5, wherein the grouping of the file data into the groups characterized by the temporal parameters comprises grouping the file data by month based on the file-generation-time values.

[0063] Clause 9. The computer-implemented method of anyone of Claims 2-8, wherein identifying the historic patterns further comprises applying a clustering algorithm to the file data to identify a plurality of clusters based on the file data.

[0064] Clause 10. A data storage management system for optimizing a data retention policy of a distributed file system, the data storage management system comprising: a data processing module to: retrieve file data of log files associated with the distributed file system, the file data comprising size values and temporal values associated with the log files; and partition the log files based on the temporal values to create temporal-based partitions; a predictive module to communicate with the data processing module, wherein the predictive module is to identify historic patterns in the file data; and an optimizer module to communicate with the predictive module, wherein the optimizer module is to optimize the data retention policy of the distributed file system based on the historic patterns.

[0065] Clause 11. The data storage management system of Claim 10, wherein the predictive module is to: group the file data into groups characterized by temporal parameters based on the temporal values; and determine average sizes associated with the groups, wherein the average sizes are based on the size values.

[0066] Clause 12. The data storage management system of Claim 11, wherein the predictive module is to identify the historic patterns based on the average sizes and the temporal parameters.

[0067] Clause 13. The data storage management system of Claim 12, wherein the predictive module is to identify the historic patterns further based on file-access frequencies of the log files.

[0068] Clause 14. The data storage management system of anyone of Claims 10-14, wherein the predictive module is to identify the historic patterns by applying a clustering algorithm to the file data to identify a plurality of clusters based on the file data.

[0069] Clause 15. The data storage management system of Claim 14, wherein the plurality of clusters comprises at least one of a regular-operation cluster, a minor-system-update cluster, a major-system-update, or a peak-user-activity cluster.

[0070] Clause 16. The data storage management system of Claim 14, wherein each of the plurality of clusters is assigned a log generation rate and a retention period.

[0071] Clause 17. The data storage management system of anyone of Claims 10-16, wherein the optimizer module is to optimize the data retention policy of the distributed file

system by predicting the data retention policy associated with a newer subset of the log files based on the historic patterns associated with an older subset of the log files.

[0072] Clause 18. The data storage management system of Clause 17, wherein the optimizer module is to: determine a log generation rate associated with the newer subset of the log files; assign the new file data to one of the plurality of clusters based on the log generation rate; and automatically adjust data retention of the newer subset of the log files based on the assigned one of the plurality of clusters.

[0073] Clause 19. The computer-implemented method of Clause 14, further comprising a user interface to communicate with the optimizer module, wherein the optimizer module is to output an alert signal to cause the user interface to issue an alert based on the assigned one of the plurality of clusters.

[0074] Clause 20. The computer-implemented method of Claim 19, wherein the user interface is to: receive a user input to adjust the data retention policy; and transmit an output signal to the optimizer module based on the user input, the output signal is to cause the optimizer module to adjust an optimization of the data retention policy, the optimization based on the historic patterns.

[0075] The foregoing detailed description has set forth various forms of the systems and/or processes via the use of block diagrams, flowcharts, and/or examples. Insofar as such block diagrams, flowcharts, and/or examples contain one or more functions and/or operations, it will be understood by those within the art that each function and/or operation within such block diagrams, flowcharts, and/or examples can be implemented, individually and/or collectively, by a wide range of hardware, software, firmware, or virtually any combination thereof. Those skilled in the art will recognize that some aspects of the forms disclosed herein, in whole or in part, can be equivalently implemented in integrated circuits as one or more computer programs running on one or more computers (e.g., as one or more programs running on one or more computer systems), as one or more programs running on one or more processors (e.g., as one or more programs running on one or more microprocessors), as firmware, or as virtually any combination thereof, and that designing the circuitry and/or writing the code for the software and or firmware would be well within the skill of one of skilled in the art in light of this disclosure. In addition, those skilled in the art will appreciate that the mechanisms of the subject matter described herein are capable of being distributed as one or more program products in a variety of forms, and an illustrative form of the subject matter described herein applies regardless of the particular type of signal-bearing medium used to actually carry out the distribution.

[0076] Instructions used to program logic to perform various disclosed aspects can be stored within a memory in the system, such as dynamic RAM, cache, flash memory, or other storage. Furthermore, the instructions can be distributed via a network or by way of other computer-readable media. Thus a machine-readable medium may include any mechanism for storing or transmitting information in a form readable by a machine (e.g., a computer), including, but not limited to, floppy diskettes, optical disks, CD-ROMs, magneto-optical disks, ROM, RAM, EPROM, EEPROM, magnetic or optical cards, flash memory, or a tangible, machine-readable storage used in the transmission of information over the Internet via electrical, optical, acoustical, or other forms of propagated signals (e.g., carrier waves, IR signals, digital signals). Accordingly, the non-transitory computer-readable medium includes any type of tangible machine-readable medium suitable for storing or transmitting electronic instructions or information in a form readable by a machine (e.g., a computer).

[0077] Any of the software components or functions described in this application may be implemented as software code to be executed by a processor using any suitable computer language, such as, for example, Python, Java, C++, or Perl, using, for example, conventional or object-oriented techniques. The software code may be stored as a series of instructions or commands on a computer-readable medium, such as RAM, ROM, a magnetic medium such as a hard drive or a floppy disk, or an optical medium such as a CD-ROM. Any such computer-readable medium may reside on or within a single computational apparatus and may be present on or within different computational apparatuses within a system or network.

[0078] As used in any aspect herein, the term “logic” may refer to an app, software, firmware, and/or circuitry configured to perform any of the aforementioned operations. Software may be embodied as a software package, code, instructions, instruction sets, and/or data recorded on a non-transitory computer-readable storage medium. Firmware may be embodied as code, instructions, instruction sets, and/or data that are hard-coded (e.g., non-volatile) in memory devices.

[0079] As used in any aspect herein, the terms “component,” “system,” “module,” and the like can refer to a computer-related entity, either hardware, a combination of hardware and software, software, or software in execution.

[0080] As used in any aspect herein, an “algorithm” refers to a self-consistent sequence of steps leading to a desired result, where a “step” refers to a manipulation of physical quantities and/or logic states that may, though need not necessarily, take the form of electrical or magnetic signals capable of being stored, transferred, combined, compared, and

otherwise manipulated. It is common usage to refer to these signals as bits, values, elements, symbols, characters, terms, numbers, or the like. These and similar terms may be associated with the appropriate physical quantities and are merely convenient labels applied to these quantities and/or states.

[0081] A network may include a packet-switched network. The communication devices may be capable of communicating with each other using a selected packet-switched network communications protocol. One example communications protocol may include an Ethernet communications protocol, which may be capable of permitting communication using a Transmission Control Protocol/Internet Protocol. The Ethernet protocol may comply or be compatible with the Ethernet standard published by the IEEE titled "IEEE 802.3 Standard," published in December 2008 and/or later versions of this standard. Alternatively or additionally, the communication devices may be capable of communicating with each other using an X.25 communications protocol. The X.25 communications protocol may comply or be compatible with a standard promulgated by the International Telecommunication Union-Telecommunication Standardization Sector. Alternatively or additionally, the communication devices may be capable of communicating with each other using a frame relay communications protocol. The frame relay communications protocol may comply or be compatible with a standard promulgated by Consultative Committee for International Telegraph and Telephone and/or the American National Standards Institute. Alternatively or additionally, the transceivers may be capable of communicating with each other using the ATM communications protocol. The ATM communications protocol may comply or be compatible with an ATM standard published by the ATM Forum titled "ATM-MPLS Network Interworking 2.0," published August 2001, and/or later versions of this standard. Of course, different and/or after-developed connection-oriented network communication protocols are equally contemplated herein.

[0082] Unless specifically stated otherwise as apparent from the foregoing disclosure, it is appreciated that, throughout the present disclosure, discussions using terms such as "processing," "computing," "calculating," "determining," "displaying," 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, registers, or other such information storage, transmission, or display devices.

[0083] One or more components may be referred to herein as "configured to," "configurable to," "operable/operative to," "adapted/adaptable," "able to,"

“conformable/conformed to,” etc. Those skilled in the art will recognize that “configured to” can generally encompass active-state components, inactive-state components, and/or standby-state components, unless context requires otherwise.

[0084] Those skilled in the art will recognize that, in general, terms used herein, and especially in the appended claims (e.g., bodies of the appended claims) are generally intended as “open” terms (e.g., the term “including” should be interpreted as “including, but not limited to”; the term “having” should be interpreted as “having at least”; the term “includes” should be interpreted as “includes, but is not limited to”). It will be further understood by those within the art that if a specific number of an introduced claim recitation is intended, such an intent will be explicitly recited in the claim, and in the absence of such recitation, no such intent is present. For example, as an aid to understanding, the following appended claims may contain usage of the introductory phrases “at least one” and “one or more” to introduce claim recitations. However, the use of such phrases should not be construed to imply that the introduction of a claim recitation by the indefinite articles “a” or “an” limits any particular claim containing such introduced claim recitation to claims containing only one such recitation, even when the same claim includes the introductory phrases “one or more” or “at least one” and indefinite articles such as “a” or “an” (e.g., “a” and/or “an” should typically be interpreted to mean “at least one” or “one or more”); the same holds true for the use of definite articles used to introduce claim recitations.

[0085] In addition, even if a specific number of an introduced claim recitation is explicitly recited, those skilled in the art will recognize that such recitation should typically be interpreted to mean at least the recited number (e.g., the bare recitation of “two recitations,” without other modifiers, typically means at least two recitations, or two or more recitations). Furthermore, in those instances where a convention analogous to “at least one of A, B, and C, etc.” is used, in general, such a construction is intended in the sense one having skill in the art would understand the convention (e.g., “a system having at least one of A, B, and C” would include, but not be limited to, systems that have A alone, B alone, C alone, A and B together, A and C together, B and C together, and/or A, B, and C together). In those instances where a convention analogous to “at least one of A, B, or C, etc.” is used, in general, such a construction is intended in the sense one having skill in the art would understand the convention (e.g., “a system having at least one of A, B, or C” would include, but not be limited to, systems that have A alone, B alone, C alone, A and B together, A and C together, B and C together, and/or A, B, and C together). It will be further understood by those skilled in the art that typically a disjunctive word and/or phrase presenting two or more alternative terms, whether in the description, claims, or drawings, should be understood to

contemplate the possibilities of including one of the terms, either of the terms, or both terms unless context dictates otherwise. For example, the phrase “A or B” will be typically understood to include the possibilities of “A,” “B,” or “A and B.”

[0086] With respect to the appended claims, those skilled in the art will appreciate that recited operations therein may generally be performed in any order. Also, although various operational flow diagrams are presented in a sequence(s), it should be understood that the various operations may be performed in other orders than those that are illustrated or may be performed concurrently. Examples of such alternate orderings may include overlapping, interleaved, interrupted, reordered, incremental, preparatory, supplemental, simultaneous, reverse, or other variant orderings, unless context dictates otherwise. Furthermore, terms like “responsive to,” “related to,” or other past-tense adjectives are generally not intended to exclude such variants, unless context dictates otherwise.

[0087] It is worthy to note that any reference to “one aspect,” “an aspect,” “an exemplification,” “one exemplification,” and the like means that a particular feature, structure, or characteristic described in connection with the aspect is included in at least one aspect. Thus, appearances of the phrases “in one aspect,” “in an aspect,” “in an exemplification,” and “in one exemplification” in various places throughout the specification are not necessarily all referring to the same aspect. Furthermore, the particular features, structures, or characteristics may be combined in any suitable manner in one or more aspects.

[0088] As used herein, the singular form of “a,” “an,” and “the” include the plural references unless the context clearly dictates otherwise.

[0089] Any patent application, patent, non-patent publication, or other disclosure material referred to in this specification and/or listed in any Application Data Sheet is incorporated by reference herein, to the extent that the incorporated materials is not inconsistent herewith. As such, and to the extent necessary, the disclosure as explicitly set forth herein supersedes any conflicting material incorporated herein by reference. Any material, or portion thereof, that is said to be incorporated by reference herein, but which conflicts with existing definitions, statements, or other disclosure material set forth herein, will only be incorporated to the extent that no conflict arises between that incorporated material and the existing disclosure material. None is admitted to be prior art.

[0090] In summary, numerous benefits have been described that result from employing the concepts described herein. The foregoing description of the one or more forms has been presented for purposes of illustration and description. It is not intended to be exhaustive or

limiting to the precise form disclosed. Modifications or variations are possible in light of the above teachings. The one or more forms were chosen and described in order to illustrate principles and practical application to thereby enable one of ordinary skill in the art to utilize the various forms with various modifications as are suited to the particular use contemplated. It is intended that the claims submitted herewith define the overall scope.

CLAIMS

What is claimed is:

1. A computer-implemented method, comprising:
 - collecting, by a data storage management system, file data of log files associated with a distributed file system, the file data comprising size values and temporal values associated with the log files;
 - partitioning, by the data storage management system, the log files based on the temporal values to create temporal-based partitions;
 - identifying, by the data storage management system, historic patterns in the file data;
 - and
 - optimizing, by the data storage management system, a data retention policy of the distributed file system based on the historic patterns.
2. The computer-implemented method of Claim 1, wherein the optimizing of the data retention policy comprises adjusting the data retention policy of a subset of the log files.
3. The computer-implemented method of Claim 2, wherein collecting the file data of the log files comprises reading metadata of the log files on the distributed file system.
4. The computer-implemented method of Claim 1, wherein the temporal values comprise file-generation-time values of the log files.
5. The computer-implemented method of Claim 4, wherein identifying the historic patterns comprises:
 - grouping the file data into groups characterized by temporal parameters, wherein the grouping of the file data is based on the file-generation-time values; and
 - determining average sizes associated with the groups, wherein the average sizes are based on the size values.
6. The computer-implemented method of Claim 5, wherein identifying the historic patterns are based on the average sizes and the temporal parameters.
7. The computer-implemented method of Claim 6, wherein identifying the historic patterns is further based on file-access frequencies of the log files.

8. The computer-implemented method of Claim 5, wherein the grouping of the file data into the groups characterized by the temporal parameters comprises grouping the file data by month based on the file-generation-time values.
9. The computer-implemented method of Claim 1, wherein identifying the historic patterns further comprises applying a clustering algorithm to the file data to identify a plurality of clusters based on the file data.
10. A data storage management system for optimizing a data retention policy of a distributed file system, the data storage management system comprising:
a data processing module to:
retrieve file data of log files associated with the distributed file system, the file data comprising size values and temporal values associated with the log files; and
partition the log files based on the temporal values to create temporal-based partitions;
a predictive module to communicate with the data processing module, wherein the predictive module is to identify historic patterns in the file data; and
an optimizer module to communicate with the predictive module, wherein the optimizer module is to optimize the data retention policy of the distributed file system based on the historic patterns.
11. The data storage management system of Claim 10, wherein the predictive module is to:
group the file data into groups characterized by temporal parameters based on the temporal values; and
determine average sizes associated with the groups, wherein the average sizes are based on the size values.
12. The data storage management system of Claim 11, wherein the predictive module is to identify the historic patterns based on the average sizes and the temporal parameters.
13. The data storage management system of Claim 12, wherein the predictive module is to identify the historic patterns further based on file-access frequencies of the log files.
14. The data storage management system of Claim 10, wherein the predictive module is to identify the historic patterns by applying a clustering algorithm to the file data to identify a plurality of clusters based on the file data.

15. The data storage management system of Claim 14, wherein the plurality of clusters comprises at least one of a regular-operation cluster, a minor-system-update cluster, a major-system-update, or a peak-user-activity cluster.
16. The data storage management system of Claim 14, wherein each of the plurality of clusters is assigned a log generation rate and a retention period.
17. The data storage management system of Claim 10, wherein the optimizer module is to optimize the data retention policy of the distributed file system by predicting the data retention policy associated with a newer subset of the log files based on the historic patterns associated with an older subset of the log files.
18. The data storage management system of Claim 17, wherein the optimizer module is to:
- determine a log generation rate associated with the newer subset of the log files;
 - assign the new file data to one of the plurality of clusters based on the log generation rate; and
 - automatically adjust data retention of the newer subset of the log files based on the assigned one of the plurality of clusters.
19. The data storage management system of Claim 14, further comprising a user interface to communicate with the optimizer module, wherein the optimizer module is to output an alert signal to cause the user interface to issue an alert based on the assigned one of the plurality of clusters.
20. The data storage management system of Claim 19, wherein the user interface is to:
- receive a user input to adjust the data retention policy; and
 - transmit an output signal to the optimizer module based on the user input, the output signal is to cause the optimizer module to adjust an optimization of the data retention policy, the optimization based on the historic patterns.