

METHOD AND APPARATUS FOR MANAGEING FILES, AND DEVICE AND STORAGE MEDIUM

TECHNICAL FIELD

[0001] Embodiments of the present disclosure relate to the technical field of computers, and particularly relate to a method and apparatus for managing files, and a device and a storage medium thereof.

RELATED ART

[0002] With the development of Internet of Things (IoT) technologies, IoT devices are widely applied in various industries, applications, and environments. IoT is a network that allows all common physical objects capable of performing independent functions to achieve interconnection based on an information carrier such as an internet and a traditional telecommunication network.

[0003] In the related art, in an IoT system, an edge device achieves data access and data persistence to multiple storage systems through an edge gateway, which stores and reads data according to the unique reading and writing methods of different storage systems.

[0004] In the related art, the access methods are different for different file systems, and the edge gateway needs to manage data according to the file management methods for different file systems. As a result, the process processing is much more complicated.

SUMMARY

[0005] The embodiments of the present disclosure provide method and apparatus for managing files, and a device and a storage medium thereof. The technical solutions are as follows.

[0006] According to an aspect of the embodiments of the present disclosure, a method for managing files is provided. The method is performed by an edge gateway, and the method includes:

[0007] acquiring a first operating parameter over a first hybrid shared interface, wherein the first hybrid shared interface is configured for implementation of executing a first operation in at least two file systems, and the first operating parameter comprises a system parameter and a

configuration parameter, wherein the system parameter indicates a file system in which the first operation is executed, and the configuration parameter describes an action of the first operation;

[0008] determining, based on the system parameter in the first operating parameter, a first file system in which the first operation is executed; and

[0009] executing the first operation in the first file system based on the configuration parameter in the first operating parameter.

[0010] According to an aspect of the embodiments of the present disclosure, an apparatus for managing files is provided. The apparatus includes:

[0011] a parameter acquiring module, configured to acquire a first operating parameter over a first hybrid shared interface, wherein the first hybrid shared interface is configured for implementation of executing a first operation in at least two file systems, and the first operating parameter comprises a system parameter and a configuration parameter, wherein the system parameter indicates a file system in which the first operation is executed, and the configuration parameter describes an action of the first operation;

[0012] a system determining module, configured to determine, based on the system parameter in the first operating parameter, a first file system in which the first operation is executed; and

[0013] an operation executing module, configured to execute the first operation in the first file system based on the configuration parameter in the first operating parameter.

[0014] According to an aspect of the embodiments of the present disclosure, a computer device is provided. The computer device includes a processor and a memory; wherein the memory stores one or more computer programs that, when loaded and run by the processor, cause the computer device to perform the method for managing files as described above.

[0015] According to an aspect of the embodiments of the present disclosure, a computer-readable storage medium is provided. The computer-readable storage medium stores one or more computer programs, wherein the one or more computer programs, when run by a processor, cause the processor to perform the method for managing files as described above.

[0016] According to an aspect of the embodiments of the present disclosure, a computer program product is provided. The computer program product includes one or more computer programs, wherein the one or more computer programs, when loaded and run by a processor, cause the processor to perform the method for managing files as described above.

[0017] The technical solutions according to the embodiments of the present disclosure achieve at least the following beneficial effects.

[0018] Files stored in different file systems are uniformly managed with smaller operating differences through a hybrid shared interface, which effectively reduces the complexity of processing files in different file systems.

BRIEF DESCRIPTION OF DRAWINGS

[0019] FIG. 1 is a schematic diagram of a solution implementation environment according to some embodiments of the present disclosure;

[0020] FIG. 2 is a flowchart of a method for managing files according to some embodiments of the present disclosure;

[0021] FIG. 3 is a schematic diagram of application of a first hybrid shared interface according to some embodiments of the present disclosure;

[0022] FIG. 4 is a schematic diagram of file tag management according to some embodiments of the present disclosure;

[0023] FIG. 5 is a bar graph of a dynamically adjusted thread number according to some embodiments of the present disclosure;

[0024] FIG. 6 is a block diagram of an apparatus for managing files according to some embodiments of the present disclosure; and

[0025] FIG. 7 is a structural block diagram of a computer device according to some embodiments of the present disclosure.

DETAILED DESCRIPTION

[0026] For clearer descriptions of the objectives, technical solutions, and advantages of the present disclosure, the embodiments of the present disclosure are further described in detail hereinafter with reference to the accompanying drawings.

[0027] FIG. 1 illustrates a schematic diagram of a solution implementation environment according to some embodiments of the present disclosure. The solution implementation environment is implemented as an IoT system. The solution implementation environment includes an edge gateway 10, an edge server 20, and at least one edge device 30.

[0028] The edge gateway 10 is an electronic device that connects the edge device 30 and the edge server 20 to provide support for data transmission, pre-processing, and management.

[0029] The edge server 20 is an electronic device that provides backend services for the edge device 30. Optionally, the backend services include, but are not limited to, file management, data processing and analysis, device management and control, network management, and the backend services may also include other backend services, which is not limited in the embodiments of the present disclosure.

[0030] The edge server 20 is a computing and storage device deployed at the edge of a network, and the edge server 20 is generally configured to perform more powerful and complex computing tasks to support functions such as local processing, big data analysis, application deployment, and storage management. For example, the edge server 20 is a standalone physical server, a server cluster or distributed system consisting of multiple physical servers, or a cloud server that provides cloud services, cloud databases, cloud computing, cloud functions, cloud storage, network services, cloud communications, middleware services, domain name services, security services, content delivery network (CDN), and basic cloud computing services such as big data and artificial intelligence platforms, which are not limited herein.

[0031] In some embodiments, the edge gateway 10 is configured to run a target application. The target application is configured to provide a file management function. In some embodiments, the target application may also run in the edge server 20.

[0032] The edge device 30 is an electronic device having a computing power, and storage and communication functions, and the edge device 30 is responsible for data collection, initial data processing, and analysis of part of data at the data source. For example, the edge device 30 includes, but is not limited to, a camera, a sensor, an industrial equipment, an industrial controller, a wearable device, a furniture appliance, an intelligent terminal device, an industrial robot, a medical device, a smart grid device, a wind power generation device, or a vehicle terminal device, and the edge device 30 may also include other edge devices, which is not limited in the embodiments of the present disclosure.

[0033] For example, data collected by the edge device 30 is uploaded, via the edge gateway 10, to the edge server 20, and the data is processed, analyzed, or stored by the edge server 20.

[0034] The edge gateway 10 and the edge server 20 communicate with each other over a network. Optionally, the edge device 30 communicates with the edge gateway 10 over a network so as to communicate with the edge server 20. The network is a wired or wireless network.

[0035] FIG. 2 illustrates a flowchart of a method for managing files according to some embodiments of the present disclosure. The steps of the method are executed by a computer device. For example, the computer device is the edge gateway 10 or edge server 20 in the solution implementation environment illustrated in FIG. 1. The method includes at least one of the following steps (210~230).

[0036] In step 210, a first operating parameter is acquired over a first hybrid shared interface, wherein the first hybrid shared interface is configured for implementation of executing a first operation in at least two file systems, and the first operating parameter includes a system parameter and a configuration parameter, wherein the system parameter indicates a file system in which the first operation is executed, and the configuration parameter describes an action of the first operation.

[0037] A file system is a core mechanism for storing and organizing files in an operating system, and a file system engine is a software component for managing and operating a file system. Optionally, the file system includes, but is not limited to, a linux file system, a seaweedfs file system, an Amazon S3 object file system, and a new technology file system (NTFS), and the file system may also include other file systems, which is not limited in embodiments of the present disclosure. In some embodiments, the functions of the file system include, but are not limited to, management and scheduling of stored resources, permission management, and file access, file creation, and the functions of the file system may also include other functions, which is not limited in the embodiments of the present disclosure.

[0038] In an IoT system, different edge devices may use different operating systems, resulting in compatible file systems being different. Therefore, it's necessary to use different file system engines to manage files in the IoT system.

[0039] The hybrid shared interface is an interface for implementing specified functions for the at least two file systems. The specified function is an operation of file access, file management, or security control on a file in the file system. In some embodiments, the specified function includes, but is not limited to, reading a file, writing into a file, modifying a file, deleting a file, creating a directory, deleting a directory, copying a directory, moving a directory, renaming, and authority management, and the specified function may include other functions, which are implemented by the relevant technical personnel in accordance with requirements. The embodiments of the present disclosure are not limited thereto. The target application is integrated with the file system engines of the at least two file systems described above, and the functions of the file system engines of two file systems are achieved by the hybrid shared interface.

[0040] In some embodiments, the target application includes at least one hybrid shared interface, and different operations on the file system can be implemented using different hybrid shared interfaces. That is, for the same operation, the operation can be executed in any one of the at least two file systems using the same hybrid shared interface. In this case, the first hybrid shared interface is any one of the at least one hybrid shared interface.

[0041] For example, in the case where the first hybrid shared interface is configured to implement the operation of reading a file, the operation of reading a specified file in the at least two file systems can be implemented using the first hybrid shared interface. It is understandable that the first operation is a file reading operation in this example.

[0042] In some embodiments, the target application only includes the first hybrid shared interface, and different operations on the file system are implemented using the first hybrid shared interface. In some embodiments, the configuration parameter in the first operating parameter further includes an operation identifier, and the operation identifier indicates a first operation. In this case, the first operation is any operation of at least one operation implemented by the first hybrid shared interface.

[0043] For example, the first hybrid shared interface is configured to implement a file reading operation, a file writing operation, a file deleting operation, and a file modifying operation, and the respective operation identifiers are an identifier 1, an identifier 2, an identifier 3, and an identifier 4. In the case where the operation identifier of the first operating parameter is the identifier 3, the first operation is a file deleting operation.

[0044] The first operating parameter is a parameter that is input when the first hybrid shared interface is called, and the first operating parameter is used to control the action of the first operation. In some embodiments, the first operating parameter indicates at least one of: an object on which the first operation is executed, an execution object for the first operation, an operation type of the first operation, or file modification content, and the first operating parameter may further indicate other information, which is not limited in the embodiments of the present disclosure.

[0045] The configuration parameter in the first operating parameter is a parameter related to the first operation, and the configuration parameter is used to control the manner of executing the first operation. In some embodiments, the configuration parameter may also include at least one of: a file path, a file identifier, a target path, access permission information, operating mode information, a metadata field, a file tag, or file modification content, and the configuration parameter may also include other information, as determined by the function of the first operation, which is not limited in the embodiments of the present disclosure.

[0046] The object on which the first operation is executed refers to the object to be processed by the first operation, which determines the scope of influence of the first operation. In some embodiments, the object on which the first operation is executed includes, but is not limited to, a file, a directory, a file system, a storage device, storage analysis, and metadata. The object on which the first operation is executed may be an object in another file system, which is not limited in the embodiments of the present disclosure. For example, the file path and/or the file identifier in the configuration parameter indicates the object on which the first operation is executed.

[0047] The operation type of the first operation refers to a specific operation mode or action specified by the first operation. For example, the operation identifier of the configuration parameter in the first operating parameter indicates the operation type of the first operation.

[0048] For example, FIG. 3 illustrates a schematic diagram of application of the first hybrid shared interface according to some embodiments of the present disclosure. The first hybrid shared interface is configured for implementation of executing the first operation in the three file systems as illustrated in FIG. 3. The three file systems are a first file system, a second file system, and a third file system. The edge device accesses files in the three file systems through the first hybrid shared interface.

[0049] In step 220, the first file system in which the first operation is executed is determined based on the system parameter in the first operating parameter.

[0050] The system parameter in the first operating parameter indicates the execution object for the first operation. The execution object for the first operation is an object in which the first operation is actually executed when the first hybrid shared interface is called. In the present disclosure, the execution object for the first operation is any one of the at least two file systems.

[0051] In step 230, the first operation is executed in the first file system based on the configuration parameter in the first operating parameter.

[0052] In some embodiments, the first operation may be executed through an operation instruction or an operation interface. The specific manner for executing the first operation is determined by the file operating system.

[0053] The operation instruction is an instruction for managing and operating files in a file system. That is, the first operation may be executed based on an operation instruction in a corresponding operation instruction set. In some embodiments, the target application includes respective operation instruction sets for the at least two file systems. The operation instruction set includes at least one operation instruction, and the operation instruction is used to execute

a specified operation in the corresponding file system. The operation instruction set for a file system is a set of instructions for managing and operating files in the file system.

[0054] In some embodiments, the first file system is determined based on the system parameter in the first operating parameter; a first operation instruction set is determined based on the first file system; and a first operation instruction for executing the first operation is determined from the at least one operation instruction included in the first operation instruction set based on the configuration parameter in the first operating parameter. It is understandable that the first operation instruction set is an operation instruction set for the first file system.

[0055] In some embodiments, there is a corresponding relationship between the respective operation instructions for the at least two file systems and the at least one hybrid shared interface. In some embodiments, operation instructions having the same or similar function for the at least two file systems correspond to the same hybrid shared interface.

[0056] In some embodiments, in the case where different hybrid shared interfaces correspond to different operations, the first operation instruction is determined from the at least one operation instruction included in the first operation instruction set based on the first hybrid shared interface and the system parameter in the first operating parameter. The first operation instruction is used for executing the first operation. That is, the operation instruction for the corresponding file system can be called based on the operating parameter input via the hybrid shared interface to execute the first operation.

[0057] For example, in the case where the first file system is a Linux file system, a corresponding operation instruction is called from a Linux operation instruction set based on a first operating parameter input via the first hybrid shared interface to execute the first operation.

[0058] For example, in the case where the first file system is an Amazon S3 file system, a corresponding operation instruction is called from an Amazon S3 software development kit (SDK) based on a first operating parameter input via the first hybrid shared interface to execute the first operation.

[0059] The operation interface is an interface for managing and operating files in a file system. That is, the first operation is executed by calling a corresponding operation interface. In some embodiments, the target application is integrated with at least one operation interface of each of the at least two file systems.

[0060] In some embodiments, the first file system is determined based on the system parameter in the first operating parameter; and a first operation interface corresponding to the

first operation in the first file system is called based on the configuration parameter in the first operating parameter to execute the first operation.

[0061] For example, in the case where the first file system is a SeaweedFS file system, an operation instruction corresponding to the first operation in the first file system is called based on a first operating parameter input via the first hybrid shared interface to execute the first operation.

[0062] In some embodiments, the first operation is a file query operation, and the configuration parameter in the first operating parameter includes a first file identifier and a first file path. The first file identifier indicates a first file, and the first file path indicates a location of the first file in the first file system. The first file is searched from the first file system based on the first file identifier and the first file path, and the first file is returned over the first hybrid shared interface.

[0063] The file identifier, which is also referred to as a file name, is used to distinguish between different files. The file path, which refers to the location of a file in a file system, describes how to locate a target file from a root directory or a current directory. Optionally, the file path may be an absolute path or a relative path, which is not limited in the embodiments of the present disclosure.

[0064] In some embodiments, the first operation is a file modifying operation, and the configuration parameter in the first operating parameter includes a first file identifier, a first file path, and first file modification content. The first file modification content indicates the content to be modified for the first file. The first file is searched from the first file system based on the first file identifier and the first file path, and the first file is modified based on the first file modification content.

[0065] The file modification content refers to the content to be modified for data or metadata of the file, which includes, but is not limited to, content addition, content deletion, content replacement, and content insertion. The file modification content may also include other content, which is not limited in the embodiments of the present disclosure.

[0066] It should be noted that the execution process of other operations in at least two file systems through the first hybrid shared interface is similar to the above, which is not repeated any further.

[0067] In summary, according to the technical solution provided in the embodiments of present disclosure, files stored in different file systems are uniformly managed with smaller operating differences through a hybrid shared interface, which effectively reduces the complexity of processing files in different file systems.

[0068] Content related to the management of file tags is introduced below.

[0069] In some embodiments, in the case where a first tag record in the file tag information satisfies a change condition, a first notification message is sent. The first notification message indicates a change in the first tag record, the first tag record indicates a first file tag of a second file, and the first file tag indicates an operating state of at least one edge device. The change condition is related to at least one of: a file path of the second file, a file type of the second file, a change type of the first tag record, or the first file tag.

[0070] A file tag indicates an operating state of an edge device. The operating state of the edge device refers to an operating condition of the edge device. In some embodiments, the file tag includes, but is not limited to, an operating state tag, a fatigue tag, and a wear tag, and the file tag may also include other text tags, which is not limited in the embodiments of the present disclosure. The operating state tag indicates the operating state of the edge device, and the operating state of the edge device refers to information related to the operating environment, state, load and the like of the edge device during operation. The fatigue tag indicates the fatigue state of the edge device, and the fatigue state of the edge device refers to the state of the edge device after a long time of operation. The wear tag indicates the wear state of the edge device, and the wear state of the edge device refers to the state of functional decrease, performance degradation, or physical wear and tear caused by long-term use, external environmental factors, workload, and the like.

[0071] The second file is any file stored in any one of the at least two file systems. The change condition refers to the condition that triggers the sending of the first notification message when the file tag is changed.

[0072] The file tag information is used for storing the tag records of file tags of respective files in various file systems. The tag record of the file indicates the file tag of the edge device corresponding to the file.

[0073] The file type is used for distinguishing between different file contents, formats, or purposes. In some embodiments, the file type of a file is stored in the metadata of the file. Optionally, the file type is predefined by relevant technical personnel in accordance with requirements. For example, in the field of wind power, the file type includes, but is not limited to, a time-domain type, a frequency-domain type, a normal type, and the like.

[0074] In some embodiments, the change condition includes a change in second tag information, and the second file is a file indicated by change configuration information. In some embodiments, the change configuration information is at least one of: a specified file

path, a specified file type, a specified change type, or a specified tag type, which is set by relevant technical personnel in accordance with requirements, and is not limited in the embodiments of the present disclosure.

[0075] For example, in the case where the second file is a file having a specified file path indicated by the change configuration information, the first notification message is sent when the first tag record of the second file is changed.

[0076] For example, in the case where the second file is a file having a specified file type indicated by the change configuration information, the first notification message is sent when the first tag record of the second file is changed.

[0077] The first notification message is used to prompt that a file tag of a second file type is changed. In some embodiments, the first notification message is triggered by a message queue (MQ) message-oriented middleware.

[0078] In some embodiments, the file tag information includes at least one tag record, and the tag record includes a file identifier, a file path, and a tag key-value pair. The tag key-value pair includes a file tag identifier and tag information, the tag identifier indicates a file tag, and the tag information is content of a file tag indicated by the tag identifier.

[0079] In some embodiments, the file tag information is implemented in the form of a relational database. In some embodiments, at least one tag key-value pair is stored in the file tag information in the form of column storage of a relational database. In some embodiments, at least one tag key-value pair is indexed taking the file identifier and the file path as an association condition.

[0080] In some embodiments, a second tag record is acquired by searching for a tag record matching the second file identifier and the second file path from the at least one tag record included in the file tag information; and an edit operation is executed on the tag key-value pair in the second tag record; wherein the edit operation includes at least one of: a new add operation, a delete operation, a modify operation, or a query operation.

[0081] In some embodiments, in the case where the operating state of the edge device corresponding to the second file is changed, the first tag record of the second file is updated through an edit operation to acquire a changed first tag record; and in the case where the changed first tag record meets the change condition, the first notification message is sent.

[0082] For example, FIG. 4 illustrates a schematic diagram of file tag management according to some embodiments of the present disclosure. As illustrated in FIG. 4, the file tag includes an operating state tag, a fatigue tag, and a wear tag. In the case where the operating

state tag of the second file satisfies the change condition, the first notification message is sent to the corresponding edge device.

[0083] According to the above method, at least one file tag is set for the file, and the tags for different file systems are uniformly managed. When a file tag satisfies the change condition, a tag change notification message is triggered, and the corresponding user can quickly obtain the notification of change in the file tag of the edge device as needed.

[0084] Content related to the load management of the edge gateway is introduced below.

[0085] In some embodiments, first load information of the edge gateway is acquired, wherein the first load information indicates resource usage of the edge gateway; and the number of parallel threads is adjusted to a first thread number corresponding to the first load information based on thread configuration information, wherein the thread configuration information includes at least one group of load information and thread number that have a corresponding relationship.

[0086] The first load information refers to the resource usage of the edge gateway. In some embodiments, in the case where the method for managing files is performed by an edge server, the first load information refers to the resource usage of the edge server. In some embodiments, the first load information includes, but is not limited to, a central processing unit (CPU) utilization rate, the number of processes, an average load, a memory utilization rate, an available memory, a disk utilization rate, a read/write speed, a network bandwidth utilization rate, and a network latency, and the first load information may also include other load information, which is not limited in the embodiments of the present disclosure.

[0087] The thread configuration information refers to the pre-configured number of threads that are allowed to run in parallel under a specified load. The at least one group of load information and thread number that have a corresponding relationship included in the thread configuration information is preset by relevant technical personnel, which is not limited in the embodiments of the present disclosure.

[0088] The number of parallel threads refers to the number of threads that are currently used in parallel by the device in the edge gateway.

[0089] In some embodiments, in the case where the first load information satisfies a first condition, a first number of threads is added to the number of parallel threads to acquire an adjusted number of parallel threads; and in the case where the first load information satisfies a second condition, the number of parallel threads is halved to acquire an adjusted number of parallel threads.

[0090] In some embodiments, the first condition refers to that the first load information does not exceed a first load threshold and the first load information is increased by a first numerical value relative to the first load information when the thread number is adjusted last time. In some embodiments, the first load threshold, the first numerical value, and the first number are preset by relevant technical personnel.

[0091] For example, the first condition may be that the first load information is less than or equal to 50% and the first load information is increased by 5% relative to the first load information when the thread number is adjusted last time. Each time the first load information satisfies the first condition, one thread is added to the current parallel threads.

[0092] In some embodiments, the second condition refers to that the first load information exceeds a second load threshold.

[0093] For example, assuming that the number of the current parallel threads is 10, then the second condition is that the first load information is greater than or equal to 80%. When the first load information satisfies the second condition, the number of current parallel threads is halved, i.e., reduced to 5.

[0094] It should be noted that in the case where the first load information does not satisfy either the first condition or the second condition, the number of parallel threads is not adjusted.

[0095] For example, FIG. 5 illustrates a bar graph of a dynamically adjusted thread number according to some embodiments of the present disclosure. Taking an example where the first load information is presented in a percentage, from the first load information being 0 and the number of parallel threads being 0, in the case where the first load information is less than or equal to 50%, one parallel thread is added whenever the first load information is increased by 5%; and in the case where the first load information is greater than or equal to 80%, the number of parallel threads is halved. The number of parallel threads changes with the system load as follows.

[0096] When the first load information is equal to 5%, 1 thread is executed in parallel.

[0097] When the first load information is equal to 10%, 2 threads are executed in parallel.

[0098] When the first load information is equal to 15%, 3 threads are executed in parallel.

[0099] When the first load information is equal to 20%, 4 threads are executed in parallel.

[0100] When the first load information is equal to 25%, 5 threads are executed in parallel.

[0101] When the first load information is equal to 30%, 6 threads are executed in parallel.

[0102] When the first load information is equal to 35%, 7 threads are executed in parallel.

[0103] When the first load information is equal to 40%, 8 threads are executed in parallel.

[0104] When the first load information is equal to 45%, 9 threads are executed in parallel.

[0105] When the first load information is equal to 50%, 10 threads are executed in parallel.

[0106] When the first load information is equal to 80%, 5 threads are executed in parallel.

[0107] According to the above method, the parallelism of threads used for file upload and download is adaptively adjusted based on the load condition of the edge gateway. In an initial state, the concurrency level is gradually increased based on the load condition, once the load reaches a stable state, a certain concurrency level is maintained, and in a high-load state, the concurrency level is halved to quickly reduce the concurrency consumption. According to this method, the concurrency of threads is dynamically adjusted based on the load condition, thereby improving the resource utilization efficiency and improving the performance and throughput of the edge gateway.

[0108] Content related to file invalidation management is introduced below.

[0109] In some embodiments, it is periodically checked whether at least one file included in each of at least two file systems is valid or not based on a period threshold; and files included in the at least two file systems are deleted based on the check result for the at least one file included in each of the at least two file systems.

[0110] That is, files stored in each file system are regularly traversed to determine the validity of each file, and invalid files are deleted according to a deletion strategy.

[0111] In some embodiments, each file corresponds to a validity period. The validity period of the file refers to the time during which the file is allowed to be continuously stored in the corresponding file system. Optionally, a validity period is set for the at least one file included in each of the at least two file systems.

[0112] In some embodiments, in the case where the time interval between a first moment and a second moment is equal to the period threshold, the metadata of the at least one file stored in a second file system is traversed; and whether the at least one file is valid or not is determined based on the creation time and the validity period of the metadata of the at least one file.

[0113] The metadata of the file refers to a set of data used for describing the attribute of the file. In some embodiments, the attribute included in the metadata includes, but is not limited to, a file identifier, a file path, a creation time, a modification time, a metadata modification time, a file permission, a creation user identifier, a file size, and a file type, and the attribute may also include other attribute, which is not limited in the embodiments of the present disclosure.

[0114] It should be noted that the second moment may be the current moment, and the first moment is the moment when it is checked whether the file is invalid or not last time.

[0115] In the present disclosure, when it is checked whether a file is valid or not, files in different file systems are scanned according to the same deletion strategy, without the need to configure different deletion strategies for different file systems.

[0116] In some embodiments, when the time interval between the first moment and the second moment is equal to the period threshold, the metadata of the at least one file stored in the second file system is traversed. The metadata of each file is information related to the file, and the metadata includes the creation time of the file. The period threshold refers to the time interval at which it is checked whether a file is invalid or not. The validity period matching a third file in the at least one file is searched from the validity period configuration information to acquire a first validity period. The first validity period is a duration of validity of the third file stored in the second file system, and the validity period configuration information includes at least one group of file scope information and validity period. The file scope information indicates a location of the file configured with the validity period in the at least two file systems. The third file is deleted from the second file system in the case where a first storage duration is greater than the first validity period, and the first storage duration is determined based on the creation time of the third file. The first moment is earlier than the second moment, and the first moment is a moment of checking whether the file is invalid or not most recently.

[0117] The validity period configuration information indicates the validity periods of files stored in the at least two file systems. The file scope information indicates the effect scope of the validity period corresponding to the file scope information in the at least two file systems. In some embodiments, the file scope information includes at least one of: a system identifier, a file directory, a file path, a file name, a file type, etc. For example, in the case where the file scope information is a file path, each group of file scope information and validity period refers to setting the validity periods of all files under the path indicated by the file scope information to the validity period corresponding to the file scope information. The unit of the validity period may be any of: a minute, a day, a month, a year, or the like.

[0118] The first storage duration refers to the duration when the third file is continuously stored in the second file system. In some embodiments, the validity period configuration information may be implemented as a relational database.

[0119] For example, the validity period configuration information is implemented as shown in Table 1 below. Each row in Table 1 represents a group of file scope information and

validity period, and the file scope information includes a file path and a file system. Taking the first row in Table 1 as an example, when the metadata corresponding to the file path of /path/to/path1 in the first file system is scanned, the validity of each file under the file path is judged based on the file creation time in the metadata and the validity period shown in Table 2. In the case where a file is invalid, the file is deleted from the first file system.

Table 1

File Path	File System	Validity Period (days)
/path/to/path1	First file system	60
/path/to/path2	First file system	180
/path3/	Second file system	90
/path4/	Second file system	180
/path5/	Third file system	365
/path6/	Third file system	90

[0120] According to the above method, different files are configured with different validity periods based on needs, and the validity of each file stored in each file system is judged based on the creation time and validity period of the file. In this way, the workload of manually deleting invalid files, and the space utilization ratio of each file system is improved.

[0121] The following is an apparatus embodiment of the present disclosure, which may be used to implement the method embodiment of the present disclosure. For details not disclosed in the apparatus embodiment of the present disclosure, please refer to the method embodiment of the present disclosure.

[0122] FIG. 6 illustrates a block diagram of an apparatus for managing files according to some embodiments of the present disclosure. The apparatus has the function of implementing the above examples, and the function may be implemented by hardware or by software executed by corresponding hardware. The apparatus may be the computer device described above or arranged in the computer device. As illustrated in FIG. 6, the apparatus 600 includes a parameter acquiring module 610, a system determining module 620, and an operation executing module 630.

[0123] The parameter acquiring module 610 is configured to acquire a first operating parameter over a first hybrid shared interface, wherein the first hybrid shared interface is configured for implementation of executing a first operation in at least two file systems, and the first operating parameter includes a system parameter and a configuration parameter,

wherein the system parameter indicates a file system in which the first operation is executed, and the configuration parameter describes an action of the first operation.

[0124] The system determining module 620 is configured to determine, based on the system parameter in the first operating parameter, a first file system in which the first operation is executed.

[0125] The operation executing module 630 is configured to execute the first operation in the first file system based on the configuration parameter in the first operating parameter.

[0126] In some embodiments, the first operation is a file query operation, and the configuration parameter in the first operating parameter includes a first file identifier and a first file path. The first file identifier indicates a first file, and the first file path indicates a location of the first file in the first file system. The operation executing module 630 is configured to search for the first file from the first file system based on the first file identifier and the first file path, and return the first file over the first hybrid shared interface.

[0127] In some embodiments, the apparatus 600 further includes a message sending module (not shown in FIG. 6), and the message sending module is configured to send a first notification message in the case where a first tag record in the file tag information satisfies a change condition. The first notification message indicates a change in the first tag record, the first tag record indicates a first file tag of a second file, and the first file tag indicates an operating state of an edge device corresponding to the second file. The change condition is related to at least one of: a file path of the second file, a file type of the second file, a change type of the first tag record, or the first file tag.

[0128] In some embodiments, the file tag information includes at least one tag record, and the tag record includes a file identifier, a file path, and a tag key-value pair. The tag key-value pair includes a file tag identifier and tag information, the file tag identifier indicates a file tag, and the tag information is content of a file tag indicated by the file tag identifier. The apparatus 600 further includes a tag editing module (not shown in FIG. 6), and the tag editing module is configured to: acquire a second tag record by searching for a tag record matching the second file identifier and the second file path from the at least one tag record included in the file tag information; and execute an edit operation on the tag key-value pair in the second tag record; wherein the edit operation includes at least one of: a new add operation, a delete operation, a modify operation, or a query operation.

[0129] In some embodiments, the apparatus 600 further includes a load adjusting module (not shown in FIG. 6), and the load adjusting module is configured to: acquire first load information of the edge gateway, wherein the first load information indicates resource usage

of the edge gateway; and adjust the number of parallel threads to a first thread number corresponding to the first load information based on thread configuration information, wherein the thread configuration information includes at least one group of load information and thread number that have a corresponding relationship.

[0130] In some embodiments, the apparatus 600 further includes a file detecting module (not shown in FIG. 6), and the file detecting module is configured to: traverse metadata of at least one file stored in a second file system in a case where a time interval between a first moment and a second moment is equal to a period threshold, wherein metadata of each file is information related to the file, the metadata includes a creation time of the file, and the period threshold is a time interval at which it is checked whether a file is invalid or not; acquire a first validity period by searching for an invalidity period that matches a third file in the at least one file from validity period configuration information, wherein the first validity period is a duration of validity of the third file stored in the second file system, and the validity period configuration information includes at least one group of file scope information and validity period, wherein the file scope information indicates a location of a file configured with the validity period in the at least two file systems; and delete the third file from the second file system in a case where a first storage duration is greater than the first validity period, wherein the first storage duration is determined based on a creation time of the third file; wherein the first moment is earlier than the second moment, and the first moment is a moment of checking whether a file is invalid or not most recently.

[0131] In summary, according to the technical solution provided in the embodiments of present disclosure, files stored in different file systems are uniformly managed with smaller operating differences through a hybrid shared interface, which effectively reduces the complexity of processing files in different file systems.

[0132] It should be noted that in the case where the apparatus according to the above embodiments implements its functions, the division of the functional modules is only for illustrative purposes. In practice, the functions may be distributed to different functional modules based on actual needs, that is, the internal structure of the apparatus may be divided into different functional modules to complete all or part of the functions described above. In addition, the apparatus embodiments share the same concept as the method embodiments, and for details of the implementation process, please refer to the method embodiments, which are not repeated any further.

[0133] FIG. 7 illustrates a structural block diagram of a computer device according to some embodiments of the present disclosure. The computer device 700 may be the edge gateway 10 or the edge server 20 in the implementation environment shown in FIG. 1, and the computer device 700 is configured to implement the method for managing files provided in the above embodiments.

[0134] Generally, the computer device 700 includes a processor 710 and a memory 720.

[0135] The processor 710 includes one or more processing cores, such as a 4-core processor, an 8-core processor, and the like. The processor 710 may be implemented in at least one of the following hardware: a digital signal processing (DSP), a field programmable gate array (FPGA), a programmable logic array (PLA), or a logic array (PLA). The processor 710 may further include a main processor and a coprocessor, the main processor is a processor configured to process data in a wake-up state, which is also referred to as a central processing unit (CPU), and the coprocessor is a low power consumption processor configured to process data in a standby state. In some embodiments, the processor 710 is integrated with a graphics processor processing unit (GPU), which is responsible for rendering and drawing content to be displayed on a display screen. In some embodiments, the processor 710 further includes an AI processor, which is used to perform computational operations related to machine learning.

[0136] The memory 720 includes one or more computer-readable storage media, which may be non-transitory. The memory 720 may also include a high-speed random access memory and a non-volatile memory, such as one or more disk storage devices or flash storage devices. In some embodiments, the non-transitory computer-readable storage medium in the memory 720 is used to store one or more computer programs that are configured to be executed by one or more processors to implement the method for managing files.

[0137] Those skilled in the art will appreciate that the structure shown in FIG. 7 does not constitute a limitation on the computer device 700, and the computer device 700 may include more or fewer components than shown, or certain components may be combined, or different component arrangements may be adopted.

[0138] In some embodiments, a computer-readable storage medium is provided. The computer-readable storage medium stores one or more computer programs that are configured to be executed by a processor to implement the above method for managing files. The computer-readable storage medium includes a read-only memory (ROM), a random-access memory (RAM), a solid state drive (SSD), an optical disk, or the like. The RAM includes a resistance random access memory (ReRAM) and a dynamic random access memory (DRAM).

[0139] In some embodiments, a computer program product is provided. The computer program product includes one or more computer programs stored in a computer-readable storage medium. The one or more computer programs, when read and executed by a processor of a computer device from the computer-readable storage medium, cause the computer device to implement the above method for managing files.

[0140] It should be noted that the collection and processing of relevant data in the present disclosure should strictly comply with the requirements of relevant national laws and regulations when applied in practice. The informed consent or separate consent of the individual information subject should be obtained, and subsequent data use and processing should be carried out within the scope of laws and regulations and the authorization of the individual information subject.

[0141] It is understandable that the mentioned term "a plurality of" herein means two or more. The term "and/or" describes the association relationship between the associated objects, and indicates that three relationships may be present. For example, the phrase "A and/or B" means (A), (B), or (A and B). The symbol "/" generally indicates an "or" relationship between the associated objects. In addition, serial numbers of the steps described herein only show an exemplary possible sequence of performing the steps. In some other embodiments, the steps may also be performed out of the numbering sequence, for example, two steps with different serial numbers are performed simultaneously, or two steps with different serial numbers are performed in a reverse order to the illustrated sequence, which is not limited in the present disclosure.

[0142] Described above are merely optional embodiments of the present disclosure and are not intended to limit the present disclosure. Any modifications, equivalent substitutions, improvements, and the like, made within the spirit and principle of the present disclosure should fall within the protection scope of the present disclosure.

CLAIMS

1. A method for managing files, performed by an edge gateway, the method comprising:
 - acquiring a first operating parameter over a first hybrid shared interface, wherein the first hybrid shared interface is configured for implementation of executing a first operation in at least two file systems, and the first operating parameter comprises a system parameter and a configuration parameter, wherein the system parameter indicates a file system in which the first operation is executed, and the configuration parameter describes an action of the first operation;
 - determining, based on the system parameter in the first operating parameter, a first file system in which the first operation is executed; and
 - executing the first operation in the first file system based on the configuration parameter in the first operating parameter.
2. The method according to claim 1, wherein the first operation is a file query operation, and the configuration parameter in the first operating parameter comprises a first file identifier and a first file path, wherein the first file identifier indicates a first file, and the first file path indicates a location of the first file in the first file system;
 - executing the first operation in the first file system based on the configuration parameter in the first operating parameter comprises:
 - searching for the first file from the first file system based on the first file identifier and the first file path; and
 - returning the first file over the first hybrid shared interface.
3. The method according to claim 1, further comprising:
 - sending a first notification message in a case where a first tag record in file tag information satisfies a change condition, wherein the first notification message indicates a change in the first tag record, and the first tag record indicates a first file tag of a second file, wherein the first file tag indicates an operating state of an edge device corresponding to the second file;
 - wherein the change condition is related to at least one of: a file path of the second file, a file type of the second file, a change type of the first tag record, or the first file tag.
4. The method according to claim 3, wherein the file tag information comprises at least one tag record, and the tag record comprises a file identifier, a file path, and a tag key-value pair, wherein the tag key-value pair comprises a file tag identifier and tag information, the file tag identifier indicating a file tag, and the tag information being content of a file tag indicated by the file tag identifier; the method further comprising:

acquiring a second tag record by searching for a tag record matching a second file identifier and a second file path in the at least one tag record comprised in the file tag information; and
executing an edit operation on a tag key-value pair in the second tag record;
wherein the edit operation comprises at least one of: a new add operation, a delete operation, a modify operation, or a query operation.

5. The method according to claim 1, further comprising:

acquiring first load information of the edge gateway, wherein the first load information indicates resource usage of the edge gateway; and

adjusting, based on thread configuration information, the number of parallel threads to a first thread number corresponding to the first load information, wherein the thread configuration information comprises at least one group of load information and thread number that have a corresponding relationship.

6. The method according to claim 1, further comprising:

traversing metadata of at least one file stored in a second file system in a case where a time interval between a first moment and a second moment is equal to a period threshold, wherein metadata of each file is information related to the file, the metadata comprises a creation time of the file, and the period threshold is a time interval at which it is checked whether a file is invalid or not;

acquiring a first validity period by searching for an invalidity period that matches a third file in the at least one file from validity period configuration information, wherein the first validity period is a duration of validity of the third file stored in the second file system, and the validity period configuration information comprises at least one group of file scope information and validity period, wherein the file scope information indicates a location of a file configured with the validity period in the at least two file systems; and

deleting the third file from the second file system in a case where a first storage duration is greater than the first validity period, wherein the first storage duration is determined based on a creation time of the third file;

wherein the first moment is earlier than the second moment, and the first moment is a moment of checking whether a file is invalid or not most recently.

7. An apparatus for managing files, comprising:

a parameter acquiring module, configured to acquire a first operating parameter over a first hybrid shared interface, wherein the first hybrid shared interface is configured for

implementation of executing a first operation in at least two file systems, and the first operating parameter comprises a system parameter and a configuration parameter, wherein the system parameter indicates a file system in which the first operation is executed, and the configuration parameter describes an action of the first operation;

a system determining module, configured to determine, based on the system parameter in the first operating parameter, a first file system in which the first operation is executed; and

an operation executing module, configured to execute the first operation in the first file system based on the configuration parameter in the first operating parameter.

8. A computer device, comprising: a processor and a memory; wherein the memory stores one or more computer programs that, when loaded and run by the processor, cause the computer device to perform the method as defined in any one of claims 1 to 6.

9. A computer-readable storage medium storing one or more computer programs, wherein the one or more computer programs, when run by a processor, cause the processor to perform the method as defined in any one of claims 1 to 6.

10. A computer program product, comprising: one or more computer programs, wherein the one or more computer programs, when loaded and run by a processor, cause the processor to perform the method as defined in any one of claims 1 to 6.