(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
12 May 2016 (12.05.2016)

WIPO | PCT

(10) International Publication Number
WO 2016/073029 A1

(51) International Patent Classification:
G06F 17/30 (2006.01) *G06F 11/07* (2006.01)

(21) International Application Number:
PCT/US2015/027419

(22) International Filing Date:
24 April 2015 (24.04.2015)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
5500/CHE/2014 3 November 2014 (03.11.2014) IN

(71) Applicant: **HEWLETT PACKARD ENTERPRISE DEVELOPMENT LP** [US/US]; 11445 Compaq Center Drive West, Houston, TX 77070 (US).

(72) Inventors: **DOVAL, Shruti**; Sy. No.192, Whitefield Road, Mahadevapura Post, Bangalore, Karnataka 560048 (IN). **KAMALAKSHA, Venkataraman**; Sy. No.192, Whitefield Road, Mahadevapura Post, Bangalore, Karnataka 560048 (IN). **BALAKRISHNA, Rohit**; Sy. No.192, Whitefield Road, Mahadevapura Post, Bangalore, Karnataka 560048 (IN). **VERMA, Rajat**; Sy. No.192, Whitefield Road, Mahadevapura Post, Bangalore, Karnataka 560048 (IN).

(74) Agents: **ORTEGA, Arthur** et al.; Hewlett Packard Enterprise, 3404 E. Harmony Road, Mail Stop 79, Fort Collins, CO 80528 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

— *as to the identity of the inventor (Rule 4.17(i))*

Published:

— *with international search report (Art. 21(3))*

(54) Title: DETECTING INCONSISTENCIES IN HIERARCHICAL ORGANIZATION DIRECTORIES

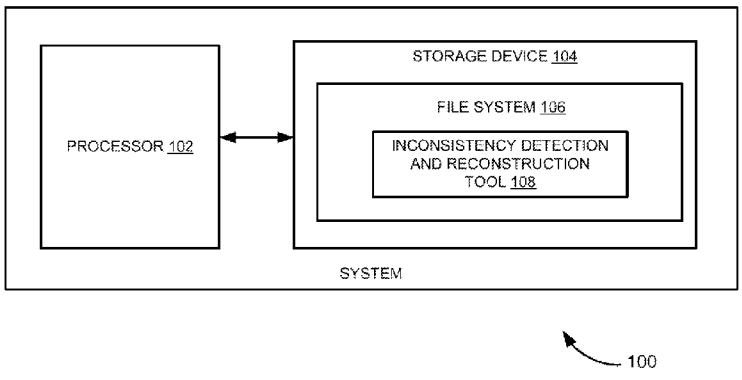


FIG. 1

(57) Abstract: In one example, a system is described in which a storage device, communicatively coupled to a processor, includes an inconsistency detection and reconstruction tool. The inconsistency detection and reconstruction tool may create a database including metadata associated with hierarchical organization directories and policy and configuration data files in a file system. Further, the inconsistency detection and reconstruction tool may perform consistency check of one or more of the hierarchical organization directories, the policy and configuration data files and the metadata to detect inconsistencies. Furthermore, the inconsistency detection and reconstruction tool may reconstruct one or more of the hierarchical organization directories, the policy and configuration data files and the metadata based on the outcome of the consistency check.

WO 2016/073029 A1

DETECTING INCONSISTENCIES IN HIERARCHICAL ORGANIZATION DIRECTORIES

Cross reference to related applications

[0001] A reference is made to application No. 5491/CHE/2014 entitled "POLICY AND CONFIGURATION DATA FOR A USER DIRECTORY" by Hewlett Packard Development Company, L.P filed on November 3, 2014.

Background

[0002] A file system can be used to store data in files and organize the files into hierarchical directories. Typically, a file system may get corrupted due to hardware, firmware and/or software errors. In such situations, a file system consistency check (FSCK) tool may be used to find and fix inconsistencies in the file system.

Brief Description of the Drawings

[0003] Examples of the disclosure will now be described in detail with reference to the accompanying drawings, in which:

[0004] FIG. 1 illustrates a block diagram of an example system for detecting inconsistencies and reconstructing hierarchical organization directories, policy and configuration data files and/or metadata in a file system;

[0005] FIG. 2 illustrates a block diagram of another example system for detecting inconsistencies and reconstructing the hierarchical organization directories, the policy and configuration data files and/or the metadata in the file system;

[0006] FIG. 3 illustrates a block diagram of an example file system, such as those shown in FIGs. 1 and 2;

[0007] FIG. 4 illustrates a flow chart of an example method for detecting inconsistencies and reconstructing hierarchical organization directories, policy and configuration data files and/or metadata in a file system;

[0008] FIG. 5 illustrates a flow chart of an example detailed method for detecting inconsistencies and reconstructing hierarchical organization directories or metadata in a file system; and

[0009] FIG. 6 illustrates a block diagram of an example computing device for detecting inconsistencies and reconstructing hierarchical organization directories, policy and configuration data files and/or metadata in a file system.

Detailed Description

[0010] In the following detailed description of the examples of the present subject matter, references are made to the accompanying drawings that form a part hereof, and in which are shown by way of illustration specific in which the present subject matter may be practiced. These examples are described in sufficient detail to enable those skilled in the art to practice the present subject matter, and it is to be understood that other examples may be utilized and that changes may be made without departing from the scope of the present subject matter. The following detailed description is, therefore, not to be taken in a limiting sense, and the scope of the present subject matter is defined by the appended claims.

[0011] Examples described herein provide enhanced methods, techniques, and systems for detecting inconsistencies and reconstructing hierarchical organization directories, policy and configuration data files and/or metadata in a file system. Generally, a file system consistency check (FSCK) tool is invoked to ensure consistency in a file system. The FSCK tool may be invoked after a system crash, a periodic health check for the file system, a user initiated consistency check and the like. To ensure consistency, the FSCK tool considers some piece of metadata associated with the file system as primal to detect and repair inconsistencies in the file system. After a

successful run of the FSCK tool, the user data in the file system may be accessible without any issues. However, the FSCK tool may not detect or repair some or all inconsistencies in user data.

[0012] In certain types of corruption scenarios, namespace co-relation associated with a file/directory may be completely lost. In these situations, the FSCK tool may recover the file/directory whose original name and/or parent file association cannot be determined and may place the file/directory in an "orphaned objects directory" in the file system. The FSCK tool may not have the ability to recover such files/directories to the right name and/or hierarchy in namespace of the file system.

[0013] In other scenarios, the FSCK tool may not be able to perform consistency check if the primal metadata is corrupted. Some existing solutions use replicas of the primal metadata to overcome this problem. However, if the replicated primal metadata is lost, the FSCK tool may not be able to validate and repair the file system. Further, such replications may result in performance and storage overheads.

[0014] To address these issues, the present disclosure describes various examples for detecting inconsistencies and reconstructing hierarchical organization directories, policy and configuration data files and/or metadata in a file system. In one example, the file system includes hierarchical organization directories, which are computing device and operating system (OS) independent, under a root directory in a namespace of the file system. Further, the file system includes user directories located under the hierarchical organization directories. Furthermore, the file system includes policy and configuration data files. For example, the policy and configuration data files include policy and configuration data associated with the hierarchical organization directories and policy and configuration data associated with the user directories. In addition, attributes (also referred to as metadata) associated with the hierarchical organization directories and the policy and configuration data files are stored in hierarchical organization directories namespace in the file system. The hierarchical organization directories, the policy and configuration data files and the metadata must be accessible and consistent in order for any user data in the file system to be accessible.

[0015] For example, in a system crash scenario, the file system may be in an inconsistent state. In such a scenario, the FSCK tool may be invoked to bring the file system to a consistent state. Further, the hierarchical organization directories, the policy and configuration data files and the metadata associated with the hierarchical organization directories and the policy and configuration data files are detected and reconstructed as described in the present disclosure.

[0016] The present disclosure describes creating a database including metadata associated with the hierarchical organization directories and the policy and configuration data files in the file system. Further, consistency check of the hierarchical organization directories, the policy and configuration data files and/or the metadata is performed. Furthermore, the hierarchical organization directories, the policy and configuration data files and/or the metadata are reconstructed based on the outcome of the consistency check.

[0017] The example implementations mentioned herein provides ability to create a database including the metadata associated with the hierarchical organization directories and the policy and configuration data files in the file system. The present disclosure further provides ability to choose between the metadata in the database and information in the hierarchical organization directories namespace as authoritative source for performing the consistency check. In one example, the hierarchical organization directories namespace in the file system includes the hierarchical organization directories and the policy and configuration data files. Further, the information in the hierarchical organization directories namespace includes the metadata associated with the hierarchical organization directories and the policy and configuration data files stored in associated inodes, extended attributes and/or any other way of associating metadata with the hierarchical organization directories and the policy and configuration data files. For example, the metadata includes attributes such as, a name, a unique identifier (ID) and an inode ID associated with the hierarchical organization directories and the policy and configuration files. The present disclosure further describes reconstructing the metadata in the database based

on the information in the hierarchical organization directories namespace when the metadata in the database is inconsistent.

[0018] The present disclosure also describes recovering orphaned hierarchical organization directories, i.e., hierarchical organization directories placed in the orphaned objects directory to their correct name hierarchy in the namespace in the file system using metadata in the database. Similarly, any orphaned policy and configuration data files may be recovered. In one example, the present disclosure describes recovering orphaned hierarchical organization directories and policy and configuration data files, using inode (index node) IDs stored in inodes, extended attributes and/or any other metadata that is part of the hierarchical organization directories and policy and configuration data files, when the associated metadata in the database is lost.

[0019] FIG. 1 illustrates a block diagram of an example system 100 for detecting inconsistencies and reconstructing hierarchical organization directories, policy and configuration data files and/or metadata in a file system 106. The system 100 may represent any type of computing device capable of reading machine-executable instructions. Examples of computing device may include, without limitation, a server, a desktop computer, a notebook computer, a tablet computer, a thin client, a mobile device, a personal digital assistant (PDA), a phablet, and the like.

[0020] In the example of FIG. 1, the system 100 may include a processor 102 and storage device 104 coupled to the processor 102. In an example, the storage device 104 may be a machine readable storage medium (e.g., a disk drive). The machine-readable storage medium may also be an external medium that may be accessible to the system 100. Further, the storage device 104 may include the file system 106. Furthermore, the file system 106 may include an inconsistency detection and reconstruction tool 108.

[0021] For example, the inconsistency detection and reconstruction tool 108 may refer to software components (machine executable instructions), a hardware component or a combination thereof. The inconsistency detection and reconstruction tool 108 may include, by way of example, components, such as software components, processes, tasks, co-routines, functions, attributes, procedures, drivers, firmware, data, databases, data structures and Application Specific Integrated Circuits (ASIC). The inconsistency detection and reconstruction tool 108 may reside on a volatile or non-volatile storage medium and configured to interact with a processor 102 of the system 100.

[0022] In one example, the file system 106 may include hierarchical organization directories (e.g., hierarchical organization directories 304 shown in FIG. 3). For example, the hierarchical organization directories are top level directory hierarchies in the file system 106 as shown in FIG. 3. The hierarchical organization directories may be located between a root directory (e.g., a root directory 302 shown in FIG. 3) and user directories (e.g., user directories (UDs) 312A- E shown in FIG. 3) in the file system 106. This is explained in detail with reference to FIG. 3. For example, the hierarchical organization directories serve as an entry point to user data stored in the user directories. These hierarchical organization directories may be marked with a flag and a unique ID for uniquely identifying the hierarchical organization directories. For example, the unique IDs associated with the hierarchical organization directories are monotonically increasing to ensure that there are no collisions.

[0023] In this example, each of the hierarchical organization directories is associated with metadata. For example, the metadata includes attributes, such as a name, a parent ID, an inode ID, the unique ID and so on. In one example, metadata associated with a hierarchical organization directory may be stored in an inode and/or extended attributes associated with the hierarchical organization directory. An inode is a data structure used to represent file system objects, such as files and directories. The inode and the extended attributes associated with the hierarchical organization directories may be stored in the file system 106.

[0024] Further in this example, the hierarchical organization directories may be associated with policy and configuration data files. The policy and configuration data files may include default policy and configuration data associated with the hierarchical organization directories and policy and configuration data associated with the user directories. For example, the policy and configuration data files may be stored in any directory in the file system 106. Furthermore, metadata associated with the configuration and policy data files may be stored as extended attributes and/or inode associated with the configuration and policy data files. For example, the metadata associated with the policy and configuration data files includes a name (e.g., a policy name and a configuration name), a unique ID, an inode ID, a unique ID of the associated hierarchical organization directory and so on.

[0025] In operation, the inconsistency detection and reconstruction tool 108 may create a database (e.g., a database 310 shown in FIG. 3) to include the metadata associated with the hierarchical organization directories and the policy and configuration data files in the file system 106. For example, the database may be a directory under the root directory in the file system 106 as shown in FIG. 3. Further, the database may include a first file to store the metadata associated with the hierarchical organization directories. The inconsistency detection and reconstruction tool 108 may create a record for each of the hierarchical organization directories in the first file. Each record may include metadata associated with the hierarchical organization directories. For example, the records may be created as and when the hierarchical organization directories are created in the file system 106. Furthermore, the database may include a second file to store the metadata associated with the policy and configuration data files. It may be understood that, the database is made self sufficient to reconstruct hierarchical organization directories namespace and policies and configuration data stored in it.

[0026] Further, each of the first and the second files in the database are associated with a unique ID for its identification and a checksum for validating its integrity. A checksum is a small-size datum that is computed from arbitrary blocks of data for the

purpose of detecting errors. The checksum may be computed using standard checksum algorithms, such as cyclic redundancy check 32 (CRC32) and the like.

[0027] In a system crash scenario, the file system 106 may be in an inconsistent state. In such a scenario, a FSCK tool (e.g., a FSCK tool 206 shown in FIG. 2) may be invoked to bring the file system 106 to a consistent state. The FSCK tool may use the metadata in the database to bring the file system 106 to a consistent state. In one example, the FSCK tool may recover one or more objects (e.g., hierarchical organization directories and policy and configuration data files) whose name cannot be determined. Such objects are referred to as orphaned objects. In this example, the FSCK tool may place the orphaned objects in an orphaned objects directory (e.g., an orphaned objects directory 316, shown in FIG. 3) under the root directory of the file system 106. For example, the FSCK tool may name an orphaned object using an inode ID associated with the orphaned object.

[0028] Further, OS (e.g., OS 204 residing in memory 202 (e.g., an internal memory) of a system 200, shown in FIG. 2) may send a request to the inconsistency detection and reconstruction tool 108 to perform consistency check of the hierarchical organization directories, policy and configuration data files and the metadata stored in the database. For example, the OS may also send the request during a periodic consistency check of the file system, a user initiated consistency check and the like.

[0029] To perform the consistency check of the metadata in the database, the inconsistency detection and reconstruction tool 108 validates integrity of the metadata. For example, the inconsistency detection and reconstruction tool 108 may validate integrity of the metadata using checksum. In this example, the inconsistency detection and reconstruction tool 108 computes a checksum for each of the first and the second files in the database. Further, the inconsistency detection and reconstruction tool 108 determines whether the computed checksum for each of the first and the second files matches with the checksum associated with the first and second files, respectively. In one example, if it is determined that the checksums match, the inconsistency detection and reconstruction tool 108 may conclude that the

metadata is consistent and may perform consistency check of the hierarchical organization directories and the policy and configuration data files using the metadata in the first and the second files, respectively.

[0030] During the consistency check of the hierarchical organization directories using the metadata in the first file, it is determined whether a hierarchical organization directory corresponding to each record in the first file is present in the file system 106. If it is determined that a hierarchical organization directory corresponding to each record in the first file is present in the file system 106, the inconsistency detection and reconstruction tool 108 may conclude that the hierarchical organization directories in the file system 106 are consistent. If it is determined that a hierarchical organization directory corresponding to each record in the first file is not present in the file system 106, the inconsistency detection and reconstruction tool 108 may conclude that the hierarchical organization directories in the file system 106 are inconsistent and may require repair and/or reconstruction.

[0031] To reconstruct the hierarchical organization directories using the metadata in the first file, the inconsistency detection and reconstruction tool 108 may determine whether a hierarchical organization directory corresponding to a record in the first file is present in the file system 106. For example, the inode ID stored in the record is used to determine whether a hierarchical organization directory with the same inode ID is present in the file system 106.

[0032] If it is determined that a hierarchical organization directory corresponding to the record is not present, the inconsistency detection and reconstruction tool 108 checks the orphaned objects directory to determine whether a hierarchical organization directory with the same in ode ID is present. If the hierarchical organization directory with the same inode ID is present in the orphaned objects directory, the inconsistency detection and reconstruction tool 108 may restore the hierarchical organization directory to correct location in the hierarchical organization directory namespace. If the hierarchical organization directory with the same inode ID is not present in the orphaned objects directory, the inconsistency detection and reconstruction tool 108

may create the hierarchical organization directory with the unique ID associated with the hierarchical organization directory in the hierarchical organization directory namespace and restore contents of the hierarchical organization directory from a backup.

[0033] Further, if it is determined that the hierarchical organization directory corresponding to the record is present, the inconsistency detection and reconstruction tool 108 may compare the metadata associated with the hierarchical organization directory, that is stored in it's inode, extended attributes and/or any other way of associating metadata with file system objects, to the metadata stored in the record. Furthermore, the inconsistency detection and reconstruction tool 108 may repair or reconstruct the metadata in the hierarchical organization directory when the metadata does not match. In addition, the inconsistency detection and reconstruction tool 108 may repeat the step of determining whether a hierarchical organization directory is present for each record in the first file.

[0034] Similarly, the inconsistency detection and reconstruction tool 108 may detect inconsistencies and reconstruct the policy and configuration data files using the metadata associated with the policy and configuration data files stored in the second file.

[0035] In another example, if it is determined that the checksum of the first file does not match, the inconsistency detection and reconstruction tool 108 may conclude that the first file is inconsistent. In one example, the inconsistency detection and reconstruction tool 108 may reconstruct the first file by creating a record for each of the hierarchical organization directories in the file system 106. In this example, the inconsistency detection and reconstruction tool 108 may reconstruct the first file using information in the inode, and/or extended attributes in the hierarchical organization directories namespace.

[0036] Upon reconstructing the first file, the inconsistency detection and reconstruction tool 108 may determine whether records for all the hierarchical organization directories in the file system 106 are created. To determine whether records for all the hierarchical organization directories in the file system 106 are created, the inconsistency detection and reconstruction tool 108 may search the orphaned objects directory. Further, when a missing hierarchical organization directory is found in the orphaned objects directory, the hierarchical organization directory is restored to correct location in the hierarchical organization directory namespace and the metadata associated with the hierarchical organization directory is updated in the first file.

[0037] Similarly, if it is determined that the checksum of the second file does not match, the inconsistency detection and reconstruction tool 108 may conclude that the second file is inconsistent. Further, the inconsistency detection and reconstruction tool 108 may reconstruct the second file using metadata associated with policy and configuration data files which are stored as extended attributes, inode and/or any other mechanism for associating metadata with the policy and configuration data files.

[0038] FIG. 3 illustrates a block diagram of an example file system 300, such as shown in FIGs. 1 and 2. The file system 300 may be created on one or more segments of a storage device. As shown in FIG. 3, the file system 300 may include hierarchical organization directories 304 under a root directory 302. In the example illustrated in FIG. 3, the hierarchical organization directories 304 include organization directories (ODs) 306A and 306B and a default OD 306C under the root directory 302. Further, the hierarchical organization directories 304 include organization sub-directories (OSDs) 308A and 308B under the OD 306A, an OSD 308C under the OD 306B and a default OSD 308D under the default OD 306C. In an example, an organization directory can be a directory associated with an organization and an organization sub-directory can be a department within the organization. Furthermore as shown in FIG. 3, user directories (UDs) 312A-E including files 314A-E, respectively, are located under the associated OSDs 308A-C. For example, a user directory can be a directory associated with a user within the department in the organization.

[0039] In one example, the ODs 306A-C and OSDs 308A-D include policy and configuration data files. For example, the ODs 306A and 306B may include policy and configuration data files associated with authentication and the OSDs 308A-D may store policy and configuration data files associated with storage allocation. Example policy data includes hierarchical name and value pairs. The "name" is a policy name and "value" is the policy applicable. Further, the default OD 306C may store policy and configuration data files that is applicable to ODs 306A and 306B. Furthermore, the default OSD 308D may store policy and configuration data files that are applicable to OSDs 308A-C.

[0040] In an example, the database 310 may include metadata associated with the hierarchical organization directories 304 (ODs 306A and 306B and OSDs 308A-C) and policy and configuration data files. For example, the unique ID and the hierarchy of the ODs 306A and 306B and OSDs 308A-C are recorded in a file under the database. The file is used as the authentic source of information about the hierarchy and can be recreated. Further, for each policy hive (e.g., a group of policies) that includes a set of name and value pairs, a file is kept in the database 310.

[0041] FIG. 4 illustrates a flow chart of an example method 400 for detecting inconsistencies and reconstructing hierarchical organization directories, policy and configuration data files and/or metadata in a file system. The method 400, which is described below, may be executed on a system such as a system 100 of FIG. 1 or system 200 of FIG. 2. However, other systems may be used as well. At block 402, a database including metadata associated with hierarchical organization directories and policy and configuration data files in the file system may be created. For example, the hierarchical organization directories are located between a root directory and user directories in the file system. In one example, metadata associated with the hierarchical organization directories and the policy and configuration data files are stored in extended attributes and/or inodes associated with the hierarchical organization directories and the policy and configuration data files, respectively. The metadata may include attributes, such as a name, a unique ID and an inode ID associated with the hierarchical organization directories and the policy and

configuration files and ID of the hierarchical organization directories associated with the policy and configuration data files.

[0042] At block 404, consistency check of the hierarchical organization directories, the policy and configuration data files and/or the metadata in the database are performed to detect inconsistencies. For example, the consistency check is performed in response to receiving a request from an OS during one of a system crash, a periodic consistency check of file system and a user initiated consistency check. In one example, integrity of the metadata in the database is validated. For example, the integrity of the metadata is validated using checksum. This is explained in detail with reference to FIG. 1. Further, consistency check of the hierarchical organization directories and the policy and configuration data files are performed using the metadata in the database when the validation is successful.

[0043] At block 406, the hierarchical organization directories, the policy and configuration data files and/or the metadata may be reconstructed based on the outcome of the consistency check. In one example, the metadata may be reconstructed using the hierarchical organization directories and the policy and configuration data files when the validation is not successful. In this example, the metadata may be reconstructed by creating a record for each of the hierarchical organization directories and the policy and configuration data files. For example, each record may include metadata of an associated one of the hierarchical organization directories and the policy and configuration data files. The metadata includes attributes, such as of a name, a unique ID and an inode ID associated with the hierarchical organization directories and the policy and configuration files.

[0044] In another example, the hierarchical organization directories and the policy and configuration data files may be reconstructed using the metadata when the hierarchical organization directories or the policy and configuration data files are inconsistent. In this example, it may be determined whether a hierarchical organization directory or the policy and configuration data file corresponding to a record in the metadata is present in the file system. For example, the record may

include metadata of the hierarchical organization directory or the policy and configuration data file. The metadata includes attributes, such as a name, a unique ID and an inode ID associated with the hierarchical organization directory or the policy and configuration file. Further, if it is determined that a hierarchical organization directory or the policy and configuration data file corresponding to the record in the metadata is not present, the hierarchical organization directory or the policy and configuration data file is created using the metadata. Furthermore, if it is determined that the hierarchical organization directory or the policy and configuration data file corresponding to the record in the metadata is present the step of determining is repeated for each record in the metadata.

[0045] FIG. 5 illustrates a flow chart of an example detailed method 500 for detecting inconsistencies and reconstructing hierarchical organization directories or metadata in a file system. At block 502, integrity of metadata associated with hierarchical organization directories (HOD) (e.g., hierarchical organization directories 304 shown in FIG. 3) is validated. For example, the integrity of the metadata is validated using checksum. This is explained in detail with reference to FIG. 1.

[0046] At block 504, a check is made to determine whether the validation is successful. If the validation is not successful, discard the metadata at bock 506. At block 508, populate attributes of HOD in the metadata for all HOD present in the hierarchical organization directories namespace. For example, the attributes of HOD include a name, a unique ID, an inode ID and the like. If the validation is successful, obtain a record from the metadata at block 510. At block 512, a check is made to determine whether a HOD corresponding to the obtained record is found in the hierarchical organization directories namespace. If the HOD is found in the hierarchical organization directories namespace, match and fix attributes of HOD from the record obtained at block 514. At block 516, a check is made to determine whether any more records are to be processed in the metadata. If so, repeat the process steps from the block 512. If not, end the process at block 526.

[0047] If the HOD is not found in the hierarchical organization directories namespace, search an orphaned objects directory (e.g., a orphaned objects directory 316 shown in FIG. 3) at block 518. At block 520, a check is made to determine if the HOD is found in the orphaned objects directory. If the HOD is found in the orphaned objects directory, rename the HOD at block 522 and repeat the process steps from the block 514. If the HOD is not found in the orphaned objects directory, recreate the HOD at block 524 and repeat the process steps from the block 516.

[0048] FIG. 6 illustrates a block diagram of an example computing device 600 for detecting inconsistencies and reconstructing hierarchical organization directories, policy and configuration data files and/or metadata in a file system. The computing device 600 includes a processor 602 and a machine-readable storage medium 604 communicatively coupled through a system bus. The processor 602 may be any type of central processing unit (CPU), microprocessor, or processing logic that interprets and executes machine-readable instructions stored in the machine-readable storage medium 604. The machine-readable storage medium 604 may be a random access memory (RAM) or another type of dynamic storage device that may store information and machine-readable instructions that may be executed by the processor 602. For example, the machine-readable storage medium 604 may be synchronous DRAM (SDRAM), double data rate (DDR), rambus DRAM (RDRAM), rambus RAM, etc., or storage memory media such as a floppy disk, a hard disk, a CD-ROM, a DVD, a pen drive, and the like. In an example, the machine-readable storage medium 604 may be a non-transitory machine-readable medium. In an example, the machine-readable storage medium 604 may be remote but accessible to the computing device 600.

[0049] The machine-readable storage medium 604 may store instructions 606, 608 and 610. In an example, instructions 606 may be executed by processor 602 to create a database including metadata associated with hierarchical organization directories and policy and configuration data files in a file system. Instructions 608 may be executed by processor 602 to perform consistency check of the hierarchical organization directories, the policy and configuration data files and/or the metadata. Instructions

610 may be executed by processor 602 to reconstruct the hierarchical organization directories, the policy and configuration data files and/or the metadata based on the outcome of the consistency check.

[0050] It may be noted that the above-described examples of the present solution is for the purpose of illustration only. Although the solution has been described in conjunction with a specific embodiment thereof, numerous modifications may be possible without materially departing from the teachings and advantages of the subject matter described herein. Other substitutions, modifications and changes may be made without departing from the spirit of the present solution. All of the features disclosed in this specification (including any accompanying claims, abstract and drawings), and/or all of the steps of any method or process so disclosed, may be combined in any combination, except combinations where at least some of such features and/or steps are mutually exclusive.

Claims:

1. A system, comprising:

a processor; and

a storage device communicatively coupled to the processor, wherein the storage device comprises an inconsistency detection and reconstruction tool to:

create a database including metadata associated with hierarchical organization directories and policy and configuration data files in a file system;

perform consistency check of at least one of the hierarchical organization directories, the policy and configuration data files and the metadata to detect inconsistencies; and

reconstruct at least one of the hierarchical organization directories, the policy and configuration data files and the metadata based on the outcome of the consistency check.

2. The system of claim 1, wherein the inconsistency detection and reconstruction tool validates integrity of the metadata in the database and performs consistency check of the hierarchical organization directories and the policy and configuration data files using the metadata in the database when the validation is successful.

3. The system of claim 2, wherein the inconsistency detection and reconstruction tool reconstructs the hierarchical organization directories and the policy and configuration data files using the metadata in the database when the hierarchical organization directories and the policy and configuration data files are inconsistent.

4. The system of claim 2, wherein the inconsistency detection and reconstruction tool reconstructs the metadata in the database using the hierarchical organization directories and the policy and configuration data files when the validation is not successful.

5. The system of claim 4, wherein the inconsistency detection and reconstruction tool reconstructs the metadata in the database by creating a record for each of the hierarchical organization directories and the policy and configuration data files in the file system,

wherein each record includes metadata associated with the hierarchical organization directories and the policy and configuration data files, wherein the metadata includes attributes selected from the group consisting of a name, a unique identifier (ID) and an inode ID associated with the hierarchical organization directories and the policy and configuration files.

6. The system of claim 3, wherein the inconsistency detection and reconstruction tool determines whether a hierarchical organization directory or a policy and configuration data file is present in the file system for a record in the metadata in the database, wherein the record includes metadata associated with the hierarchical organization directory or the policy and configuration data file, wherein the metadata includes attributes selected from the group consisting of a name, a unique ID and an inode ID associated with the hierarchical organization directory or the policy and configuration file, reconstructs the hierarchical organization directory or the policy and configuration data file using the metadata when the hierarchical organization directory or the policy and configuration data file is not present and repeats the step of determining for each record in the metadata in the database.

7. The system of claim 1, wherein the inconsistency detection and reconstruction tool performs the consistency check in response to receiving a request from an operating system during one of a system crash, a periodic consistency check of file system and a user initiated consistency check.

8. The system of claim 1, wherein the hierarchical organization directories are located between a root directory and user directories in the file system, wherein the metadata associated with the hierarchical organization directories and the policy and configuration data files are stored in extended attributes and/or inodes of the hierarchical organization directories and the policy and configuration data files, respectively, wherein the metadata comprises attributes selected from the group consisting of a name, a unique ID and an inode ID associated with the hierarchical organization directories and the policy and configuration files and a unique ID of the hierarchical organization directories associated with the policy and configuration data files.

9. A method, comprising:

creating a database including metadata associated with hierarchical organization directories and policy and configuration data files in a file system;

performing consistency check of at least one of the hierarchical organization directories, the policy and configuration data files and the metadata to detect inconsistencies; and

reconstructing at least one of the hierarchical organization directories, the policy and configuration data files and the metadata based on the outcome of the consistency check.

10. The method of claim 9, wherein performing consistency check of at least one of the hierarchical organization directories, the policy and configuration data files and the metadata to detect inconsistencies comprises:

validating integrity of the metadata in the database; and

performing consistency check of the hierarchical organization directories and the policy and configuration data files using the metadata in the database when the validation is successful.

11. The method of claim 10, wherein reconstructing at least one of the hierarchical organization directories, the policy and configuration data files and the metadata based on the outcome of the consistency check comprises:

reconstructing the hierarchical organization directories and the policy and configuration data files using the metadata in the database when the hierarchical organization directories and the policy and configuration data files are inconsistent.

12. The method of claim 10, wherein reconstructing at least one of the hierarchical organization directories, the policy and configuration data files and the metadata based on the outcome of the consistency check comprises:

reconstructing the metadata in the database using the hierarchical organization directories and the policy and configuration data files when the validation is not successful.

13. The method of claim 12, wherein reconstructing the metadata using the hierarchical organization directories and the policy and configuration data files when the validation is not successful comprises:

reconstructing the metadata by creating a record for each of the hierarchical organization directories and the policy and configuration data files in the file system, wherein each record includes metadata associated with one of the hierarchical organization directories and the policy and configuration data files, wherein the metadata includes attributes selected from the group consisting of a name, a unique identifier (ID) and an inode ID associated with the hierarchical organization directories and the policy and configuration files.

14. The method of claim 11, wherein reconstructing the hierarchical organization directories and the policy and configuration data files using the metadata when the hierarchical organization directories and the policy and configuration data files are inconsistent comprises:

determining whether a hierarchical organization directory or a policy and configuration data file is present in the file system for a record in the metadata in the database, wherein the record includes metadata associated with the hierarchical organization directory or the policy and configuration data file, wherein the metadata includes attributes selected from the group consisting of a name, a unique ID and an inode ID associated with the hierarchical organization directory or the policy and configuration file;

reconstructing the hierarchical organization directory or the policy and configuration data file using the metadata when the hierarchical organization directory or the policy and configuration data file is not present; and

repeating the step of determining for each record in the metadata in the database.

15. A non-transitory machine-readable storage medium comprising instructions for detecting inconsistencies and reconstructing hierarchical organization directories, policy and configuration data files and/or metadata, the instructions executable by a processor to:

create a database including metadata associated with hierarchical organization directories and policy and configuration data files in a file system;

perform consistency check of at least one of the hierarchical organization directories, the policy and configuration data files and the metadata to detect inconsistencies; and

reconstruct at least one of the hierarchical organization directories, the policy and configuration data files and the metadata based on the outcome of the consistency check.

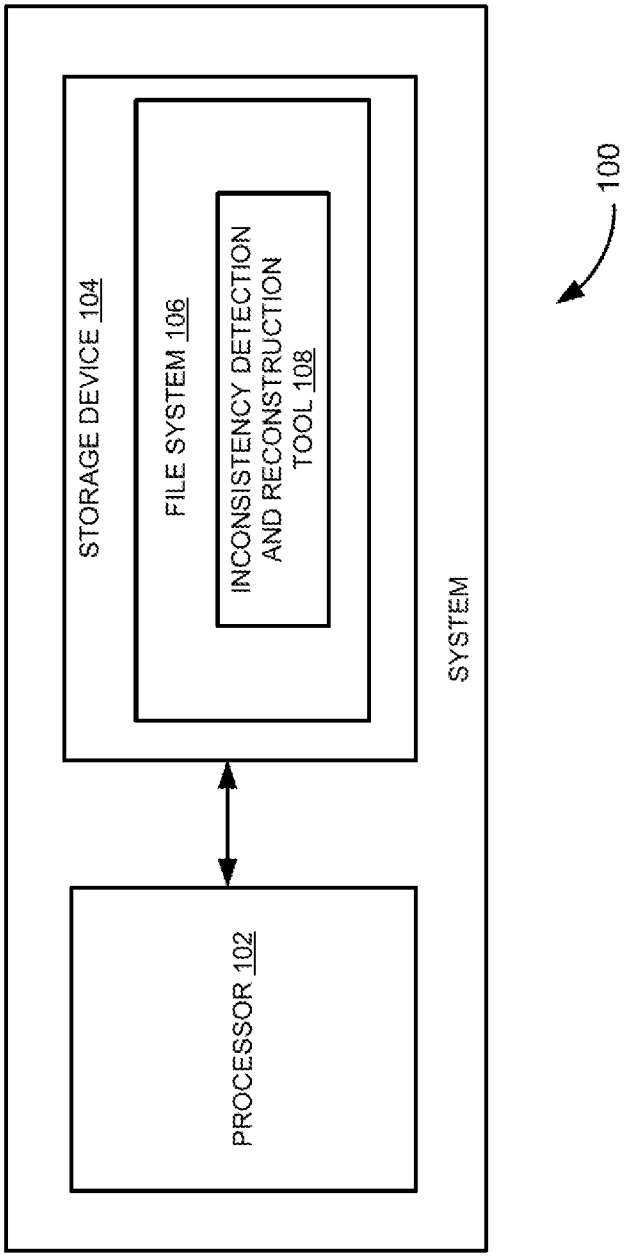

100
SYSTEM
STORAGE DEVICE 104
FILE SYSTEM 106
INCONSISTENCY DETECTION AND RECONSTRUCTION TOOL 108
PROCESSOR 102

FIG. 1

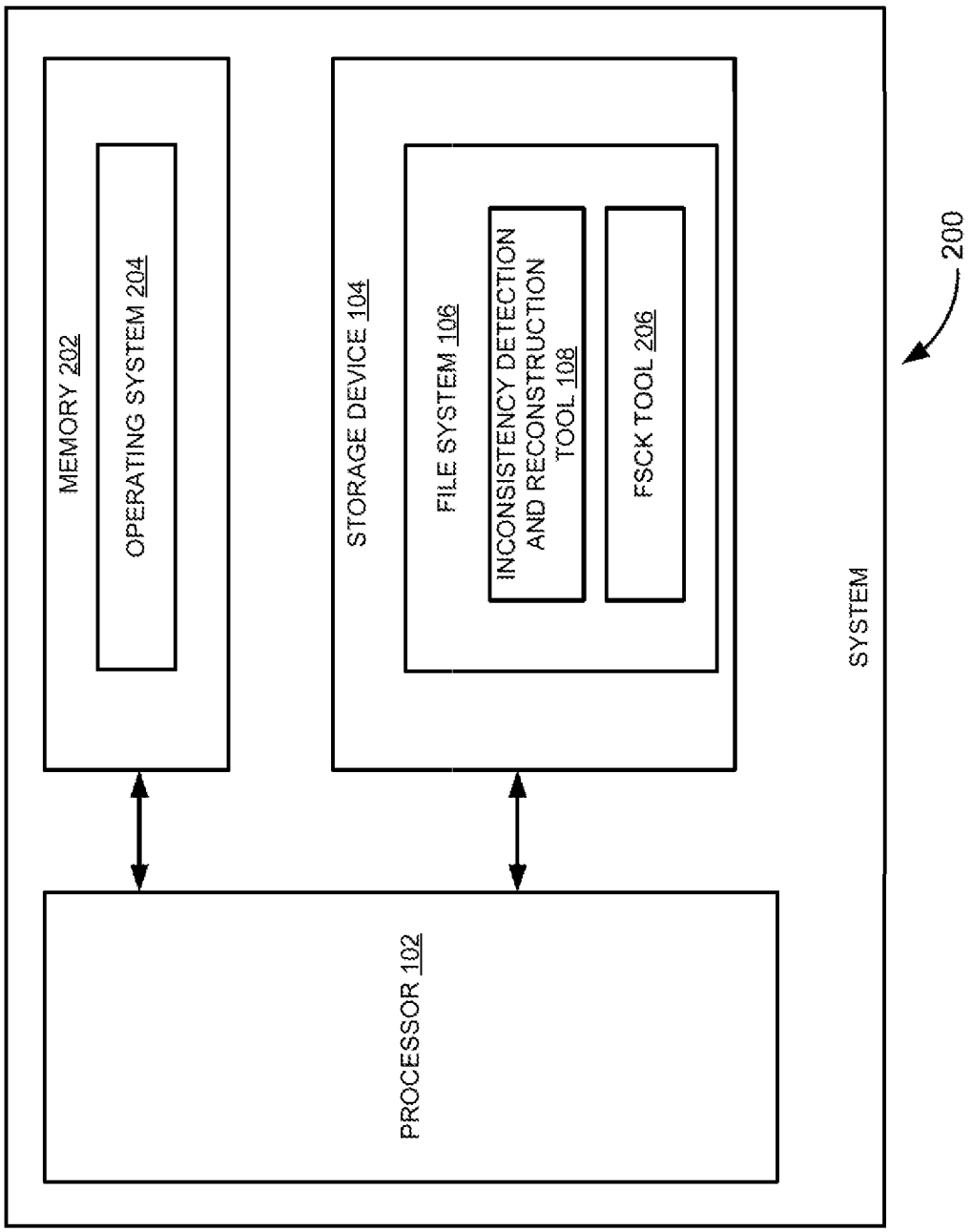

MEMORY 202
OPERATING SYSTEM 204
STORAGE DEVICE 104
FILE SYSTEM 106
INCONSISTENCY DETECTION AND RECONSTRUCTION TOOL 108
FSCK TOOL 206
PROCESSOR 102
SYSTEM
200

FIG. 2

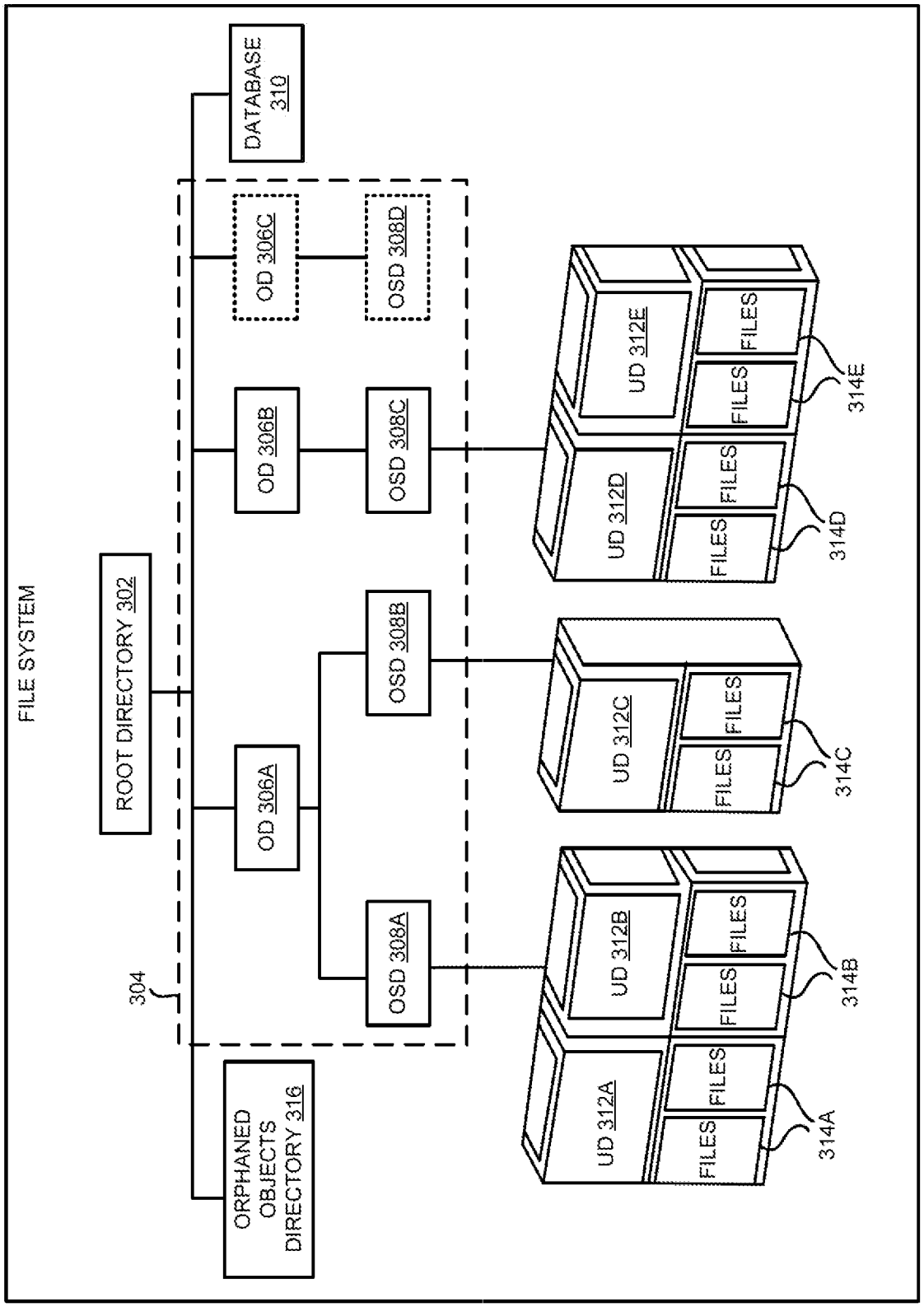

FILE SYSTEM
DATABASE 310
ROOT DIRECTORY 302
OD 306A
OD 306B
OD 306C
OSD 308A
OSD 308B
OSD 308C
OSD 308D
304
ORPHANED OBJECTS DIRECTORY 316
UD 312A
UD 312B
UD 312C
UD 312D
UD 312E
FILES
314A
314B
314C
314D
314E
300

FIG. 3

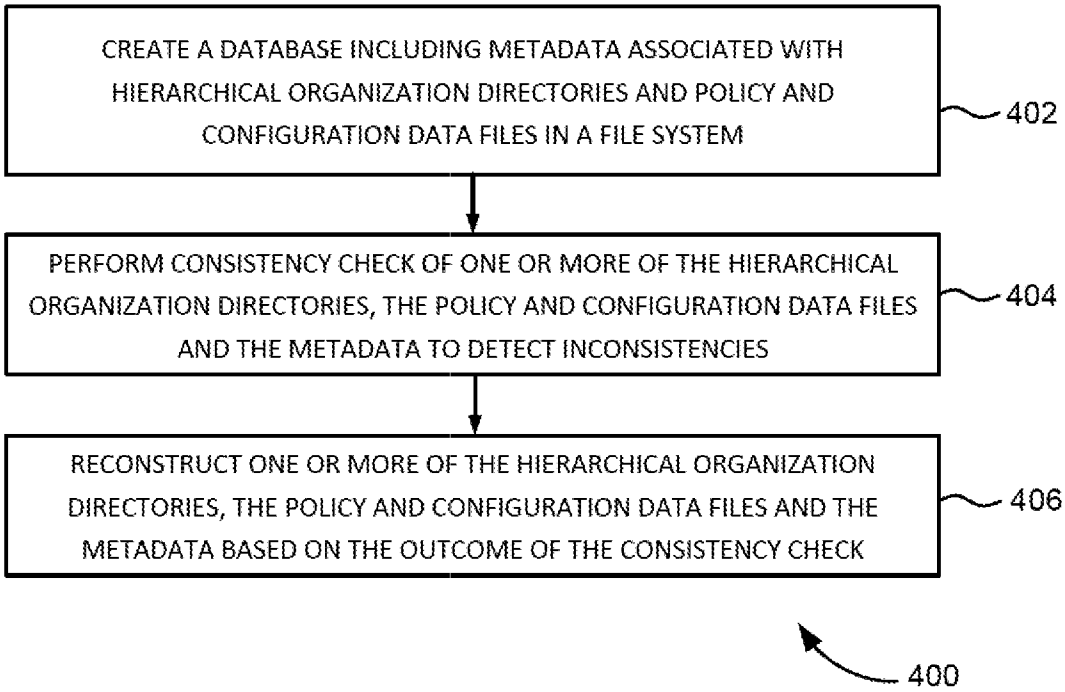

CREATE A DATABASE INCLUDING METADATA ASSOCIATED WITH HIERARCHICAL ORGANIZATION DIRECTORIES AND POLICY AND CONFIGURATION DATA FILES IN A FILE SYSTEM
402
PERFORM CONSISTENCY CHECK OF ONE OR MORE OF THE HIERARCHICAL ORGANIZATION DIRECTORIES, THE POLICY AND CONFIGURATION DATA FILES AND THE METADATA TO DETECT INCONSISTENCIES
404
RECONSTRUCT ONE OR MORE OF THE HIERARCHICAL ORGANIZATION DIRECTORIES, THE POLICY AND CONFIGURATION DATA FILES AND THE METADATA BASED ON THE OUTCOME OF THE CONSISTENCY CHECK
406
400

FIG. 4

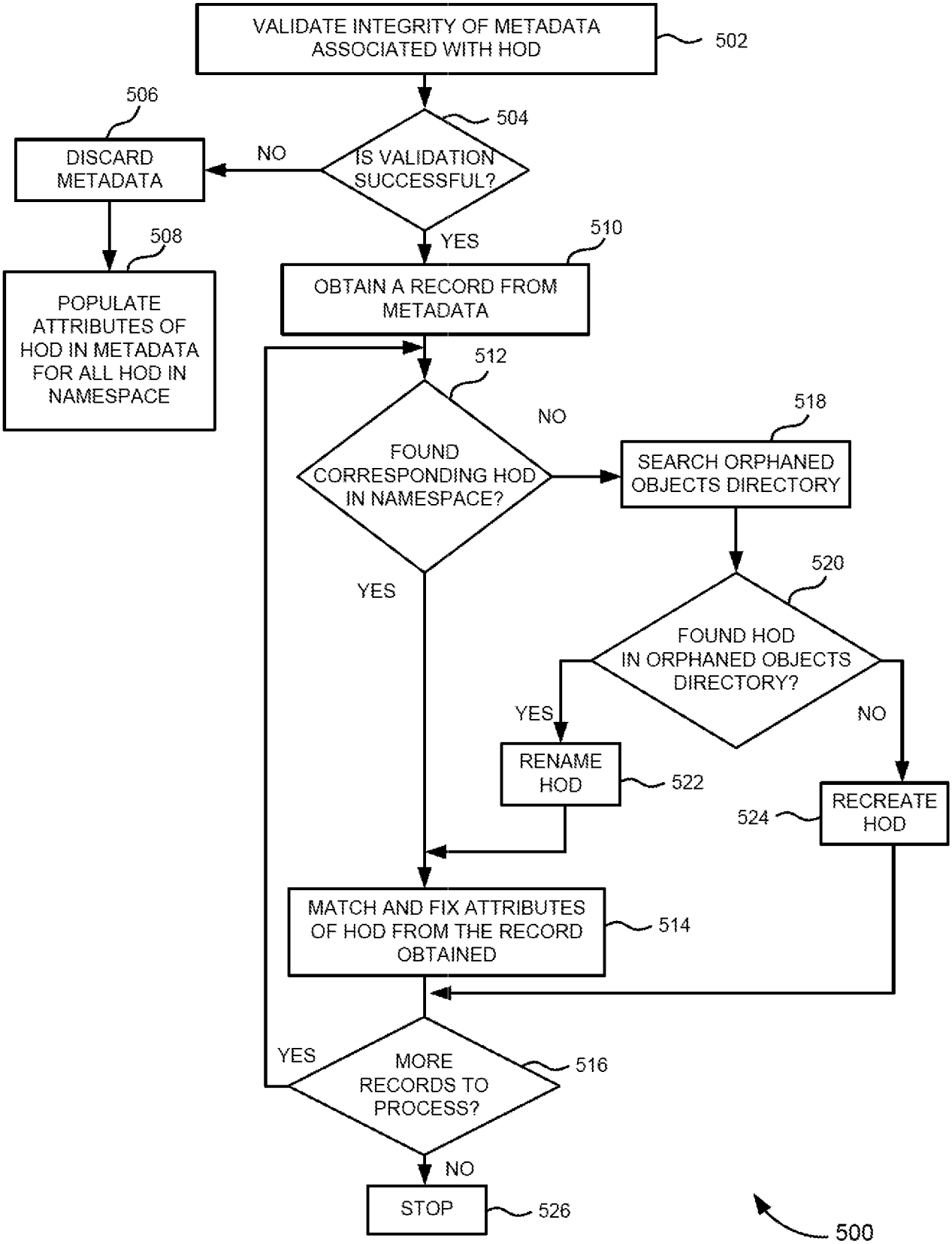

VALIDATE INTEGRITY OF METADATA ASSOCIATED WITH HOD
502
IS VALIDATION SUCCESSFUL?
504
NO
DISCARD METADATA
506
YES
POPULATE ATTRIBUTES OF HOD IN METADATA FOR ALL HOD IN NAMESPACE
508
OBTAIN A RECORD FROM METADATA
510
FOUND CORRESPONDING HOD IN NAMESPACE?
512
NO
SEARCH ORPHANED OBJECTS DIRECTORY
518
YES
FOUND HOD IN ORPHANED OBJECTS DIRECTORY?
520
YES
NO
RENAME HOD
522
RECREATE HOD
524
MATCH AND FIX ATTRIBUTES OF HOD FROM THE RECORD OBTAINED
514
MORE RECORDS TO PROCESS?
516
YES
NO
STOP
526
500

FIG. 5

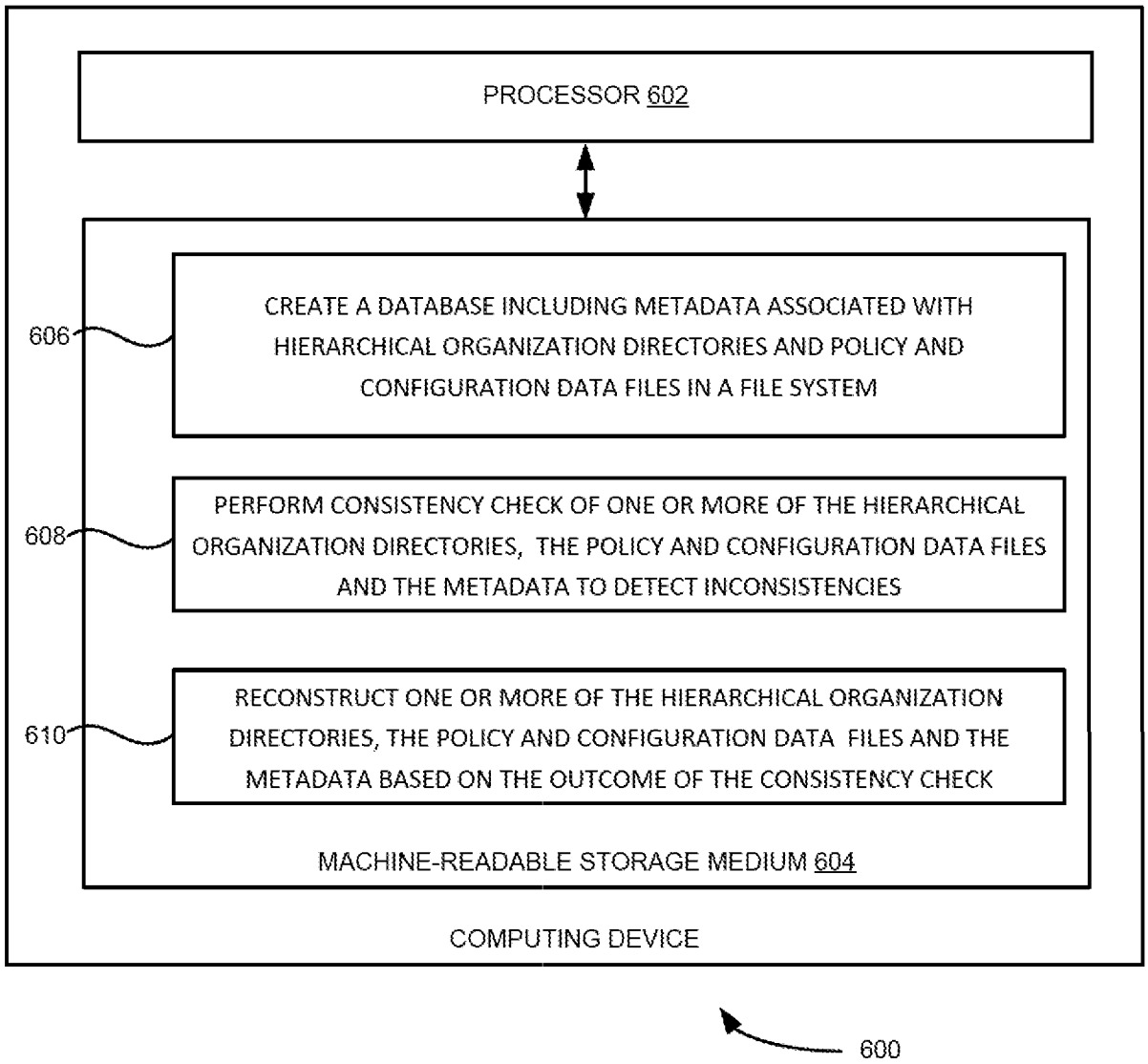

PROCESSOR 602
606
CREATE A DATABASE INCLUDING METADATA ASSOCIATED WITH HIERARCHICAL ORGANIZATION DIRECTORIES AND POLICY AND CONFIGURATION DATA FILES IN A FILE SYSTEM
608
PERFORM CONSISTENCY CHECK OF ONE OR MORE OF THE HIERARCHICAL ORGANIZATION DIRECTORIES, THE POLICY AND CONFIGURATION DATA FILES AND THE METADATA TO DETECT INCONSISTENCIES
610
RECONSTRUCT ONE OR MORE OF THE HIERARCHICAL ORGANIZATION DIRECTORIES, THE POLICY AND CONFIGURATION DATA FILES AND THE METADATA BASED ON THE OUTCOME OF THE CONSISTENCY CHECK
MACHINE-READABLE STORAGE MEDIUM 604
COMPUTING DEVICE
600

FIG. 6

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2015/027419

A. CLASSIFICATION OF SUBJECT MATTER

G06F 17/30(2006.01)i, G06F 11/07(2006.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G06F 17/30; G06F 17/00; G06F 7/00; G06F 12/00; G06F 15/167; G06F 11/07

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
Korean utility models and applications for utility models
Japanese utility models and applications for utility models

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
eKOMPASS(KIPO internal) & Keywords: file system, consistency check, hierarchical organization directories, metadata, policy, configuration, reconstruct, and similar terms.

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 2012-0158652 A1 (PAVAN PS et al.) 21 June 2012 See paragraphs [0020], [0022]-[0023], [0026], [0032]-[0037], [0040], and [0042]-[0044]; claims 17 and 21-22; and figures 1 and 5-6.	1-4,7-12,15
A		5-6,13-14
Y	WO 2010-050944 A1 (HEWLETT-PACKARD DEVELOPMENT COMPANY, L.P.) 06 May 2010 See page 3, lines 3-12; page 7, lines 12-24; page 8, lines 2-8; and figures 2-3.	1-4,7-12,15
A	US 2006-0282471 A1 (TIMOTHY W. MARK et al.) 14 December 2006 See paragraphs [0021]-[0029] and figure 2.	1-15
A	US 2013-0325817 A1 (STEVEN JOHN WHITEHOUSE) 05 December 2013 See paragraphs [0020] and [0022]-[0028]; and figures 1-2.	1-15
A	US 7,610,307 B2 (SAROSH CYRUS HAVEWALA et al.) 27 October 2009 See column 7, line 5 - column 8, line 22; column 9, lines 10-63; and figures 5 and 7.	1-15

☐ Further documents are listed in the continuation of Box C. ☒ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents,such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
29 July 2015 (29.07.2015)	**29 July 2015 (29.07.2015)**

Name and mailing address of the ISA/KR
International Application Division
Korean Intellectual Property Office
189 Cheongsa-ro, Seo-gu, Daejeon Metropolitan City, 302-701, Republic of Korea
Facsimile No. +82-42-472-7140

Authorized officer
NHO, Ji Myong
Telephone No. +82-42-481-8528

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2015/027419

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2012-0158652 A1	21/06/2012	None	
WO 2010-050944 A1	06/05/2010	None	
US 2006-0282471 A1	14/12/2006	None	
US 2013-0325817 A1	05/12/2013	US 8793224 B2	29/07/2014
US 7610307 B2	27/10/2009	US 2006-0117056 A1	01/06/2006