



- (51) International Patent Classification:
G06F 17/30 (2006.01)
- (21) International Application Number:
PCT/US2015/023649
- (22) International Filing Date:
31 March 2015 (31.03.2015)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:
424/CHE/2015 29 January 2015 (29.01.2015) IN
- (71) Applicant: **HEWLETT PACKARD ENTERPRISE DEVELOPMENT LP** [US/US]; 11445 Compaq Center Drive West, Houston, TX 77070 (US).
- (72) Inventors: **PENDYALA, Jaipal**; Sy.No.192, Whitefield Road Mahadevapura, Post, Karnataka, Bangalore 560048 (IN). **SIVASHANMUGAM, Jothivelavan**; Sy.No.192, Whitefield Road, Mahadevapura Post, Karnataka, Bangalore 560048 (IN). **K, Anmary Justine**; Sy.No.192, Whitefield Road, Mahadevapura Post, Karnataka, Bangalore 560048 (IN). **KANNAN, Rajkumar**; Sy.No.192, Whitefield Road, Mahadevapura Post, Karnataka, Bangalore 560048 (IN). **K, Ramesh Kannan**; Sy.No.192, Whitefield Road, Mahadevapura Post, Karnataka, Bangalore 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))*
 - *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

[Continued on next page]

(54) Title: FILE SYSTEM REPLICATION

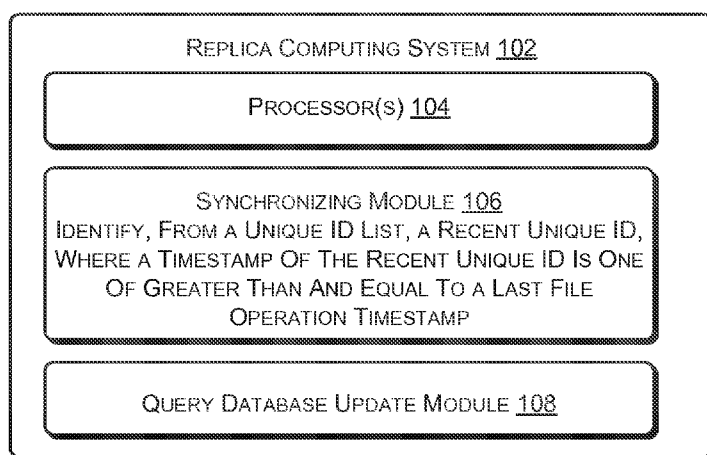


Fig. 1A

(57) **Abstract:** Replica computing system comprising a processor and a synchronizing module coupled to the processor to scan a replica query database associated with the replica computing system to identify a last file operation timestamp corresponding to a unique ID of a file whose file object attributes were recorded as a last updated entry in the replica query database. The synchronizing module identifies a recent unique ID from a unique ID list. A timestamp of the recent unique ID is one of greater than and equal to the last file operation timestamp. The unique ID list includes unique IDs of files updated in a source computing system and timestamps corresponding to each unique ID. A query database update module coupled to the processor updates the replica query database to include the recent unique ID and file object attributes having the most recent updates of file for the recent unique ID.



Published:

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

5

FILE SYSTEM REPLICATION

BACKGROUND

[0001] Increasing use of data processing and data generation in enterprises produces ever-increasing amounts of data which may be stored for short, medium, or long periods. Such data may be used by the organization for carrying out one or more operations. In order to ensure uninterrupted functioning, and to prevent any data loss owing to malfunctioning of computing systems managing such data, the enterprises deploy a network of computing systems.

15

BRIEF DESCRIPTION OF FIGURES

[0002] Fig. 1A illustrates a schematic diagram of an example replica computing system for file system replication, in accordance with an example of the present subject matter.

[0003] Fig. 1B illustrates various example components of a replica computing system and a source computing system for file system replication, in accordance with an example of the present subject matter.

[0004] Fig. 2 illustrates an example method for file system replication, in accordance with an example of the present subject matter.

[0005] Fig. 3 illustrates another example method for file system replication, in accordance with an example of the present subject matter.

[0006] Fig. 4 illustrates an example network environment implementing a non-transitory computer readable medium for file system replication, according to an example of the present subject matter.

30

DETAILED DESCRIPTION

[0007] Enterprises are nowadays deploying large data storage systems for storage and management of data generated and processed within the

5 enterprise. The data storage systems generally include a network of source computing systems and replica computing systems as a part of disaster recovery measures. A source computing system may be considered as a computing system that processes a variety of tasks and queries related to storage and management of the data. In some examples, a source computing
10 system is a computing system that is actively employed in the enterprises for implementing the data management and file storage systems. The replica computing systems are dormant computing systems deployed in the data storage systems. The replica computing system includes a replica of a file system of the source computing system. The replica computing system can be
15 used in place of the source computing system, whenever the source computing system is down or not available. In one example, replica computing systems are dormant computing systems in the network which can be used in place of the source computing system, in case the source computing system is down or not available, thereby ensuring seamless availability of stored data.

20 **[0008]** A file system may be understood as an abstraction implemented onto disk storage for organizing and storing data, stored as files. The file system may facilitate in controlling the manner in which the data, such as a file is stored, modified, or retrieved from the disk storage. Such file related actions may be carried out by one or more applications which may be deployed on a
25 computing system onto which the file system is implemented. Occurrence of any event that causes updates to files, such as creation of a new file or modification of an existing file may be captured and saved in an audit journal. Along with data related to occurrence of the event, file object attributes corresponding to the files that have been updated are also saved. The file
30 object attributes form metadata which in turn capture modifications made to the file. Generally, such file object attributes are obtained from the audit journal and saved in a query database. The query database is provided to facilitate a

5 quick access of the file object attributes by a user of the computing system. Further, a unique ID corresponding to each of the files within the file system, may be also saved in the query database. In such a case, the unique ID is mapped to each of its corresponding file object attributes.

[0009] Further, to keep data at the replica computing system up-to-date
10 with the source computing system, all updates made to data stored at the source computing system are periodically replicated onto the replica computing system. For instance, all newly created files and modifications made to existing files may be replicated from the source computing system to the replica computing system. Generally, snapshots of the file system and, including the
15 query database, of the source computing system are periodically generated and transmitted to the replica computing system. The snapshots may be used for synchronizing the replica computing system with the source computing system. The replica computing system may therefore include the latest file updates whenever the replica computing system is requested to become active
20 in place of the source computing system.

[0010] As explained, the file object attributes corresponding to the files that have been updated are first saved in the audit journal before being moved to the query database. In such a case, it may happen that some of the file object attributes may not get captured in the query database in the snapshot currently
25 being taken. Although, such file object attributes may get captured in a subsequent snapshot, a synchronization error may occur in case the replica computing system becomes active for replacing the source computing system before such subsequent snapshot is generated. The query database in the replica computing system may thus not include the file object attributes that
30 were not captured during the last snapshot. As a result, the file system at the replica computing system may incorrectly consider that the file object attributes corresponding to latest update, have been captured in the query database. This

5 occurs as the file system of the source computing system would have indicated that these file object attributes had been captured in audit journal before the current snapshot was generated.

[0011] Omission of such file object attributes may result in data loss, making the replica computing system out of sync with the source computing
10 system. The file system at the replica computing system would thus have to be entirely scanned to identify all such file object attributes that are indicated as being captured in the query database in the source computing system but have actually not been captured in the query database in the replica computing system. Scanning the entire file system would result in an increase in
15 computational sources and time utilized before the replica computing system is actively deployed.

[0012] Approaches for file system replication are described. The present subject matter facilitates in preventing data loss during file system replication. In one example, a replica computing system is dynamically updated before
20 being actively deployed in a communication network for replacing a source computing system based on a unique identifier (ID) list for each snapshot generated at the source computing system. The unique ID list along with the snapshot may be shared with the replica computing system.

[0013] The unique ID list (as described above) may include a list of
25 unique IDs of files updated at the source computing system, and corresponding timestamps indicating a time instant at which the files were updated. The replica computing system may use the unique ID list to identify the files for which file object attributes are either not present in a replica query database of the replica computing system or have not been updated. The replica computing
30 system may accordingly update the replica query database. Using the unique ID list to identify file object attributes that are either missing or are not updated

5 facilitates in minimizing computational sources and time utilized for synchronizing the replica computing system.

[0014] In accordance to an example of the present subject matter, a source query database at the source computing system is updated to include file object attributes of files which have been updated during a file operation. In
10 one example, the file operation may be one of creation of a new file, modification of an existing file, and deletion of an existing file. Further, a unique ID list is updated to include a unique ID of the file being updated and a timestamp corresponding to the unique ID. The timestamp indicates a time instant at which the file was updated.

15 **[0015]** Subsequently a snapshot of a source file system of the source computing system is generated and transmitted to the replica computing system. The snapshot may be saved by the replica computing system for creating a replica of the source file system. In one example, the snapshot includes files saved in the source file system, the source query database, and
20 the unique ID list. The replica computing system may thus have a replica file system which will be a replica of the source file system. Further, the replica file system may have the replica query database which will be a replica of the source query database.

[0016] The replica computing system may use the unique ID list to
25 synchronize the replica computing system with a state of the source computing system as it existed at the time of generation of the snapshot. Initially a last file operation timestamp is obtained from the replica query database of the replica computing system. In one example, the last file operation timestamp is a timestamp corresponding to a unique ID of a file whose file object attributes
30 were recorded as a last updated entry in the replica query database before the snapshot was generated.

- 5 **[0017]** Subsequently, a unique ID having a timestamp greater than or equal to the last file operation timestamp is identified from the unique ID list. As previously described, the unique ID corresponds to a file that has been updated at the source computing system. The corresponding timestamp indicates a time instant at which the file was updated. Upon identifying the
- 10 unique ID, the replica file system is queried to determine if the file, and the file object attributes corresponding to the identified unique ID, are present in the replica file system. The file object attributes corresponding to the file and the file object attributes subsequently obtained and used to update the replica query database.
- 15 **[0018]** Further, if the file and the file object attributes not present in the replica file system, it is determined that the file was deleted at the source computing system. The unique ID, the file object attributes corresponding to the file, and a timestamp corresponding to the unique ID are thus deleted from the replica query database.
- 20 **[0019]** The present subject matter thus facilitates in preventing data loss during file system replication. As will be understood based on the above description, the data loss is prevented by using the unique ID and the corresponding timestamp to identify the files for which the most recent file object attributes are not stored in the replica query database. The unique ID
- 25 and the corresponding timestamp are obtained from the unique ID list provided by the source computing system. The unique ID list is updated every time a file is updated, irrespective of whether the updates have been captured in the source query database or not. The unique ID list thus includes unique IDs for all the files that were updated before the snapshot was generated for being
- 30 transmitted to the replica computing system. Using the unique ID list to identify missing or old versions of file object attributes for files updated at the source computing system ensures that the replica computing system is updated and

5 synchronized with the source computing system. Further, using the unique ID list to identify missing or old versions of file object attributes instead of scanning each entry of the replica query database and the replica file system helps in minimizing computational sources and time utilized for synchronizing the replica computing system.

10 **[0020]** The systems and methods are further described in conjunction with Fig. 1 to Fig. 4, as examples of the present subject matter. It should be understood that the description and figures merely illustrate the principles of the present subject matter. It will thus be appreciated that various arrangements that embody the principles of the present subject matter,
15 although not explicitly described or shown herein, can be devised from the description and are included within its scope. Furthermore, all examples recited herein are for pedagogical purposes to aid the reader in understanding the principles of the present subject matter. Moreover, all statements herein reciting principles, aspects, and examples of the present subject matter, as well
20 as specific examples thereof, are intended to encompass equivalents thereof.

[0021] Fig. 1A illustrates a schematic diagram of an example replica computing system 102 for file system replication, in accordance with an example of the present subject matter. The replica computing system 102 is hereinafter interchangeably referred to as system 102. The system 102 may
25 be implemented in, for example, desktop computers, multiprocessor systems, personal digital assistants (PDAs), laptops, network computers, cloud servers, mainframe computers, and computing based devices in general. The system 102 may also be hosting a plurality of applications. The system 102 may further be implemented in a networked environment (not shown in the figure).

30 **[0022]** The system 102 may include, for example, processor(s) 104, a synchronizing module 106 communicatively coupled to the processor(s) 104,

5 and a query database update module 108 communicatively coupled to the processor(s) 104.

[0023] The processor(s) 104 may include microprocessors, microcomputers, microcontrollers, digital signal processors, central processing units, state machines, logic circuitries, and/or any other devices that manipulate
10 signals and data based on computer-readable instructions. Further, functions of the various elements shown in the figures, including any functional blocks labeled as "processor(s)", may be provided through the use of dedicated hardware as well as hardware capable of executing computer-readable instructions.

15 **[0024]** In operation, the synchronizing module 106 may scan a replica query database (not shown in this figure) of the replica computing system 102 to identify a last file operation timestamp. In one example, the last file operation timestamp is a timestamp corresponding to a unique Identifier (ID) of a file whose file object attributes were recorded as a last updated entry in the replica
20 query database.

[0025] The synchronizing module 106 subsequently identifies a recent unique ID from a unique ID list. In one example, the unique ID list includes unique IDs of files updated in a source computing system (not shown in this figure) and timestamps corresponding to each unique ID. The corresponding
25 timestamp indicates a time instant at which the file was updated at the source computing system. Further, the recent unique ID is identified such that a timestamp of the recent unique ID is one of greater than and equal to the last file operation timestamp.

[0026] The query database update module 108 may subsequently
30 update the replica query database to include the recent unique ID and file object attributes having the most recent updates for a file corresponding to the recent unique ID.

5 **[0027]** Fig. 1B illustrates various example components of the replica computing system 102 and a source computing system 110 for file system replication, in accordance with an example of the present subject matter. Each of the replica computing system 102 and the source computing system 110 includes processor(s), interface(s), memory 112, modules, and data. For instance, the replica computing system 102 includes the processor(s) 104-1, interface(s) 112-1, memory 114-1, modules 116-1, and data 118-1. The source computing system 110 includes the processor(s) 104-2, interface(s) 112-2, memory 114-2, modules 116-2, and data 118-2.

15 **[0028]** The interfaces 112-1 and 112-2 are hereinafter collectively referred to as the interfaces 112. The interfaces 112 may include a variety of commercially available interfaces, for example, interfaces for peripheral device(s), such as data input output devices, referred to as I/O devices, interface cards, storage devices, and network devices.

20 **[0029]** The memory 114-1 and 114-2 is hereinafter collectively referred to as the memory 114. The memory 114 may be communicatively coupled to the processor(s) 104 and may include any non-transitory computer-readable medium known in the art including, for example, volatile memory, such as static random access memory (SRAM) and dynamic random access memory (DRAM), and/or non-volatile memory, such as read only memory (ROM), erasable programmable ROM, flash memories, hard disks, optical disks, and magnetic tapes. In one example, the memory 114 may include a storage device and main memory. For instance, the replica computing system 102 includes storage device 120-1 and main memory 122-1. The source computing system 110 includes storage device 120-2 and main memory 122-2.

30 **[0030]** The storage device 120-1 and 120-2 are hereinafter collectively referred to as the storage device 120. The storage device 120, such as hard disks and magnetic tapes may be used for storing content, such as files in the

5 replica computing system 102 and the source computing system 110. The main memory 122-1 and 122-2 are hereinafter referred to as the main memory 122. The main memory 122, such as RAM may be used for temporary storage of content for processing by the replica computing system 102 and the source computing system 110.

10 **[0031]** The modules 116-1 and 116-2 are hereinafter collectively referred to as the modules 116. The modules 116, amongst other things, include routines, programs, objects, components, and data structures, which perform particular tasks or implement particular abstract data types. The modules 116 may also be implemented as, signal processor(s), state machine(s), logic
15 circuitries, and/or any other device or component that manipulate signals based on operational instructions. Further, the modules 116 can be implemented by hardware, by computer-readable instructions executed by a processing unit, or by a combination thereof.

[0032] The modules 116-1 of the replica computing system 102 may
20 include a communication module 124, the synchronizing module 106, the query database update module 108, and other module(s) 126. The other modules 126 may include programs or coded instructions that supplement applications and functions, for example, programs in an operating system of the replica computing system 102. Further, the data 118-1 may include a replica unique ID
25 list 128, replica file data 130, replica query database 132, and other data 134.

[0033] The modules 116-2 of the source computing system 110 may include a file system update module 136, an archive journal scanner 138, an interaction module 140, and other module(s) 142. The other modules 142 may include programs or coded instructions that supplement applications and
30 functions, for example, programs in an operating system of the replica computing system 102. Further, the data 118-2 may include a source unique

5 ID list 144, source file data 146, source query database 148, and other data 140.

[0034] As previously described, the present subject matter facilitates in minimizing resources and time utilized in synchronizing the replica computing system 102 with the source computing system 110. The subject matter further facilitates in preventing data loss during such synchronization. In one example, the replica computing system 102 uses the unique ID list provided by the source computing system 110 for synchronization. The source computing system 110, in one example, may update the unique ID list each time a file is updated during a file operation.

15 **[0035]** In one example, the file system update module 136 may ascertain the occurrence of the file operation. Examples of the file operation include, but are not limited to, creation of a new file, modification of an existing file, and deletion of an existing file. The file system update module 136 may subsequently update an audit journal of the source file system to include the file operation along with file object attributes corresponding to the files that have been updated. As previously described, the file object attributes of a file are metadata capturing the modifications made to the file. In one example, the source file system and the audit journal of the source file system are stored in the storage device 120-2.

25 **[0036]** Further, the file system update module 136 may obtain the unique ID corresponding to the file that has been updated during the file operation. The file system update module 136 may then update the unique ID list to include the unique ID and the corresponding timestamp indicating a time instant at which the file was updated. Further, in case no unique ID list exists at the time of occurrence of the file operation, the file system update module 136 may create a new unique ID list and save the unique ID and the corresponding timestamp in the unique ID list. For instance, in case a snapshot

30

5 had been taken just before the occurrence of the file operation, the unique ID list existing at that would have been shared with the replica computing system 102 and the source computing system 110 would thus not have any unique ID list. The file system update module 136 in such a case may create a new unique ID list and save the unique ID and the corresponding timestamp in the unique ID list. In one example, the unique ID list is saved in the source unique ID list 144.

[0037] Further, the audit journal is periodically processed by the archive journal scanner 138 to obtain data related to the file operation and the corresponding file object attributes. The archive journal scanner 138 may save the data related to the file operation and the corresponding file object attributes in the source query database 148. The archive journal scanner 138 may further save the unique ID corresponding to the file in the source query database 148.

[0038] Subsequently, the file system update module 136 may generate a snapshot of the source file system. In one example, the snapshot of the source file system may include source file data 146, i.e., files saved in the source file system, the source query database 148, and the unique ID list. The snapshot is then transmitted to the replica computing system 102 by the interaction module 140. In one example, the snapshot is received by the communication module 124 of the replica computing system 102. The communication module 124 may save the snapshot in the replica file data 130. As the snapshot is saved in the replica computing system 102, the replica query database 132 and the replica file system are dynamically updated to replicate the source query database 148 and the source file system. Further the unique ID list is saved in the replica unique ID list 132.

30 **[0039]** In one example, the synchronizing module 106 may use the unique ID list to synchronize the replica computing system 102 with the source computing system 110 upon receiving a synchronization request. The

5 synchronization request may be either an activation indication indicating failover of the source computing system 110 or a snapshot update indication indicating receipt of the snapshot. On receiving the synchronization request, the synchronizing module 106 may scan the replica query database 132 to identify a last file operation timestamp.

10 **[0040]** The last file operation timestamp is a timestamp corresponding to a unique ID of a file whose file object attributes were recorded as a last updated entry in the replica query database 132. The last file operation timestamp may thus indicate a time instant at which a last update was made to the source query database 148 before the snapshot was generated at the
15 source computing system 110. In one example, the last file operation timestamp may correspond to a unique ID for which a new record was created at the time instant indicated by the last file operation timestamp. In another example, the last file operation timestamp may correspond to a unique ID for which an existing record was updated at the time instant indicated by the last
20 file operation timestamp.

[0041] The synchronizing module 106 may obtain the last file operation timestamp to identify a recent unique ID from the unique ID list. In one example, the recent unique ID is a unique ID having a timestamp greater than or equal to the last file operation timestamp. The recent unique ID may thus be defined
25 as unique ID corresponding to a file that was updated after the last update of the source query database 148 and before the generation of the snapshot at the source computing system 110. The recent unique ID and the timestamp corresponding to the file are thus updated in the unique ID list, however, most recent file object attributes corresponding to the recent unique ID are missing
30 from the replica query database 132.

[0042] The synchronizing module 106 may initially compare the last file operation timestamp with the timestamp corresponding to each unique ID

5 present in the unique ID list to determine the recent unique ID. Each unique ID having a timestamp greater than or equal to the last file operation timestamp is identified as the recent unique ID. Further, the unique IDs having a timestamp less than the last file operation timestamp are determined to be updated. Such unique IDs are determined to be corresponding to files for which the most
10 recent file object attributes are present in the replica query database 132. In case the last file operation timestamp is greater than all timestamps for all unique IDs present in Unique ID list, the synchronizing module 106 may conclude that the replica computing system 102 is synchronized.

[0043] Upon identifying the recent unique ID, the query database update
15 module 108 may query the replica file system to determine if the file object attributes of the file corresponding to the recent unique ID are present in the replica file system. If the file corresponding to the unique ID is present in the replica file system, the query database update module 108 may obtain the file object attributes of the file corresponding to the unique ID. The query database
20 update module 108 may further update the replica query database 132 to include the file object attributes having the most recent updates for the unique ID.

[0044] In one example, updating the replica query database 132 may include either updating existing file storage attributes or creating a new entry
25 for the recent unique ID and the file object attributes. For instance, in case the recent unique ID and file object attributes of the file corresponding to the recent unique ID are present in the replica query database 132, the query database update module 108 may update the existing file object attributes for the recent unique ID. In case the recent unique ID and file object attributes of the file
30 corresponding to the recent unique ID are not present in the replica query database 132, the query database update module 108 may add the file object attributes for the recent unique ID in the replica query database.

5 **[0045]** Further, if the file and file object attributes corresponding to the recent unique ID are not present in the replica file system, the query database update module 108 may determine that the file has been deleted in the source computing system 110. The recent unique ID and the corresponding timestamp in the unique ID list may thus correspond to the file operation related to deletion
10 of the file from the source file system 110. The query database update module 108 may thus delete the recent unique ID and the last updated timestamp from the replica query database 132.

[0046] Figs. 2 and 3 illustrate example methods 200 and 300, respectively, for file system replication, in accordance with an example of the
15 present subject matter. The order in which the methods are described is not intended to be construed as a limitation, and any number of the described method blocks may be combined in any order to implement the aforementioned methods, or an alternative method. Furthermore, the methods 200 and 300 may be implemented by processing resource or computing device(s) through
20 any suitable hardware, non-transitory machine readable instructions, or combination thereof.

[0047] It may also be understood that the methods 200 and 300 may be performed by programmed computing devices, such as the source computing system 110 and the replica computing system 102, respectively as depicted in
25 Fig. 1B. Furthermore, the methods 200 and 300 may be executed based on instructions stored in a non-transitory computer readable medium, as will be readily understood. The non-transitory computer readable medium may include, for example, digital memories, magnetic storage media, such as one or more magnetic disks and magnetic tapes, hard drives, or optically readable
30 digital data storage media.

[0048] Further, the methods 200 and 300 are described below with reference to the source computing system 110 and the replica computing

5 system 102 as described above, other suitable systems for the execution of these methods can be utilized. Additionally, implementation of these methods is not limited to such examples.

[0049] Referring to Fig. 2, at block 202, a source query database is updated to include file object attributes of a file updated during a file operation.
10 In one example, the source query database is associated with a source computing system, such as the source computing system 110. Examples of the file operation include, but are not limited to, creation of a new file, modification of an existing file, and deletion of an existing file.

[0050] At block 204, a unique Identifier (ID) list is updated to include a
15 unique ID of the file being updated and a timestamp corresponding to the unique ID. The timestamp indicates a time instant at which the file was updated. In one example, the unique ID list is updated at the source computing system 110 by a file system update module, for example, the file system update module 136 of the source computing system 110.

20 **[0051]** At block 206, a snapshot of a source file system is transmitted to a replica computing system for creating a replica of the source query database. In one example, the snapshot includes files saved in the source file system, the source query database, and the unique ID list. Further the snapshot is transmitted by the source computing system 110 to the replica computing
25 system, for example, the replica computing system 102.

[0052] Referring to Fig. 3, at block 302, a snapshot is obtained by a replica computing system, for example, the replica computing system 102. In one example, the snapshot is received from a source computing system, for example, the source computing system 110. The snapshot may include files
30 saved in a source file system, a source query database, and a unique Identifier (ID) list. The unique ID list includes unique IDs of files updated in the source computing system and timestamps corresponding to each unique ID.

5 **[0053]** At block 304, a replica query database associated with the replica computing system is scanned to identify a last file operation timestamp. The last file operation timestamp is a timestamp corresponding to a unique ID of a file whose file object attributes were recorded as a last updated entry in the replica query database. In one example, the last file operation timestamp is
10 obtained upon receiving a synchronization request.

[0054] At block 306, a recent unique ID is identified from a unique ID list. In one example, a synchronizing module of the replica computing system identifies the recent unique ID such that a timestamp of the recent unique ID is one of greater than and equal to the last file operation timestamp.

15 **[0055]** At block 308, it is determined whether a file and file object attributes corresponding to the recent unique ID are present in the replica file system. If in case it is determined that the file and the file object attributes corresponding to the recent unique ID are not present in the replica file system, ('No' path from block 308), the recent unique ID, the file object attributes, and
20 a timestamp corresponding to the unique ID are deleted from the replica query database at block 310.

[0056] If in case it is determined that the file and the file object attributes corresponding to the file are present in the replica file system ('Yes' path from block 308), the file object attributes having most recent updates of the file
25 corresponding to the recent unique ID are obtained from the replica file system at block 312.

[0057] At block 314, the replica query database is updated to include the recent unique ID and file object attributes having the most recent updates for a file corresponding to the recent unique ID. In case the recent unique ID and file
30 object attributes of the file corresponding to the recent unique ID are present in the replica query database, an existing file object attributes for the recent unique ID is updated. In case the recent unique ID and file object attributes of

5 the file corresponding to the recent unique ID are not present in the replica query database, the file object attributes for the recent unique ID are added in the replica query database.

[0058] Fig. 4 illustrates an example network environment implementing a non-transitory computer readable medium for file system replication, according to an example of the present disclosure. The system environment 10 400 may comprise at least a portion of a public networking environment or a private networking environment, or a combination thereof. In one implementation, the system environment 400 includes a processing resource 402 communicatively coupled to a computer readable medium 404 through a 15 communication link 406.

[0059] For example, the processing resource 402 can include one or more processors of a computing device for file system replication. The computer readable medium 404 can be, for example, an internal memory device of the computing device or an external memory device. In one 20 implementation, the communication link 406 may be a direct communication link, such as any memory read/write interface. In another implementation, the communication link 406 may be an indirect communication link, such as a network interface. In such a case, the processing resource 402 can access the computer readable medium 404 through a network 408. The network 408 may 25 be a single network or a combination of multiple networks and may use a variety of different communication protocols.

[0060] The processing resource 402 and the computer readable medium 404 may also be coupled to requested data sources 410 through the communication link 406, and/or to communication devices 412 over the 30 network 408. The coupling with the requested data sources 410 enables in receiving the requested data in an offline environment, and the coupling with

5 the communication devices 412 enables in receiving the requested data in an online environment.

[0061] In one implementation, the computer readable medium 404 includes a set of computer readable instructions, implementing a synchronizing module 414 and a query database update module 416. The set of computer readable instructions can be accessed by the processing resource 402 through
10 the communication link 406 and subsequently executed to process requested data communicated with the requested data sources 410 in order to facilitate file system journaling. When executed by processing resource 402, the instructions of the synchronizing module 414 may perform the functionalities
15 described above in relation to the synchronizing module 106. When executed by processing resource 402, the instructions of the query database update module 416 may perform the functionalities described above in relation to the query database update module 108.

[0062] For example, in response to a synchronization request, the
20 synchronizing module 414 may scan a replica query database associated with a replica computing system to identify a last file operation timestamp. The last file operation timestamp is a timestamp corresponding to a unique Identifier (ID) of a file whose file object attributes were recorded as a last updated entry in the replica query database.

[0063] The synchronizing module 414 may subsequently identify a
25 recent unique ID from a unique ID list based on the last file operation timestamp and a timestamp of the recent unique ID. The timestamp of the recent unique ID is one of greater than and equal to the last file operation timestamp. The unique ID list includes unique IDs and timestamps of files updated in a source
30 computing system. The unique ID is associated with a file that has been updated at a source computing system, while the timestamp indicates a time instant at which the file was updated.

5 **[0064]** Further, the query database update module 416 may query a replica file system of the replica computing system to determine if the file and the file object attributes corresponding to the recent unique ID are present in the replica file system. If the file and the file object attributes are present in the replica file system, the query database update module 416 may obtain the file
10 object attributes corresponding to the file. The query database update module 416 may subsequently update the replica query database to include the file object attributes having the most recent updates for the recent unique ID.

[0065] If the file and the file object attributes are not present in the replica file system, the query database update module 416 may determine the file to
15 be deleted. The query database update module 416 may delete the recent unique ID, the file object attributes corresponding to the file, as saved in the replica query database, and the last updated timestamp, from the replica query database.

[0066] Although examples for the present disclosure have been
20 described in language specific to structural features and/or methods, it should be understood that the appended claims are not necessarily limited to the specific features or methods described. Rather, the specific features and methods are disclosed and explained as examples of the present disclosure.

5 I/We claim:

1. A replica computing system comprising:

a processor;

a synchronizing module coupled to the processor to:

10 scan a replica query database associated with the replica computing system to identify a last file operation timestamp, wherein the last file operation timestamp is a timestamp corresponding to a unique Identifier (ID) of a file whose file object attributes were recorded as a last updated entry in the replica query database; and

15 identify, from a unique ID list, a recent unique ID, wherein a timestamp of the recent unique ID is one of greater than and equal to the last file operation timestamp, wherein the unique ID list includes unique IDs of files updated in a source computing system and timestamps corresponding to each unique ID; and

a query database update module coupled to the processor to:

20 update the replica query database to include the recent unique ID and file object attributes having the most recent updates for a file corresponding to the recent unique ID.

2. The computing system as claimed in claim 1, wherein the query database
25 update module further is to query a replica file system of the replica computing system to obtain the file object attributes having the most recent updates for the file corresponding to the recent unique ID.

3. The computing system as claimed in claim 2, wherein the query database
30 update module further is to:

for the file and the file object attributes not present in the replica file system, determine the file to be deleted; and

5 delete, from the replica query database, the recent unique ID, the file object attributes corresponding to the file, and a timestamp corresponding to the recent unique ID.

4. The computing system as claimed in claim 1, wherein the synchronizing
10 module further is to:

compare the last file operation timestamp with the timestamp corresponding to each unique ID present in the unique ID list to determine the recent unique ID.

15 5. A method comprising:

updating, at a source computing system, a source query database to include file object attributes of a file updated during a file operation;

updating a unique Identifier (ID) list to include a unique ID of the file being updated and a timestamp corresponding to the unique ID, wherein the
20 timestamp indicates a time instant at which the file was updated; and

transmitting a snapshot of a source file system, to a replica computing system for creating a replica of the source file system and the source query database, wherein the snapshot includes files saved in the source file system, the source query database, and the unique ID list.

25

6. The method as claimed in claim 5 further comprising:

generating the snapshot of the source file system, wherein the snapshot includes files saved in the source file system, the source query database, and the unique ID list since generation of a previous snapshot.

30

7. The method as claimed in claim 5 further comprising:

5 obtaining, at the replica computing system, the snapshot from the
source computing system; and
 saving the snapshot in the replica computing system.

8. The method as claimed in claim 7 further comprising:

10 at the replica computing system, scanning a replica query database of
the replica computing system to identify a last file operation timestamp; wherein
the last file operation timestamp is a timestamp corresponding to a unique
Identifier (ID) of a file whose file object attributes were recorded as a last
updated entry in the replica query database;

15 identifying a recent unique ID from the unique ID list based on the last
file operation timestamp and a timestamp corresponding to the recent unique
ID, wherein the last file operation timestamp is less than the timestamp
corresponding to the recent unique ID; and

 updating the replica query database to include the file object attributes
20 obtained from a replica file system of the replica computing system, having the
most recent updates for the recent unique ID.

9. The method as claimed in claim 8, wherein the updating comprises one of:

 for the unique ID being present in the replica query database, updating
25 existing file object attributes for the recent unique ID; and

 for the unique ID not present in the replica query database, adding the
file object attributes for the recent unique ID in the replica query database.

10. The method as claimed in claim 8, wherein the updating the replica query
30 database further comprising:

5 querying a replica file system of the replica computing system to determine if the file and the file object attributes corresponding to the file are present in the replica file system; and

 obtaining the file object attributes corresponding to the file if the file and the file object attributes are present in the replica file system.

10

11. The method as claimed in claim 8 further comprising;

 querying a replica file system of the replica computing system to determine if the file and the file object attributes corresponding to the file are present in the replica file system;

15 for the file and the file object attributes not present in the replica file system, determining the file to be deleted; and

 deleting, from the replica query database, the recent unique ID, the file object attributes corresponding to the file, and a timestamp corresponding to the unique ID.

20

12. The method as claimed in claim 8, wherein identifying the unique ID having the timestamp greater than the last file operation timestamp further comprises
 comparing the last file operation timestamp with each unique ID present in the unique ID list.

25

13. A non-transitory computer readable medium having a set of computer readable instructions that, when executed, cause a processor to:

 scan a replica query database associated with a replica computing system to identify a last file operation timestamp; wherein the last file operation
30 timestamp is a timestamp corresponding to a unique Identifier (ID) of a file whose file object attributes were recorded as a last updated entry in the replica query database; and

5 identify, from a unique ID list, a recent unique ID, wherein a timestamp of the recent unique ID is one of greater than and equal to the last file operation timestamp, wherein the unique ID list includes unique IDs of files updated in a source computing system and timestamps corresponding to each unique ID;
 query a replica file system of the replica computing system to determine
10 if file and file object attributes corresponding to the recent unique ID are present in the replica file system;
 for the file and the file object attributes not present in the replica file system, determine the file to be deleted in a source computing system; and
 delete, from the replica query database, the unique ID, the file object
15 attributes corresponding to the file, and a timestamp corresponding to the recent unique ID.

14. The non-transitory computer readable medium of claim 13, wherein the
computer readable instructions, when executed, further cause the processor
20 to:

 obtain the file object attributes corresponding to the file if the file and the
file object attributes are present in the replica file system; and
 update the replica query database to include the file object attributes
obtained from the replica file system, having the most recent updates for the
25 recent unique ID.

15. The non-transitory computer readable medium of claim 13, wherein the
computer readable instructions, when executed, further cause the processor
to:

30 obtain, at the replica computing system, a snapshot from the source
computing system for synchronizing the replica computing system with the
source computing system; and

5 save the snapshot in the replica computing system.

1/5

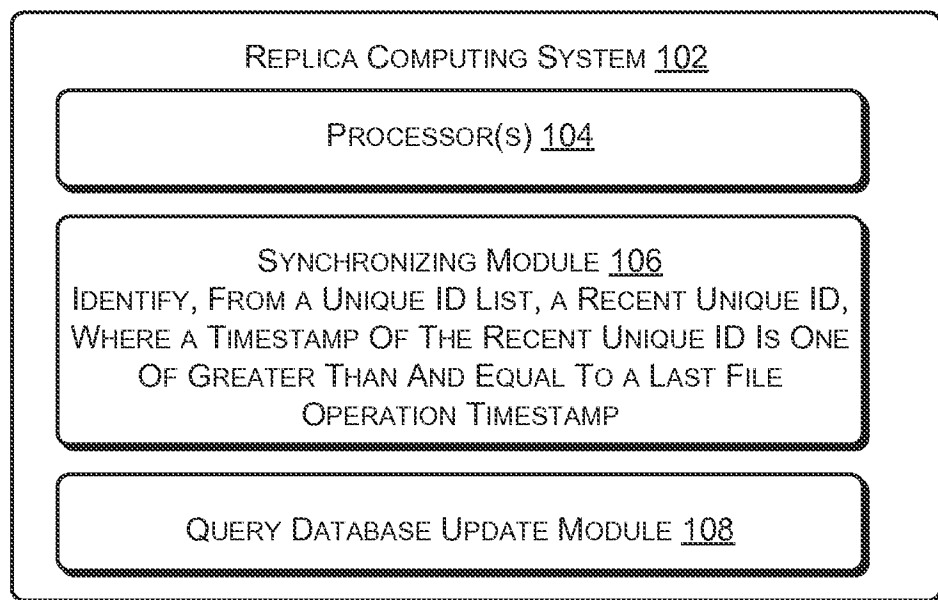


Fig. 1A

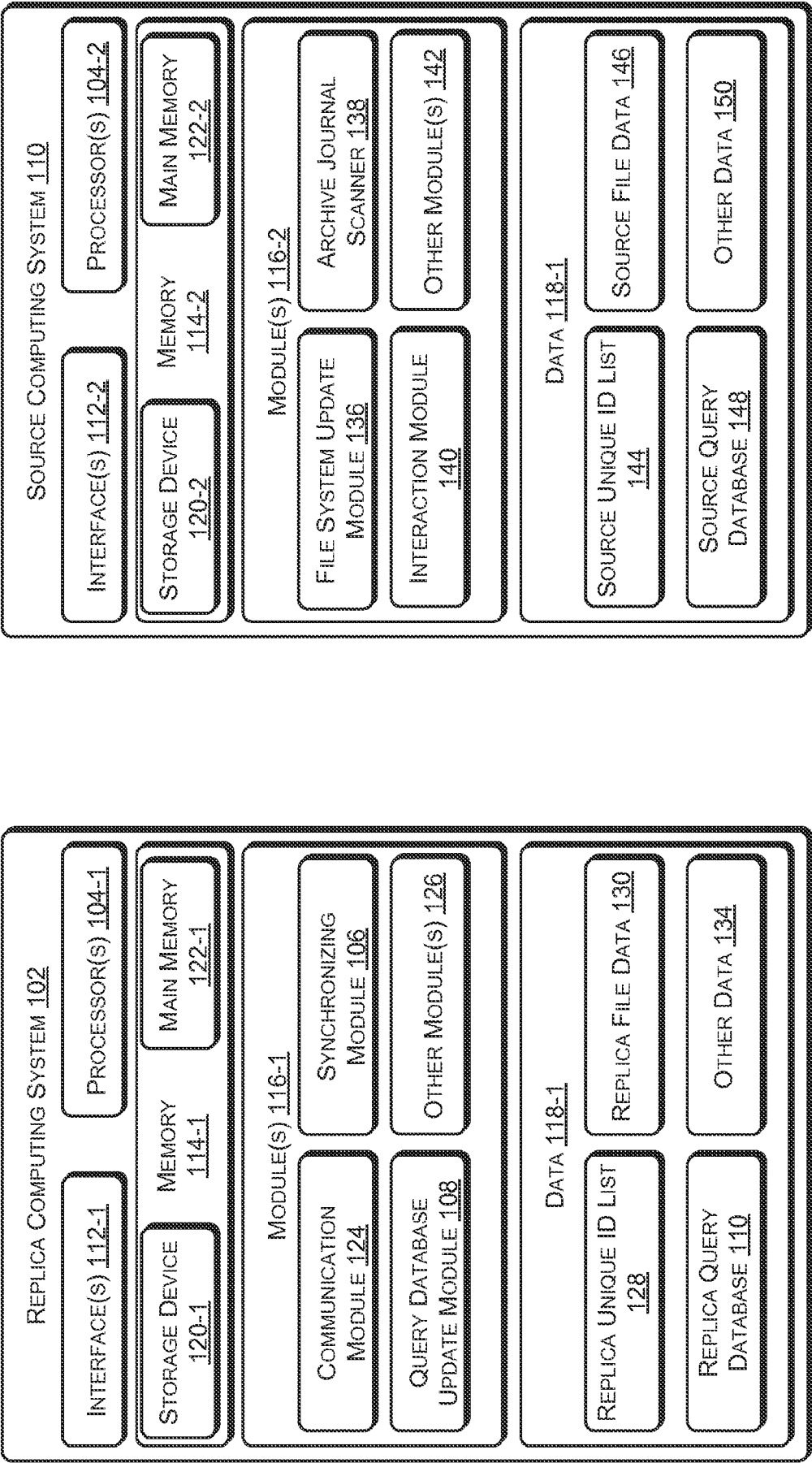


Fig. 1B

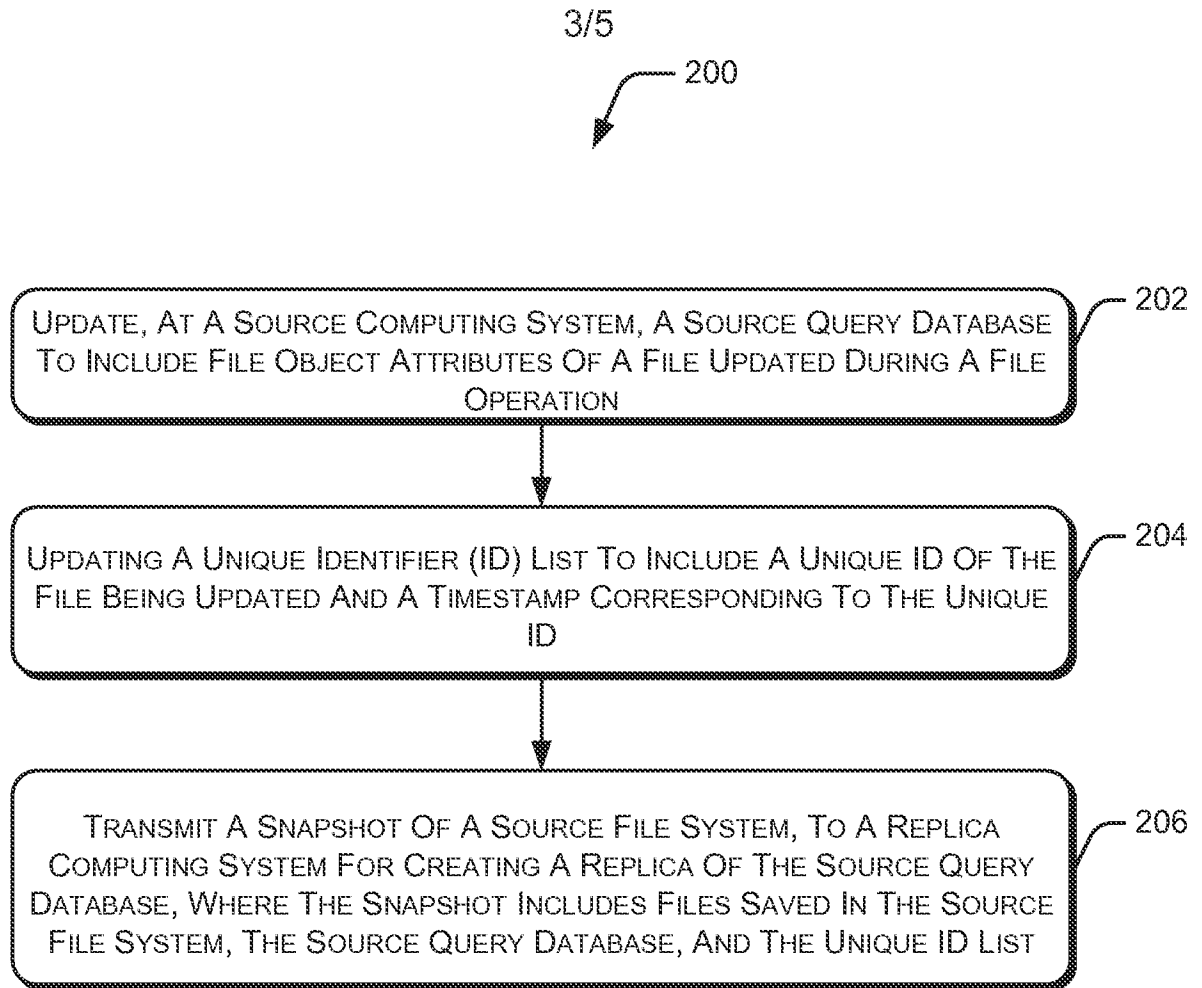


Fig. 2

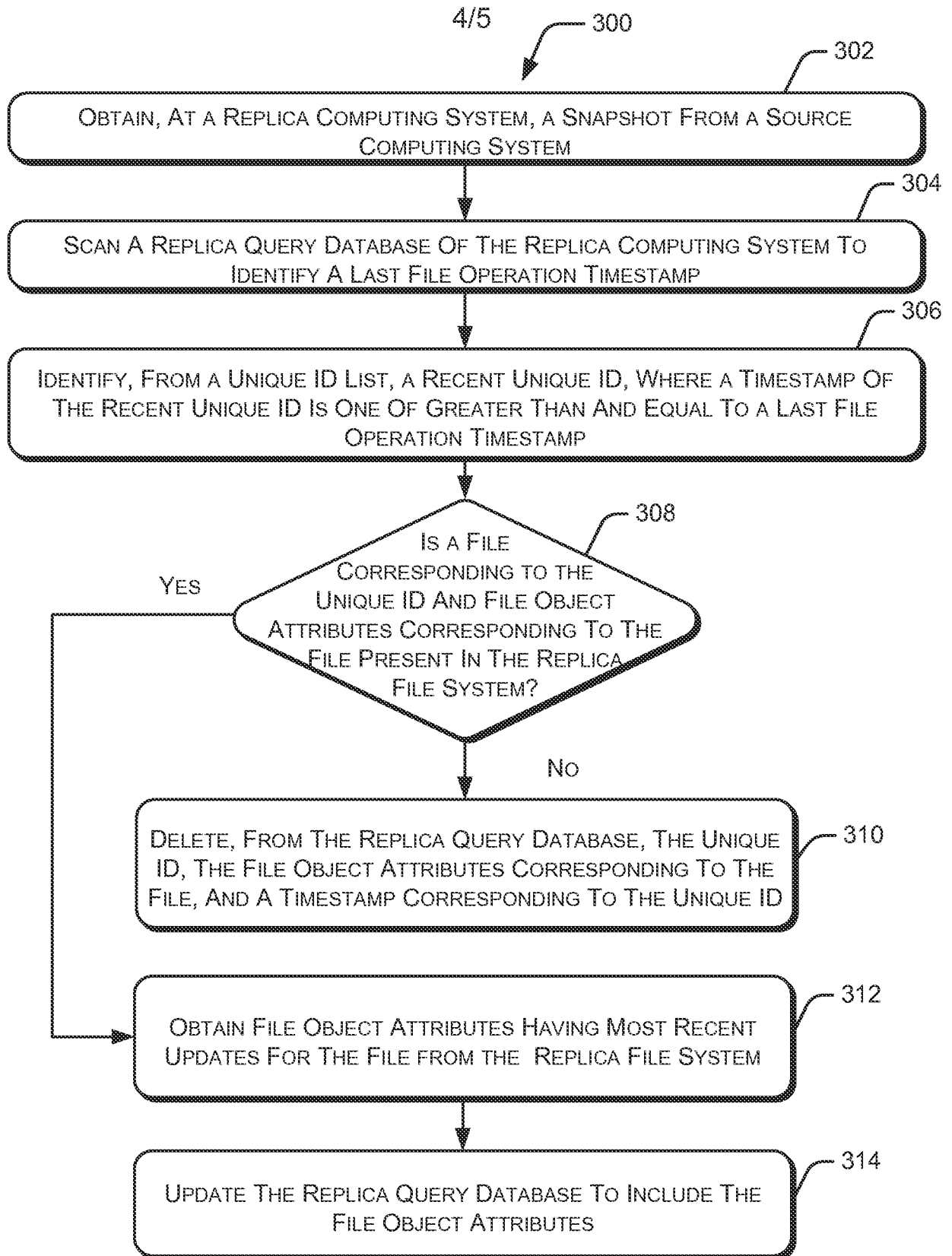


Fig. 3

5/5

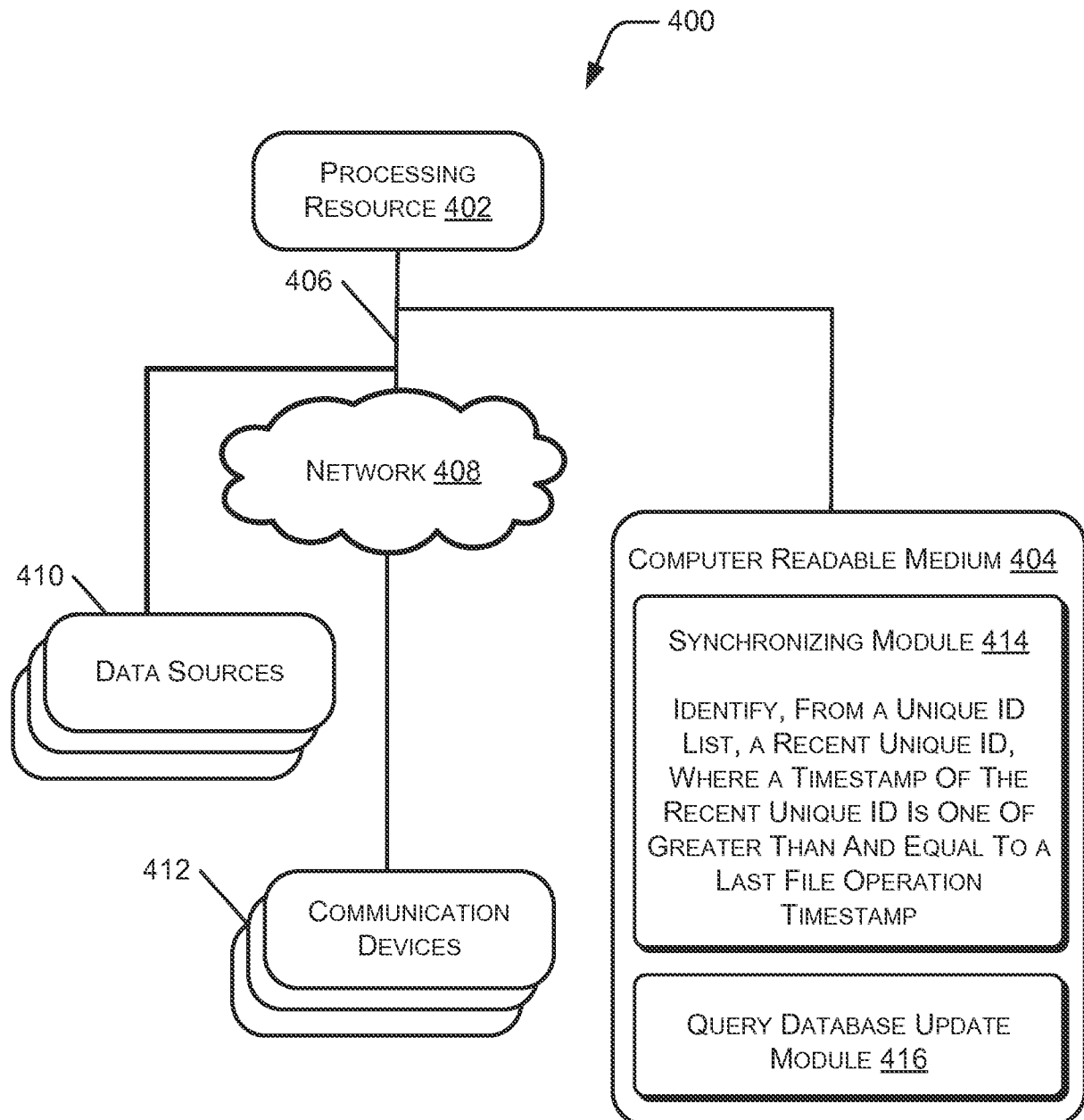


Fig. 4

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2015/023649**A. CLASSIFICATION OF SUBJECT MATTER****G06F 17/30(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 15/16; G06F 3/06; G06F 17/00; G06Q 10/00

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: replica database, timestamp, snapshot, ID, comparing, update, 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.
X	US 2007-0255763 A1 (MARC BEYERLE et al.) 01 November 2007 See paragraphs [0048]-[0049] and [0052]; claims 1-2; and figure 2.	5-7
A		1-4, 8-15
A	US 2006-0168120 A1 (JEFFREY B. PARHAM) 27 July 2006 See paragraphs [0045] and [0053]-[0056]; claims 1-2; and figure 7.	1-15
A	JP 2009-116722 A (NEC CORP.) 28 May 2009 See paragraphs [0038]-[0045] and figure 5.	1-15
A	US 8,849,777 B1 (JIAN XING) 30 September 2014 See column 5, line 63 - column 6, line 27; and figures 5-6.	1-15
A	US 2009-0157537 A1 (BARRICK H. MILLER) 18 June 2009 See paragraphs [0096]-[0102] and figure 14.	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

31 August 2015 (31.08.2015)

Date of mailing of the international search report

01 September 2015 (01.09.2015)

Name and mailing address of the ISA/KR

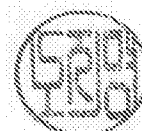
International Application Division
Korean Intellectual Property Office
189 Cheongsu-ro, Seo-gu, Daejeon Metropolitan City, 35208,
Republic of Korea

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/US2015/023649

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2007-0255763 A1	01/11/2007	None	
US 2006-0168120 A1	27/07/2006	US 2003-0101258 A1 US 7035922 B2 US 7519736 B2	29/05/2003 25/04/2006 14/04/2009
JP 2009-116722 A	28/05/2009	JP 4888347 B2	29/02/2012
US 8849777 B1	30/09/2014	US 2015-046401 A1	12/02/2015
US 2009-0157537 A1	18/06/2009	US 8468211 B2	18/06/2013