



- (51) **International Patent Classification:**
G06F 12/06 (2006.01)
- (21) **International Application Number:**
PCT/US2014/060284
- (22) **International Filing Date:**
13 October 2014 (13.10.2014)
- (25) **Filing Language:** English
- (26) **Publication Language:** English
- (30) **Priority Data:**
3522/CHE/2014 17 July 2014 (17.07.2014) IN
- (71) **Applicant:** HEWLETT-PACKARD DEVELOPMENT COMPANY, L.P. [US/US]; 11445 Compaq Center Drive W., Houston, Texas 77070 (US).
- (72) **Inventors:** SRIVILLIPUTTUR MANNARSWAMY, Sandya; Sy.No.192, Whitefield Road, Mahadevapura Post, Bangalore 560048 (IN). PHATAK, Anand Satish; Sy.No.192, Whitefield Road, Mahadevapura Post, Bangalore 560048 (IN).

(74) **Agents:** ORTEGA, Arthur et al.; Hewlett-Packard Company, Intellectual Property Administration, 3404 East Harmony Road, Mail Stop 35, Fort Collins, Colorado 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,

[Continued on next page]

(54) **Title:** PARTIAL SNAPSHOT CREATION

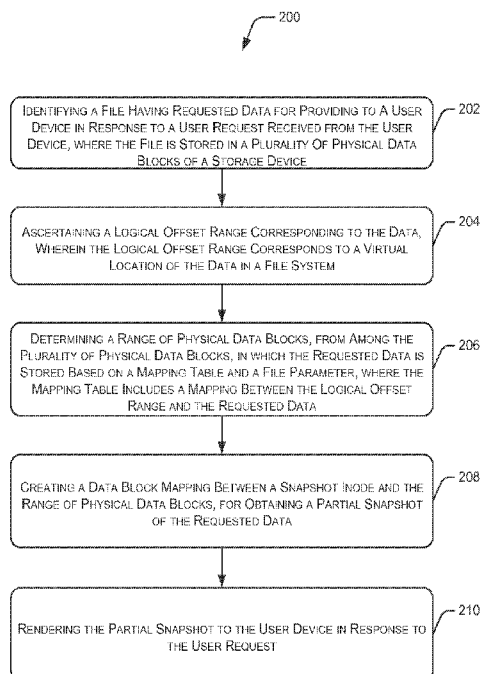


Figure 2

(57) **Abstract:** Partial snapshot creation includes identifying a file having requested data for providing to a user device in response to a user request. The file is stored in a plurality of physical data blocks of a storage device. A logical offset range corresponding to the requested data is ascertained. The logical offset range corresponds to a virtual location of the requested data in a file system. Range of physical data blocks in which the requested data is stored is determined based on a mapping table and a file parameter. The mapping table includes a mapping between the logical offset range and the requested data. A data block mapping between a snapshot inode, having metadata associated with partial snapshot, and the range of physical data blocks is created for obtaining a partial snapshot of the requested data. The partial snapshot is rendered to the user device in response to the user request.



SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ,
GW, KM, ML, MR, NE, SN, TD, TG).

— *as to applicant's entitlement to apply for and be granted
a patent (Rule 4.17(ii))*

Declarations under Rule 4.17:

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

Published:

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

5

PARTIAL SNAPSHOT CREATION

BACKGROUND

[0001] Increasing use of data processing and data generation in enterprises produces ever-increasing amounts of information which are stored for short, medium, or long periods. The information is typically stored for re-use by the organization. To maintain such information, such as data logs and files, enterprises generally implement data management and file storage systems for managing such information. Such systems further facilitate in sharing the stored information between various applications hosted, or used by the enterprise and its users. Generally, the stored information is shared such that an original copy of the information is not affected. For example, upon receiving a request for accessing certain data from a user, the systems may share a copy of the data instead of sharing original documents containing the data in order to prevent any modification by the user.

20

BRIEF DESCRIPTION OF FIGURES

[0002] The detailed description is described with reference to the accompanying figures. In the figures, the left-most digit(s) of a reference number identifies the figure in which the reference number first appears. The same numbers are used throughout the figures to reference like features and components. Systems and/or methods, in accordance with examples of the present subject matter are now described, by way of example, and with reference to the accompanying figures, in which:

[0003] Fig. 1(a) illustrates a block diagram of an example system for partial snapshot creation, in accordance with an example of the present subject matter.

30

5 **[0004]** Fig. 1(b) illustrates various example components of the example system for partial snapshot creation, in accordance with an example of the present subject matter.

[0005] Fig. 2 illustrates an example method for partial snapshot creation, in accordance with an example of the present subject matter.

10 [0006] Fig. 3 illustrates another example method for partial snapshot
creation, in accordance with an example of the present subject matter.

[0007] Fig. 4 illustrates an example network environment implementing a non-transitory computer readable medium for partial snapshot creation, according to an example of the present subject matter.

15	DETAILED DESCRIPTION
----	----------------------

[0008] Systems and methods for partial snapshot creation are described herein. The methods can be implemented in various systems, such as computing devices connected through various networks. Although the description herein is with reference to systems used for partial snapshot creation for multi-user file sharing, the methods and described techniques may be implemented in other systems, albeit with a few variations.

[0009] Typically, data is stored in a storage device through a file system. A file system may be understood as an abstraction for organizing and storing data on the storage device. The file system may also facilitate controlling how the data is stored and retrieved from the storage device. A storage device includes many data storage units known as “blocks”. When a file is stored in the storage device, the file data is written and stored over multiple blocks. Therefore, each file is usually considered to be associated with a list of blocks storing the file data. A list of the blocks associated with the file is maintained and stored in an index node (inode) associated with the

5 file. The inode includes metadata associated with the file. The metadata may include, inode number, attributes, number of blocks, file size, file type, block map information, and the like. The inode does not store the file data.

[0010] While storing, the file data in each file is allocated virtual addresses, also known as logical offsets that correspond to a virtual location
10 of the data in a file system. For instance, considering a file to be a stream of bytes, a first byte of the stream will have a logical offset zero, a second byte will have a logical offset two, a third byte will have a logical offset three, and so on till the n^{th} byte that will have a logical offset n .

[0011] When the data is accessed by different applications, for
15 example, by analytical and data presentation applications, a copy of the file containing the data accessed is created and written to new blocks, in addition to the blocks which may be retaining the actual data. For instance, depending on one or more queries requesting access to data, or portion thereof, data stored in the files may be processed and subsequently rendered on a user
20 device in multiple formats based on a query executed on the data by the analytics or data presentation applications. To this end, portion of the data may be copied and stored in other blocks, and processed. Upon processing, the output data may be provided to the requesting user in a different form or format. However, it should be noted that as a result of the processing initiated
25 by the executing queries, the original data is not updated. However, these applications typically create new files by fully copying data based on the presentation changes needed in the layout. Such a process of creating new file and writing the file to new blocks may result in unwanted duplication of data, leading to wastage of storage space. This, in turn may result in increase
30 in storage costs as the enterprise may need to deploy additional storage

5 devices for the files. Creation of the new file and writing the new file to the new blocks may further increase processing time of the applications.

[0012] The present subject matter, describes a system and a method for partial snapshot creation. The system and the method facilitate in creating a partial snapshot for rendering data to a user in response to a user request
10 received from the user. The user request may be, for example, a query, such as a search query and an analytical query. Based on the user request, relevant requested data may be identified from a storage device associated with the system for creating the partial snapshot. The partial snapshot creation facilitates in rendering the requested data without creating a new
15 data file having a copy of the requested data, thus preventing data duplication, in turn providing memory space utilization.

[0013] In accordance to an example of the present subject matter, upon receiving the user request, requested data, i.e., data that has to be provided to the user device in response to the user request is identified.
20 Subsequently, a file having the requested data is identified from amongst a plurality of files stored in the storage device. In one example, the file may be stored in a plurality of physical data blocks of the storage device of the system. Once the file has been identified, the file and file data may be obtained and copied to a main memory, for example, random access memory
25 of the system. The file may be subsequently analyzed to ascertain a logical offset range corresponding to the requested data. The logical offset range corresponds to a virtual location of the requested data in a file system. As previously described, a file system is used for organizing data on the storage device.

30 **[0014]** Further, from among the plurality of physical data blocks in which the file data of the file is stored, a range of physical data blocks in which the requested data is stored, is determined. In one example, the range

5 of physical data blocks is determined based on a mapping table and a file parameter. Examples of the file parameter include, but are not limited to, a file inode and a file name corresponding to the file in which the requested data is stored. The mapping table includes a mapping between the logical offset range and the requested data. Upon determining the range of physical data
10 blocks in which the requested data is stored, a data block mapping between a snapshot inode and the range of physical data blocks is created for obtaining a partial snapshot of the requested data. The snapshot inode includes metadata associated with the partial snapshot. Subsequently, the partial snapshot of the requested data is generated based on the snapshot inode and the data block mapping. As will be understood, the partial snapshot
15 includes the requested data. The partial snapshot is subsequently rendered to the user device in response to the user request.

[0015] The present subject matter thus facilitates in creating partial snapshots for providing requested data to a user device. As will be
20 understood based on the above description, the partial snapshot is created such that the snapshot inode is linked to the existing physical data blocks having the requested data. Linking the snapshot inode to the existing physical data blocks instead of copying the requested data to new physical data blocks facilitates in avoiding data duplication. Avoiding data duplication
25 further facilitates in memory space utilization as memory space wastage due to the data duplication is avoided. Linking new inode, i.e., the snapshot inode to the existing physical data blocks instead of copying the requested data to new physical data blocks and then linking the snapshot inode to the new physical data blocks reduces processing time of the applications. The
30 applications can now perform analytical and search functionalities faster as compared to known technologies. Further, the subject matter facilitates in creation of dynamic partial snapshots as the partial snapshots are created

5 based on the user request. Since the partial snapshots are created based on the user request, a different snapshot may be created in response to each user request.

[0016] The above systems and methods are further described in conjunction with Fig. 1 to Fig. 4. It should be noted that the description and
10 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, 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
15 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 as specific examples thereof, are intended to encompass equivalents thereof.

[0017] Fig. 1(a) illustrates a block diagram of an example system 102
20 for partial snapshot creation, in accordance with an example of the present subject matter. The system 102 may be implemented in, for example, desktop computers, multiprocessor systems, personal digital assistants (PDAs), laptops, network computers, cloud servers, minicomputers, mainframe computers, hand-held devices, such as tablets. The system 102 may also be
25 hosting a plurality of applications.

[0018] The system 102 may include, for example, processor(s) 104, a mapping module 106 communicatively coupled to the processor 104, and a partial snapshot module 108 communicatively coupled to the processor 104.

[0019] The processor(s) 104 may include microprocessors,
30 microcomputers, microcontrollers, digital signal processors, central processing units, state machines, logic circuitries, and/or any other devices

5 that manipulate 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.

10 **[0020]** In operation, upon receiving a user request from a user, through a user device of the user, the user request is processed by the system 102. The user request may be processed to determine a file having requested data that has to be shared with the user device in response to the user request. The file may be identified from amongst a plurality of files stored in a storage
15 device (not shown in the figure) associated with the system 100. The mapping module 106 subsequently ascertains a logical offset range corresponding to the requested data that has to be shared with the user device in response to the user request. The logical offset range corresponds to a virtual location of the requested data in a file system used for organizing
20 information on the storage device.

[0021] The partial snapshot module 108 may subsequently use the logical offset range to determine a range of physical data blocks, from among a plurality of physical data blocks, in which the requested data is stored. The plurality of physical data blocks are the physical data blocks in which file data
25 of the file is stored. In one example, the partial snapshot module 108 may determine the range of physical data blocks based on a mapping table and a file parameter. The mapping table, in one example, includes a mapping between the logical offset range and the requested data. Examples of the file parameter include, but are not limited to, as a file inode and a file name
30 corresponding to the file in which the requested data is stored.

[0022] Further, the partial snapshot module 108 may create a data block mapping between a snapshot inode and the range of physical data

5 blocks for obtaining a partial snapshot of the requested data. The snapshot inode includes metadata associated with the partial snapshot. The partial snapshot of the requested data thus created may be rendered to the user device.

[0023] Fig. 1(b) illustrates various example components of the system 102, in accordance with an example of the present subject matter. The system 102 includes the processor(s) 104, interface(s) 110, memory 112, modules 114, and data 116. The interfaces 110 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.

[0024] The memory 112 may be communicatively coupled to the processor 102 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 112 may include a storage device 118 and main memory 120. The storage device 118, such as hard disks and magnetic tapes may be used for storing content, such as files in the system 102. The main memory 120, such as RAM may be used for temporary storage of content for processing by the system 102.

[0025] The modules 114, amongst other things, include routines, programs, objects, components, and data structures, which perform particular tasks or implement particular abstract data types. The modules 114 may also be implemented as, signal processor(s), state machine(s), logic circuitries, and/or any other device or component that manipulate signals based on operational instructions. Further, the modules 114 can be implemented by

5 hardware, by computer-readable instructions executed by a processing unit, or by a combination thereof.

[0026] The modules 114 may include an interaction module 122, an analysis module 124, the mapping module 106, the partial snapshot module 108, and other module(s) 126. The other modules 126 may include programs
10 or coded instructions that supplement applications and functions, for example, programs in an operating system of the system 102. Further, the data 116 may include file(s) 128, analysis data 130, mapping data 132, snapshot data 134, and other data 136.

[0027] As previously described, the system 102 facilitates in providing
15 partial snapshots of data to a user device in response to a user request received from the user device. The user request may be a query by a user intending to search for certain data. For instance, a user working on an analytical or research related project may wish to search for certain data, from among the vast data stored in the memory 114. The user in such a case
20 may provide the user request to the system 102. For example, a user intending to analyze literacy rate of countries having names starting with the alphabet 's' may send a user request to the system 102 for providing data related to the literacy rate of such countries.

[0028] In one example, the user request is received by the interaction
25 module 122. The interaction module 122 may save the user request in the analysis data 130. The user request is subsequently analyzed by the analysis module 124. The analysis module 124 may analyze the user request to determine requested data, i.e., the data that has to be shared with the user device. Upon determining the requested data that has to be shared, the
30 analysis module 124 may perform a search to identify a file having the requested data. In one example, the file may be stored in a plurality of physical data blocks of the storage device 118. For instance, in the above

5 described example of the analysis of the literacy rate, the analysis module 124 may analyze the user request to ascertain that user request is for requested data related to literacy rate of the countries having names starting with the alphabet 's'. The analysis module 124 may thus perform a search for identifying the file, from among the files 128, having requested data about the
10 literacy rate of the countries across the world.

[0029] Upon identifying the file having the requested data, the analysis module 124 obtains the file for further analysis. In one example, the analysis module 124 copies the file and its file data into the main memory 120 from the storage device 118. In one example, the entire file maybe copied at once in
15 the main memory 120. In another example, in case of large file size, parts of the file may be copied sequentially. The file may be subsequently analyzed by the mapping module 106 to determine the logical offset, i.e., the virtual location of the requested data in the file. As will be understood, the file will include the literacy rate for almost all the countries with the literacy rate being
20 arranged, for example, in an alphabetical order of the names of the countries. The virtual location of the requested data in such a case may be the place in the file where the countries with their name starting from the alphabet 's' are listed.

[0030] The mapping module 106 in such a case may subsequently
25 ascertain the logical offset range based on the virtual location of the requested data. For instance, considering the file having the requested data to be a stream of bytes and the requested data is stored at the virtual locations 25-60 bytes. The requested data in such a case will have a logical offset range of 25-60. The mapping module 106 may further generate the
30 mapping table based on the logical offset range of the requested data. In one example, the mapping table may include a beginning position of the logical offset range and a bit length corresponding to the logical offset range. The

5 beginning position of the logical offset range may be the virtual location at which the requested data begins, for instance, the offset 25 in the above example. The bit length of the logical offset range may be difference between the beginning position and a last position of the logical offset range, for instance, bit length will be 35 in the above example. The mapping module
10 106 may subsequently save the mapping table in the mapping data 132. The mapping module 132 may further save the file parameter in the mapping module 132.

[0031] In one example, the partial snapshot module 108 may obtain the mapping table and the file parameter from the mapping data 132 to
15 determine the range of physical data blocks of the storage device 118 in which the requested data is stored. As previously described, the range of physical data blocks is determined from among the plurality of physical data blocks in which the file is stored in the storage device 118. For instance, if the storage device includes physical data blocks numbered one to ten lakhs, the
20 file may be stored in the plurality of physical data blocks numbered from, say, 100 to 150. Further, the range of physical data blocks having the requested data that has to be shared with the user may be numbered, say, 130 to 135. The partial snapshot module 108 may thus determine physical data blocks 130 to 135 as the range of physical data blocks corresponding to the
25 requested data that has to be shared.

[0032] Subsequently, the partial snapshot module 108 may create a snapshot inode for obtaining the partial snapshot of the requested data. The snapshot inode includes metadata about the partial snapshot being created for the requested data. The metadata may include inode number, attributes,
30 number of blocks, file size, file type, and the like. The partial snapshot module 108 may then create a data block mapping between the snapshot inode and the physical data blocks. The data block mapping may define a linking

5 between the snapshot inode and the physical data blocks. The partial snapshot module 108 may subsequently generate the partial snapshot based on the snapshot inode and the data block mapping for being rendered to the user device. Further, the partial snapshot module 108 may save the partial snapshot in the snapshot data 134. Although the present subject matter has
10 been described in the context of physical disk blocks, the present subject may be implemented for file systems which allocate data storage in terms of disk extents, i.e., a set of contiguous physical disk blocks.

[0033] In one example, the interaction module 122 renders the partial snapshot to the user device as a response to the user request received from
15 the user device. For instance, in the above described example of the user request for obtaining the literacy rate of the countries having names starting with the alphabet 's', the partial snapshot module 108 may generate a partial snapshot for literacy rate of the countries having names starting with the alphabet 's'. In said example, the snapshot inode may refer to the physical
20 data blocks storing the requested data about literacy rate of the countries having names starting with the alphabet 's'. The interaction module 122 may subsequently render the partial snapshot of the literacy rates to the user device. Further, since the partial snapshots are created based on the user request, a different snapshot may be created in response to each user
25 request. The partial snapshot module 108 may thus facilitate in providing dynamic partial snapshots.

[0034] Figs. 2 and 3 illustrate example methods 200 and 300, respectively, for partial snapshot creation, according to an implementation of the present subject matter. The order in which the methods are described is
30 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

5 methods 200 and 300 may be implemented by processing resource or computing device(s) through any suitable hardware, non-transitory machine readable instructions, or combination thereof.

[0035] It may also be understood that the methods 200 and 300 may be performed by programmed computing devices, such as the system 102 as depicted in Figs. 1A and 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
10 optically readable digital data storage media.
15

[0036] Further, the methods 200 and 300 are described below with reference to the 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.

20 **[0037]** Referring to Fig. 2, at block 202, a file having requested data for providing to a user device is identified. In one example, the file is identified for providing requested data in response to a user request received from the user device. Further, the file is stored in a plurality of physical data blocks of a storage device associated with a computing system, for example, the system
25 102. In one example, upon receiving the user request, analysis module, such as the analysis module 124 of the system 102 analyzes the user request to determine the requested data has to be shared. The file having the requested data is subsequently searched from the storage device.

[0038] At block 204, a logical offset range corresponding to the
30 requested data is ascertained. In one example, the system 102 may analyze the file to ascertain the logical offset range corresponding to the requested

5 data. The logical offset range corresponds to a virtual location of the requested data in a file system that is used for organizing information on the storage device of the system 102.

[0039] At block 206, a range of physical data blocks in which the requested data is stored is determined, from among the plurality of physical
10 data blocks. In one example, the range of physical data blocks is determined based on a mapping table and a file parameter, such as a file inode and a file name of the file having the requested data that has to be shared. The mapping table includes a mapping between the logical offset range and the requested data. In one example, the system 102 may analyze the logical
15 offset range and the file parameter to determine the range of physical blocks of a storage device of the system 102 in which the requested data is stored.

[0040] At block 208, a data block mapping between a snapshot inode and the range of physical data blocks is created for obtaining a partial snapshot of the requested data. In one example, the snapshot inode includes
20 metadata associated with the partial snapshot. The snapshot inode is created based on the determining of the range of physical data blocks.

[0041] At block 210, the partial snapshot is rendered to the user device in response to the user request. In one example, once the data block mapping between the snapshot inode and the range of physical data blocks is
25 created, the partial snapshot is created by the system 102.

[0042] Referring to Fig. 3, at block 302, a file having requested data that has to be shared with a user device is identified. In one example, the file is identified for providing requested data in response to a user request received from the user device. Further, the file is stored in a plurality of
30 physical data blocks of a storage device associated with a computing system, for example, the system 102. In one example, upon receiving the user

5 request, the system 102 analyzes the user request to determine the requested data has to be shared. The file having the requested data is subsequently searched from the storage device.

[0043] At block 304, the file and its file data are copied into a main memory. In one example, the file is obtained from the storage device of the
10 system, say, the system 102 and copied into the main memory. Main memory may be, for example, a RAM of the system 102. Further, in one example, the file maybe copied entirely at once. In another example, parts of the file may be copied sequentially instead of copying the entire file at a once.

[0044] At block 306, the file and its file data are analyzed to ascertain a
15 logical offset range corresponding to the requested data. In one example, the system 102 may analyze the file to ascertain the logical offset range corresponding to the requested data. The logical offset range corresponds to a virtual location of the requested data in a file system that is used for organizing information on the storage device of the system 102.

20 **[0045]** At block 308, a range of physical data blocks in which the requested data is stored is determined. In one example, the range of physical data blocks is determined based on a mapping table and a file parameter, such as a file inode and a file name of the file having the requested data that has to be shared. The mapping table includes a mapping between the logical
25 offset range and the requested data. In one example, the system 102 may analyze the logical offset range and the file parameter to determine the range of physical blocks of the storage device of the system 102 in which the requested data is stored.

[0046] At block 310, a snapshot inode referring to the physical data
30 blocks is created. In one example, the snapshot inode is created based on

5 the determining of the range of physical data blocks. For example, the system 102 creates the snapshot inode based on the physical data blocks.

[0047] At block 312, a data block mapping is created. In one example, the data block mapping is created between the snapshot inode and the physical data blocks. The data block mapping links the snapshot inode to the
10 physical data blocks for further processing.

[0048] At block 314, a partial snapshot of the requested data is generated based on the snapshot inode and the data block mapping. In one example, the system 102 generates the partial snapshot for being rendered to user device in response to the user request.

15 [0049] FIG. 4 illustrates a system environment 400 for partial snapshot creation, according to an example of the present disclosure. The system environment 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
20 resource 402 communicatively coupled to a computer readable medium 404 through a communication link 406.

[0050] For example, the processing resource 402 can include one or more processors of a computing device for partial snapshot creation. The computer readable medium 404 can be, for example, an internal memory
25 device of the computing device or an external memory device. In one 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
30 the computer readable medium 404 through a network 408. The network 408

5 may be a single network or a combination of multiple networks and may use a variety of different communication protocols.

[0051] 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
10 network 408. The coupling with the requested data sources 410 enables in receiving the requested data in an offline environment, and the coupling with the communication devices 412 enables in receiving the requested data in an online environment.

[0052] In one implementation, the computer readable medium 404
15 includes a set of computer readable instructions, implementing a mapping module 414 and a partial snapshot module 416. The set of computer readable instructions can be accessed by the processing resource 402 through the communication link 406 and subsequently executed to process requested data communicated with the requested data sources 410 in order
20 to facilitate partial snapshot creation. When executed by processing resource 402, the instructions of mapping module 414 may perform the functionalities described above in relation to mapping module 106. When executed by processing resource 402, the instructions of partial snapshot module 416 may perform the functionalities described above in relation to partial snapshot
25 module 108.

[0053] For example, in response to a user request for accessing requested data, the mapping module 414 may identify logical offset range corresponding to the requested data in a file having the requested data. Initially, an analysis module (not shown in the figure) may analyze the user
30 request to determine the requested data that has to be shared with the user device. The analysis module may subsequently perform a search to identify the file having the requested data. In one example, the file is stored in a

5 plurality of physical data blocks of a storage device associated with the communication devices 412.

[0054] The mapping module 414 may subsequently analyze the file and its file data to determine logical offset range corresponding to the requested data. The logical offset range corresponds to a virtual location of
10 the requested data in a file system. The mapping module 414 further creates a mapping table having a mapping between the logical offset range and the requested data. The mapping table may be further used by the partial snapshot module 416 to determine a range of physical data blocks of the storage device in which requested data, that has to be shared with the user
15 device, is stored. In one example, the partial snapshot module 416 may determine the range of physical data blocks based on the mapping table and a file parameter, such as a file inode and a file name of the file having the requested data.

[0055] The partial snapshot module 416 may further create a snapshot
20 inode referring to the physical data blocks. Further, the partial snapshot module 416 may create a data block mapping between the snapshot inode and the physical data blocks. The partial snapshot module 416 may subsequently generate a partial snapshot for the requested data based on the snapshot inode and the data block mapping for being rendered to the user
25 device. As previously described, creating the partial snapshot facilitates avoiding creation of a copy of the requested data.

[0056] Although examples for the present disclosure have been described in language specific to structural features and/or methods, it should
30 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

PAGE INTENTIONALLY LEFT BLANK

5 I/We claim:

1. A method comprising:

identifying a file having requested data for providing to a user device in response to a user request received from the user device, wherein the file is stored in a plurality of physical data blocks of a storage device;

10 ascertaining a logical offset range corresponding to the requested data, wherein the logical offset range corresponds to a virtual location of the requested data in a file system;

 determining a range of physical data blocks, from among the plurality of physical data blocks, in which the requested data is stored, based on a mapping table and a file parameter, wherein the mapping table includes a mapping between the logical offset range and the requested data;

15 creating a data block mapping between a snapshot inode and the range of physical data blocks, for obtaining a partial snapshot of the requested data, wherein the snapshot inode includes metadata associated with the partial snapshot; and

20 rendering the partial snapshot to the user device in response to the user request.

2. The method of claim 1, wherein the identifying further comprises:

25 analyzing the user request to determine the requested data that has to be shared with the user device; and

 performing a search to identify, from among a plurality of files in a storage device, the file having the requested data.

30 3. The method of claim 1, wherein the ascertaining further comprises:

 copying, into a main memory, the file and file data from the storage device; and

5 analyzing the file and the file data to ascertain the logical offset range.

4. The method of claim 1, further comprising generating the mapping table based on the ascertaining, wherein the mapping table includes a beginning position of the logical offset range and bit length corresponding to
10 the logical offset range.

5. The method of claim 1, wherein the creating further comprising generating the partial snapshot based on the snapshot inode and the data block mapping.
15

6. The method of claim 1, the creating the data block mapping further comprising creating the snapshot inode based on the determining. .

7. A system comprising:
20 a processor;
a mapping module coupled to the processor to:
ascertain a logical offset range corresponding to requested data that has to be shared with a user device in response to a user request received from the user device, wherein the logical offset range corresponds to a virtual location of the requested data in a file system;
25 and
a partial snapshot module coupled to the processor to:
determine a range of physical data blocks, from among a plurality of physical data blocks, in which the requested data is stored based on a mapping table and a file parameter, wherein the mapping
30 table includes a mapping between the logical offset range and the requested data; and

5 create a data block mapping between a snapshot inode and the range of physical data blocks to obtain a partial snapshot of the requested data for being rendered to the user device, wherein the snapshot inode includes metadata associated with the partial snapshot.

10

8. The system as claimed in claim 7, further comprising an analysis module coupled to the processor to:

 analyze the user request to determine the requested data that has to be shared with the user device; and

15 perform a search to identify a file having the requested data, wherein the file is stored in the plurality of physical data blocks of a storage device associated with the system.

9. The system as claimed in claim 8, wherein the analysis module further
20 is to copy, into a main memory of the system, the file and file data from the storage device.

10. The system as claimed in claim 9, wherein the mapping module further is to:

25 analyze the file and file data to ascertain the logical offset range; and
 generate the mapping table based on the logical offset range, wherein the mapping table includes a beginning position of the logical offset range and bit length corresponding to the logical offset range.

30 11. The system as claimed in claim 7, wherein the partial snapshot module further is to:

5 generate the partial snapshot based on the snapshot inode and the data block mapping for being rendered to the user device.

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

10 determine, based on a mapping table and a file parameter, a range of physical data blocks of a storage device in which requested data, that has to be shared with a user device, is stored, wherein the mapping table includes a mapping between a logical offset range and the requested data, and wherein the logical offset range corresponds to a virtual location of the requested data
15 in a file system;

 create a snapshot inode referring to the physical data blocks, based on the determining;

 create a data block mapping between the snapshot inode and the physical data blocks; and

20 generate a partial snapshot based on the snapshot inode and the data block mapping for being rendered to the user device.

13. The non-transitory computer readable medium of claim 12, wherein the computer readable instructions, when executed, further cause the processor
25 to:

 receive a user request from the user device;

 analyze the user request to determine the requested data that has to be shared with the user device; and

 perform a search to identify a file having the requested data, wherein
30 the file is stored in a plurality of physical data blocks of the storage device.

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

 copy, into a main memory of the system, the file and file data from the storage device; and

10 analyze the file and the file data to ascertain the logical offset range corresponding to the requested data.

 15. The non-transitory computer readable medium of claim 12, wherein the computer readable instructions, when executed, further cause the processor
15 to generate the mapping table based on the logical offset range, wherein the mapping table includes a beginning position of the logical offset range and bit length corresponding to the logical offset range.

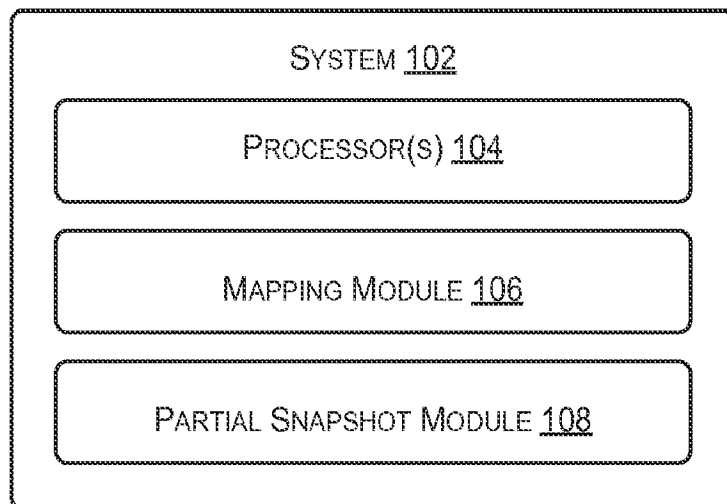


Figure 1A

2/5

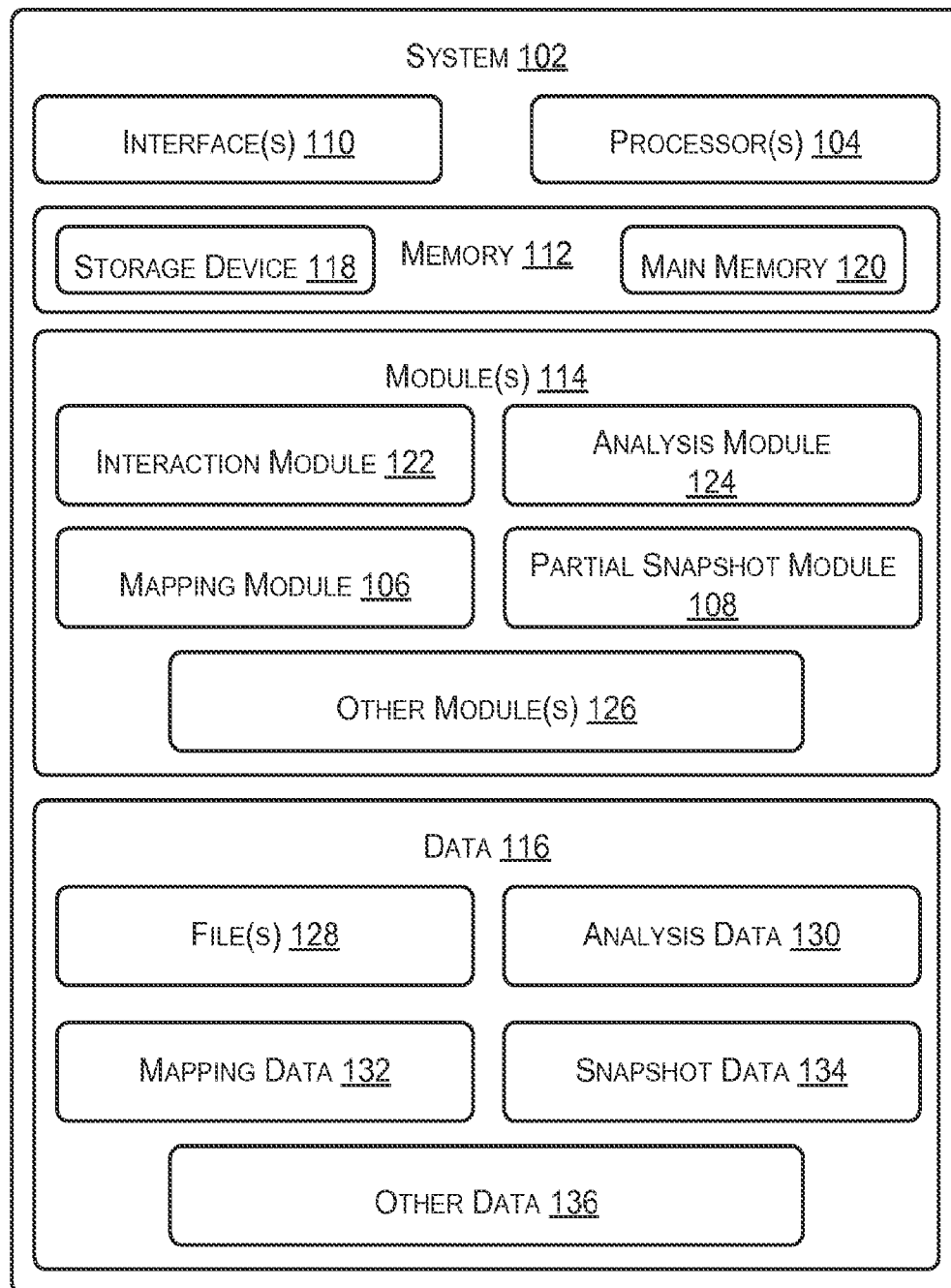


Figure 1B

3/5

200

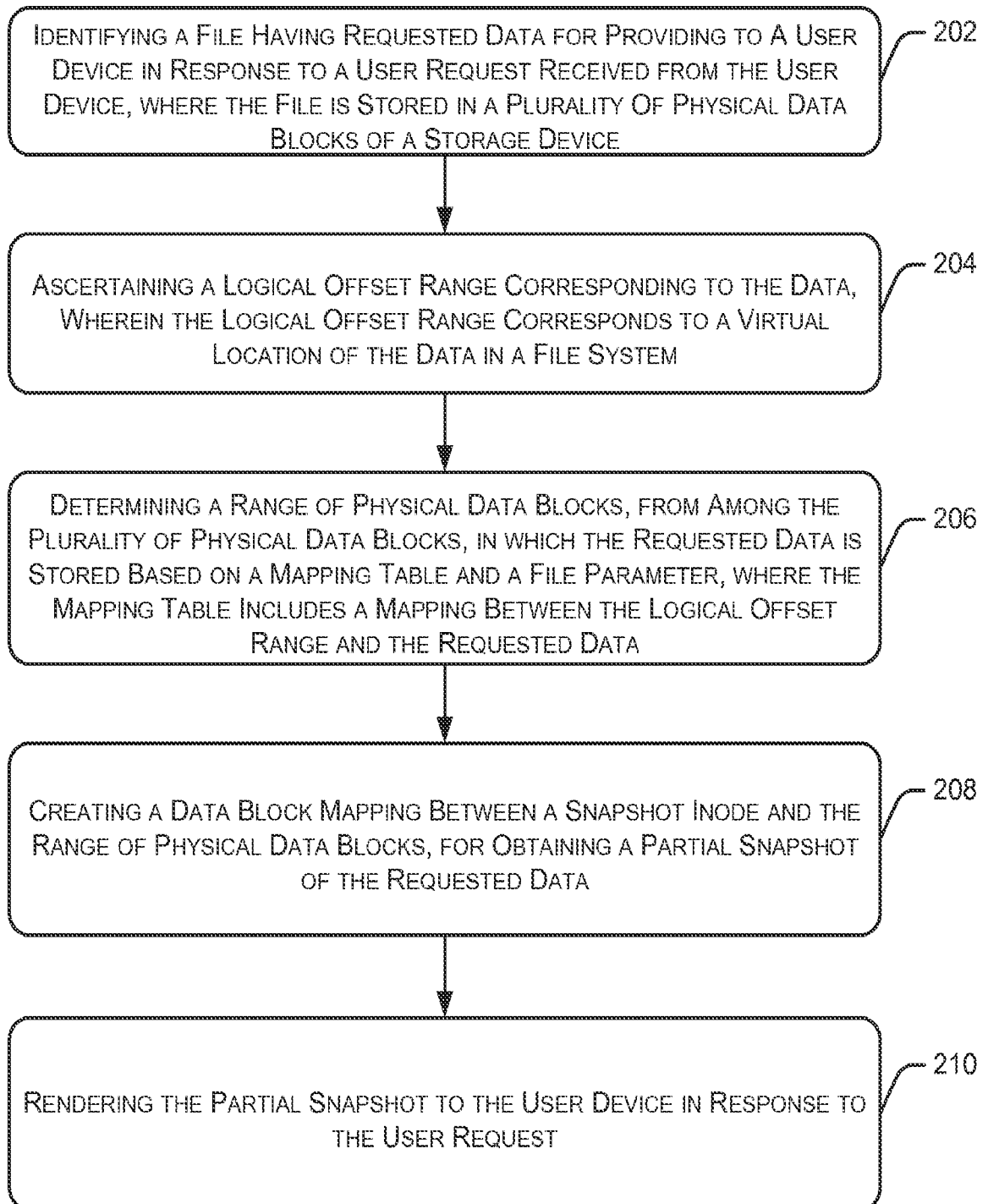


Figure 2

4/5

300

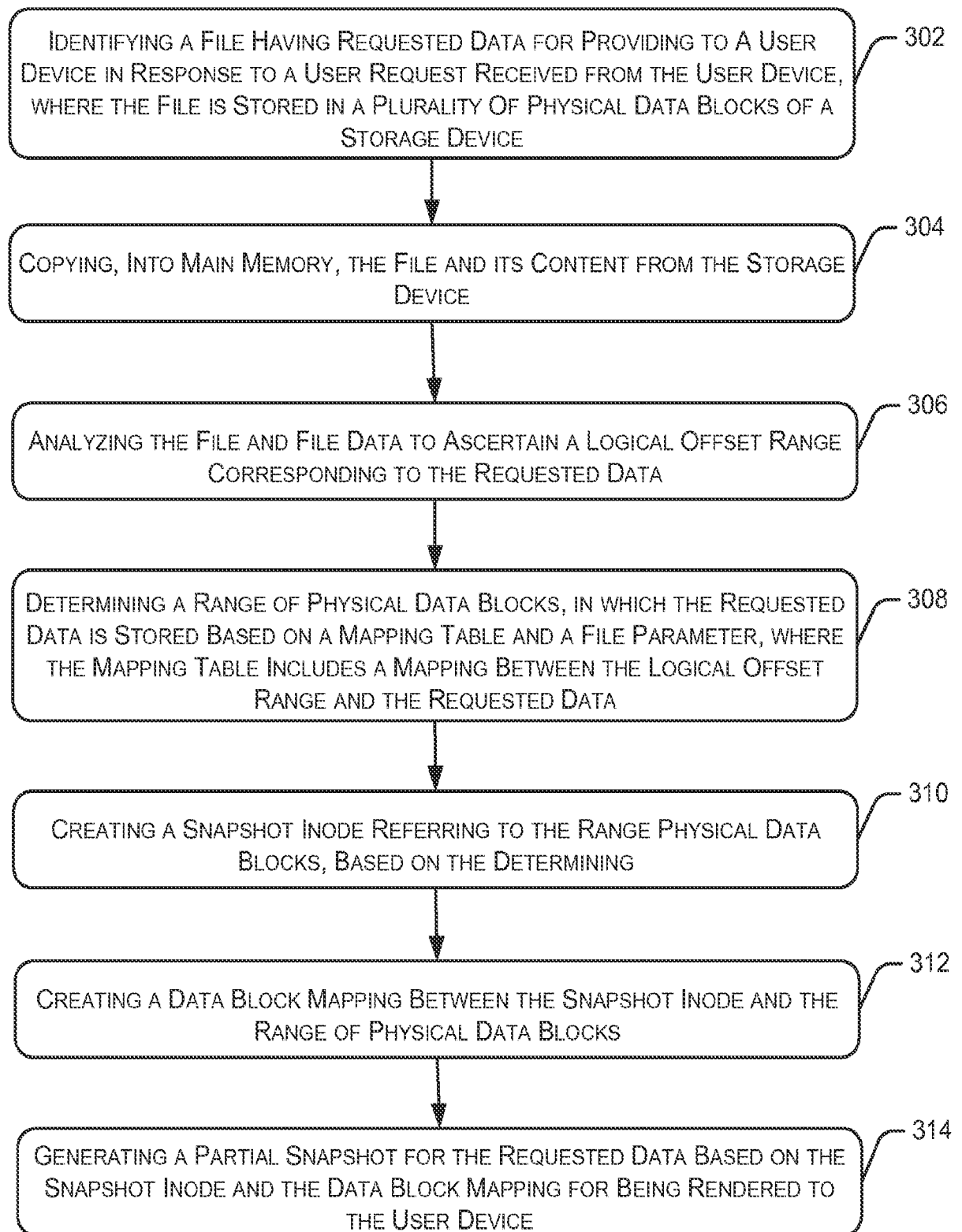


Figure 3

5/5

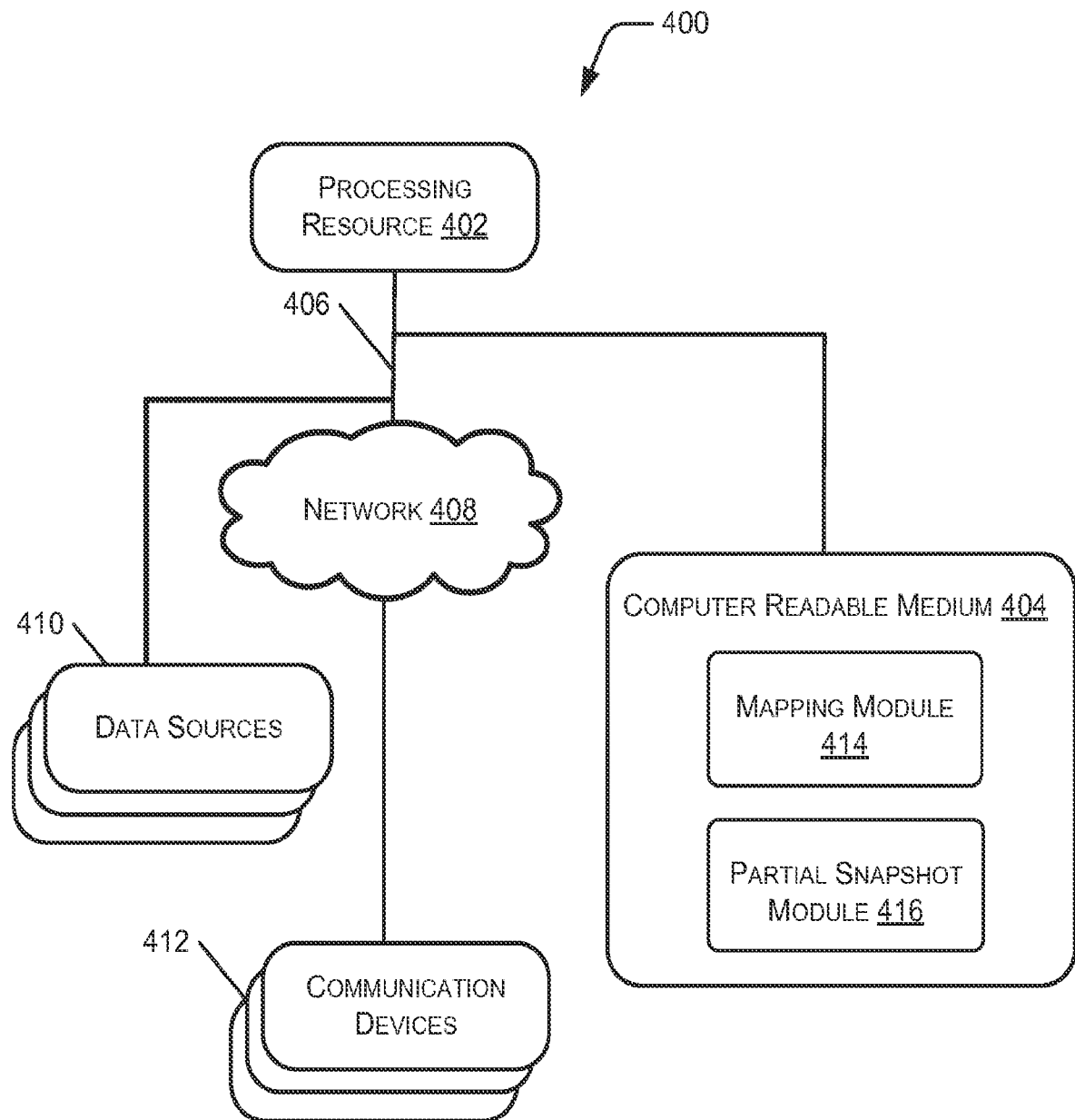


Figure 4

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2014/060284**A. CLASSIFICATION OF SUBJECT MATTER****G06F 12/06(2006.01)i**

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

B. FIELDS SEARCHEDMinimum documentation searched (classification system followed by classification symbols)
G06F 12/06; G06F 7/10; G06F 12/16; G06K 9/46; G06F 17/30; G06K 9/00; G06F 9/00Documentation 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 modelsElectronic data base consulted during the international search (name of data base and, where practicable, search terms used)
eKOMPASS(KIPO internal) & Keywords: partial, snapshot, map, iNode, block,**C. DOCUMENTS CONSIDERED TO BE RELEVANT**

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2011-0282841 A1 (NOBUYUKI SAIKA et al.) 17 November 2011 See paragraphs [0001]-[0003], [0011]-[0012], [0057], [0062]-[0064], [0070]-[0071], [0090], [0112], [0130], [0146], [0163]; claims 1-2; and figures 4, 6.	1-6, 12-15
Y		7-11
Y	US 2008-0250086 A1 (RICHARD VICTOR KISLEY) 09 October 2008 See paragraphs [0027], [0040]; claim 1; and figure 6.	7-11
A	US 2011-0078131 A1 (JI-RONG WEN et al.) 31 March 2011 See paragraphs [0005], [0022]; and figure 1.	1-15
A	US 2005-0238200 A1 (RAKESH GUPTA et al.) 27 October 2005 See paragraphs [0009], [0022]; and figure 1.	1-15
A	WO 01-50250 A2 (SUN MICROSYSTEMS, INC.) 12 July 2001 See page 4, lines 13-23; and figure 1.	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

26 February 2015 (26.02.2015)

Date of mailing of the international search report

26 February 2015 (26.02.2015)

Name and mailing address of the ISA/KR

International Application Division
Korean Intellectual Property Office
189 Cheongsu-ro, Seo-gu, Daejeon Metropolitan City, 302-701,
Republic of Korea

Facsimile No. ++82 42 472 3473

Authorized officer

BYUN, Sung Cheal

Telephone No. +82-42-481-8262



INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2014/060284

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2011-0282841 A1	17/11/2011	US 8352426 B2 WO 2011-145132 A1	08/01/2013 24/11/2011
US 2008-0250086 A1	09/10/2008	US 2005-0027748 A1 US 7398285 B2 US 7984016 B2	03/02/2005 08/07/2008 19/07/2011
US 2011-0078131 A1	31/03/2011	None	
US 2005-0238200 A1	27/10/2005	JP 04644248 B2 JP 2007-535765 A US 7831094 B2 WO 2005-104737 A2 WO 2005-104737 A3	02/03/2011 06/12/2007 09/11/2010 10/11/2005 16/04/2009
WO 01-50250 A2	12/07/2001	AU 2001-27651 A1 US 7120572 B1 WO 01-50250 A3	16/07/2001 10/10/2006 06/12/2001