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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
6 October 2016 (06.10.2016)

WIPO | PCT

(10) International Publication Number
WO 2016/160027 A1

(51) International Patent Classification:
G06F 12/08 (2006.01)

(21) International Application Number:
PCT/US2015/024159

(22) International Filing Date:
2 April 2015 (02.04.2015)

(25) Filing Language: English

(26) Publication Language: English

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

(72) Inventors: **VOIGT, Douglas L.**; 11311 Chinden Blvd., Boise, Idaho 83714-0021 (US). **ZOU, Meng**; 4209 Technology Drive, Fremont, California 94538 (US).

(74) Agents: **SURESH, Anup** 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))*

Published:

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

WO 2016/160027 A1

(54) Title: PAGE CACHE ON PERSISTENT MEMORY

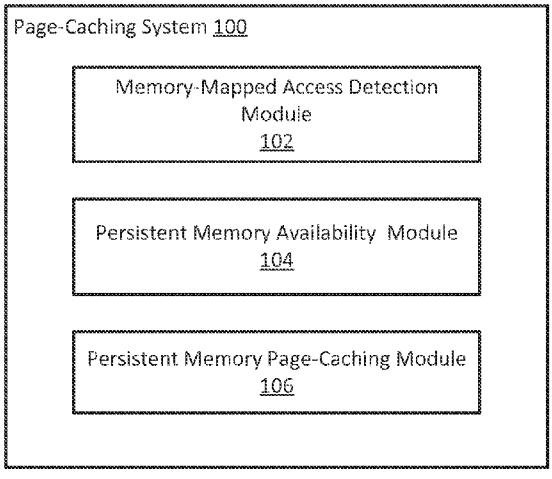

FIG. 1

(57) Abstract: Various examples described herein provide for caching a page on persistent memory for memory-mapped access of a file from a non-persistent memory file system or a remote file system having a non-persistent memory page cache. In particular, some examples detect memory-mapped access of a file from a non-persistent memory file system or a remote file system having a non-persistent memory page cache and, based on availability of persistent memory, caches a page associated with the memory-mapped access on the persistent memory.

PAGE CACHE ON PERSISTENT MEMORY

BACKGROUND

[1] Various memory-based technologies are utilized to improve the speed at which files can be accessed from a file system. For instance, using memory mapping to provide access to a file permits an application to access the mapped portions of the file as if they are in primary memory, which can improve input/output (I/O) performance when accessing the file, especially when the file is large in size.

BRIEF DESCRIPTION OF THE DRAWINGS

[2] Certain examples are described in the following detailed description in reference to the following drawings.

[3] FIGs. 1 and 2 illustrate example page-caching systems for page caching using persistent memory during memory-mapped access of a file.

[4] FIG. 3 illustrates an example computer system including an example page-caching system.

[5] FIG. 4 illustrates example data flow in an example computing environment that includes an example page-caching system.

[6] FIG. 5 illustrates an example computer system for page caching using persistent memory during memory-mapped access of a file.

[7] FIGs. 6 and 7 illustrate example methods performed by an example computer system to facilitate page-caching using persistent memory during memory-mapped access of a file.

DETAILED DESCRIPTION

[8] Though traditional uses of memory-based technologies can improve access to files on file systems, they also have their drawbacks. For instance, using with Non-volatile memory (NVM) to improve memory-mapped access of file generally requires an application be modified to synchronize or flush writes to NVM, or requires the application use libraries to do so. With respect to using memory mapping to access files, various data storage devices do not permit

direct memory mapping and achieve the memory mapping by allocating volatile memory (e.g., dynamic random access memory [DRAM]) and synchronizing data to non-volatile data storage (e.g., hard-disk drive or solid-state drive). Use of the volatile memory for page caching usually leads to repeated writes of page data from the volatile memory to non-volatile data storage, partially due to the lack of data persistent on volatile memory (e.g., when the volatile memory is no longer powered, page data thereon is lost).

[9] Various examples described herein provide use of persistent memory for page-caching during memory-mapped access of a file stored on and accessed from a non-persistent memory file system or a remote file system. According to some examples, when a file is determined to have been memory-mapped while it is being accessed from a non-persistent memory file system or a remote file system, persistent memory is utilized for caching a page involved in the memory-mapped access of the file. In this way, various examples use memory-mapping of a file as a trigger for selecting persistent memory to page cache a portion of the file when the file is stored on and accessed from a non-persistent memory file system, such as a hard-disk drive-based file system, or a remote file system, such as a file system (e.g., persistent or non-persistent memory file system) accessed over a communications network.

[10] In some examples, a memory-mapped access of a file from a non-persistent memory file system or a remote file system having a non-persistent memory page cache is detected, availability of persistent memory for caching a page in association with the memory-mapped access is determined, and based on this availability, the page is cached on the persistent memory. Depending on the example, where the page is a new page and the memory-mapped access comprises a request to cache the new page in association with the memory-mapped access, the page may be cached on the persistent memory in response to the request. Accordingly, the availability of the persistent memory may be determined in response to a request for caching a new page in association with the memory-mapped access of the file. Where the page may be an existing page cached on the non-persistent memory page cache and the memory-mapped access comprises a request to access the existing page in association with the

memory-mapped access, the page may be cached on the persistent memory in response to the access of the page (e.g., read or written to by an application). Where the page is an existing page, caching the page may involve migrating the page from the non-persistent memory page cache to the persistent memory. Accordingly, the availability of the persistent memory may be determined in response to an existing page on the non-persistent page cache being accessed (e.g., read or written to by an application).

[11] For some examples, the page cached on the persistent memory is flushed from the persistent memory to the non-persistent memory file system or the remote file system from where the file is accessed. Additionally, for some examples, a page cached on persistent memory in association with memory-mapped access of a file may be flushed when the persistent memory experiences resource pressure (e.g., lack of data storage space), when the memory-mapped access of the file is disestablished (e.g., un-mapped), when the file is closed, or when it is determined (e.g., by the computer system including the persistent memory) that other uses of the persistent memory would provide an operational advantage over using it for page caching. Additionally, for some examples, persistent memory may be utilized for page caching in place of, or in addition, to a non-persistent memory page cache, which may be based on volatile memory (e.g., DRAM).

[12] Unlike a page cache based on volatile memory (e.g., dynamic random access memory [DRAM]) used by traditional memory-mapping, persistent memory is non-volatile and, as such, a page cached on persistent memory inherently gains persistence while avoiding, reducing, or at least delaying the need to write (e.g., flush) the cached page (e.g., by way of an msync system call) back to a non-persistent memory storage device (e.g., hard-disk drive or solid-state drive) or a remote file system. By avoiding, reducing, or delaying the need to write a cached page to the non-persistent memory storage device, memory-mapped access to a file can be accelerated and wear on a store device (e.g., solid-state drive [SSD]) underlying a non-persistent memory device or a remote file system can be reduced.

[13] As used herein, persistent memory may include a persistent memory device, such as a phase change memory (PCM) device. As used herein, a non-persistent memory file system can include a file system that is implemented using a non-persistent memory device, such as a hard disk drive (HDD), a solid-state drive (SSD), or some other non-volatile memory device, and that is capable of providing memory-mapped access to a file stored thereon using a non-persistent memory page cache. As used herein, access of a file can include reading data from or writing data to a portion of the file.

[14] As also used herein, a remote file system may be one accessible by a first computer system over a communications network and through a second computer system that maintains the remote file system. A remote file system may be associated with (e.g., implemented using) a persistent memory device or a non-persistent memory device, which may be local to the second computer system. For various examples, the first computer system utilizes a page-caching technique described herein when using memory mapping to access a file stored on the remote file system.

[15] The following describes an example page-caching system in the context of an example life cycle of a particular file. The example life cycle may begin with the particular file being stored on a non-persistent memory file system or a remote file system, and the particular file being accessed (e.g., opened) from the non-persistent memory file system or the remote file system (e.g., by an application). The non-persistent memory file system or the remote file system may include a file system capable of providing memory-mapped access to a file using a non-persistent memory page cache. For the example page-caching system, when memory-mapped access of the particular file is detected, the page-caching system determines availability of persistent memory for caching a page in association with the memory-mapped access (e.g., availability of data storage space on the persistent memory for page caching) and, based on the availability, caches the page on the persistent memory. The determination of availability of the persistent memory and caching of the page may be in response to a request to cache a new page generated in association with the memory-mapped access,

or in response to an existing page on the non-persistent memory page cache being accessed.

[16] Accordingly, where the particular file is opened and the persistent memory is available at the time a new page needs to be cached in association with the memory-mapped access of the particular file (e.g., using a mmap system call), the example page-caching system may cache the new page on the persistent memory. Where the persistent memory is not available at the time, the example page-caching system may cache the new page on the non-persistent memory page cache. Where the persistent memory becomes available after an existing page has been cached on the non-persistent memory page cache in association with the memory-mapped access of the particular file, the example page-caching system may migrate the existing page from the non-persistent memory page cache to the persistent memory. This existing page may be migrated to the persistent memory after or upon subsequent access of the existing page (e.g., access virtual memory area claimed by a previous mmap request involving the non-persistent memory page cache). Migrating the existing page from the non-persistent memory page cache to the persistent memory may involve copying the page from the non-persistent memory page cache to the persistent memory, and then removing the page from the non-persistent memory page cache.

[17] After a page has been cached on the persistent memory, the page can be accessed from the persistent memory through a direct memory-mapping with the persistent memory, where a memory address can be directly mapped to a location on the persistent memory device where the page is stored. Such a direct mapping can permit (e.g., an application to) access to the page using load (e.g., ld) and store (e.g., st) machine code instructions of a processor (e.g., central processing unit [CPU]), and can also permit access to the page with no need of a non-persistent memory page cache (thereby avoiding drawbacks associated therewith).

[18] The following provides a detailed description of the examples illustrated by FIGs. 1-7.

[19] FIG. 1 illustrates an example page-caching system 100 for page caching using persistent memory during memory-mapped access of a file. As shown, the page-caching system 100 includes a memory-mapped access detection module 102, a persistent memory availability module 104, and a persistent memory page-caching module 106. Depending on the example, the page-caching system 100 may be part of a computer system, such as a desktop, laptop, hand-held computing device (e.g., personal digital assistants, smartphones, tablets, etc.), workstation, server, or other device that includes a processor. In various examples, the components or the arrangement of components in the page-caching system 100 may differ from what is depicted in FIG. 1.

[20] As used herein, modules and other components of various examples may comprise, in whole or in part, machine-readable instructions or electronic circuitry. For instance, a module may comprise computer-readable instructions executable by a processor to perform one or more functions in accordance with various examples described herein. Likewise, in another instance, a module may comprise electronic circuitry to perform one or more functions in accordance with various examples described herein. The elements of a module may be combined in a single package, maintained in several packages, or maintained separately.

[21] The memory-mapped access detection module 102 may facilitate detecting memory-mapped access of a file, by a computer system, from a non-persistent memory file system, or remote file system, having a non-persistent memory page cache. For various examples, the non-persistent memory page cache assists in the non-persistent memory file system providing memory-mapped access to a file stored thereon. The computer system may initiate memory-mapped access of the file by opening the file and establishing memory-mapped access of the file, which can result in a new page being allocated in association with the memory-mapped access. The memory-mapped access detected by memory-mapped access detection module 102 may comprise the computer system establishing a new memory-mapping of the file, or the computer system requesting access to a memory address (e.g., virtual memory address) of a page stored on a non-persistent memory page cache and associated with an existing (e.g., previously established) memory mapping.

[22] The persistent memory availability module 104 may facilitate determining availability of persistent memory, on a computer system, for page caching. For some examples, the persistent memory availability module 104 determines availability of persistent memory prior to caching a page, associated with memory-mapped access of a file, on the persistent memory. Depending on the example, the persistent memory availability module 104 may determine availability of the persistent memory in response to a request to cache a new page in association with the memory-mapped access, or in response to an existing page on the non-persistent memory page cache being accessed in association with the memory-mapped access.

[23] Depending on the example, the availability of persistent memory for caching a page in association with the memory-mapped access may be based on whether the persistent memory is present on the computer system. The availability of persistent memory for caching a page in association with the memory-mapped access may be based on whether the persistent memory is enabled for page caching (e.g., persistent memory may be disabled for page caching but enabled for long-term file storage). Additionally, the availability of persistent memory for caching a page in association with the memory-mapped access may be based on whether the persistent memory possess available data storage space to cache a page associated with the memory-mapped access of the file.

[24] The persistent memory page-caching module 106 may facilitate caching a page, associated with memory-mapped access of a file, on persistent memory of a computer system. Where the page is a new page and the memory-mapped access comprises a request to cache the new page in association with the memory-mapped access, the persistent memory page-caching module 106 may cache the page on the persistent memory in response to the request. Where the page is an existing page and the memory-mapped access comprises accessing the existing page in association with the memory-mapped access, the persistent memory page-caching module 106 may cache the page on the persistent memory in response to the existing page being accessed (e.g., by an application).

[25] For some examples, the page may constitute at least a portion, if not all, of the file being accessed via the memory-mapped access. Where the file is opened by the computer system, the file is memory-mapped by the computer system, and persistent memory is determined to be available (e.g., by the persistent memory availability module 104), when a new page needs to be cached in association with the memory mapping (e.g., as a result using a mmap system call), the persistent memory page-caching module 106 may cause the new page to be cached on the persistent memory. Where the persistent memory is determined not to be available when a new page needs to be cached, the persistent memory page-caching module 106 may cause the new page to be cached on a non-persistent memory page cache of the computer system. Where the persistent memory is determined to be available after an existing page has been cached on a non-persistent memory page cache of the computer system, the persistent memory page-caching module 106 may cause the existing page to migrate from the non-persistent memory page cache to the persistent memory. The persistent memory page-caching module 106 may cause this migration after or upon subsequent access of the existing page (e.g., access virtual memory area claimed by a previous mmap request involving the non-persistent memory page cache). As described herein, migrating the existing page from the non-persistent memory page cache to the persistent memory may involve copying the page from the non-persistent memory page cache to the persistent memory, and then removing the page from the non-persistent memory page cache.

[26] FIG. 2 illustrates an example page-caching system 200 for page caching using persistent memory during memory-mapped access of a file. As shown, the page-caching system 200 includes a memory-mapped access detection module 202, a persistent memory availability module 204, a persistent memory page-caching module 206, and a persistent memory page-flushing module 208. Depending on the example, the page-caching system 200 may be part of a computer system, such as a desktop, laptop, hand-held computing device (e.g., personal digital assistants, smartphones, tablets, etc.), workstation, server or other device that includes a processor. In various examples, the components or

the arrangement of components in the page-caching system 200 may differ from what is depicted in FIG. 2.

[27] The memory-mapped access detection module 202, the persistent memory availability module 204, and the persistent memory page-caching module 206 may be respectively similar to the memory-mapped access detection module 102, the persistent memory availability module 104, and the persistent memory page-caching module 106 described above with respect to the page-caching system 100 of FIG. 1.

[28] The persistent memory page-flushing module 208 may facilitate flushing a page from the persistent memory to a non-persistent memory file system, where the flushing may be based on a set of criteria. The page may be one previously cached on the persistent memory by the persistent memory page-caching module 206. As described herein, the persistent memory page-caching module 206 may have cached the page on the persistent memory in response to the memory-mapped access detection module 202 detecting memory-mapped access of a file, where the page is associated with the memory-mapped access. As also described herein, the persistent memory page-caching module 206 may have cached the page on the persistent memory based on availability of the persistent memory as determined by the persistent memory availability module 204.

[29] Depending on the example, the set of criteria for flushing the page may include a criterion relating to a condition of the memory-mapped access, such as whether the memory-mapped access has been disestablished (e.g., unmapped), frequency of use of the memory-mapped access, or data storage space of the persistent memory utilized for the memory-mapped access. For instance, the persistent memory page-flushing module 208 may cause the page to be flushed from the persistent memory to the non-persistent memory file system when the memory-mapped access of the file is disestablished, which may occur when the computer system explicitly executes an un-mapping system call. The set of criteria may include a criterion relating to whether the file associated with the memory-mapped access has been closed by the computer system. For example, the persistent memory page-flushing module 208 may cause the page to be flushed from the persistent memory to the non-persistent memory file system

upon the computer system closing the file, where such closure may be detected by the persistent memory page-flushing module 208. Additionally, the set of criteria may include a criterion relating to a condition of the persistent memory. For instance, the persistent memory page-flushing module 208 may cause the page to be flushed from the persistent memory to the non-persistent memory file system when availability of the persistent memory is determined (e.g., by the persistent memory availability module 204) to be limited or exhausted. In another instance, the persistent memory page-flushing module 208 may cause the page to be flushed from the persistent memory to the non-persistent memory file system when it is determined (e.g., by the persistent memory page-flushing module 208) that data store space occupied on the persistent memory by the page may be better utilized for another use on the computer system. The better utilization of the persistent memory may include any alternative use of the persistent memory that would provide an operational advantage over using it to specifically cache the page to be flushed, or over using it for page caching in general. Examples of alternative utilizations may include long term storage of a file (e.g., the file associated with the memory-mapped access), caching of a new page (e.g., associated with the memory-mapped access), caching of an existing page currently on the non-persistent memory page cache, or caching of a page associated with a higher priority memory-mapped access.

[30] FIG. 3 illustrates an example computer system 300 including the page-caching system 100. The computer system 300 may be any computing device having a processor and memory, such as a desktop, laptop, hand-held computing device (e.g., personal digital assistants, smartphones, tablets, etc.), workstation, server, or other device that includes a processor. As shown, the computer system 300 includes an application module 302, a persistent memory module 304, the page-caching system 200, a file system module 306, and a communications module 308. In various examples, the components or the arrangement of components in the computer system 300 may differ from what is depicted in FIG. 3.

[31] The application module 302 represents any firmware or software (e.g., software application or operating system) operable on the computer system 300

and capable of accessing (e.g., read from, write to, or modify) a file stored on a file system. For some examples, the application module 302 opens a file from a file system when the application module 302 wishes to access the file and, eventually, may close the file once the application module 302 has completed its access of the file. When the application module 302 accesses a file from a non-persistent memory file system or a remote file system, the file may be memory-mapped using a non-persistent memory page cache (e.g., based on volatile memory) included by the computer system 300 (not shown).

[32] The persistent memory module 304 may comprise persistent memory to store a page. The page may be a new page cached on the persistent memory by the page-caching system 200, or one migrated by the page-caching system 200 to the persistent memory from a non-persistent memory page cache included by the computer system 300.

[33] The file system module 306 may provide the application module 302 with access to a local non-persistent memory file system of the computer system 300, or with access to a remote file system that is remote with respect to the computer system 300. The local non-persistent memory file system of the computer system 300 may include a file system that is implemented using a non-persistent memory device (e.g., a hard disk drive [HDD] or a solid-state drive [SSD]) local to the computer system 300, and that is capable of providing memory-mapped access to a file stored thereon using a non-persistent memory page cache of the computer system 300. The remote file system may be a file system that is maintained by another computer system with which the computer system 300 can communicate (e.g., over a communications network), and that is capable providing memory-mapped access to a file stored thereon using a non-persistent memory page cache of the computer system 300. The file system of the remote file system may be associated with a non-persistent memory device or a persistent memory device accessible by the other computer system.

[34] The page-caching system 200 facilitates various operations described herein on the computer system 300. For instance, the page-caching system 200 may detect memory-mapped access of a file by the application module 302, from a local non-persistent memory file system of the computer system 300 or a remote

file system, through the file system module 306. In response, the page-caching system 200 may determine availability of the persistent memory of the computer system 300 accessible through the persistent memory module 304. Based on this determination, the page-caching system 200 may cache a page, associated with the memory-mapped access, on the persistent memory of the computer system 300. As described herein, caching the page may involve caching a new page associated with the memory-mapped access on the persistent memory of the computer system 300, or migrating a page existing on a non-persistent memory page cache of the computer system 300 to the persistent memory. Eventually, based on a set of criteria, the page-caching system 200 may flush a page from the persistent memory of the computer system 300 to the non-persistent memory file system or the remote file system from which the file is being accessed by the computer system 300.

[35] For some examples, the file system module 306 may include a virtual file system module that facilitates the page-caching system 200 caching a page on persistent memory of the computer system 300. The virtual file system may provide the application module 302 with a virtual file system having a single file namespace but being associated with (e.g., based on) a plurality of separate file systems of similar (e.g., two or more non-persistent memory file systems) or different types (e.g., a non-persistent memory file system and a persistent memory file system).

[36] The communications module 308 may facilitate communication of the computer system 300 with another computer system over a communications network that that permits data communication. The communications network may comprise one or more local or wide-area communications networks, such as the Internet, WiFi networks, cellular networks, private networks, public networks, and the like. As described herein, the computer system 300 may access a remote file system maintained by another computer system, and the communications module 308 may facilitate the exchange of network data packets between the computer system 300 and the other computer system that facilitate the access.

[37] FIG. 4 illustrates example data flow in an example computing environment 400 that includes the page-caching system 200. As shown, the

computing environment 400 includes an application 402, a virtual file system module 404, the page-caching system 200, a non-persistent memory page cache 406, a file system 408, a non-volatile memory device 410, and persistent memory 412. According to some examples, the virtual file system module 404 provides the application 402 with a virtual file system having a single file namespace based at least on the file system 408, which is associated with the non-volatile memory device 410. The non-persistent memory page cache 406 may support memory-mapped access, by the application 402, of files stored on the file system 408.

[38] During an operation in the computing environment 400, the application 402 may open and access a given file stored on the file system 408 and provided by the virtual file system module 404 as part of a single name space. When the given file is accessed by the application 402 through the virtual file system module 404, the given file may be memory-mapped. The page-caching system 200 may detect the memory-mapped access of the given file and, in response, determine availability of the persistent memory 412 for caching a page in association with the memory-mapped access of the given file. Where the page-caching system 200 determines that the persistent memory 412 is available, the page-caching system 200 may cache the page on the persistent memory 412. Where the page-caching system 200 determines that the persistent memory 412 is available, the page-caching system 200 may cache the page on the non-persistent memory page cache 406.

[39] As described herein, where the memory-mapped access detected by the page-caching system 200 comprises a request to cache a new page in association with the memory-mapped access, the new page may be cached to the persistent memory 412 and the application 402 may access the new page from the persistent memory 412 via a direct memory mapping to the persistent memory 412 (as illustrated in FIG. 4). Where the memory-mapped access detected by the page-caching system 200 comprises a request (e.g., by the application 402) to access an existing page on the non-persistent memory page cache 406, the page-caching system 200 may cache the existing page on the persistent memory 412, and may do so by migrating the existing page from the non-persistent memory page cache 406 to the persistent memory 412 (as

illustrated in FIG. 4). After the existing page has been migrated to the persistent memory 412, the existing page may be accessed (e.g., by the application 402) via a direct memory mapping to the persistent memory 412. Eventually, based on a set of criteria described herein, the page-caching system 200 may flush a page cached on the persistent memory 412 from the persistent memory 412 to the file system 408, thereby causing the page to be stored on the non-volatile memory device 410.

[40] With respect to a computer system, the non-persistent memory page cache 406 and the persistent memory 412 may be local to the computer system. For some examples, the file system 408 is a local non-persistent memory file system of the computer system and the non-volatile memory device 410 associated with the file system 408 is also local to the computer system. For various examples, the file system 408 is a remote file system maintained by another computer system and the non-volatile memory device 410 associated with the file system 408 is remote with respect to the computer system.

[41] FIG. 5 illustrates an example computer system 500 for page caching using persistent memory during memory-mapped access of a file. As shown, the computer system 500 includes a computer-readable medium 502, a processor 504, persistent memory 506, and non-persistent memory 508. In various examples, the components or the arrangement of components of the computer system 500 may differ from what is depicted in FIG. 5. For instance, the computer system 500 can include more or less components than those depicted in FIG. 5.

[42] The computer-readable medium 502 may be any electronic, magnetic, optical, or other physical storage device that stores executable instructions. For example, the computer-readable medium 502 may be a Random Access Memory (RAM), an Electrically-Erasable Programmable Read-Only Memory (EEPROM), a storage drive, an optical disc, or the like. The computer-readable medium 502 can be encoded to store executable instructions that cause the processor 504 to perform operations in accordance with various examples described herein. In various examples, the computer-readable medium 502 is non-transitory. As shown in FIG. 5, the computer-readable medium 502 includes memory-mapped

access detection instructions 510 and persistent memory page-caching instructions 512.

[43] The processor 504 may be one or more central processing units (CPUs), microprocessors, or other hardware devices suitable for retrieval and execution of one or more instructions stored in the computer-readable medium 502. The processor 504 may fetch, decode, and execute the instructions 510 and 512 to enable the computer system 500 to perform operations in accordance with various examples described herein. For some examples, the processor 504 includes one or more electronic circuits comprising a number of electronic components for performing the functionality of one or more of the instructions 510 and 512.

[44] The persistent memory 506 may include a persistent memory device, such as a phase change memory (PCM) device. In some examples, the persistent memory 506 may support a local persistent memory file system at the computer system 500, which may be provided to an application on the computer system 500 as part of a virtual file system.

[45] The non-persistent memory 508 may include random access memory device, such as dynamic random access memory (DRAM), which may serve as the basis for a non-persistent memory page cache on the computer system 500. As described herein, a non-persistent memory page cache, based on the non-persistent memory 508, may support memory-mapped access of files from a file system (e.g., non-persistent memory file system or remote file system) accessible by the computer system 500.

[46] The memory-mapped access detection instructions 510 may cause the processor 504 to detect memory-mapped access of a file from a remote file system having a non-persistent memory page cache. As described herein, the remote file system may be a file system maintained by a computer system remote with which the computer system 500, and memory-mapped access of files on the remote file system may be supported by a non-persistent memory page cache based on the non-persistent memory 508. The persistent memory page-caching instructions 512 may cause the processor 504 to cache a page on the persistent

memory 506 based on availability of the persistent memory 506 for caching the page in association with the memory-mapped access of the file.

[47] FIG. 6 illustrates an example method 600 performed by an example computer system to facilitate page-caching using persistent memory during memory-mapped access of a file. Although execution of the method 600 is described below with reference to the page-caching system 100 of FIG. 1 and a computer system, execution of the method 600 by other suitable systems or devices may be possible. The method 600 may be implemented in the form of executable instructions stored on a computer-readable medium or in the form of electronic circuitry.

[48] In FIG. 6, the method 600 begins at block 602, with the memory-mapped access detection module 102 detecting memory-mapped access of a file, by a computer system, from a remote file system having a non-persistent memory page cache. At block 604, the method 600 continues with the persistent memory page-caching module 106 caching a page, associated with the memory-mapped access of the file, on persistent memory of the computer system based on availability of the persistent memory for page caching. As described herein, for some examples, the availability of the persistent memory on the computer system is determined by the persistent memory availability module 104.

[49] FIG. 7 illustrates an example method 700 performed by an example computer system to facilitate page-caching using persistent memory during memory-mapped access of a file. Although execution of the method 700 is described below with reference to the page-caching system 200 of FIG. 2 and a computer system, execution of the method 700 by other suitable systems or devices may be possible. The method 700 may be implemented in the form of executable instructions stored on a computer-readable medium or in the form of electronic circuitry.

[50] In FIG. 7, the method 700 begins at block 702, with the memory-mapped access detection module 202 detecting memory-mapped access of a file, by a computer system, from a remote file system having a non-persistent memory page cache. At block 704, the method 700 continues with the persistent memory availability module 204 determining availability of persistent memory of the

computer system for page caching. At block 706, the method 700 continues with the persistent memory page-caching module 206 caching a page, associated with the memory-mapped access of the file, on the persistent memory based on the availability determined at block 704. At block 708, the method 700 continues with the persistent memory page-flushing module 208 flushing the page, associated with the memory-mapped access of the file, from the persistent memory to the remote file system.

[51] In the foregoing description, numerous details are set forth to provide an understanding of the subject disclosed herein. However, various examples may be practiced without some or all of these details. Some examples may include modifications and variations from the details discussed above. It is intended that the appended claims cover such modifications and variations.

CLAIMS

1. A page-caching system, comprising:

a memory-mapped access detection module to detect memory-mapped access of a file from a non-persistent memory file system having a non-persistent memory page cache;

a persistent memory availability module to determine availability of persistent memory for caching a page in association with the memory-mapped access of the file; and

a persistent memory page-caching module to cache the page on the persistent memory based on the availability of the persistent memory.

2. The page-caching system of claim 1, wherein the page is a new page, and the memory-mapped access comprises a request to cache the new page in association with the memory-mapped access of the file.

3. The page-caching system of claim 1, wherein the page is an existing page cached on the non-persistent memory page cache, and the memory-mapped access comprises a request to access the existing page in association with the memory-mapped access of the file.

4. The page-caching system of claim 3, wherein caching the existing page on the persistent memory comprises migrating the existing page from the non-persistent memory page cache to the persistent memory.

5. The page-caching system of claim 1, comprising a persistent memory page-flushing module to flush the page from the persistent memory to the non-persistent memory file system based on a set of criteria that includes a criterion relating to a condition of the memory-mapped access.

6. The page-caching system of claim 1, comprising a persistent memory page-flushing module to flush the page from the persistent memory to the non-persistent memory file system based on a set of criteria that includes a criterion relating to whether the file is closed.

7. The page-caching system of claim 1, comprising a persistent memory page-flushing module to flush the page from the persistent memory to the non-persistent memory file system based on a set of criteria that includes a criterion relating to a condition of the persistent memory.

8. A non-transitory computer readable medium having instructions stored thereon, the instructions being executable by a processor of a computer system, the instructions causing the processor to:

detect memory-mapped access of a file from a non-persistent memory file system having a non-persistent memory page cache on the computer system; and

cache a page on persistent memory based on availability of the persistent memory to cache the page in association with the memory-mapped access of the file.

9. The non-transitory computer readable medium of claim 8, wherein the page is a new page, and the memory-mapped access comprises a request to cache the new page in association with the memory-mapped access of the file.

10. The non-transitory computer readable medium of claim 8, wherein the page is an existing page cached on the non-persistent memory page cache, and the memory-mapped access comprises a request to access the existing page in association with the memory-mapped access of the file.

11. A method, comprising:

detecting, by a computer system, memory-mapped access of a file from a remote file system having a non-persistent memory page cache; and

caching, by the computer system, a page on persistent memory based on availability of the persistent memory for caching the page in association with the memory-mapped access of the file.

12. The method of claim 11, comprising determining, by the computer system, the availability of the persistent memory.

13. The method of claim 11, comprising flushing, by the computer system, the page from the persistent memory to the remote file system based on a set of criteria that includes a criterion relating to a condition of the memory-mapped access of the file.

14. The method of claim 11, comprising flushing, by the computer system, the page from the persistent memory to the remote file system based on a set of criteria that includes a criterion relating to whether the file is closed.

15. The method of claim 11, comprising flushing, by the computer system, the page from the persistent memory to the remote file system based on a set of criteria that includes a criterion relating to a condition of the persistent memory.

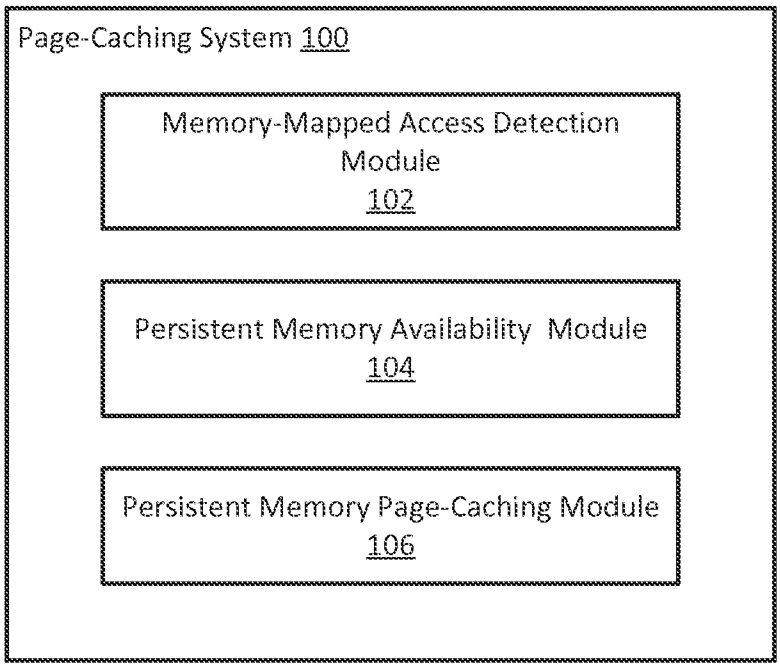
Page-Caching System 100
Memory-Mapped Access Detection Module 102
Persistent Memory Availability Module 104
Persistent Memory Page-Caching Module 106

FIG. 1

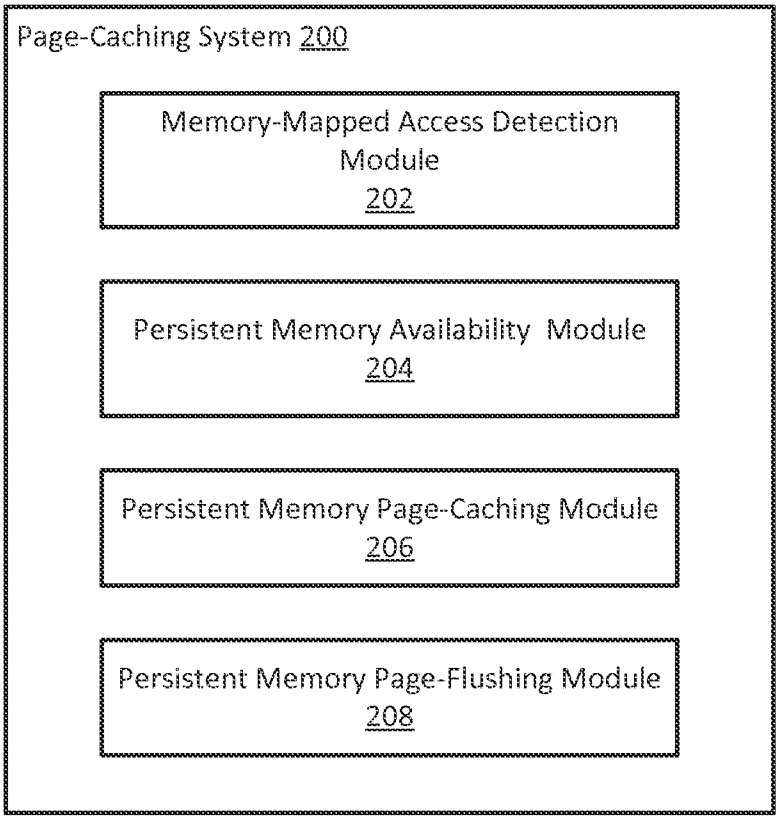
Page-Caching System 200
Memory-Mapped Access Detection Module 202
Persistent Memory Availability Module 204
Persistent Memory Page-Caching Module 206
Persistent Memory Page-Flushing Module 208

FIG. 2

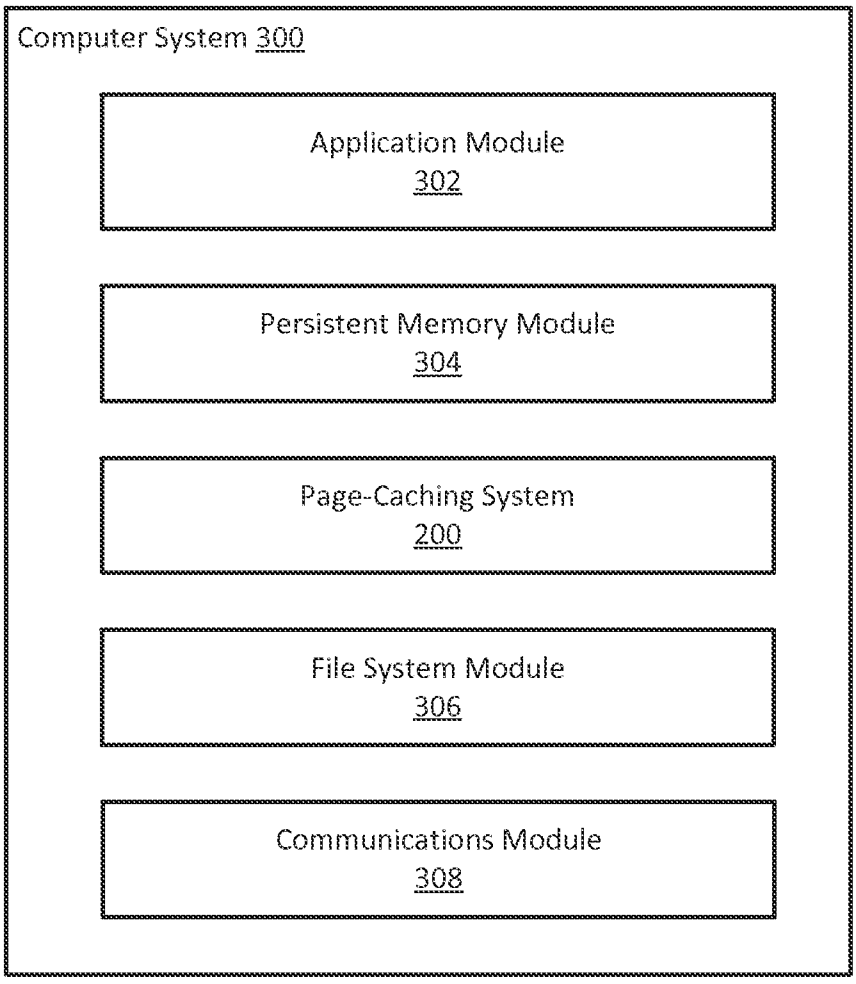
Computer System 300
Application Module
302
Persistent Memory Module
304
Page-Caching System
200
File System Module
306
Communications Module
308

FIG. 3

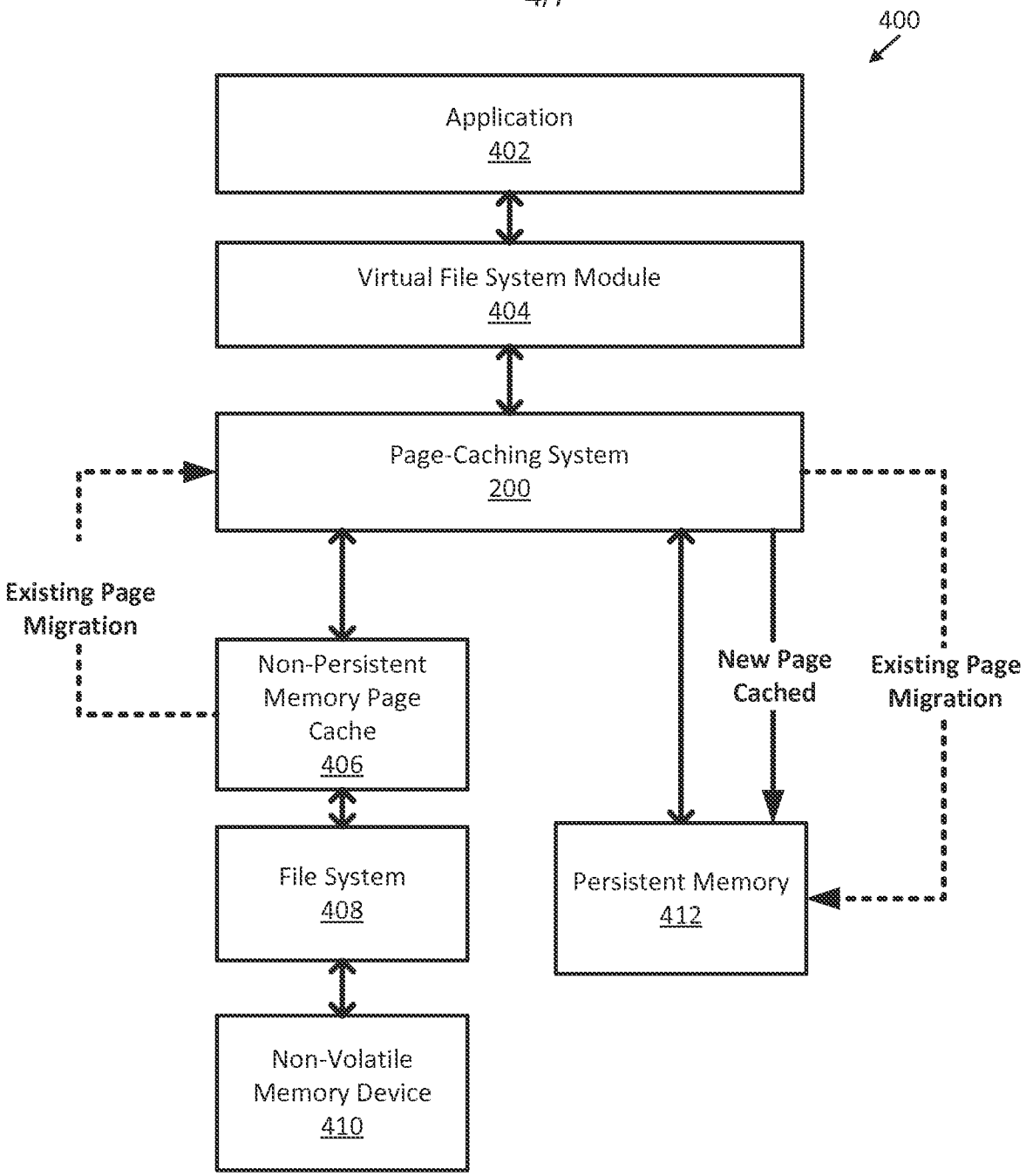
400
Application
402
Virtual File System Module
404
Page-Caching System
200
Existing Page Migration
Non-Persistent Memory Page Cache
406
File System
408
Non-Volatile Memory Device
410
New Page Cached
Existing Page Migration
Persistent Memory
412

FIG. 4

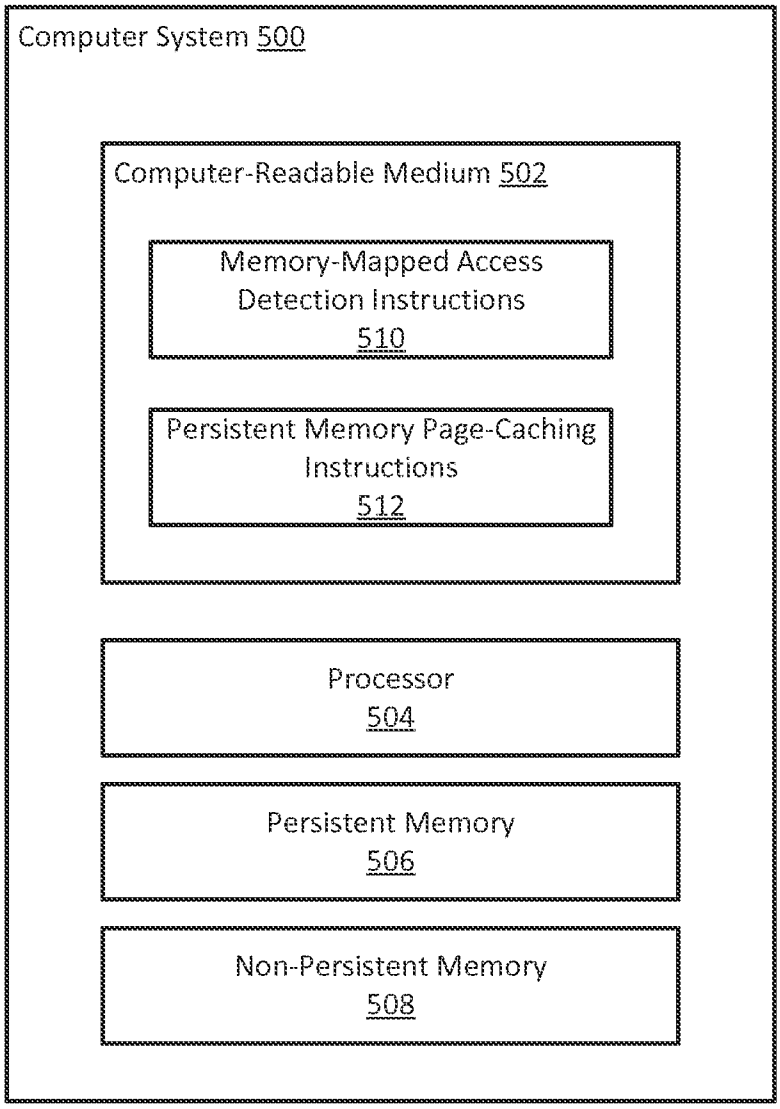
Computer System 500
Computer-Readable Medium 502
Memory-Mapped Access Detection Instructions 510
Persistent Memory Page-Caching Instructions 512
Processor 504
Persistent Memory 506
Non-Persistent Memory 508

FIG. 5

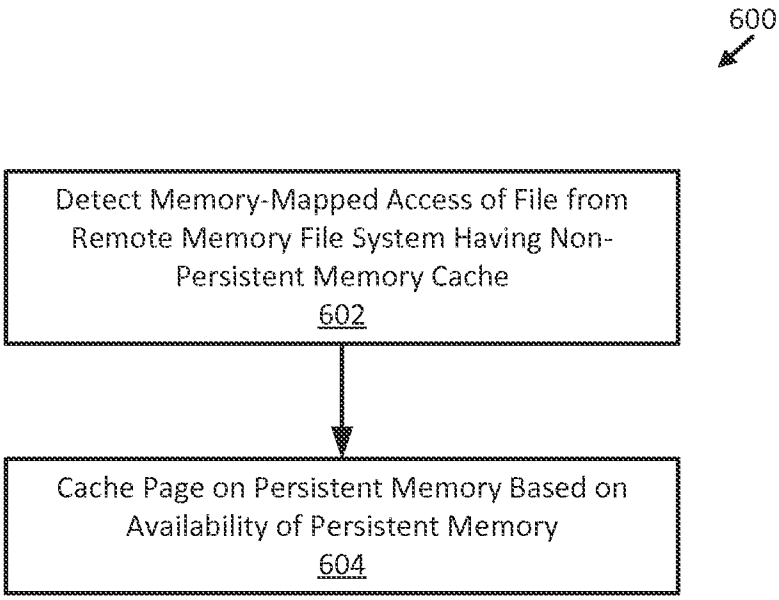
600
Detect Memory-Mapped Access of File from Remote Memory File System Having Non-Persistent Memory Cache
602
Cache Page on Persistent Memory Based on Availability of Persistent Memory
604

FIG. 6

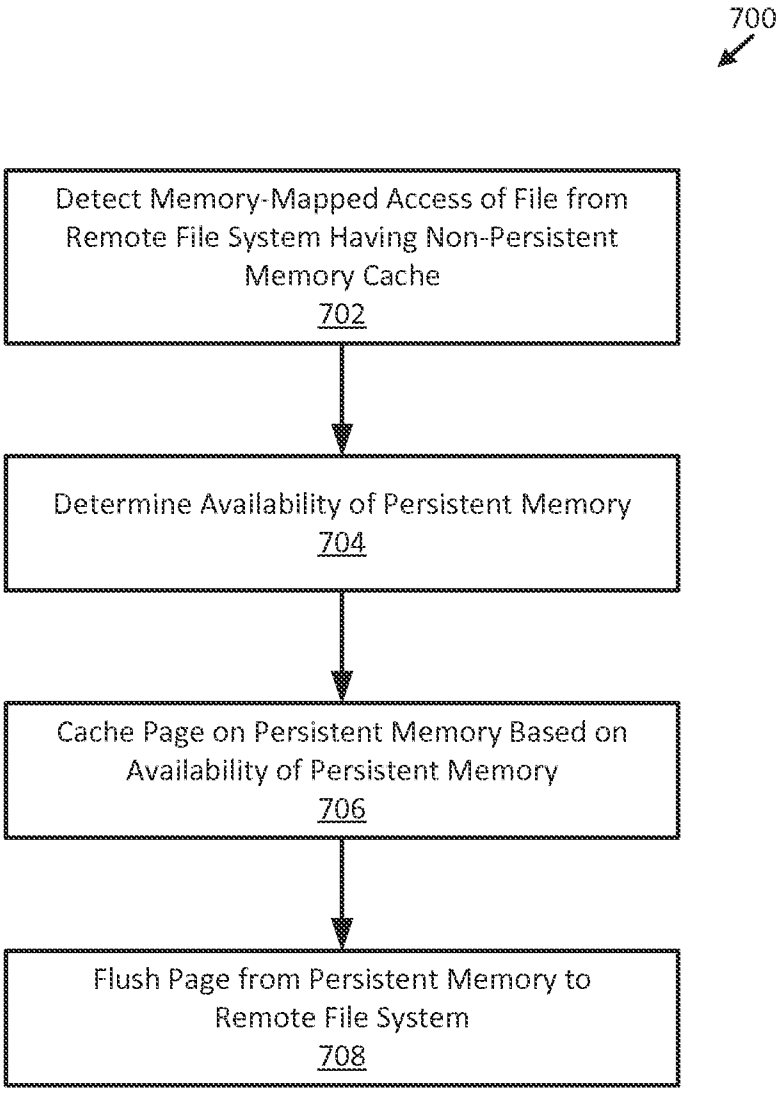
700
Detect Memory-Mapped Access of File from Remote File System Having Non-Persistent Memory Cache
702
Determine Availability of Persistent Memory
704
Cache Page on Persistent Memory Based on Availability of Persistent Memory
706
Flush Page from Persistent Memory to Remote File System
708

FIG. 7

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2015/024159

A. CLASSIFICATION OF SUBJECT MATTER

G06F 12/08(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 12/08; G06F 3/06; G06F 12/00

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:memory-mapped access, persistent memory, non-persistent memory, availability, page-caching, non-volatile cache, volatile cache, storage medium

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2015-0012690 A1 (ROLANDO H. BRUCE et al.) 08 January 2015 See paragraphs [0050], [0094], [0128], [0142]-[0144], [0153], [0197]; claim 1; and figures 1-2.	1-15
A	US 2012-0297147 A1 (KIMMO JUHANI MYLLY et al.) 22 November 2012 See paragraph [0020]; and figure 1.	1-15
A	US 2012-0290785 A1 (CENK ERGAN et al.) 15 November 2012 See paragraph [0067]; and figure 4.	1-15
A	US 2005-0262306 A1 (ILIYAN N. NENOV et al.) 24 November 2005 See paragraph [0035]; and figure 4.	1-15
A	WO 2013-055312 A1 (INTEL CORPORATION) 18 April 2013 See page 2, lines 19-29; 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	Date of mailing of the international search report
22 February 2016 (22.02.2016)	**23 February 2016 (23.02.2016)**

Name and mailing address of the ISA/KR
International Application Division
Korean Intellectual Property Office
189 Cheongsa-ro, Seo-gu, Daejeon Metropolitan City, 35208,
Republic of Korea
Facsimile No. +82-42-472-7140

Authorized officer
COMMISSIONER
Telephone No. +82-42-481-8112

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.
PCT/US2015/024159

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2015-0012690 A1	08/01/2015	None	
US 2012-0297147 A1	22/11/2012	None	
US 2012-0290785 A1	15/11/2012	CN 100470508 C	18/03/2009
		CN 1801121 A	12/07/2006
		EP 1594064 A2	09/11/2005
		EP 1594064 A3	12/09/2007
		JP 2006-004407 A	05/01/2006
		JP 2013-047979 A	07/03/2013
		KR 10-1044220 B1	29/06/2011
		KR 10-2006-0047704 A	18/05/2006
		TW 201227293 A	01/07/2012
		US 2005-246487 A1	03/11/2005
		US 2010-077197 A1	25/03/2010
		US 2012-005422 A1	05/01/2012
		US 7644239 B2	05/01/2010
		US 8041904 B2	18/10/2011
		US 8255645 B2	28/08/2012
US 2005-0262306 A1	24/11/2005	US 7330938 B2	12/02/2008
WO 2013-055312 A1	18/04/2013	US 2013-0268731 A1	10/10/2013