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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
22 December 2016 (22.12.2016)

WIPO | PCT

(10) International Publication Number
WO 2016/202393 A1

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

(21) International Application Number:
PCT/EP2015/063723

(22) International Filing Date:
18 June 2015 (18.06.2015)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant: **HUAWEI TECHNOLOGIES CO., LTD.** [CN/CN]; Huawei Administration Building Bantian, Longgang District, Shenzhen, Guangdong 518129 (CN).

(72) Inventors; and
(71) Applicants *(for US only)*: **WU, Zuguang** [CN/DE]; Huawei Technologies Duesseldorf GmbH, Riesstr.25, 80992 Munich (DE). **METUKI, Assaf** [IL/DE]; Huawei Technologies Duesseldorf GmbH, Riesstr.25, 80992 Munich (DE). **GOCHMAN, Simcha** [IL/DE]; Huawei Technologies Duesseldorf GmbH, Riesstr.25, 80992 Munich (DE).

(74) Agent: **KREUZ, Georg**; Huawei Technologies Duesseldorf GmbH, Riesstr. 8, 80992 Munich (DE).

(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).

Published:

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

(54) **Title**: SYSTEMS AND METHODS FOR DIRECTORY BASED CACHE COHERENCE

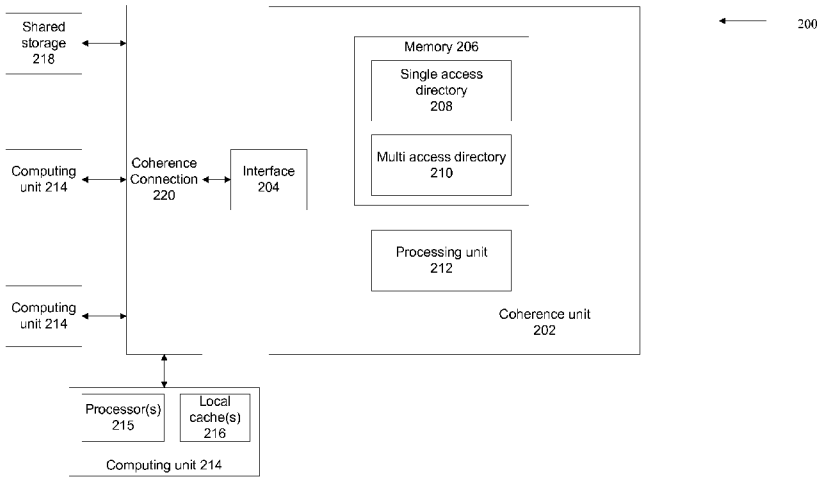

FIG. 2

(57) **Abstract**: A system for memory management accessible by a plurality of computing units, comprises: an interface adapted to receive a request to access one of a plurality of storage regions of a cache storage; and a memory for storing: a single access 5 directory documenting which of the plurality of storage regions is currently assigned to one of the plurality of computing units; and a multi-access directory documenting which of the plurality of storage regions is accessed by at least one of a plurality of computing units.

WO 2016/202393 A1

SYSTEMS AND METHODS FOR DIRECTORY BASED CACHE COHERENCE

BACKGROUND

The present invention, in some embodiments thereof, relates to cache coherence and, more specifically, but not exclusively, to systems and methods for directory based cache coherence.

In a multiprocessing system that includes multiple processing units, cache coherence protocols maintain consistency between local caches (each associated with a processing unit) and a shared memory (accessible by all processing units). Directory-based cache coherence protocols are based on a common directory that maintains the coherence between the local caches. The directory keeps track of the shared memory blocks that are stored in the local caches. Directory based cache coherence protocols may provide local processors with fast coherent access to the shared memory without necessarily broadcasting a request to all local caches.

When the number of processing units and corresponding local caches in the multiprocessing system is increased and/or the size of the local caches is increased, the rate of cache evictions may increase, which may lead to invalidation of cached blocks and reduction in performance. One solution is to increase the size of the common directory, however, the resulting directory size may be larger than the combined local cache size.

As such, different methods have been developed to try and improve performance of the directory rather than increasing the size of the directory, for example, reducing the number of directory entries by avoiding tracking of noncoherent cache blocks, and for example, increasing entry granularity by combining several directory entries into a single one. However, such solutions may provide non-precise tracking, and may require additional snooping mechanisms, coherence recovery mechanisms, or broadcast and flush-based page eviction mechanisms. The mechanisms may degrade performance of the common directory, and/or may not be scalable.

SUMMARY

It is an object of the present invention to provide an apparatus, a system, a computer program product, and a method for management of a memory accessible by multiple computing units.

The foregoing and other objects are achieved by the features of the independent claims. Further implementation forms are apparent from the dependent claims, the description and the figures.

According to a first aspect, a system for memory management accessible by a plurality of computing units, comprises an interface adapted to receive a request to access one of a plurality of storage regions of a cache storage; and a memory for storing: a single access directory documenting which of the plurality of storage regions is currently assigned to one of the plurality of computing units; and a multi-access directory documenting which of the plurality of storage regions is accessed by at least one of a plurality of computing units.

The total size of the directories, including the single access directory and the multi-access directory, may be reduced, as compared to, for example, other directory cache coherence protocol based methods. The size reduction may be achieved while maintaining precise information of each cache block. The size reduction may be achieved with no significant performance penalty. For example, broadcast snooping mechanisms, which use large bandwidth that limits scalability, are not necessarily required.

The implementation based on the single access and multi-access directories is scalable when additional computing units are added to the system. The implementation may reduce the eviction rate, and may reduce the rate of invalidation of cached blocks.

The implementation based on the single access and multi-access directories may be more easily integrated with existing systems based on an existing directory cache coherence protocol.

In a first possible implementation of the system according to the first aspect, the system further comprises a processing unit adapted to manage the single access directory and the multi-access directory to reflect a current allocation of the plurality of storage regions in light of the request.

In a second possible implementation form of the system according to the first aspect as such or according to any of the preceding implementation forms of the first aspect, the system is adapted to manage the single access directory and the multi-access directory by: searching at least one of the single access directory and the multi-access directory for a current allocation status of the requested storage region, according to an outcome of the search performing a new allocation of the requested storage region to one of the plurality of computing units, updating at least one of the single access directory and the multi-access directory to reflect the new allocation.

The single access and multi-access directory design allows for parallel searching of both directories, which may be searched faster as compared to, for example, a single directory design. Searching may be performed based on non-overlapping region tags, which may be searched faster as compared to, for example, overlapped directory structures.

In a third possible implementation form of the system according to the first or second implementation forms of the first aspect, the processing unit is adapted to perform the new allocation according an analysis of at least one sharer bitmap of each of at least some of the plurality of regions.

The sharer bitmap may be analyzed to reduce cache block replacement and/or evictions (to provide space for the new allocation), such as by avoiding replacing frequently accessed regions, which then would need a re-replacement due to their frequent use.

Different eviction (and/or replacement) strategies may be implemented for the single and multi-access directories based on the sharer bitmap, to optimize the eviction rate for both the single and multi-access directories.

In a fourth possible implementation form of the system according to the second or third implementation forms of the first aspect, the new allocation is to a computing unit submitting the request.

In a fifth possible implementation form of the system according to the first aspect as such or according to the first or second implementation forms of the first aspect, the system is adapted to manage the single access directory and the multi-access directory by migrating at least one entry from one of the single access directory and the multi-access directory to the other.

Entries migration between the single access directory and multi-access directory may reduce storage size and/or bandwidth requirements, such as by selecting the optimal entry representation.

In a sixth possible implementation form of the system according to the fifth implementation form of the first aspect, the migrating is triggered when the request is for accessing a region documented in the single access directory by a computing unit which is not the single computing unit.

Entry migration from the single access directory to the multi-access directory may improve precision tracking of the sharers for each block in the region, which may be performed with reduced total bandwidth.

In a seventh possible implementation form of the system according to the fifth or sixth implementation form of the first aspect, the migrating is triggered when the number of blocks of a region documented in the multi-access directory which are cached by the same computing unit of the plurality of computing units exceeds a threshold.

Entry migration from the multi-access directory to the single access directory may require a smaller storage size.

An entry in the multi-access directory may be migrated to the single access directory instead of evicting the multi-access directory entry, which may improve performance as evicting the entry may require more computing resources than migrating and maintaining the entry.

In an eighth possible implementation form of the system according to the first aspect as such or according to any of the preceding implementation forms of the first aspect, the system is configured to manage the single access directory and the multi-access directory by updating an entry associated with the requested storage region in both the single access directory and the multi-access directory for removing a usability indication of the requested storage region from one of the single access directory and the multi-access directory to the other.

When both the single access and multi-access directories are full, the new allocation may be made to the directory having the least number of blocks in the region that are used, optionally with a least recently used strategy, which may reduce the eviction rate.

In a ninth possible implementation form of the system according to the first aspect as such or according to any of the preceding implementation forms of the first aspect, the system is adapted to monitor at least one sharer bitmap of each of at least some of the plurality of regions and to reallocate any of the plurality of storage regions among the plurality of computing units accordingly.

The reallocation may improve efficiency of storage in the directories, by reducing the bandwidth penalty.

In a tenth possible implementation form of the system according to the first aspect as such or according to any of the preceding first to eighth implementation forms of the first aspect, the system is adapted to manage the single access directory and the multi-access directory by identifying a least recently used region among the plurality of regions according to an analysis of the single access directory and allocating the least recently used region as the new allocation.

Selection of the least recently used region in the single access directory may reduce eviction rates.

In an eleventh possible implementation form of the system according to the first aspect as such or according to any of the preceding implementation forms of the first aspect, the system is adapted to manage the single access directory and the multi-access directory by identifying a least shared region among the plurality of regions according to an analysis of the multi-access directory and allocating the least shared region as the new allocation.

Selection of the least shared region in the multi access directory may reduce eviction rates.

In a twelfth possible implementation form of the system according to the first aspect as such or according to any of the preceding first to tenth implementation forms of the first aspect, the single access directory and the multi-access directory comprises a plurality of status values each indicative of a usability status for another of the plurality of storage regions.

In a thirteenth possible implementation form of the system according to the first aspect as such or according to any of the preceding implementation forms of the first aspect, the single access directory comprises a plurality of single access records each document a usage of a plurality of region blocks of one of the plurality of regions by a single computing unit of the plurality of computing units and the multi-

access directory comprises a plurality of multiple access records each document for each one of the plurality of computing units a usage state of a plurality of region blocks of one of the plurality of regions.

Although the number of single access records in the single access directory may be much larger than the number of multiple access records in the multi-access directory, the size of each single access record may be much smaller than the size of each multi-access record, especially when the region granularity is larger (i.e., regions defined by records of the single access directory are associated with many blocks that are converged into the same region).

In a fourteenth possible implementation form of the system according to the first aspect as such or according to any of the preceding implementation forms of the first aspect, the cache storage is a last level cache (LLC) storage and the request is a request to access the LLC storage and wherein the plurality of computing units are processors of a multi processor unit, the plurality of computing units share the cache storage for performing multiple processing tasks in parallel.

The single access and multi-access directory implementation is scalable, such as at different architecture levels, allowing for tracking of sharing by individual processors, and/or sharing by a cluster of processors. For example, at a low architectural level, the single access directory may track individual processors, with the multi-access directory tracking multiple processors. In another example, which may be implemented as a higher architectural level optionally over the lower level, the single access directory may track individual clusters (each cluster including multiple processors), and the multi-access directory tracking multiple clusters.

According to a second aspect, a method for memory management comprises: receiving requests to access a plurality of storage regions of a cache storage from a plurality of computing units; managing a single access directory documenting which of the plurality of storage regions is currently assigned to a single computing unit of the plurality of computing units and a multi-access directory documenting which of the plurality of storage regions is accessed by at least some of the plurality of computing units to reflect an access given in response to the requests.

According to a third aspect, a computer program comprises a readable storage medium storing program code thereon for managing storage region allocation, the program code comprises: instructions for receiving requests to access a plurality of

storage regions of a cache storage from a plurality of computing units; instructions for managing a single access directory documenting which of the plurality of storage regions is currently assigned to a single computing unit of the plurality of computing units and a multi-access directory documenting which of the plurality of storage regions is accessed by at least some of the plurality of computing units to reflect an access given in response to the requests.

Unless otherwise defined, all technical and/or scientific terms used herein have the same meaning as commonly understood by one of ordinary skill in the art to which the invention pertains. Although methods and materials similar or equivalent to those described herein can be used in the practice or testing of embodiments of the invention, exemplary methods and/or materials are described below. In case of conflict, the patent specification, including definitions, will control. In addition, the materials, methods, and examples are illustrative only and are not intended to be necessarily limiting.

BRIEF DESCRIPTION OF THE SEVERAL VIEWS OF THE DRAWINGS

Some embodiments of the invention are herein described, by way of example only, with reference to the accompanying drawings. With specific reference now to the drawings in detail, it is stressed that the particulars shown are by way of example and for purposes of illustrative discussion of embodiments of the invention. In this regard, the description taken with the drawings makes apparent to those skilled in the art how embodiments of the invention may be practiced.

In the drawings:

FIG. 1A is a flowchart of a method for management of a shared storage accessible by multiple computing units, using a single access directory and a multi-access directory, in accordance with some embodiments of the present invention;

FIG. 1B is a flowchart of some features for management of the single access directory and the multi-access directory based on a received request to access one or more storages regions of the shared storage, in accordance with some embodiments of the present invention;

FIG. 2 is a block diagram of components of a system that manages a shared storage accessible by multiple computing units, using a single access directory and a

multi-access directory, in accordance with some embodiments of the present invention;

FIG. 3 is a schematic diagram of examples of data structures for implementing the single access directory and the multi-access directory, in accordance with some embodiments of the present invention; and

FIG. 4 is a dataflow chart of an implementation for management of the shared storage accessible by multiple computing units, using the single access directory and multi-access directory, in accordance with some embodiments of the present invention.

DETAILED DESCRIPTION

The present invention, in some embodiments thereof, relates to cache coherence and, more specifically, but not exclusively, to systems and methods for directory based cache coherence.

An aspect of some embodiments of the present invention relates to a system that includes a single access directory and a multi-access directory, for management of a common storage (e.g., memory) accessible by multiple computing units which may be organized for distributed and/or parallel processing. The single access directory documents which storage region(s) of the common storage is assigned to which single computing unit. Storage blocks within each storage region documented in the single access directory are assigned to only one computing unit. The multi-access directory documents which storage region(s) of the common memory is accessed by which computing units, optionally multiple computing units. Each storage block within each storage region documented in the multi-accessed directory may be accessed by multiple computing units.

Each directory may be implemented with different entry formats and/or different sizes for each entry. Cache coherence may be maintained using a relatively small total size (i.e., including the total size of the single access directory and the multi-access director) without additional performance penalties. Each entry of the single-access directory is designed to have a relatively small size (relative to the multi-access directory), such that a relatively large number of single-access directory entries (relative to the multi-access directory) results in a relatively small total

directory size. The multi-access directory is designed to have a relatively small number of entries (relative to the single access directory), such that a relatively large entry size (relative to the single access directory) results in a relatively small total directory size.

The single access directory and the multi-access directory may be independently processed. Optionally, the single and multi-access directories are searched in parallel for a current allocation status when a request to access the respective storage region is received from one of the computing units.

The system may migrate entries between the directories. Migration may be according to a selection of the entry representation having the smallest size in view of the accessing computer unit(s). Optionally, an entry in the single access directory is migrated to the multi-access directory. Optionally, the single to multi-access directory migration is triggered by another computing unit that attempts to access a storage region represented as an entry assigned to a single computing unit in the single access directory. Alternatively, an entry in the multi-access directory is migrated to the single access directory. Optionally, the entry is migrated to the single access directory when most or all storage blocks in the storage region are accessed by the same computing unit.

Optionally, eviction of entries from the single access directory is according to a least number of blocks used (within the storage region documented by the respective entry). Optionally, eviction of entries from the multi-access directory is according to a least number of computing units sharing the storage region documented by the respective entry. The evicted entry may be migrated into the other directory or deleted.

It is noted that the system as described herein may be implemented as an apparatus, as a program module (in hardware and/or software), as a system, as a method, and/or as a computer program product.

Before explaining at least one embodiment of the invention in detail, it is to be understood that the invention is not necessarily limited in its application to the details of construction and the arrangement of the components and/or methods set forth in the following description and/or illustrated in the drawings and/or the Examples. The invention is capable of other embodiments or of being practiced or carried out in various ways.

The present invention may be a system, a method, and/or a computer program product. The computer program product may include a computer readable storage medium (or media) having computer readable program instructions thereon for causing a processor to carry out aspects of the present invention.

The computer readable storage medium can be a tangible device that can retain and store instructions for use by an instruction execution device. The computer readable storage medium may be, for example, but is not limited to, an electronic storage device, a magnetic storage device, an optical storage device, an electromagnetic storage device, a semiconductor storage device, or any suitable combination of the foregoing.

Computer readable program instructions described herein can be downloaded to respective computing/processing devices from a computer readable storage medium or to an external computer or external storage device via a network, for example, the Internet, a local area network, a wide area network and/or a wireless network.

The computer readable program instructions may execute entirely on the user's computer, partly on the user's computer, as a stand-alone software package, partly on the user's computer and partly on a remote computer or entirely on the remote computer or server. In the latter scenario, the remote computer may be connected to the user's computer through any type of network, including a local area network (LAN) or a wide area network (WAN), or the connection may be made to an external computer (for example, through the Internet using an Internet Service Provider). In some embodiments, electronic circuitry including, for example, programmable logic circuitry, field-programmable gate arrays (FPGA), or programmable logic arrays (PLA) may execute the computer readable program instructions by utilizing state information of the computer readable program instructions to personalize the electronic circuitry, in order to perform aspects of the present invention.

Aspects of the present invention are described herein with reference to flowchart illustrations and/or block diagrams of methods, apparatus (systems), and computer program products according to embodiments of the invention. It will be understood that each block of the flowchart illustrations and/or block diagrams, and combinations of blocks in the flowchart illustrations and/or block diagrams, can be implemented by computer readable program instructions.

The flowchart and block diagrams in the Figures illustrate the architecture, functionality, and operation of possible implementations of systems, methods, and computer program products according to various embodiments of the present invention. In this regard, each block in the flowchart or block diagrams may represent a module, segment, or portion of instructions, which comprises one or more executable instructions for implementing the specified logical function(s). In some alternative implementations, the functions noted in the block may occur out of the order noted in the figures. For example, two blocks shown in succession may, in fact, be executed substantially concurrently, or the blocks may sometimes be executed in the reverse order, depending upon the functionality involved. It will also be noted that each block of the block diagrams and/or flowchart illustration, and combinations of blocks in the block diagrams and/or flowchart illustration, can be implemented by special purpose hardware-based systems that perform the specified functions or acts or carry out combinations of special purpose hardware and computer instructions.

Reference is now made to FIG. 1A which is a flowchart of a method for management of a shared storage accessible by multiple computing units, using a single access directory and a multi-access directory, in accordance with some embodiments of the present invention. Reference is also made to FIG. 2 which is a block diagram of components of a system that manages a common storage accessible by multiple computing units, by utilizing a design of a single access directory and a multi-access directory that relatively reduce the total size required for cache coherence management (i.e., the single and the multi-access directories), as compared to, for example, other directory based cache coherence protocol methods, in accordance with some embodiments of the present invention.

The total size of the directory, including the single access directory and the multi-access directory, may be reduced by the design and/or management. The size reduction may be achieved while maintaining precise information of each cache block of each region. The size reduction may be achieved with no significant performance penalty, for example, as compared to existing directory based cache coherence methods. For example, broadcast snooping mechanisms, which use large bandwidth that limits scalability, are not necessarily required.

The implementation based on the single access and multi-access directories is scalable when additional computing units are added to the system. The

implementation may reduce the eviction rate, and may reduce the rate of invalidation of cached blocks.

The implementation based on the single access and multi-access directories may be more easily integrated with existing systems based on an existing directory cache coherence protocol.

System 200 includes a coherence unit 202 (and/or processing unit 212) for maintaining cache coherence between multiple computing units 214 sometimes referred to herein as sharers (optionally each organized as a computing cluster within a distributed computing system), and a shared storage 218 (optionally a cache storage) accessible to computing units 214. Examples of shared storage 218 include: a random access memory (RAM), read-only memory (ROM), and/or a storage device, for example, non-volatile memory, magnetic media, semiconductor memory devices, hard drive, removable storage, and optical media (e.g., DVD, CD-ROM). Each computing unit 214 includes one or more processors 215, and one or more local caches 216. Examples of processors 215 include: graphics processing unit(s) (GPU), central processing unit(s) CPU, field programmable gate arrays (FPGA), digital signal processor (DSP), and application specific integrated circuits (ASIC). It is noted that computing units 214 may be homogenous with each other (i.e. similar or same architecture), or heterogeneous relative to one another (i.e., different architectures). Each computing unit 214 may include a single processor, and/or each computing unit 214 may be organized as a distributed system having multiple processors organized as computing clusters, and/or represent a hierarchy of distributed systems and/or computing units. It is noted that local caches 216 may be organized into a hierarchy.

Optionally, computing units 214 are processors organized as part of a multi processor unit. Share storage 218 may be a cache storage, optionally a last level cache (LLC) storage of the multi processor system. In such an architecture, multiple computing units 214 may share shared storage 218, (e.g., LLC) for performing multiple processing tasks in parallel.

Coherence unit 202 and/or processing unit 212 includes or is in communication with a data interface 204 for data communication with computing units 214, optionally via a cache coherence connection 220, for example, a local bus, a local network, the internet, or other communication mechanisms.

Coherence unit 202 and/or processing unit 212 includes or is in communication with a memory 206 storing a single access directory 208 and a multi-access directory 210. Single access directory 208 documents which of the multiple storage regions of shared storage 218 are currently assigned to which one of multiple computing units 214. Single access directory 208 associates a single computing unit 214 to one or more regions of storage 218. Multi-access directory 210 documents which of the multiple storage regions is accessed by which computing units 214. Multi-access directory 210 associates each storage region to multiple (or one) computing units 214.

Coherence unit 202 may include processing unit 212 to manage single access directory 208 and multi-access directory 210. Processing unit 212 may be a separate processor designated to coherence unit 202, integrated with memory 206 as coherence unit 202, or processing unit 212 may be implemented by one or more of computing units 214. It is noted that memory 206, processing unit 212 and interface 204 may be implemented as separate elements, not necessarily organized into coherence unit 202, or coherence unit 202 may be implemented using the separate elements (which may be distributed and/or implemented as components of other devices).

Coherence unit 202 may act as a filter through which computing units 214 ask permission to access entries in shared storage 218. The accessed entries may be copied from shared storage 218 to local cache 216 of computing unit 214, or copied from local cache 216 to shared storage 218. Coherence unit 202 helps ensure that relevant computing units 214 are made aware of any changes made to data in shared storage 218 by another computing unit 214, for example, when one computing unit 214 changes data in shared storage 218, coherence unit 202 may generate instructions to update or invalidate local caches 216 of other computing units 214 accessing the same data.

The single access and multi-access directory implementation is scalable, such as at different architecture levels, allowing for tracking of sharing by individual processors, and/or sharing by a cluster of processors. For example, at a low architectural level, the single access directory may track individual processors, with the multi-access directory tracking multiple processors. In another example, which may be implemented as a higher architectural level optionally over the lower level,

the single access directory may track individual clusters (each cluster including multiple processors), and the multi-access directory tracking multiple clusters.

At 102, one or more requests to access one or more of multiple storage regions of a shared storage, optionally a cache storage, optionally shared storage 218, are received. The request may include a single block in a single region, multiple blocks in a single region, or multiple blocks in multiple regions. The request(s) may be received by coherence unit 202 (and/or processing unit 212) via interface 204, from one or more of the computing units 214. The request(s) may be to access the storage region of shared storage 218.

Optionally, the request is a request to access the LLC (or other cache) storage, optionally made by one or more processors of a multi processor unit sharing the LLC (or other cache storage) for performing multiple processing tasks in parallel.

Reference is now made to FIG. 3, which is a schematic diagram of an example of data structures for implementing the single access directory and the multi-access directory, in accordance with some embodiments of the present invention.

Single access directory 208 includes multiple single access records 302 (one entry is shown for clarity). A region tag 304 of each record 302 defines the respective storage region of shared storage 218. An owner field 306 of each record 302 defines an identification representing a single computing unit of the multiple computing units having access to shared storage 218. Usage of each region block of the region (defined by tag 304) by the single computing device (defined by field 306) is defined by a respective bit map field 308 (also referred to herein as a single owner bitmap). Optionally, each bit of bit map field 308 represents whether or not the respective block is accessed by the single computing unit defined by owner field 306.

When the size of region tag 304 is T, the size of owner field 306 is M, and the number of blocks per region is N (i.e., corresponding to the size of all bit map fields 308), the size of each record 302 entry in single access directory 208 is $T+M+N$.

Multi-access directory 210 includes multiple access records 320 (one entry is shown for clarity). Each record 320 documents, for each one of the computing units accessing shared storage 218, a usage state of multiple region blocks of one of the regions of shared storage 218. Region tag 322 defines the region of shared storage 218. Each computing unit is document by a respective owner bit map field 324 (also referred to herein as a sharer bitmap). Each bit map 324 represents the usage state of

each block of region tag 322 for the respective computing unit. Optionally, each respective owner bit map field 324 (for each computing unit) has a structure corresponding to the full set of bit map fields 308, representing, for each computing unit, which respective blocks are accessed by the respective computing unit. Each bit map field 324 may include an identification representation of the respective computing unit, for example, an owner field (e.g., field 306). Alternatively, the order of each respective bit map fields 324 may represent the identification of the respective computing unit.

When the size of region tag 322 is T, the size of each respective owner bit map field 324 is N (based on the number of blocks per region), and the number of different computing units is M, the size of each record 320 entry in multi-access directory 210 is $T+M*N$.

Although the number of single access records in the single access directory may be much larger than the number of multiple access records in the multi-access directory, the size of each single access record may be much smaller than the size of each multi-access record, especially when the region granularity is larger (i.e., regions defined by records of the single access directory are associated with many blocks that are converged into the same region).

The combined size of both directories may be reduced as the percent of storage blocks (and/or regions) in the shared storage accessed by a single computing unit increases, and/or as the number of blocks in the shared storage that may be converged to regions increases.

Referring now back to FIG. 1, at 104, single access directory 208 and/or multi-access directory 210 are managed, optionally to reflect an access given in response to the request. Optionally, processing unit 212 manages single access directory 208 and/or multi-access directory 210 to reflect a current allocation of the storage regions (e.g., of shared storage 218) in light of the received request. The allocation may be, for example, migration between directories 208 and 210 (in either direction), eviction from one or both directories 208 and 210, and new entry allocation in one or both directories 208 and 210, as described with reference to FIG. 1B.

Reference is now made to FIG. 1B, which is a flowchart of some features for management (e.g., by processing unit 212 of coherence unit 202) of the single access

directory and/or the multi-access directory based on a received request to access one or more storages regions of the shared storage, in accordance with some embodiments of the present invention.

At 118, the single access directory and/or the multi-access directory are searched for a current allocation status of the requested storage region. Searching may be performed according to the region tag of respective entries of the directories. The single access and multi-access directory design allows for parallel searching of both directories, which may be searched faster as compared to, for example, a single directory design. Searching may be performed based on non-overlapping region tags, which may be searched faster as compared to, for example, overlapped directory structures.

Optionally, at 120, a new allocation is performed of the requested storage region to one of the computing units that are allowed to access the shared storage. Optionally, the new allocation is made to the computing unit submitting the request. The new allocation may performed according to the outcome of the search, such as when the search does not identify a match between the requesting computing unit and the requested storage region, which may represent, for example, that the computing unit is attempting to access a new storage region for the first time (or after a threshold time limit has passed without access activity).

Optionally, when the single access directory has available storage space for a new entry, the new allocation is performed in the single access directory. Optionally, with reference to the entry data structure of FIG. 3, a new entry 302 is generated with region tag 304 containing the value of the requested storage space, and owner field 306 containing the value representing one of the computing units, optionally the requesting computing unit. Bit maps 308 corresponding to the requested blocks of the requested region are updated.

Alternatively, when the single access directory is full, and the multi-access directory has available storage space for a new entry, the new allocation is performed in the multi-access directory. Optionally, with reference to the entry data structure of FIG. 3, a new entry 320 is generated with region tag 322 containing the value of the requested storage space, and bits of the owner bit map field 324 corresponding to the value of one the computing units, optionally the requesting computing unit, are

updated to reflect the requested blocks of the requested region as part of the new allocation.

Alternatively, when both the single access directory and the multi-access directory are full, the new allocations is made based on a replacement (or eviction) of a current entry.

Optionally, the entry for replacement (or eviction) is selected according to an analysis of the sharer bitmap that defines block usage for respective computing units (e.g., bit maps 308, and/or owner bit map fields 324), to determine which one or more regions of the shared storage are accessed the least. The sharer bitmap may be analyzed to reduce cache block replacement and/or evictions (to provide space for the new allocation) rates, such as by avoid the replacement of frequently accessed regions, which then would need replacing due to their frequent use. Different eviction and/or replacement strategies may be implemented for the single and multi-access directories based on the sharer bitmap, to optimize the eviction rate for both the single and multi-access directories.

Alternatively or additionally, the entry for replacement is selected from the single access directory by identifying a least recently used region from among the regions of the shared storage defined by the entries in the directory. Optionally, the least recently used region is selected according to a least number of blocks recently used, in the region defined by the entry. The least recently used region may be identified according to an analysis of the single access directory, for example, by tracking, for each region, time related access to any blocks in the region representing recent activity. The least recently used region having the least recently used block(s) may be used for the new allocation. Selection of the least recently used region in the single access directory may reduce eviction rates, for example, as regions that have not been recently used are not expected to be used in the near future as frequency as regions that have been recently used which may have ongoing access activity. The least number of blocks used may be combined with a least recently used policy, for example, by tracking the blocks that were least recently accessed by each or all computing units.

Alternatively or additionally, the entry for replacement is selected from the multi-access directory by identifying a least shared region among the regions of the shared storage defined by the entries in the directory. Optionally, the least shared

region is selected according to an analysis of the multi-access directory, for example, by counting the number of different computing units accessing the same region and selecting regions with the least number of sharing computing units. The least shared region may be combined with a least recently used policy, for example, by tracking the blocks that were least recently accessed by each sharing computing unit. Selection of the least shared region in the multi-access directory may reduce eviction rates, as the probability of another sharer or the same sharer accessing a block in the region may be lower than the probability of access in another region having a greater number of active sharers.

Alternatively or additionally, at 122, the single access directory and the multi-access directory are managed by migrating one or more entries from one of the directories to the other directory. Entries migration between the single access directory and multi-access directory may reduce storage size and/or bandwidth requirements, such as by selecting the optimal entry representation.

Optionally, an entry is migrated from the single access directory to the multi-access directory. The migration may be triggered when the request for accessing a region documented in the single access directory (e.g., in region tag 304 of record 302) is made by a computing unit which is not the single computing unit defined for the existing entry (e.g., in owner field 306). For example, if the request to access a region 0001 is made by computing unit *AAA*, and an entry for that region already exists in the single access directory assigned to computing unit *BBB*, the entry is migrated to the multi-access directory, and units *AAA* and *BBB* are assigned to respective fields of the entry (e.g., owner bit map field 324). Entry migration from the single access directory to the multi-access directory may improve precision tracking of the sharers for each block in the region, which may be performed with reduced total bandwidth.

Alternatively, an entry is migrated from the multi-access directory to the single access directory. The migration may be triggered when the number of blocks of a region documented in the multi-access directory, which are cached by the same computing unit, exceeds a threshold. The threshold may define, for example, the majority of blocks in the region, or all blocks in the region, for example, 70% of the blocks in the region, or 80%, or 90%, or 95%, or 100%, or absolute values may be used for the number of blocks instead of or in addition to the percentages. Migration

may be trigged when the entry in the multi-access directory only documents a single computing unit, for example, other sharers are removed based on a least recently used policy. Migration may be trigged when the entry in the multi-access directory documents multiple sharers, with one sharer dominating access to the region, for example, other sharers accessing one block, and the dominating sharer accessing 90% of the blocks. Entry migration from the multi-access directory to the single access directory may require a smaller storage size. An entry in the multi-access directory may be migrated to the single access directory instead of evicting the multi-access directory entry, which may improve performance as evicting the entry may require more computing resources than migrating and maintaining the entry.

Alternatively or additionally, at 124, an entry is evicted (or deleted or selected for replacement) from the single access directory or from the multi-access directory. Entry eviction may be performed to make room for allocation of a new entry, when one or both directories do not have sufficient storage space for the new entry. When both the single access and multi-access directories are full, the new allocation may be made to the directory having the least number of blocks in the region that are used, optionally with a least recently used strategy, which may reduce the eviction rate.

Optionally, an entry for eviction from the single access directory may be selected based on identifying candidate entries having a least number of blocks used in the region defined by the respective candidate entry.

Optionally, an entry for eviction from the multi-access directory may be selected based on identifying candidate entries having a single sharer. Alternatively, the entry is selected based on identifying regions having the least number of sharers.

Alternatively or additionally, the entry for eviction from the single access directory and/or from the multi-access is based on selecting the number of least recently used regions, optionally based on the greatest number of least recently used blocks per region.

A least recently used mechanism for selecting entries for migration and/or eviction may be implemented by status values associated with, or included within the single access directory and the multi-access directory. The status values are each indicative of a usability status for another of the storage regions. For example, when a storage region (or one or more blocks within the storage region) is accessed, the status values of other storage regions (i.e., the non-accessed storage regions) are

incremented, representing the least recently used mechanism. The status values may be queries to select the least recently used region, for example, based on the highest value (i.e., representing number of accesses to other regions). Alternatively or additionally, the least recently used mechanism may be implemented by updating an entry associated with the requested storage region in both the single access directory and the multi-access directory for removing a usability indication of the requested storage region from one of the directories to the other directory. The usability status may serve as a basis for implementing efficient eviction and/or migration algorithms.

Alternatively or additionally, at 126, one or more sharer bitmaps (e.g., field 324) of each of at least some of the storage regions of the multi-access directory are monitored, optionally, by processing unit 212. Alternatively or additionally, one or more single owner bit maps (e.g., fields 308) of the single owner of at least some of the storage regions of the single access directory are monitored. Monitoring may be performed, for example, based on predefined time intervals, and/or per received request (or after a predefined number of requests have been received).

The sharer bitmaps and/or single owner bitmaps may be analyzed by maintaining precision information for each block of each region, with details regarding the trade-off between directory storage size and bandwidth. The precision information may be stored in a precision table within or associated with coherence unit 202, and/or integrated within the single access and/or multi-access directories. Although the precision table may require additional storage size (which may add to the total directory size), the precision table may improve management of the directories, which may achieve a reduction in bandwidth penalty. Each of the single access and multi-access directories may be optimized with the goal of improving the trade-off between directory size and bandwidth, for example, by migrating entries between directories as described herein. As the directory storage size decreases, more bandwidth may be required to provide the data. As the directory storage size increases, less bandwidth may be required. However, the reduced bandwidth penalty may achieve an overall improvement of the system in terms of the trade-off.

The sharer bitmaps and/or single owner bitmaps may be analyzed to select better candidate region entries for migration and/or eviction, which may improve performance, for example, in terms of reduction of total storage space and/or bandwidth.

Optionally, one or more of the storage region are reallocated among the computing units according to the monitoring and/or analyzing. For example, as described herein, in the single access directory, the bitmaps may be analyzed to identify the least block usage regions as candidates for eviction. For example, as described herein, in the multi-access directory, the sharer bitmaps may be analyzed to identify the least block sharing regions as candidates for eviction and/or migration.

It is noted that different blocks 118-126 may be executed during different iterations, and/or as needed, based on the request. It is noted that blocks 118-126 may be executed in a different order, for example, monitoring in block 126 may occur to identify candidates for eviction (block 124) or migration (block 122), to make room for a new allocation 122. Additional examples are presented with reference to FIG. 4.

Referring now back to FIG. 1A, at 106, the single access directory and/or multi-access directory are updated to reflect the management. Optionally, the single access directory and/or multi-access directory are updated to reflect one or more of: the new allocated one or more entries, migration of one or more entries, eviction of one or more entries, and reallocation of storage. It is noted that the update may occur after the management (as in FIG. 1B), or as part of the management (i.e., the management implies an update as part of the process).

Optionally, at 108, a response is transmitted to the requesting computing unit(s), for example, via interface 204 over coherence connection 220. The response may include a signal representing authorization for the requesting computing unit to access the requested region(s) of shared storage 218.

Alternatively or additionally, coherence unit 202 executes the request from the computing unit, for example, sending a read or write request to shared memory 218 via interface 204 over coherence connection 220.

It is noted that the update of block 106 may occur after execution of block 108, instead of before block 108.

Blocks 102-108 may be iterated, for example, per received request.

Reference is now made to FIG. 4, which is a dataflow chart of an implementation for management of the shared storage accessible by multiple computing units, based on the single access directory and multi-access directory (e.g., as described with reference to FIG. 3), in accordance with some embodiments of the present invention. The method of FIG. 4 may be executed by the system of FIG. 2,

optionally based on one or more blocks described with reference to FIG. 1A and/or 1B. It is noted that in the FIG. the single access directory may be referred to as *private region table (PRT)*, and the multi-access directory may be referred to as *shared region table (SRT)*.

At 402, a last level cache (LLC) miss occurs at a local LLC of a computing unit (or a group of computing units). The miss triggers a request to the single access and/or multi-access directory (e.g., to the coherence unit) to access a shared storage (e.g., as in block 102).

At 404, a search of the directories is performed (e.g., as in block 118) to identify a match within the single access or multi-access directories.

Optionally, at 406, the search results did not identify matches in either directory. A new entry is allocated for the new private region (i.e., single owner since no matches for the same or other owners have been identified by the search), in the single access directory (e.g., as in block 120). When there is no room in the single access directory, the new allocation may be made to the multi-access directory (e.g., as in block 120). Optionally, at 408, when there is no room in either directory, an existing entry may be migrated (e.g., block as in 122) or evicted (e.g., block as in 124). The dataflow proceeds to block 410.

Alternatively, at 414, when the search identifies a matching entry, the matching directory entry region is identified, and the owner(s) accessing the matched region is identified. At 416, the owner(s) of the matching entry are compared to the requesting computing unit. When the owner of the matched entry is the same as the requesting computing unit (e.g., entry assigned to the owner in the single access directory), the dataflow proceeds to block 410. When the requesting computing unit is different than the owner(s) of the matched entry, at block 418 an entry for the matched region is allocated in the multi-access directory including the requesting computing unit as an owner and optionally including the owner and accessed region blocks identified in block 414 (e.g., as in block 120). At 420, the identified matched entry may be evicted from the relevant directory to make room for the new allocation (e.g., as in block 124), the matched entry may be migrated from the single access directory to the multi-access directory including both the new owner and previous owner (e.g. as in block 122), or the matched entry in the multi region table may be

updated to include the requesting computing unit as an owner (e.g., as in block 120). The dataflow proceeds to block 410.

At 410, a read request is transmitted (e.g., by the processing unit of the coherence unit) to the specified LLC, main memory, or other shared storage, to obtain a copy of data stored in the requested memory block and/or region.

Optionally, at 412, the one or more relevant directory entries are updated according to the executed data processing events.

The descriptions of the various embodiments of the present invention have been presented for purposes of illustration, but are not intended to be exhaustive or limited to the embodiments disclosed. Many modifications and variations will be apparent to those of ordinary skill in the art without departing from the scope and spirit of the described embodiments. The terminology used herein was chosen to best explain the principles of the embodiments, the practical application or technical improvement over technologies found in the marketplace, or to enable others of ordinary skill in the art to understand the embodiments disclosed herein.

It is expected that during the life of a patent maturing from this application many relevant caches, computing units, and storage will be developed and the scope of the terms cache, computing unit and storage are intended to include all such new technologies a priori.

As used herein the term "about" refers to ± 10 %.

The terms "comprises", "comprising", "includes", "including", "having" and their conjugates mean "including but not limited to". This term encompasses the terms "consisting of" and "consisting essentially of".

The phrase "consisting essentially of" means that the composition or method may include additional ingredients and/or steps, but only if the additional ingredients and/or steps do not materially alter the basic and novel characteristics of the claimed composition or method.

As used herein, the singular form "a", "an" and "the" include plural references unless the context clearly dictates otherwise. For example, the term "a compound" or "at least one compound" may include a plurality of compounds, including mixtures thereof.

The word "exemplary" is used herein to mean "serving as an example, instance or illustration". Any embodiment described as "exemplary" is not necessarily

to be construed as preferred or advantageous over other embodiments and/or to exclude the incorporation of features from other embodiments.

The word "optionally" is used herein to mean "is provided in some embodiments and not provided in other embodiments". Any particular embodiment of the invention may include a plurality of "optional" features unless such features conflict.

Throughout this application, various embodiments of this invention may be presented in a range format. It should be understood that the description in range format is merely for convenience and brevity and should not be construed as an inflexible limitation on the scope of the invention. Accordingly, the description of a range should be considered to have specifically disclosed all the possible subranges as well as individual numerical values within that range. For example, description of a range such as from 1 to 6 should be considered to have specifically disclosed subranges such as from 1 to 3, from 1 to 4, from 1 to 5, from 2 to 4, from 2 to 6, from 3 to 6 etc., as well as individual numbers within that range, for example, 1, 2, 3, 4, 5, and 6. This applies regardless of the breadth of the range.

Whenever a numerical range is indicated herein, it is meant to include any cited numeral (fractional or integral) within the indicated range. The phrases "ranging/ranges between" a first indicate number and a second indicate number and "ranging/ranges from" a first indicate number "to" a second indicate number are used herein interchangeably and are meant to include the first and second indicated numbers and all the fractional and integral numerals therebetween.

It is appreciated that certain features of the invention, which are, for clarity, described in the context of separate embodiments, may also be provided in combination in a single embodiment. Conversely, various features of the invention, which are, for brevity, described in the context of a single embodiment, may also be provided separately or in any suitable subcombination or as suitable in any other described embodiment of the invention. Certain features described in the context of various embodiments are not to be considered essential features of those embodiments, unless the embodiment is inoperative without those elements.

All publications, patents and patent applications mentioned in this specification are herein incorporated in their entirety by reference into the specification, to the same extent as if each individual publication, patent or patent

application was specifically and individually indicated to be incorporated herein by reference. In addition, citation or identification of any reference in this application shall not be construed as an admission that such reference is available as prior art to the present invention. To the extent that section headings are used, they should not be construed as necessarily limiting.

CLAIMS

What is claimed is:

1. A system for memory management accessible by a plurality of computing units, comprising:

an interface adapted to receive a request to access one of a plurality of storage regions of a cache storage; and

a memory for storing:

a single access directory documenting which of said plurality of storage regions is currently assigned to one of the plurality of computing units; and

a multi-access directory documenting which of said plurality of storage regions is accessed by at least one of a plurality of computing units.

2. The system of claim 1, further comprises a processing unit adapted to manage said single access directory and said multi-access directory to reflect a current allocation of said plurality of storage regions in light of said request.

3. The system of claim 1, adapted to manage said single access directory and said multi-access directory by:

searching at least one of said single access directory and said multi-access directory for a current allocation status of said requested storage region,

according to an outcome of said search performing a new allocation of said requested storage region to one of said plurality of computing units,

updating at least one of said single access directory and said multi-access directory to reflect said new allocation.

4. The system of any of claims 2-3, wherein said processing unit is adapted to perform said new allocation according an analysis of at least one sharer bitmap of each of at least some of said plurality of regions.

5. The system of any of claims 3-4, wherein said new allocation is to a computing unit submitting said request.

6. The system of any of claims 1-3, adapted to manage said single access directory and said multi-access directory by migrating at least one entry from one of said single access directory and said multi-access directory to the other.

7. The system of claim 6, wherein when said migrating is triggered when said request is for accessing a region documented in said single access directory by a computing unit which is not said single computing unit.

8. The system of any of claims 6-7, wherein when said migrating is triggered when the number of blocks of a region documented in said multi-access directory which are cached by the same computing unit of said plurality of computing units exceeds a threshold.

9. The system of any of claims 1-8, wherein the system is configured to manage said single access directory and said multi-access directory by updating an entry associated with said requested storage region in both said single access directory and said multi-access directory for removing a usability indication of said requested storage region from one of said single access directory and said multi-access directory to the other.

10. The system of any of claims 1-9 adapted to monitor at least one sharer bitmap of each of at least some of said plurality of regions and to reallocate any of said plurality of storage regions among said plurality of computing units accordingly.

11. The system of any of claims 1-9 adapted to manage said single access directory and said multi-access directory by identifying a least recently used region among said plurality of regions according to an analysis of said single access directory and allocating said least recently used region as said new allocation.

12. The system of any of claims 1-11 adapted to manage said single access directory and said multi-access directory by identifying a least shared region among said plurality of regions according to an analysis of said multi-access directory and allocating said least shared region as said new allocation.

13. The system of any of claims 1-11, wherein said single access directory and said multi-access directory comprises a plurality of status values each indicative of a usability status for another of said plurality of storage regions.

14. The system of any of the previous claims, wherein said single access directory comprises a plurality of single access records each document a usage of a plurality of region blocks of one of said plurality of regions by a single computing unit of said plurality of computing units and said multi-access directory comprises a plurality of multiple access records each document for each one of said plurality of computing units a usage state of a plurality of region blocks of one of said plurality of regions.

15. The system of any of the previous claims, wherein said cache storage is a last level cache (LLC) storage and said request is a request to access said LLC storage and wherein said plurality of computing units are processors of a multi processor unit, said plurality of computing units share said cache storage for performing multiple processing tasks in parallel.

16. A method for memory management, comprising:

receiving requests to access a plurality of storage regions of a cache storage from a plurality of computing units;

managing a single access directory documenting which of said plurality of storage regions is currently assigned to a single computing unit of said plurality of computing units and a multi-access directory documenting which of said plurality of storage regions is accessed by at least some of said plurality of computing units to reflect an access given in response to said requests.

17. A computer program comprising a readable storage medium storing program code thereon for managing storage region allocation, the program code comprising:

instructions for receiving requests to access a plurality of storage regions of a cache storage from a plurality of computing units;

instructions for managing a single access directory documenting which of said plurality of storage regions is currently assigned to a single computing unit of said plurality of computing units and a multi-access directory documenting which of said

plurality of storage regions is accessed by at least some of said plurality of computing units to reflect an access given in response to said requests.

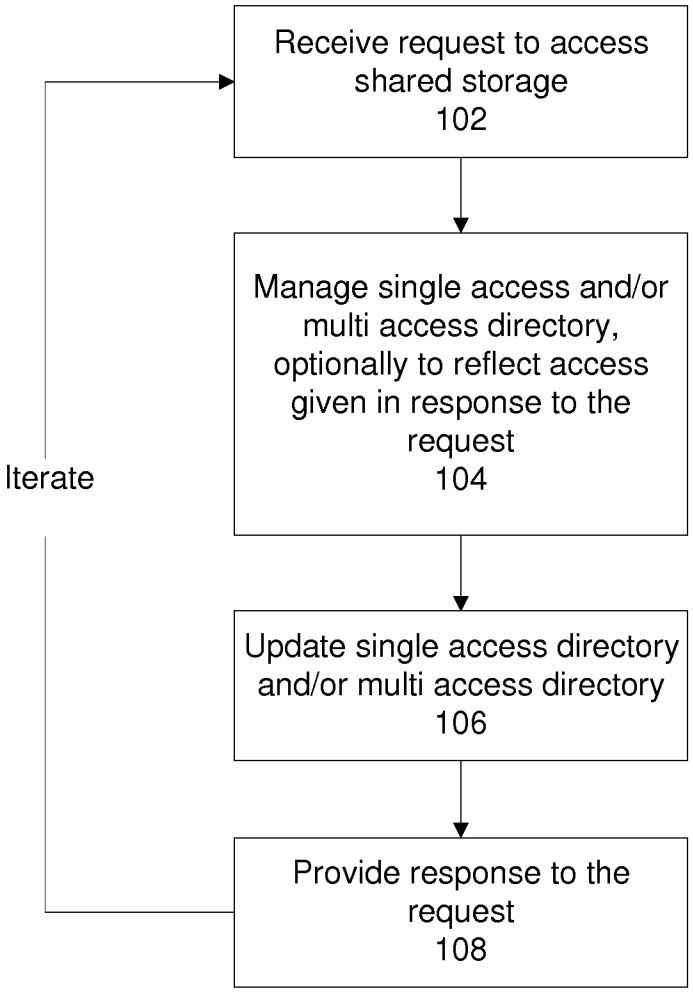
Receive request to access shared storage
102
Manage single access and/or multi access directory, optionally to reflect access given in response to the request
104
Update single access directory and/or multi access directory
106
Provide response to the request
108
Iterate

FIG. 1A

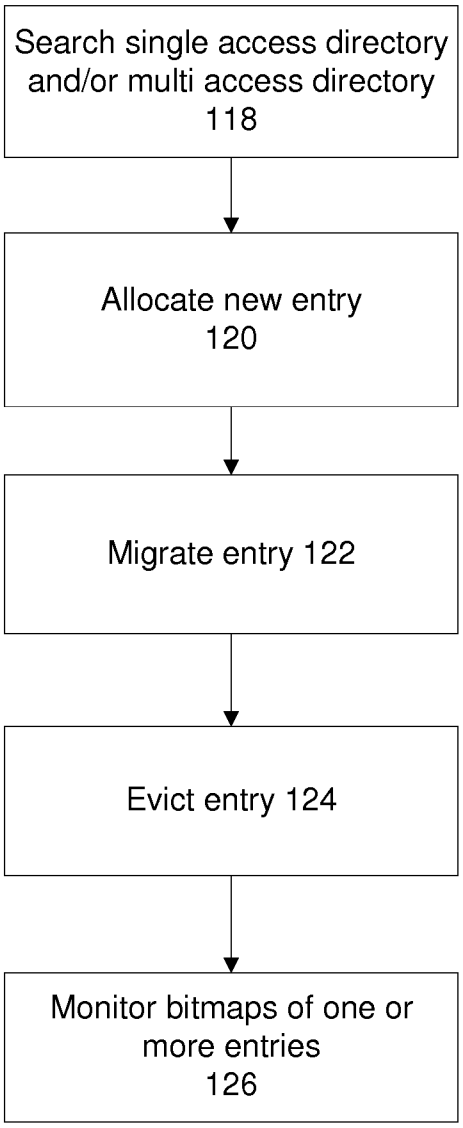
Search single access directory and/or multi access directory 118
Allocate new entry 120
Migrate entry 122
Evict entry 124
Monitor bitmaps of one or more entries 126

FIG. 1B

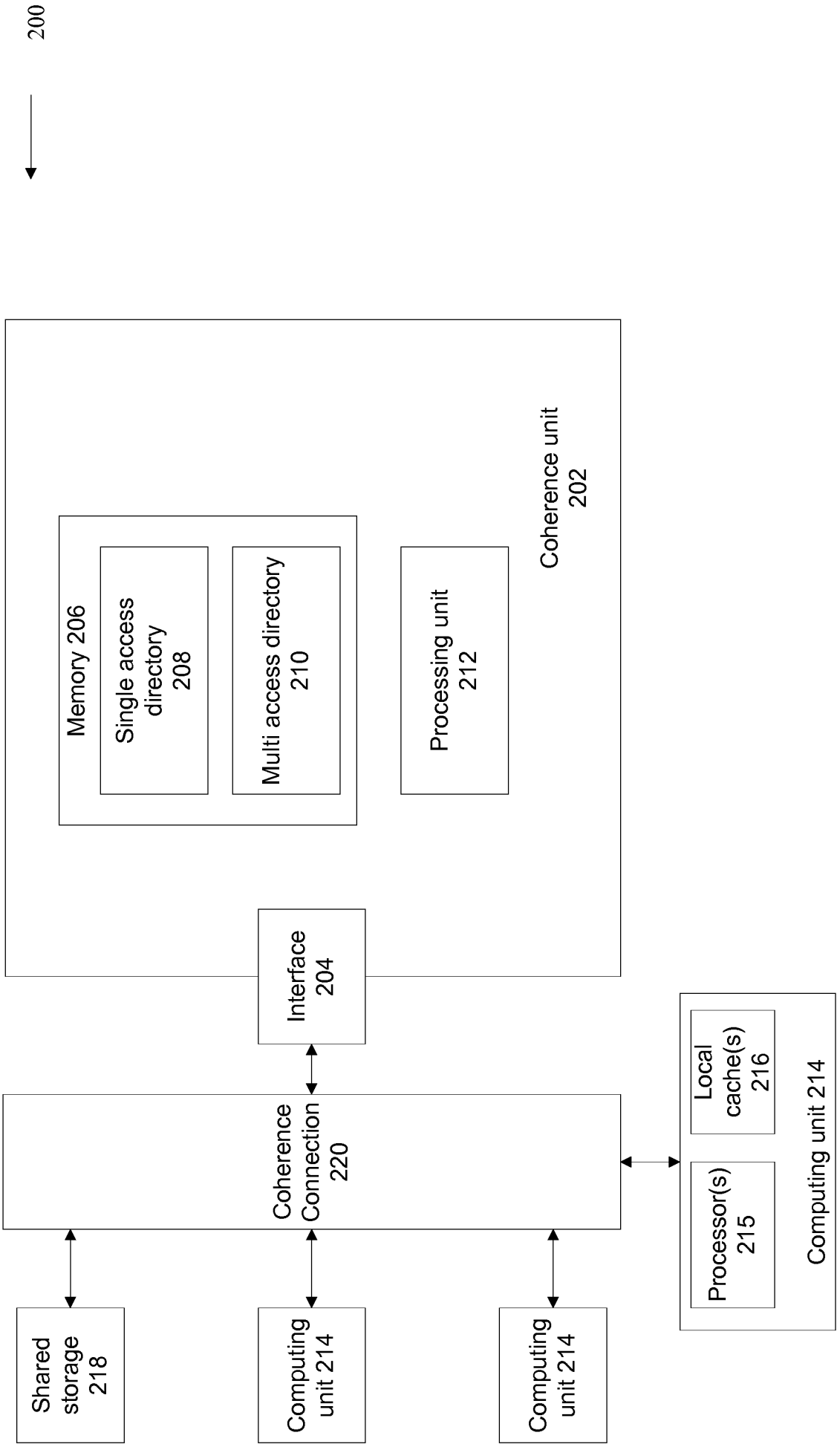
200
Coherence unit 202
Memory 206
Single access directory 208
Multi access directory 210
Processing unit 212
Interface 204
Coherence Connection 220
Shared storage 218
Computing unit 214
Computing unit 214
Computing unit 214
Processor(s) 215
Local cache(s) 216

FIG. 2

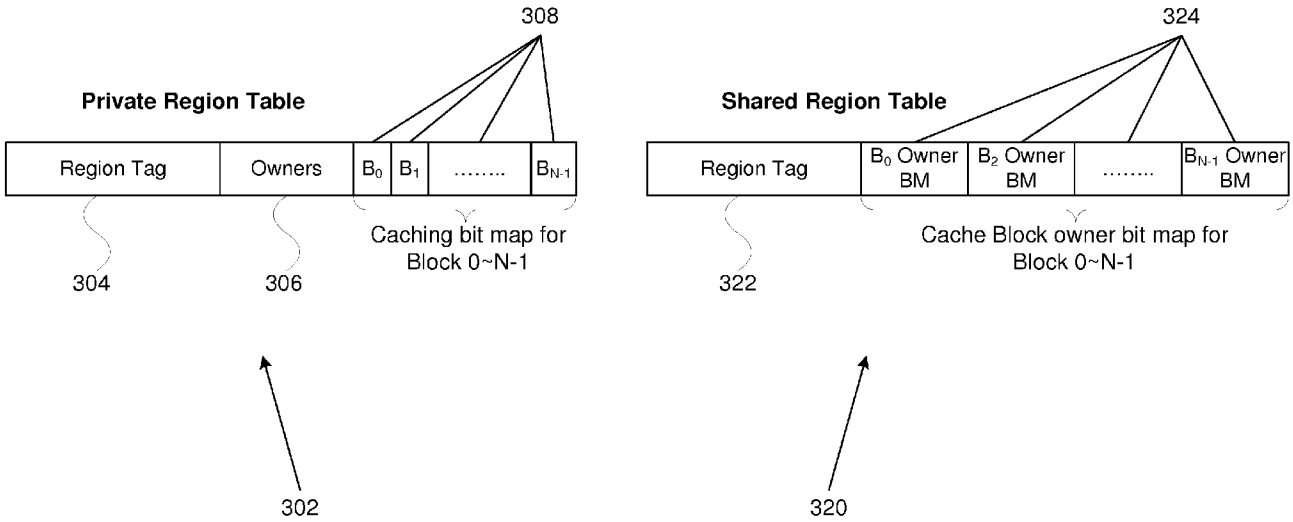
308
Private Region Table
Region Tag
Owners
B_0
B_1
……..
B_{N-1}
Caching bit map for Block 0~N-1
304
306
302
324
Shared Region Table
Region Tag
B_0 Owner BM
B_2 Owner BM
……..
B_{N-1} Owner BM
Cache Block owner bit map for Block 0~N-1
322
320

Fig. 3

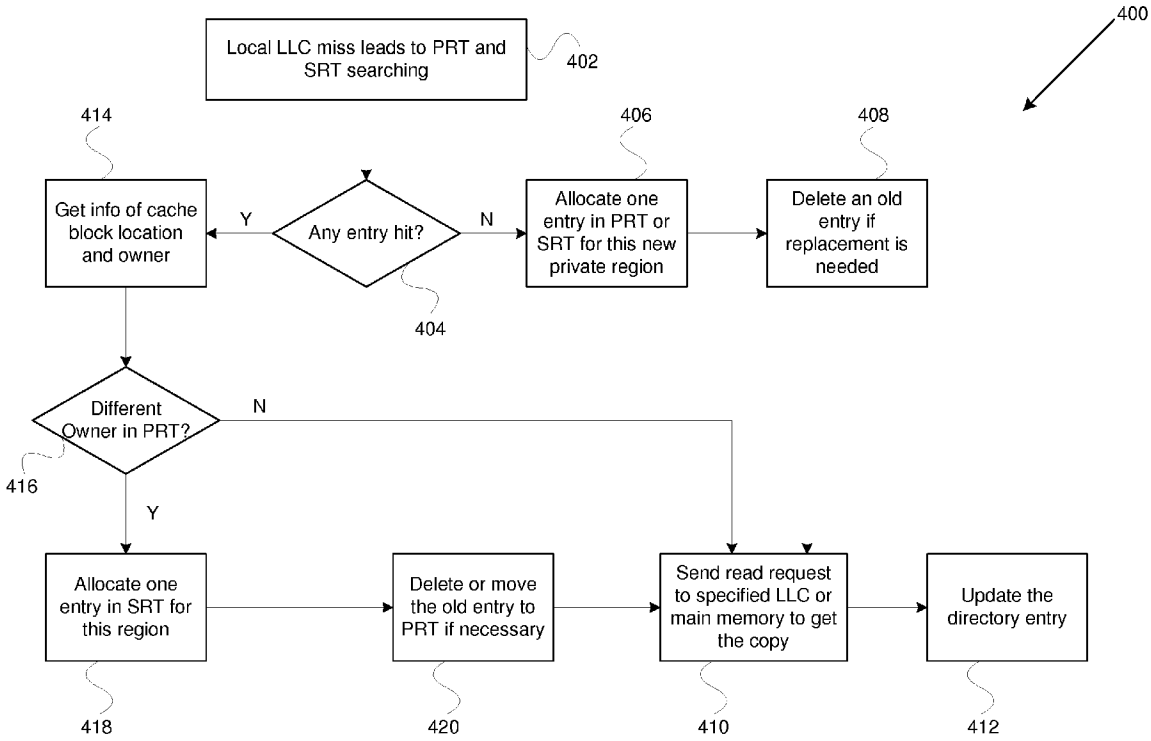
400
Local LLC miss leads to PRT and SRT searching
402
414
406
408
Get info of cache block location and owner
Y
Any entry hit?
N
Allocate one entry in PRT or SRT for this new private region
Delete an old entry if replacement is needed
404
Different Owner in PRT?
N
416
Y
Allocate one entry in SRT for this region
Delete or move the old entry to PRT if necessary
Send read request to specified LLC or main memory to get the copy
Update the directory entry
418
420
410
412

Fig. 4

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2015/063723

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F12/08
ADD.

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

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2002/138698 A1 (KALLA RONALD NICK [US]) 26 September 2002 (2002-09-26) paragraphs [0013], [0022], [0023], [0024], [0025], [0026]; claims 1-7; figures 1-6	1-17
A	US 5 148 533 A (JOYCE THOMAS F [US] ET AL) 15 September 1992 (1992-09-15) the whole document	1-17
A	US 2013/339620 A1 (LIH IULIN [US]) 19 December 2013 (2013-12-19) the whole document	1-17

☐ 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
1 September 2015	07/09/2015

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Jardon, Stéphan

1

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/EP2015/063723

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2002138698 A1	26-09-2002	NONE	
US 5148533 A	15-09-1992	NONE	
US 2013339620 A1	19-12-2013	CN 104364776 A	18-02-2015
		US 2013339620 A1	19-12-2013
		WO 2013185638 A1	19-12-2013