



(51) International Patent Classification:
G06F 11/16 (2006.01)

(21) International Application Number:
PCT/US2019/041069

(22) International Filing Date:
09 July 2019 (09.07.2019)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
62/716,846 09 August 2018 (09.08.2018) US
16/193,414 16 November 2018 (16.11.2018) US

(71) Applicant: **NETAPP, INC.** [US/US]; 1395 Crossman Avenue, Sunnyvale, California 94089 (US).

(72) Inventors: **VENKETARAMAN, Sriram**; 1395 Crossman Avenue, Sunnyvale, California 94089 (US). **GOLAN-**

DER, Amit; 1395 Crossman Avenue, Sunnyvale, California 94089 (US).

(74) Agent: **SINGH, Tejinder** et al.; Klein, O'Neill & Singh, LLP, 16755 Von Karman Avenue, Suite 275, Irvine, California 92606 (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, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, 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,

(54) Title: METHODS AND SYSTEMS FOR PROTECTING DATA OF A PERSISTENT MEMORY BASED FILE SYSTEM

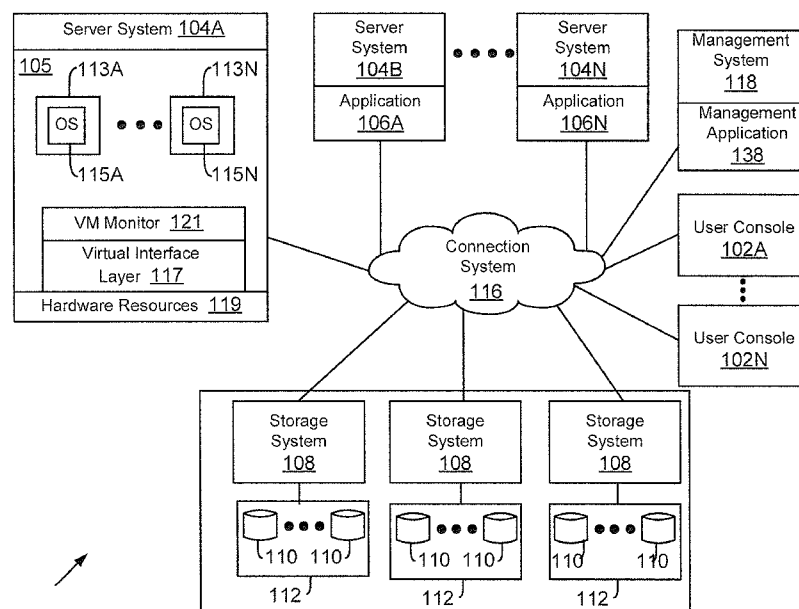


FIG. 1A

(57) Abstract: Methods and systems for a networked storage system is provided. One method includes creating a first snapshot for data units stored at a persistent memory of a computing device, the data units managed by a first file system; transferring metadata associated with the data units and the data units stored at the persistent memory to a storage device managed by a second file system using a logical object, the second file system executed by a storage system interfacing with the computing device; and generating a second snapshot of the logical object at the storage device, the second snapshot including data units and associated metadata of the first snapshot.

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 applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*
- *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))*

Published:

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

METHODS AND SYSTEMS FOR PROTECTING DATA OF A PERSISTENT MEMORY BASED FILE SYSTEM

[0001] Technical Field: The present disclosure relates to networked storage systems, and particularly, to providing
5 computing technology for taking snapshots for data units of a persistent memory based file system executed by a computing device communicating with a storage system executing another file system for managing mass storage devices.

[0002] Background: Various forms of storage systems are used
10 today. These forms include direct attached storage (DAS) network attached storage (NAS) systems, storage area networks (SANs), and others. Network storage systems are commonly used for a variety of purposes, such as providing multiple users with access to shared data, backing up data and others.

15 [0003] A storage system typically includes at least one computing system executing a storage operating system with a file system for storing and retrieving data on behalf of one or more client computing systems ("clients"). The storage operating system stores and manages shared data containers in a
20 set of mass storage devices.

[0004] Newly emerging non-volatile or persistent memory (PM) technology may be implemented through a nonvolatile media attached to a central processing unit (CPU) of a computer. PM is characterized by low RAM-like latencies, and is typically

substantially faster vis-à-vis flash-based SSD arrays (solid state devices) and HDDs (hard disk drive), respectively. PM is typically implemented using backed-up dynamic random access memory (DRAM, magneto-resistive random-access memory (MRAM) or spin-transfer torque magnetic random-access memory (STT-MRAM) technologies. Other emerging technologies, such as resistive random-access memory (ReRAM and phase-change memory (PCM)) which are very dense, may enable cheaper, though slower, PM components.

[0005] File systems are usually block-based and are architected towards using HDDs and/or SSDs, and as such, they do not store or cache user data on a PM. Typically, a separate software layer manages memory-based software caching. Emerging PM-aware file systems (e.g. EXT4-DAX) directly access PM, avoiding slow and cumbersome caching and/or memory map services of a virtual file system layer.

[0006] Typically, PM based file systems executed out of host computing systems initially store data units at the PM. The data units are later transferred to lower tier storage managed by a storage operating system of a storage system (or storage controller). Continuous efforts are being made to develop computing technology to efficiently protect data that is initially stored by the PM based file system and later moved to lower tier storage devices managed by a storage controller.

BRIEF DESCRIPTION OF THE DRAWINGS

[0007] The foregoing features and other features will now be described with reference to the drawings of the various aspects. In the drawings, the same components have the same reference numerals. The illustrated aspects are intended to illustrate, but not to limit the present disclosure. The drawings include the following Figures:

[0008] Figure 1A shows an example of an operating environment for implementing the various aspects of the present disclosure;

[0009] Figure 1B shows a block diagram of a system with a persistent memory based file system, used according to one aspect of the present disclosure;

[0010] Figure 1C shows an example of a process for taking snapshots, according to one aspect of the present disclosure;

[0011] Figures 1D-1E illustrate the process blocks of Figure 1C, according to one aspect of the present disclosure;

[0012] Figure 1F shows an example of retaining a snapshot at a persistent memory, according to one aspect of the present disclosure;

[0013] Figure 2A shows an example of a clustered storage system, used according to one aspect of the present disclosure;

[0014] Figure 2B shows an example of a storage system node, used according to one aspect of the present disclosure;

[0015] Figure 3 shows an example of a storage operating system, used according to one aspect of the present disclosure; and

[0016] Figure 4 shows an example of a processing system, used according to one aspect of the present disclosure.

5 DETAILED DESCRIPTION

[0017] As preliminary note, the terms "component", "module", "system," and the like as used herein are intended to refer to a computer-related entity, either software-executing general purpose processor, hardware, firmware and a combination thereof.

10 For example, a component may be, but is not limited to being, a process running on a processor, a processor, an object, an executable, a thread of execution, a program, and/or a computer.

[0018] By way of illustration, both an application running on a server and the server can be a component. One or more components

15 may reside within a process and/or thread of execution, and a component may be localized on one computer and/or distributed between two or more computers. Also, these components can

execute from various non-transitory, computer readable media having various data structures stored thereon. The components

20 may communicate via local and/or remote processes such as in accordance with a signal having one or more data packets (e.g., data from one component interacting with another component in a local system, distributed system, and/or across a network such as the Internet with other systems via the signal).

[0019] Computer executable components can be stored, for example, on non-transitory, computer readable media including, but not limited to, an ASIC (application specific integrated circuit), CD (compact disc), DVD (digital video disk), ROM (read only memory), floppy disk, hard disk, EEPROM (electrically erasable programmable read only memory), memory stick or any other storage device type, in accordance with the claimed subject matter.

[0020] System 100: Figure 1A shows an example of a networked operating environment 100 (also referred to as system 100), for implementing the various adaptive aspects of the present disclosure. In one aspect, system 100 may include a plurality of computing systems 104A-104N (may also be referred to and shown as server system (or server systems) 104 or as host system (or host systems) 104) that may access one or more storage systems 108 via a connection system 116 such as a local area network (LAN), wide area network (WAN), the Internet and others. The server systems 104 may communicate with each other via connection system 116, for example, for working collectively to provide data-access service to user consoles (or computing devices) 102A-102N (may be referred to as user 102 or client system 102).

[0021] Server systems 104 may be computing devices configured to execute applications 106A-106N (may be referred to as

application 106 or applications 106) over a variety of operating systems, including the UNIX® and Microsoft Windows® operating systems. Applications 106 may utilize data services of storage system 108 to access, store, and manage data in a set of storage devices 110 that are described below in detail. Applications 106 may include a database program, an email program or any other computer executable program. The server systems may also execute a persistent memory-based file system that stores data in persistent memory of the server system, as described below in detail.

[0022] Some of the server systems 104 may also utilize file-based access protocols when accessing information (in the form of files and directories) over a network attached storage (NAS)-based network. Alternatively, server systems 104 may use block-based access protocols, for example, the Small Computer Systems Interface (SCSI) protocol encapsulated over TCP (iSCSI) and SCSI encapsulated over Fibre Channel (FCP) to access storage via a storage area network (SAN).

[0023] As an example, server 104A executes a virtual machine environment 105, according to one aspect. In the virtual machine environment 105, a physical resource is time-shared among a plurality of independently operating processor executable virtual machines (VMs). Each VM may function as a self-contained platform, running its own operating system (OS)

and computer executable, application software. The computer executable instructions running in a VM may be collectively referred to herein as "guest software". In addition, resources available within the VM may be referred to herein as "guest
5 resources".

[0024] The guest software expects to operate as if it were running on a dedicated computer rather than in a VM. That is, the guest software expects to control various events and have access to hardware resources on a physical computing system (may
10 also be referred to as a host platform) which may be referred to herein as "host hardware resources". The host hardware resource may include one or more processors, resources resident on the processors (e.g., control registers, caches and others), memory (instructions residing in memory, e.g., descriptor tables), and
15 other resources (e.g., input/output devices, host attached storage, network attached storage or other like storage) that reside in a physical machine or are coupled to the host platform.

[0025] The virtual machine environment 105 includes a plurality
20 of VMs 113A-113N that execute a plurality of guest OS 115A-115N (may also be referred to as guest OS 115) to share hardware resources 119. As described above, hardware resources 119 may include CPU, memory, I/O devices, storage or any other hardware resource.

[0026] A virtual machine monitor (VMM) 121, for example, a processor executed hypervisor layer provided by VMWare Inc., Hyper-V layer provided by Microsoft Corporation (without derogation of any third party trademark rights) or any other virtualization layer type, presents and manages the plurality of guest OS 115. VMM 121 may include or interface with a virtualization layer (VIL) 117 that provides one or more virtualized hardware resource 119 to each guest OS. For example, VIL 117 presents physical storage at storage devices 110 as virtual storage (for example, as a virtual hard drive (VHD)) to VMs 113A-113N. The VMs use the VHDs to store information at storage devices 110.

[0027] In one aspect, VMM 121 is executed by server system 104A with VMs 113A-113N. In another aspect, VMM 121 may be executed by a separate stand-alone computing system, often referred to as a hypervisor server or VMM server and VMs 113A-113N are presented via another computer system. It is noteworthy that various vendors provide virtualization environments, for example, VMware Corporation, Microsoft Corporation (without derogation of any third party trademark rights) and others. The generic virtualization environment described above with respect to Figure 1A may be customized depending on the virtual environment provider.

[0028] System 100 may also include a management system 118 for managing and configuring various elements of system 100.

Management system 118 may execute or include a management application 138 for coordinating snapshots (i.e. point in time
5 copy) of data stored by the storage system 108.

[0029] In one aspect, storage system 108 is a shared storage system having access to a set of mass storage devices 110 (may be referred to as storage devices 110 or "L3 storage") within a storage subsystem 112. As an example, storage devices 110 may be
10 a part of a storage array within the storage sub-system 112.

Storage devices 110 are used by the storage system 108 for storing information. The storage devices 110 may include writable storage device media such as magnetic disks, video tape, optical, DVD, magnetic tape, non-volatile memory devices for
15 example, self-encrypting drives, flash memory devices and any other similar media adapted to store information. The storage devices 110 may be organized as one or more groups of Redundant Array of Independent (or Inexpensive) Disks (RAID). The various aspects disclosed herein are not limited to any particular
20 storage device or storage device configuration.

[0030] In one aspect, to facilitate access to storage devices 110, a storage operating system of storage system 108 "virtualizes" the storage space provided by storage devices 110. The storage system 108 can present or export data stored at

storage devices 110 to server systems 104 and VMM 121 as a storage volume or one or more qtree sub-volume units including logical unit numbers (LUNs). Each storage volume may be configured to store data files (or data containers or data units), scripts, word processing documents, executable programs, and any other type of structured or unstructured data. From the perspective of the VMS/server systems, each volume can appear to be a single disk drive. However, each volume can represent the storage space in one disk, an aggregate of some or all of the storage space in multiple disks, a RAID group, or any other suitable set of storage space.

[0031] It is noteworthy that the term "disk" as used herein is intended to mean any storage device/space and not to limit the adaptive aspects to any particular type of storage device, for example, hard disks.

[0032] The storage system 108 may be used to store and manage information at storage devices 110 based on a request generated by server system 104, management system 118, user 102 and/or a VM. The request may be based on file-based access protocols, for example, the CIFS or the NFS protocol, over TCP/IP. Alternatively, the request may use block-based access protocols, for example, iSCSI or FCP.

[0033] As an example, in a typical mode of operation, server system 104 (or VMs 113A-113N) transmits one or more input/output

(I/O) commands, such as an NFS or CIFS request, over connection system 116 to the storage system 108. Storage system 108 receives the request, issues one or more I/O commands to storage devices 110 to read or write the data on behalf of the server system 104, and issues an NFS or CIFS response containing the requested data over the connection system 116 to the respective server system 104.

[0034] In one aspect, storage system 108 may have a distributed architecture, for example, a cluster based system that may include a separate network module and storage module, described below in detail with respect to Figure 2A. Briefly, the network module is used to communicate with server systems 104 and management system 118, while the storage module is used to communicate with the storage devices 110.

[0035] Persistent Memory Based File System: Figure 1B shows a block diagram of a system where a persistent memory (PM) based file system (FS1) 144 is executed out of a PM 142 of a host system 140. The host system 140 may be similar to servers 104 described above and interfaces with the storage system 108 that executes a storage operating system 150 with a file system (may be referred to as a second file system). FS1 144 stores data units 146 at PM 142. The data units may be stored for client applications and other processes executed by host system 140. A snapshot 148 is taken by FS1 144 and may be stored at PM 142, as

described below in detail. The term snapshot means a point-in-time copy that captures a state of a file system at any given time. Snapshot 148 is a point in time copy of data units 146 stored at PM 142 by FS1 144.

5 **[0036]** The storage devices 110 managed by the storage operating system 150 store data units and associated metadata 152 using one or more LUNs/volumes. The storage devices 110 also store snapshot 154 of the active FS1 144 of host system 140, as described below in detail.

10 **[0037]** In one aspect, the system of Figure 1B uses two storage tiers; a memory addressable first tier (PM 142) and a block addressable second tier (storage devices 110). FS1 144 manages the first tier (i.e. PM 142) and the storage operating system 150 manages the second tier (i.e. storage devices 110).

15 **[0038]** When an application writes data, the data is first written to the first tier 142 as data units 146 and the write operation is acknowledged. The data units are then written to the second tier 110, when space in the first tier reaches a certain threshold level, to prevent repeated overwrites from
20 getting written to the second tier.

[0039] At any given time, a first portion of file system data (i.e. data units stored by FS1 144) (e.g. X%) may be present in the first tier and a second portion is stored at the second tier, managed by storage operating system 150. To take a

snapshot of all the data units, a snapshot operation has to take a point in time copy of data units that span across both storage tiers. This involves writing data that is present in the first tier to the second tier and then taking a snapshot of the second tier. The amount of time it takes to flush data to the second tier is proportional to the size of the first tier and the bandwidth available to write the data to the second tier. To flush the data units from the first tier to the second tier, any I/O (Input/output) requests are fenced (or held) until a flush operation has been completed.

[0040] To reduce the amount of time application I/O requests are fenced, the systems and processes disclosed herein take a snapshot of the first tier and then unfences new and pending I/O requests i.e. application requests for reading and writing data are executed. Subsequently, only the data units present in the snapshot and not yet written to the second tier have to be written to the second tier. It is noteworthy that it is not enough to write the data blocks to second tier during the snapshot operation, metadata associated with the data blocks is also written.

[0041] The snapshot at the second tier includes the snapshot of the first tier with both data units and associated metadata. The snapshot of the second tier can be used for different reasons, including, data protection, remote protection using SnapMirror

(without derogation of any trademark rights), clones, and for restore operations. A restore operation regenerates the first tier with metadata blocks included in the second tier snapshot and enables read/write access to data by FS1 144.

5 [0042] It is noteworthy that snapshots in the first tier can use limited PM 142 pages. To prevent excessive usage of PM pages in the first tier, snapshots of the first tier that have been written to the second tier may be deleted, as described below in detail.

10 [0043] Process Flow: Figure 1C shows a process flow 156, according to one aspect of the present disclosure. Process 156 begins in block B158 when the host system 140 and the storage system 108 are initialized and operational. Before taking a FS1 144 based snapshot, an optional block B160 is executed. Block
15 B160, reduces the number of data units that have a valid copy at PM 142 (also referred to as L2 storage of first tier storage) but have not been transferred or flushed to a LUN at storage devices 110 (also referred to as L3 storage or second tier storage). FS1 144 typically operates by writing to PM 142 and
20 then acknowledging modifications to an application that initiates write operations. The data unit that is written by FS1 144 is not immediately written out to storage device 110 for performance reasons. In block B160, the number of data units that are still at PM 142 are flushed to storage devices 110,

which reduces the time to take a snapshot of PM 142 that stores data units 146. Block B160 is executed before a snapshot is taken of PM 142.

[0044] In block B162, a snapshot (referred to as L2 snap-1

5 (Figures 1D-1E) is taken by FS1 144. In one aspect, before taking the snapshot, any I/O request by one or more applications is fenced (i.e. held) to complete the snapshot operation by FS1 144. The amount of time it takes to write data to the LUNs at storage devices 110 is proportional to the amount of data units
10 at PM 142. By taking an instantaneous snapshot in the PM-based filesystem, I/Os can be unfenced quickly.

[0045] After the snapshot is taken, in block B164, the metadata for the data units in L2 snapshot is encoded and written to storage devices 110. The metadata includes timestamps, access
15 rights, a unique identifier for the snapshot, storage location of the snapshot, size of the snapshot, number of data units or any other information (pointers, links and others) that may be needed to restore data units and metadata associated with the data units from the snapshot. The metadata is encoded such that
20 the metadata can be represented from a memory representation of PM 142 to a block representation at storage devices 110. This may be accomplished by encoding block numbers as offsets within the metadata. Metadata encoding details are also provided in US Patent No. 9, 678, 670, the disclosure of which is incorporated

herein by reference in its entirety. In one aspect, the data blocks that belong to the snapshot taken by FS1 144 are identified. The data units are written to a pre-determined offset at LUNs of storage devices 110, if the units were not
5 written earlier in block B160. It is noteworthy that only incremental data units across different snapshots are written to the LUNs at storage devices 110. The data units written to the LUNs are compressed, compacted and deduplicated. After writing the data blocks to the LUNs, the metadata blocks belonging to
10 the snapshot are encoded in a similar fashion and written to the LUNs.

[0046] In block B166, a snapshot of the LUNs at storage device 110 is taken. This snapshot is referred to as L3 snap-1 (Figures 1D-1E). In one aspect, the LUN based snapshot is taken after the
15 data units and the metadata blocks of the PM-filessystem snapshot (L2 snap-1) have been flushed to the LUNs of storage devices 110. The L3 snapshot is uniquely identified and indicates that the FS1 144 snapshot is included in it. Each snapshot of the LUNs of storage device 110 may have multiple FS1 144 snapshots.
20 The FS1 144 snapshots metadata links the valid PM-filessystem snapshots in a L3 snapshot.

[0047] In block B168, the FS1 144 based snapshot or its data units may be optionally deleted. This reclaims storage space at PM 142.

[0048] In block B170, storage blocks for LUNs at the storage devices 110 are released if they are no longer referenced by the deleted snapshot.

[0049] In block B172, optionally, reverse transform the snapshot data units within the L3 snap-1 to the FS1 144. To access (read and write) the data units stored at storage devices 110, a writeable copy of the LUNs is created and the FS1 144 metadata stored at the LUNs is transformed to a new FS1 144 instance.

[0050] Figure 1D shows an example of the process blocks of

Figure 1C without the steps involved in block B170. Figure 1E shows an example of the process blocks of Figure 1C without the steps involved in block B172.

[0051] Figure 1F shows an example, where snapshots are left at the host system. This allows for faster lookups in L2 snap-1, faster recovery to snap-1 and faster snapshot operation for a next snapshot.

[0052] Figure 1D shows the various process blocks of Figure 1C between time t1 and time t6. At time t1, the L2 active file system (AFS) 161 (L2 AFS 161) has a metadata node 151 and data units 153, 155 and 157. Data unit 155 has been transferred to a LUN at storage device 110 managed by L3 AFS 167. Between time t1 and t2, the L2 snap-1 159 has been taken. The user has also written to block 165 with root node 163. Between time t2 and t3,

the metadata 151 and data units 153, 155 and 157 have been transferred to the LUNs at L3 storage.

[0053] Between t3 and t4, a snapshot 169 of the L3 LUN is taken. The L3 AFS still retains 151, 153, 155 and 157 blocks. Between
5 t4 and t5, a delete process to delete the snapshot is initiated. Between t5 and t6, the L2 AFS uses root node 163 and blocks 165, 155 and 157. The L3 AFS still retains 151, 153, 155 and 157 and hence use extra storage space at L3 storage.

[0054] Figure 1E shows all the process blocks of Figure 1D but
10 also shows at time t6 that blocks 151 and 153 are released since they are no longer referenced by the L2 AFS and the L3 snap-1 has all the data units of L2 snap-1.

[0055] In one aspect, methods and systems for a networked storage system is provided. One method includes creating a first
15 snapshot for data units stored at a persistent memory of a computing device, the data units managed by a first file system; transferring metadata associated with the data units and the data units stored at the persistent memory to a storage device managed by a second file system using a logical object, the
20 second file system executed by a storage system interfacing with the computing device; and generating a second snapshot of the logical object at the storage device, the second snapshot including data units and associated metadata of the first snapshot.

[0056] Clustered Storage System: Figure 2A depicts an illustrative aspect of a shared, storage environment 200 using the L3 storage described above. The shared, storage environment 200 includes the management system 118, a plurality of server systems 204.1-204.2 (similar to server systems 104), a clustered storage system 202 and at least one computer network 206 communicably connecting the server systems 204.1-204.2 and the clustered storage system 202.

[0057] The clustered storage system 202 includes a plurality of nodes 208.1-208.3, a cluster switching fabric 210, and a plurality of mass storage devices 212.1-212.3 (similar to 110, Figure 1A). Each of the plurality of nodes 208.1-208.3 is configured to include a network module, a storage module, and a management module, each of which can be implemented as a separate processor executable or machine implemented module. Specifically, node 208.1 includes a network module 214.1, a storage module 216.1, and a management module 218.1, node 208.2 includes a network module 214.2, a storage module 216.2, and a management module 218.2, and node 208.3 includes a network module 214.3, a storage module 216.3, and a management module 218.3.

[0058] The network modules 214.1-214.3 include functionality that enables the respective nodes 208.1-208.3 to connect to one or more of the client systems 204.1-204.2 over the computer

network 206, while the storage modules 216.1-216.3 connect to one or more of the storage devices 212.1-212.3.

[0059] The management modules 218.1-218.3 provide management functions for the clustered storage system 202. Accordingly, each of the plurality of server nodes 208.1-208.3 in the clustered storage server arrangement provides the functionality of a storage server.

[0060] A switched virtualization layer including a plurality of virtual interfaces (VIFs) 220 is provided below the interface between the respective network modules 214.1-214.3 and the client systems 204.1-204.2, allowing storage 212.1-212.3 associated with the nodes 208.1-208.3 to be presented to the client systems 204.1-204.2 as a single shared storage pool. For example, the switched virtualization layer may implement a virtual interface architecture. Figure 2A depicts only the VIFs 220 at the interfaces to the network modules 214.1, 214.3 for clarity of illustration.

[0061] The clustered storage system 202 can be organized into any suitable number of virtual servers (VServer or storage virtual machines (SVM)) 222A-222N, in which each virtual storage system represents a single storage system namespace with separate network access. Each virtual storage system has a user domain and a security domain that are separate from the user and security domains of other virtual storage systems. Server

systems 204 can access storage space via a VServer from any node of the clustered system 202.

[0062] Each of the nodes 208.1-208.3 may be defined as a computer adapted to provide application services to one or more of the client systems 204.1-204.2. In this context, a SVM is an instance of an application service provided to a client system. The nodes 208.1-208.3 are interconnected by the switching fabric 210, which, for example, may be embodied as a Gigabit Ethernet switch or any other switch type.

[0063] Although Figure 2A depicts three network modules 214.1-214.3, the storage modules 216.1-216.3, and the management modules 218.1-218.3, any other suitable number of network modules, storage modules, and management modules may be provided. There may also be different numbers of network modules, storage modules, and/or management modules within the clustered storage system 202. For example, in alternative aspects, the clustered storage system 202 may include a plurality of network modules and a plurality of storage modules interconnected in a configuration that does not reflect a one-to-one correspondence between the network modules and storage modules.

[0064] The server systems 204.1-204.2 of Figure 2A may be implemented as computing devices configured to interact with the respective nodes 208.1-208.3 in accordance with a client/server

model of information delivery. In the presently disclosed aspect, the interaction between the server systems 204.1-204.2 and the nodes 208.1-208.3 enable the provision of network data storage services. Specifically, each server system 204.1, 204.2
5 may request the services of one of the respective nodes 208.1, 208.2, 208.3, and that node may return the results of the services requested by the client system by exchanging packets over the computer network 206, which may be wire-based, optical fiber, wireless, or any other suitable combination thereof. The
10 server systems 204.1-204.2 may issue packets according to file-based access protocols, such as the NFS or CIFS protocol, when accessing information in the form of files and directories.

[0065] In a typical mode of operation, one of the server systems 204.1-204.2 transmits an NFS or CIFS request for data to one of
15 the nodes 208.1-208.3 within the clustered storage system 202, and the VIF 220 associated with the respective node receives the client request. It is noted that each VIF 220 within the clustered system 202 is a network endpoint having an associated IP address. The server request typically includes a file handle
20 for a data file stored in a specified volume on at storage 212.1-212.3.

[0066] Storage System Node: Figure 2B is a block diagram of a computing system 224, according to one aspect. System 224 may be used by a stand-alone storage system 108 and/or a storage system

node operating within a cluster based storage system described above with respect to Figure 2A.

[0067] System 224 may include a plurality of processors 226A and 226B, a memory 228, a network adapter 234, a cluster access
5 adapter 238 (used for a cluster environment), a storage adapter 240 and local storage 236 interconnected by a system bus 232. The local storage 236 comprises one or more storage devices, such as disks, utilized by the processors to locally store configuration and other information.

10 [0068] The cluster access adapter 238 comprises a plurality of ports adapted to couple system 224 to other nodes of a cluster as described above with respect to Figure 2A. In the illustrative aspect, Ethernet may be used as the clustering protocol and interconnect media, although it will be apparent to
15 those skilled in the art that other types of protocols and interconnects may be utilized within the cluster architecture described herein.

[0069] System 224 is illustratively embodied as a dual processor storage system executing a storage operating system 230 (or 150,
20 Figure 1B) that preferably implements a high-level module, such as a file system, to logically organize information as a hierarchical structure of named directories, files and special types of files called virtual disks (hereinafter generally "blocks") on storage devices 110/212. However, it will be

apparent to those of ordinary skill in the art that the system
224 may alternatively comprise a single or more than two
processor systems. Illustratively, one processor 226 executes
the functions of a network module on a node, while the other
5 processor 226B executes the functions of a storage module.

[0070] The memory 228 illustratively comprises storage locations
that are addressable by the processors and adapters for storing
programmable instructions and data structures. The processor
and adapters may, in turn, comprise processing elements and/or
10 logic circuitry configured to execute the programmable
instructions and manipulate the data structures. It will be
apparent to those skilled in the art that other processing and
memory means, including various computer readable media, may be
used for storing and executing program instructions described
15 herein.

[0071] The storage operating system 230, portions of which is
typically resident in memory and executed by the processing
elements, functionally organizes the system 224 by, *inter alia*,
invoking storage operations in support of the storage service
20 provided by storage system 108. An example of operating system
230 is the DATA ONTAP® (Registered trademark of NetApp, Inc.
operating system available from NetApp, Inc. that implements a
Write Anywhere File Layout (WAFL® (Registered trademark of
NetApp, Inc.)) file system. However, it is expressly

contemplated that any appropriate storage operating system may be enhanced for use in accordance with the inventive principles described herein. As such, where the term "ONTAP" is employed, it should be taken broadly to refer to any storage operating
5 system that is otherwise adaptable to the teachings of this invention.

[0072] The network adapter 234 comprises a plurality of ports adapted to couple the system 224 to one or more server systems over point-to-point links, wide area networks, virtual private
10 networks implemented over a public network (Internet) or a shared local area network. The network adapter 234 thus may comprise the mechanical, electrical and signaling circuitry needed to connect storage system 108 to the network.

Illustratively, the computer network may be embodied as an
15 Ethernet network or a FC network.

[0073] The storage adapter 240 cooperates with the storage operating system 230 executing on the system 224 to access information requested by the server systems 104 and management system 118 (Figure 1A). The information may be stored on any
20 type of attached array of writable storage device media such as video tape, optical, DVD, magnetic tape, bubble memory, electronic random access memory, flash memory devices, micro-electro mechanical and any other similar media adapted to store information, including data and parity information.

[0074] The storage adapter 240 comprises a plurality of ports having input/output (I/O) interface circuitry that couples to the disks over an I/O interconnect arrangement, such as a conventional high-performance, FC link topology.

5 [0075] In another aspect, instead of using a separate network and storage adapter, a converged adapter is used to process both network and storage traffic.

[0076] Operating System: Figure 3 illustrates a generic example of operating system 230 executed by storage system 108,
10 according to one aspect of the present disclosure. Storage operating system 230 interfaces with the management system 118 and file system 144.

[0077] As an example, operating system 230 may include several modules, or "layers". These layers include a file system
15 manager 303 (e.g. L3 AFS) that keeps track of a directory structure (hierarchy) of the data stored in storage devices and manages read/write operations, i.e. executes read/write operations on disks in response to server system 104/140 (Figure 1B) requests.

20 [0078] Operating system 230 may also include a protocol layer 303 and an associated network access layer 305, to allow system 200 to communicate over a network with other systems, such as server system 104 and management system 118. Protocol layer 303 may implement one or more of various higher-level network

protocols, such as NFS, CIFS, Hypertext Transfer Protocol (HTTP), TCP/IP and others, as described below.

[0079] Network access layer 305 may include one or more drivers, which implement one or more lower-level protocols to communicate
5 over the network, such as Ethernet. Interactions between server systems 104 and mass storage devices 110/212 are illustrated schematically as a path, which illustrates the flow of data through operating system 230.

[0080] The operating system 230 may also include a storage
10 access layer 307 and an associated storage driver layer 309 to communicate with a storage device. The storage access layer 307 may implement a higher-level disk storage protocol, such as RAID (redundant array of inexpensive disks), while the storage driver layer 309 may implement a lower-level storage device access
15 protocol, such as FC or SCSI.

[0081] It should be noted that the software "path" through the operating system layers described above needed to perform data storage access for a client request may alternatively be implemented in hardware. That is, in an alternate aspect of the
20 disclosure, the storage access request data path may be implemented as logic circuitry embodied within a field programmable gate array (FPGA) or an ASIC. This type of hardware implementation increases the performance of the file service provided by storage system 108.

[0082] As used herein, the term "storage operating system" generally refers to the computer-executable code operable on a computer to perform a storage function that manages data access and may implement data access semantics of a general purpose operating system. The storage operating system can also be implemented as a microkernel, an application program operating over a general-purpose operating system, such as UNIX® or as a general-purpose operating system with configurable functionality, which is configured for storage applications as described herein.

[0083] In addition, it will be understood to those skilled in the art that the invention described herein may apply to any type of special-purpose (e.g., file server, filer or storage serving appliance) or general-purpose computer, including a standalone computer or portion thereof, embodied as or including a storage system. Moreover, the teachings of this disclosure can be adapted to a variety of storage system architectures including, but not limited to, a network-attached storage environment, a storage area network and a disk assembly directly-attached to a client or host computer. The term "storage system" should therefore be taken broadly to include such arrangements in addition to any subsystems configured to perform a storage function and associated with other equipment or systems.

[0084] Processing System: Figure 4 is a high-level block diagram showing an example of the architecture of a processing system, at a high level, in which executable instructions as described above can be implemented. The processing system 400 can
5 represent modules of host system 140, management system 118, user console 102, server systems 104, and others. Note that certain standard and well-known components which are not germane to the present invention are not shown in Figure 4.

[0085] The processing system 400 includes one or more processors
10 402 and memory 404, coupled to a bus system 405. The bus system 405 shown in Figure 4 is an abstraction that represents any one or more separate physical buses and/or point-to-point connections, connected by appropriate bridges, adapters and/or controllers. The bus system 405, therefore, may include, for
15 example, a system bus, a Peripheral Component Interconnect (PCI) bus, a HyperTransport or industry standard architecture (ISA) bus, a small computer system interface (SCSI) bus, a universal serial bus (USB), or an Institute of Electrical and Electronics Engineers (IEEE) standard 1394 bus (sometimes referred to as
20 "Firewire").

[0086] The processors 402 are the central processing units (CPUs) of the processing system 400 and, thus, control its overall operation. In certain aspects, the processors 402 accomplish this by executing programmable instructions stored in

memory 404. A processor 402 may be, or may include, one or more programmable general-purpose or special-purpose microprocessors, digital signal processors (DSPs), programmable controllers, application specific integrated circuits (ASICs), programmable
5 logic devices (PLDs), or the like, or a combination of such devices.

[0087] Memory 404 represents any form of random access memory (RAM), read-only memory (ROM), flash memory, or the like, or a combination of such devices. Memory 404 includes the main
10 memory of the processing system 400. Instructions 406 which implements techniques introduced above may reside in and may be executed (by processors 402) from memory 404. For example, instructions 406 may include code for executing the process blocks of Figure 1C. Memory 404 may be similar to PM 142 storing
15 data units and FS 1 144.

[0088] Also connected to the processors 402 through the bus system 405 are one or more internal mass storage devices 410, and a network adapter 412. Internal mass storage devices 410 may be or may include any conventional medium for storing large
20 volumes of data in a non-volatile manner, such as one or more magnetic or optical based disks. The network adapter 412 provides the processing system 400 with the ability to communicate with remote devices (e.g., storage servers) over a network and may be, for example, an Ethernet adapter, a FC

adapter, or the like. The processing system 400 also includes one or more input/output (I/O) devices 408 coupled to the bus system 405. The I/O devices 408 may include, for example, a display device, a keyboard, a mouse, etc.

5 [0089] Thus, methods and systems for protecting data have been described. Note that references throughout this specification to "one aspect" or "an aspect" mean that a particular feature, structure or characteristic described in connection with the aspect is included in at least one aspect of the present

10 invention. Therefore, it is emphasized and should be appreciated that two or more references to "an aspect" or "one aspect" or "an alternative aspect" in various portions of this specification are not necessarily all referring to the same aspect. Furthermore, the particular features, structures or
15 characteristics being referred to may be combined as suitable in one or more aspects of the present disclosure, as will be recognized by those of ordinary skill in the art.

[0090] While the present disclosure is described above with respect to what is currently considered its preferred aspects,
20 it is to be understood that the disclosure is not limited to that described above. To the contrary, the disclosure is intended to cover various modifications and equivalent arrangements within the spirit and scope of the appended claims.

What is Claimed is:

1. A method, comprising;

transferring at a first time via a network connection,
valid data units stored at a persistent memory of a
computing device managed by a first file system to a
5 storage device of a storage system managed by a second
file system;

generating by the first file system, at a second time,
a first snapshot of data units remaining at the persistent
10 memory after the first time;

transferring at a third time, metadata for data units
in the first snapshot, metadata associated with data units
remaining at the persistent memory between the second and
third time, and the data units remaining at the persistent
15 memory between the second and third time;

generating a second snapshot of a logical object at
the storage device by the second file system, the second
snapshot including the transferred metadata and data units
transferred at the first time and the third time; and

20 deleting the first snapshot at the computing device,
after the second snapshot is generated and releasing
storage blocks from the persistent memory associated with
the first snapshot.

2. The method of Claim 1, further comprising:

using the second snapshot for restoring data units and the associated metadata for use by the first file system.

3. The method of Claim 1, wherein after the second snapshot and upon deleting the first snapshot, data blocks at the storage device associated with data units of the first snapshot are released.

4. The method of Claim 1, wherein the metadata for the first snapshot is encoded before being transferred to the second storage device.

5. The method of Claim 1, wherein the first snapshot is retained at the first snapshot and a subsequent snapshot taken by the first file system after the second snapshot is deleted.

6. The method of Claim 1, wherein the storage system is a clustered system having a network component interfacing with the computing device and a storage component that manages the storage device.

7. The method of Claim 1, wherein a request for a data unit associated with the first snapshot is held while the first snapshot is being taken by the first file system.

8. A non-transitory machine readable storage medium having stored thereon instructions for performing a

method, comprising machine executable code which when executed by at least one machine, causes the machine to:

transfer at a first time via a network connection, valid data units stored at a persistent memory of a computing device managed by a first file system to a storage device of a storage system managed by a second file system;

generate by the first file system, at a second time, a first snapshot of data units remaining at the persistent memory after the first time;

transfer at a third time, metadata for data units in the first snapshot, metadata associated with data units remaining at the persistent memory between the second and third time, and the data units remaining at the persistent memory between the second and third time;

generate a second snapshot of a logical object at the storage device by the second file system, the second snapshot including the transferred metadata and data units transferred at the first time and the third time; and

delete the first snapshot at the computing device, after the second snapshot is generated and releasing storage blocks from the persistent memory associated with the first snapshot.

9. The non-transitory machine readable storage medium of

Claim 8, wherein the second snapshot is used for restoring data units and the associated metadata for use by the first file system.

10. The non-transitory machine readable storage medium of
5 Claim 8, wherein after the second snapshot and upon deleting the first snapshot, data blocks at the storage device associated with data units of the first snapshot are released.

11. The non-transitory machine readable storage medium of
10 Claim 8, wherein the metadata for the first snapshot is encoded before being transferred to the second storage device.

12. The non-transitory machine readable storage medium of
15 Claim 8, wherein the first snapshot is retained at the first snapshot and a subsequent snapshot taken by the first file system after the second snapshot is deleted.

13. The non-transitory machine readable storage medium of
20 Claim 8, wherein the storage system is a clustered system having a network component interfacing with the computing device and a storage component that manages the storage device.

14. The non-transitory machine readable storage medium of
Claim 8, wherein a request for a data unit associated with the first snapshot is held while the first snapshot is

being taken by the first file system.

15. A system comprising:

a memory containing machine readable medium comprising machine executable code having stored thereon instructions; and a processor module coupled to the memory to execute the machine executable code to:

transfer at a first time via a network connection, valid data units stored at a persistent memory of a computing device managed by a first file system to a storage device of a storage system managed by a second file system;

generate by the first file system, at a second time, a first snapshot of data units remaining at the persistent memory after the first time;

transfer at a third time, metadata for data units in the first snapshot, metadata associated with data units remaining at the persistent memory between the second and third time, and the data units remaining at the persistent memory between the second and third time;

generate a second snapshot of a logical object at the storage device by the second file system, the second snapshot including the transferred metadata and data units transferred at the first time and the third time; and

delete the first snapshot at the computing device,

after the second snapshot is generated and releasing storage blocks from the persistent memory associated with the first snapshot.

16. The system of Claim 15, wherein the second snapshot is
5 used for restoring data units and the associated metadata for use by the first file system.

17. The system of Claim 15, wherein after the second snapshot and upon deleting the first snapshot, data blocks at the storage device associated with data units of the
10 first snapshot are released.

18. The system of Claim 15, wherein the metadata for the first snapshot is encoded before being transferred to the second storage device.

19. The system of Claim 15, wherein the first snapshot is
15 retained at the first snapshot and a subsequent snapshot taken by the first file system after the second snapshot is deleted.

20. The system of Claim 15, wherein a request for a data unit associated with the first snapshot is held while the
20 first snapshot is being taken by the first file system.

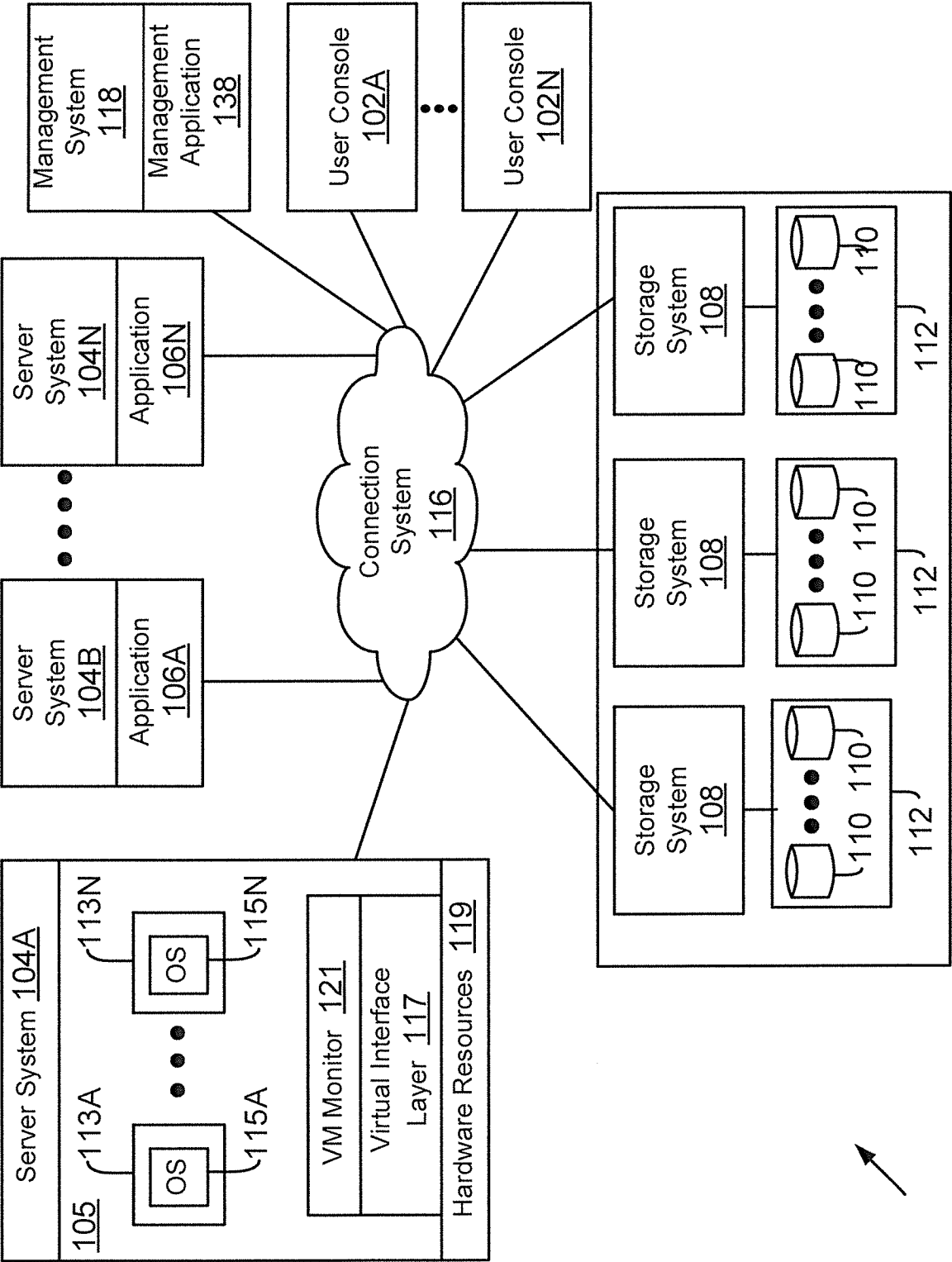


FIG. 1A

2/9

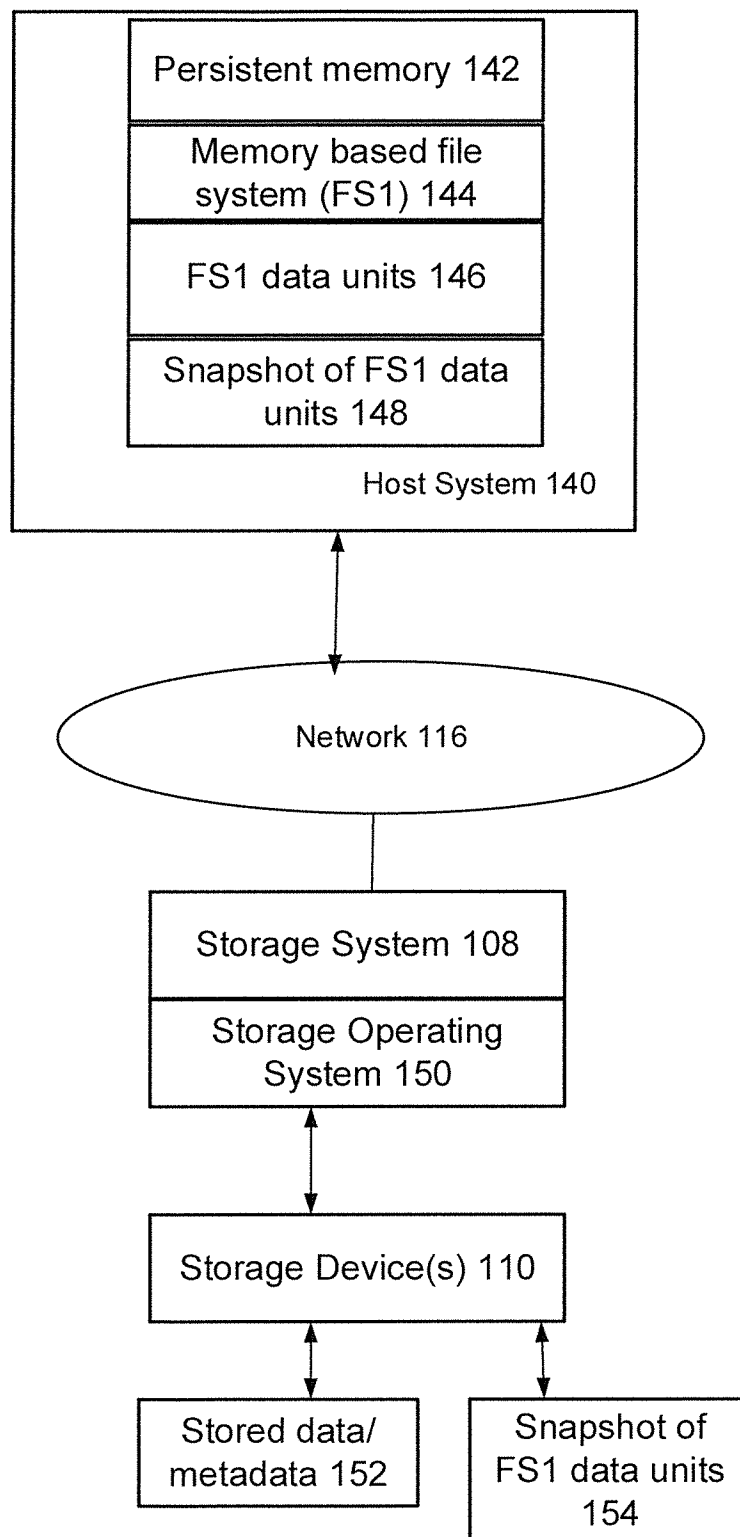


FIG. 1B

3/9

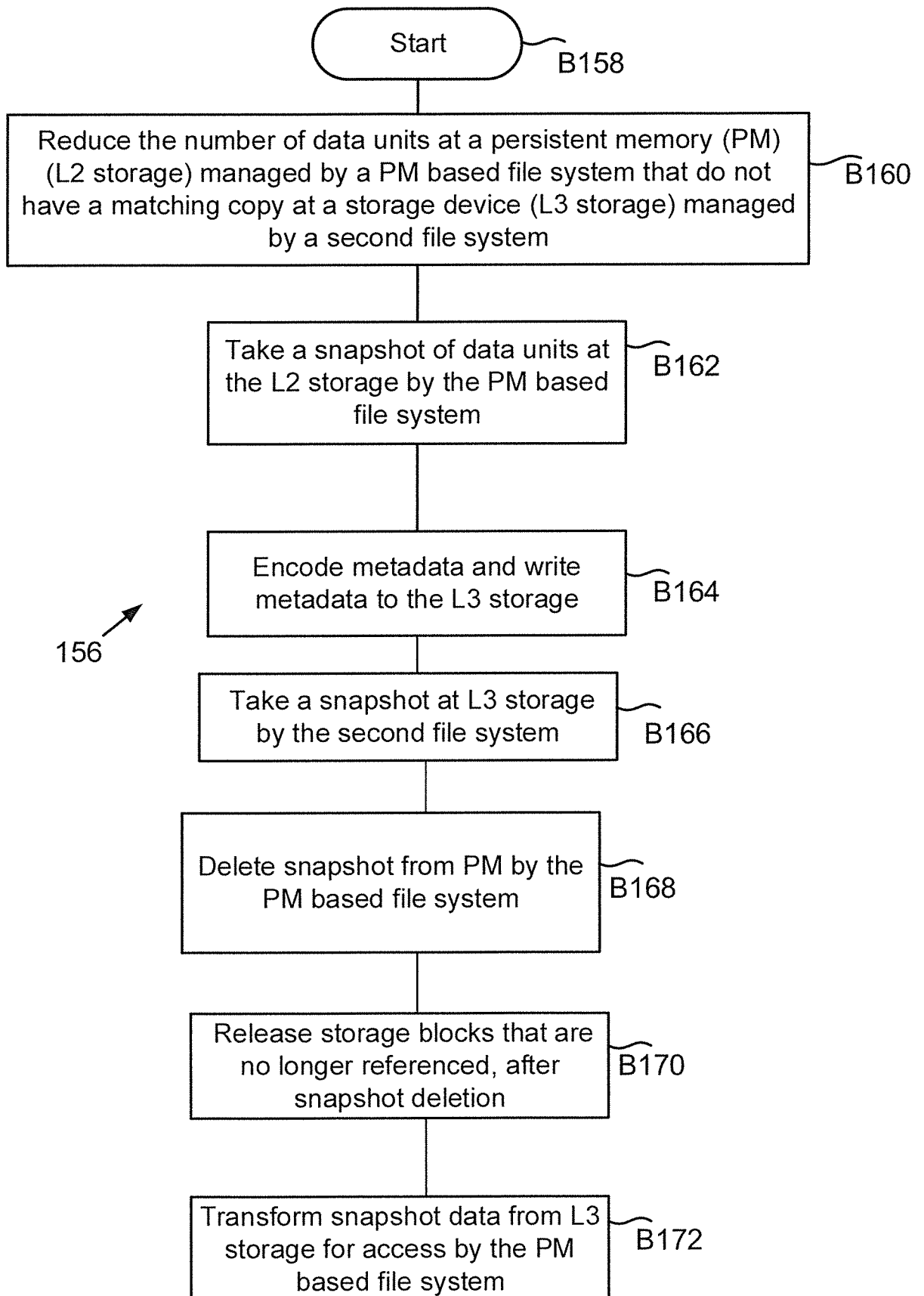


FIG. 1C

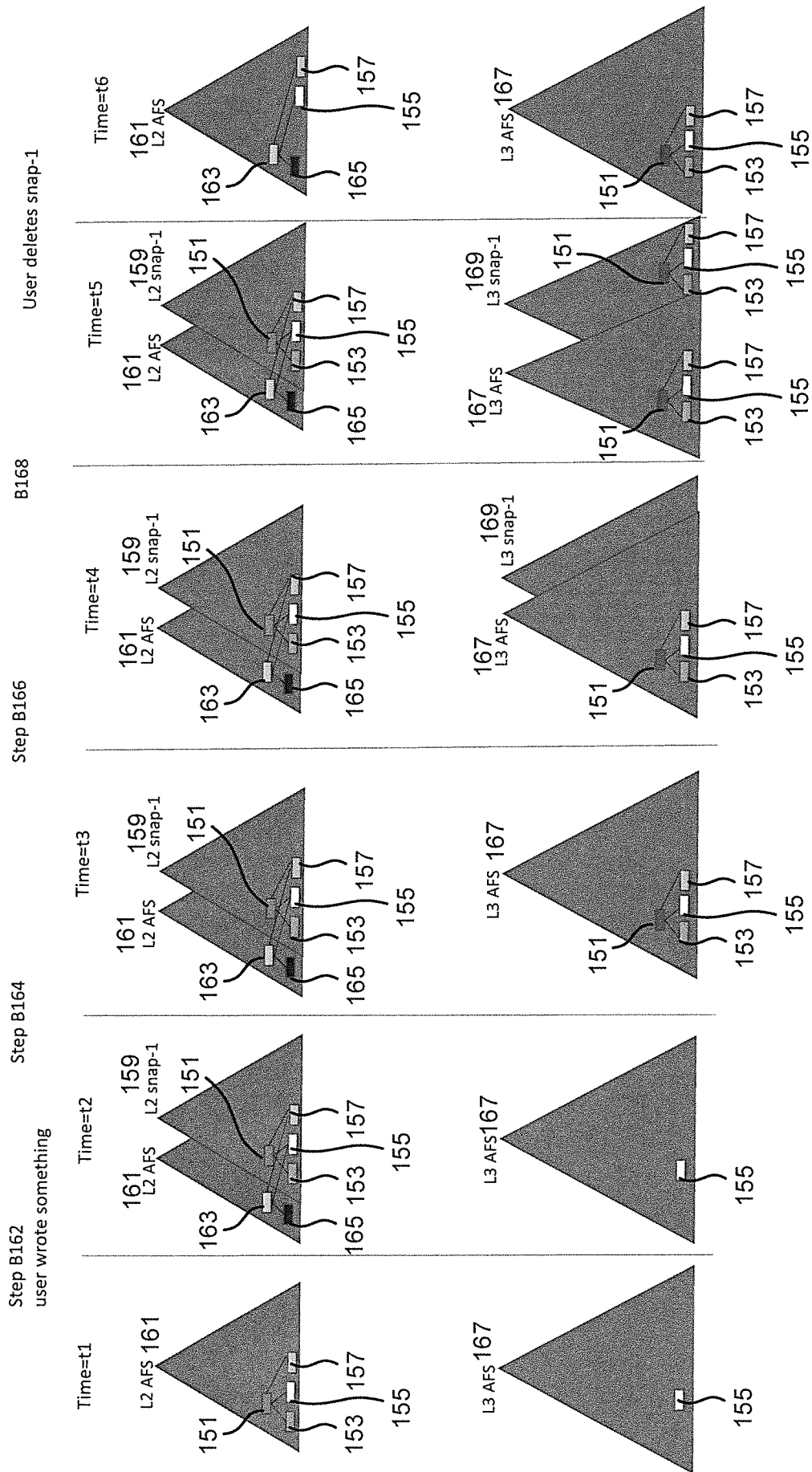


FIG. 1D

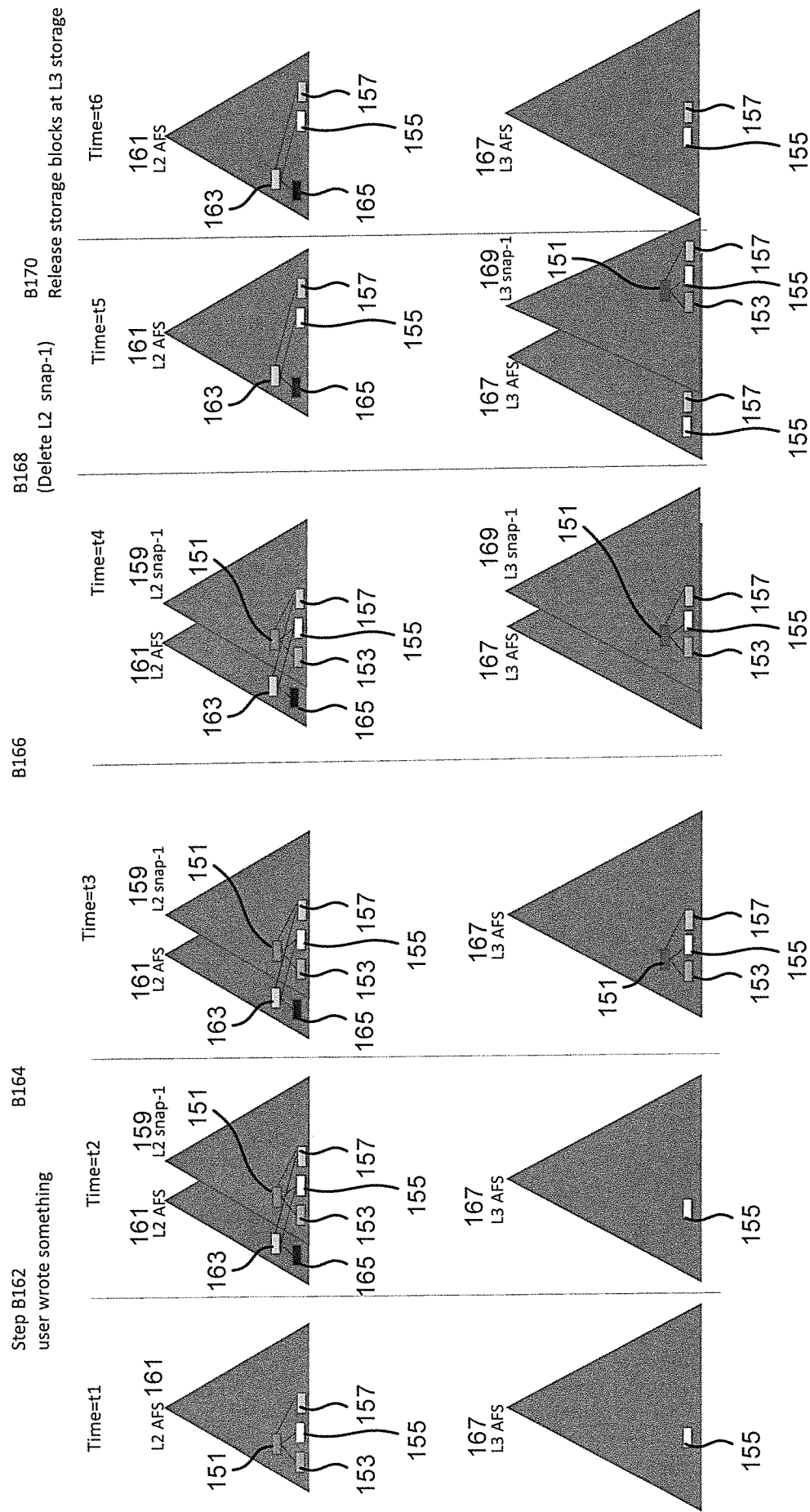


FIG. 1E

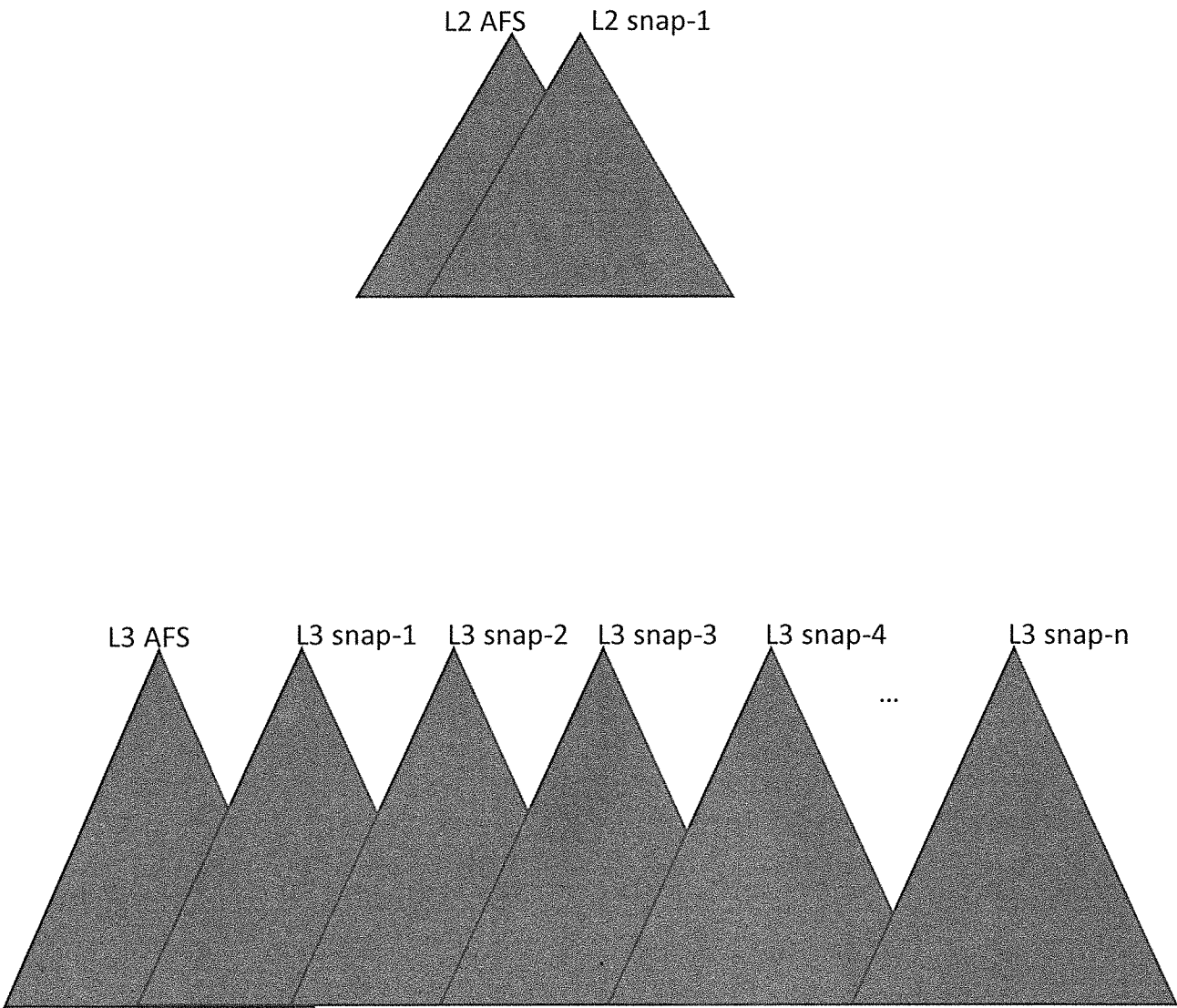


FIG. 1F

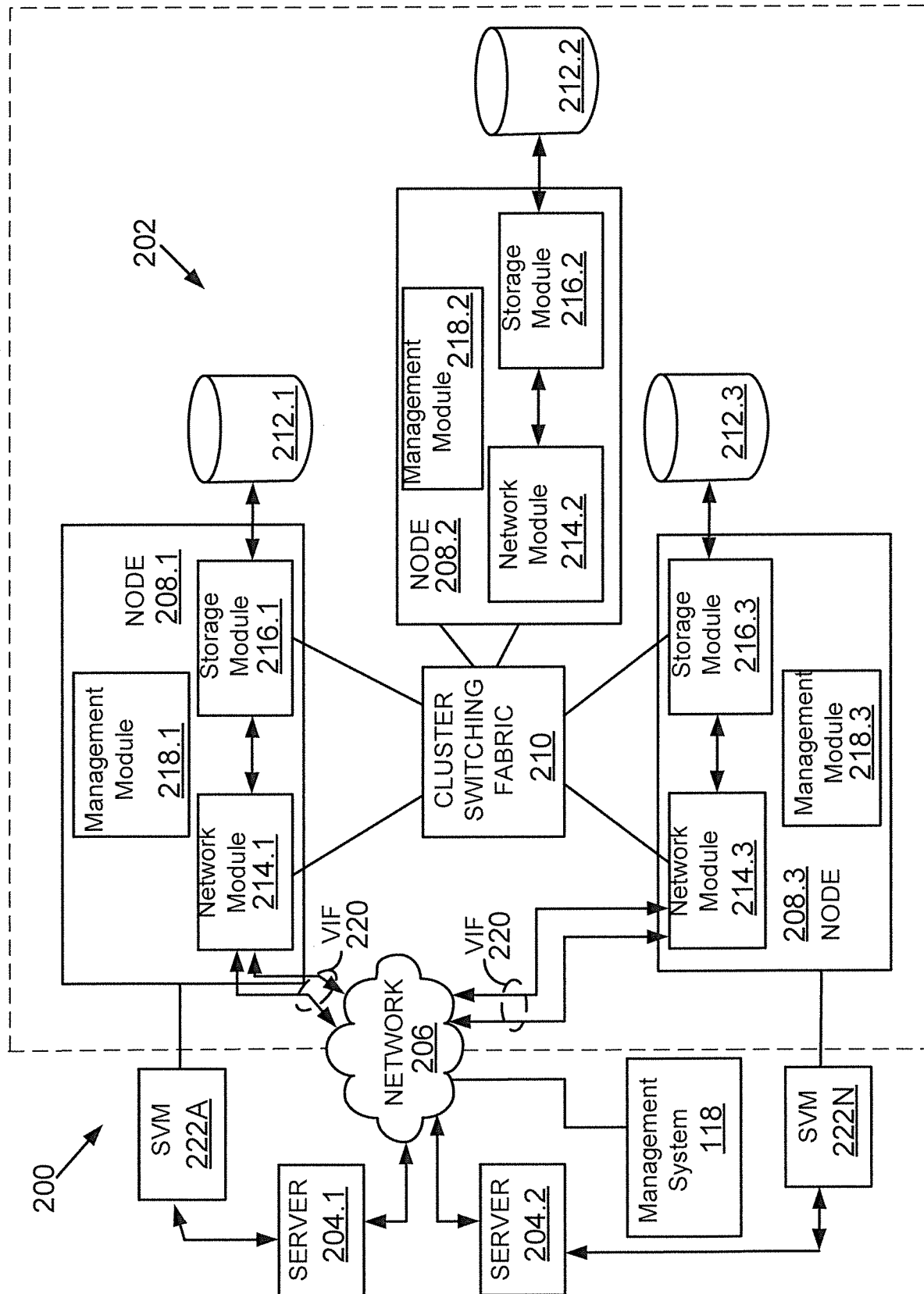


FIG. 2A

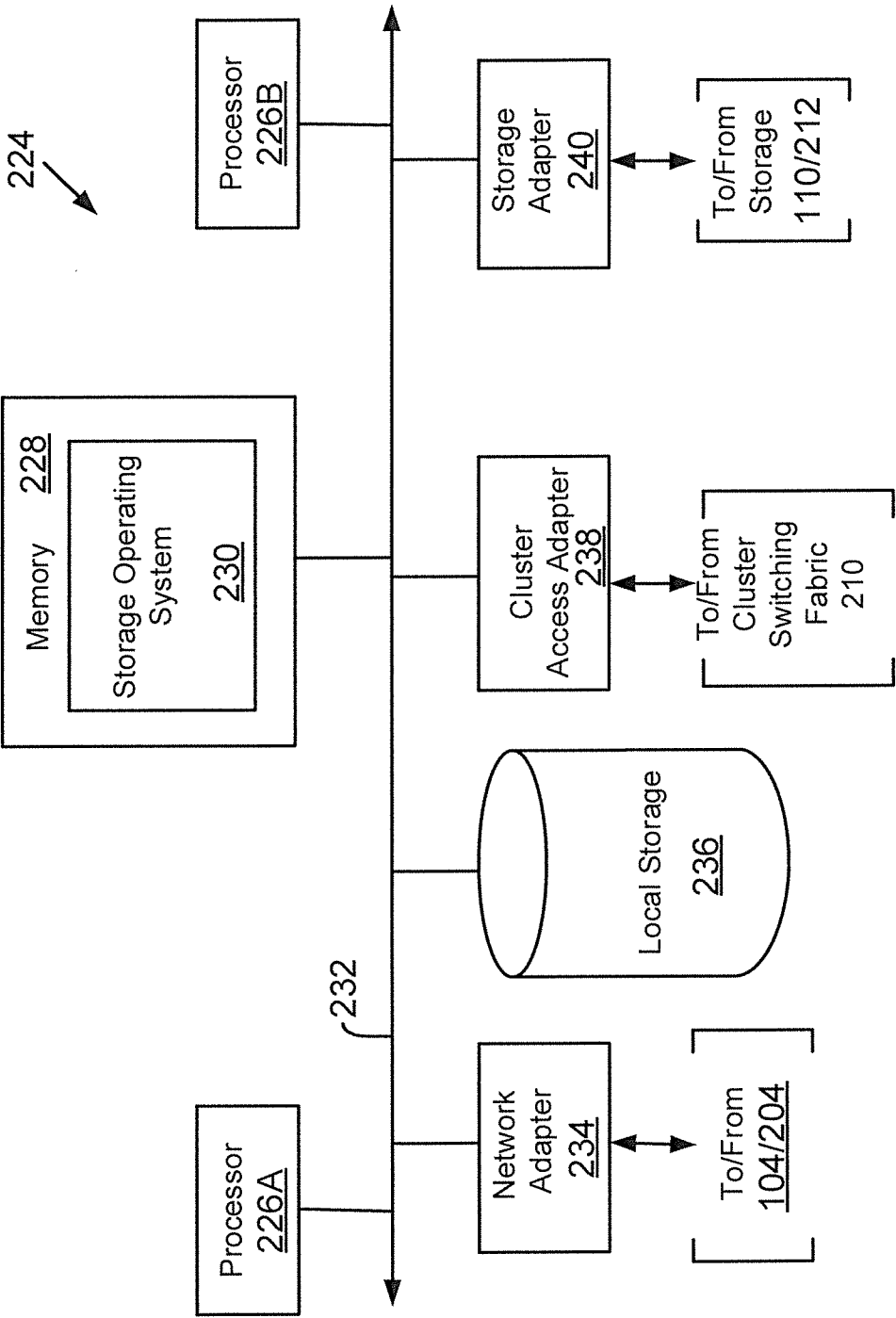


FIG. 2B

9/9

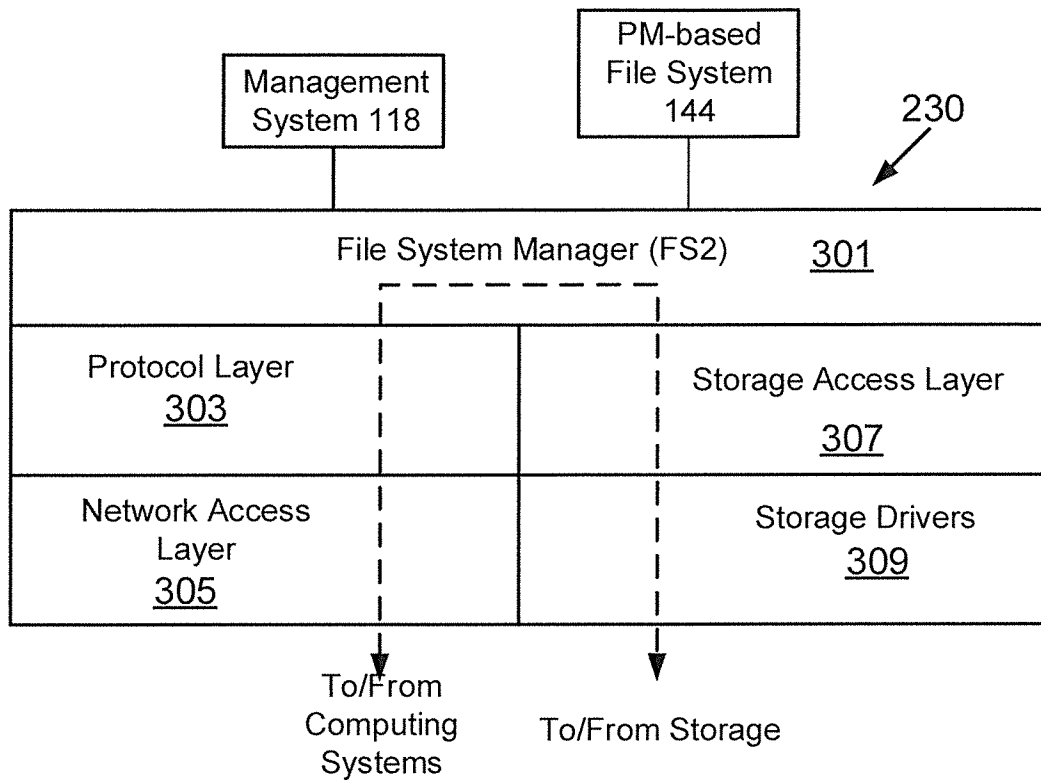


FIG. 3

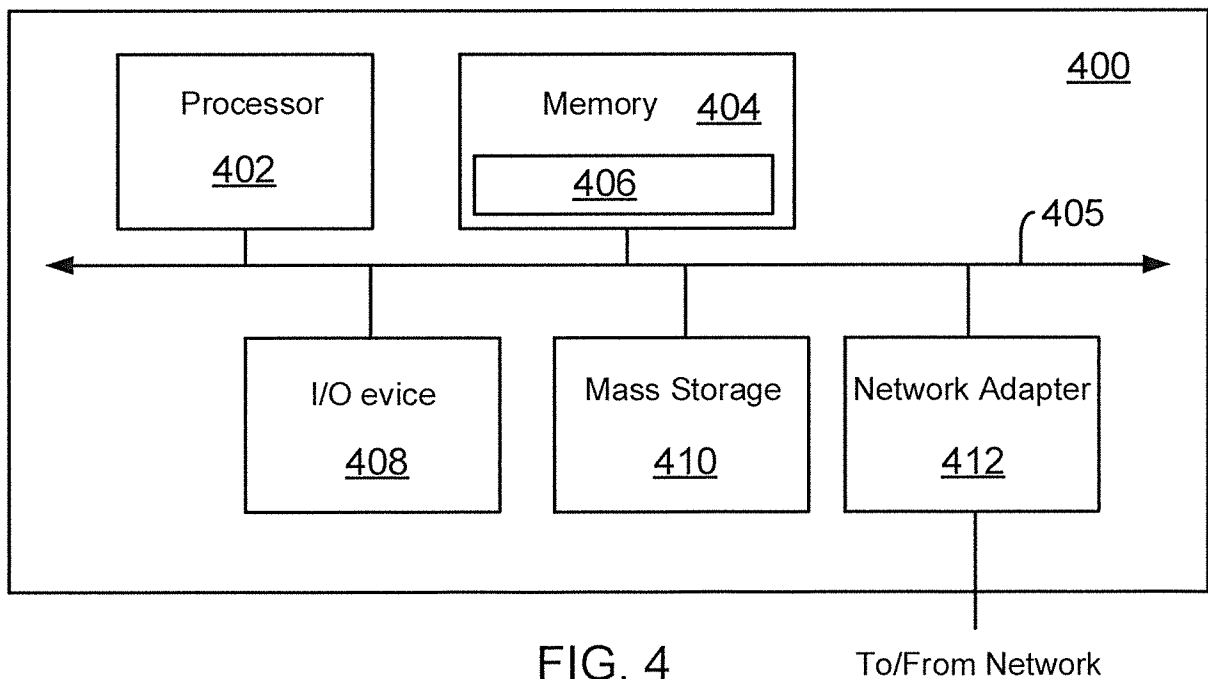


FIG. 4

INTERNATIONAL SEARCH REPORT

International application No

PCT/US2019/041069

A. CLASSIFICATION OF SUBJECT MATTER
 INV. G06F11/16
 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 2013/275674 A1 (RANGACHARI MURALIDHARAN [US] ET AL) 17 October 2013 (2013-10-17) abstract paragraph [0003] - paragraph [0012] paragraph [0028] - paragraph [0054] paragraph [0065] - paragraph [0071] paragraph [0080] - paragraph [0081] paragraph [0085] - paragraph [0089] paragraph [0092] - paragraph [0113] paragraph [0114] - paragraph [0116] ----- -/--	1-20



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

16 October 2019

Date of mailing of the international search report

25/10/2019

Name and mailing address of the ISA/

European Patent Office, P.B. 5818 Patentlaan 2
 NL - 2280 HV Rijswijk
 Tel. (+31-70) 340-2040,
 Fax: (+31-70) 340-3016

Authorized officer

Eftimescu, Nicolae

INTERNATIONAL SEARCH REPORT

International application No

PCT/US2019/041069

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2003/158862 A1 (ESHEL MARC M [US] ET AL) 21 August 2003 (2003-08-21) abstract paragraph [0002] - paragraph [0012] paragraph [0046] - paragraph [0064] paragraph [0066] - paragraph [0086] paragraph [0088] - paragraph [0100] paragraph [0103] - paragraph [0111] paragraph [0113] - paragraph [0118] paragraph [0125] - paragraph [0134] paragraph [0136] - paragraph [0144] paragraph [0163] - paragraph [0176] paragraph [0179] - paragraph [0201] claims 1-32; figures 1-17 -----	1-20

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/US2019/041069

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2013275674 A1	17-10-2013	US 8473777 B1	25-06-2013
		US 2013275674 A1	17-10-2013

US 2003158862 A1	21-08-2003	TW I260502 B	21-08-2006
		US 2003158862 A1	21-08-2003
