



(51) International Patent Classification:
G06F 9/30 (2006.01)

(21) International Application Number:

PCT/US2014/032418

(22) International Filing Date:

31 March 2014 (31.03.2014)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

61/807,673 2 April 2013 (02.04.2013) US
14/061,592 23 October 2013 (23.10.2013) US

(71) Applicant: WESTERN DIGITAL TECHNOLOGIES, INC. [US/US]; 3355 Michelson Drive, Suite 100, Irvine, California 92612 (US).

(72) Inventor: SADRY, Nauzad; 3355 Michelson Drive, Suite 100, Irvine, California 92612 (US).

(74) Agents: PUA, Don et al.; Western Digital Technologies, Inc., 3355 Michelson Drive, Suite 100, Irvine, California 92612 (US).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, 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, 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,

[Continued on next page]

(54) Title: METHODS AND SYSTEMS FOR PRIVILEGED EXECUTION SUPPORT FOR FILE SYSTEM COMMANDS ON A STORAGE DEVICE

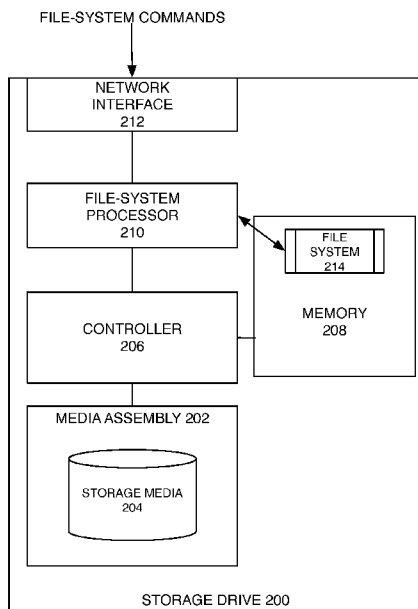


FIG. 2

(57) Abstract: The present invention relates to a storage device that is able to execute higher level commands, such as network-level, file-system commands, with privileged access to various resources, such as the storage media, hardware, memory, firmware, etc. In one embodiment, the storage device is configured to receive and execute network-level file-system commands, such as Server-Message-Block protocol commands. In particular, the storage device comprises a drive having a storage media and a communications interface, such as a network interface, and a controller. The controller is configured to interpret and execute network-level, file-system commands received from the communications interface on data stored on the storage media. Accordingly, the storage device can service the network-level, file-system commands more efficiently and without the need for user-space applications.



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:**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))*

- *with international search report (Art. 21(3))*
- *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))*

METHODS AND SYSTEMS FOR PRIVILEGED EXECUTION SUPPORT FOR FILE SYSTEM COMMANDS ON A STORAGE DEVICE

BACKGROUND

[0001] Data storage is a fundamental function of virtually all computing systems. Today, there are wide variety of data storage devices and technologies available. Typically, however, all data storage involves the use of a data storage device or drive, such as a hard disk drive, solid-state drive, etc. The known types of drives store and retrieve information at the block level and are interfaced using some form of block-level access, such as SCSI, IDE/ATA, SAS, SATA, etc. Drives may often be directly attached to a client or host computing device with a cable, and thus, are typically referred to as a direct attached storage or “DAS.” To further extend the availability of storage, one or more drives may be accessed over a network, such as, using iSCSI, network attached storage (“NAS”) or storage area networks (“SAN”).

[0002] Information, however, is usually organized based on a file system. File systems, such as network file system (“NFS”), common Internet file system (“CIFS”), server message block (“SMB”), etc., are well known file systems that employ network-level file-system commands to provide for shared access to files, etc. over a network, e.g., such as an Internet Protocol (“IP”) based network. Unfortunately, network-level file-system commands are not natively processed by storage drives.

[0003] Instead, the storage drives must be coupled with another device or computing resource that executes and translates the network-level file system commands into block-level commands. DAS drives rely on the operating system and application-level software running on their corresponding host, such as a personal computer, a server, etc. NAS devices comprise a controller that executes an operating system, such as LINUX or Windows, which supports running an application that can process network-level file-system commands. iSCSI and SAN storage systems rely on application servers to handle the file-system commands.

For example, the file-sharing software, SAMBA, is a well-known application that can receive file-system commands and interface with storage drives. However, this software runs in user space and uses device drivers and other services provided by the operating system running on the storage device. Unfortunately, user space software drivers can be unstable and/or suffer from poor performance. This performance disadvantage is especially pronounced on embedded computing devices and storage devices. Therefore, it would be desirable to provide methods and systems that improve the performance of computing devices, especially storage systems.

BRIEF DESCRIPTION OF THE DRAWINGS

[0004] Systems and methods which embody the various features of the invention will now be described with reference to the following drawings, in which:

[0005] FIGS. 1A and 1B illustrates a conventional storage device that employs a user-space application to support file system commands.

[0006] FIG. 2 illustrates an exemplary storage device that supports privileged access to resources of the storage device for execution of file system commands.

[0007] FIG. 3 illustrates an exemplary process flow in accordance with an embodiment of the present invention.

DETAILED DESCRIPTION

[0008] The embodiments relate to optimizing access to stored files, especially in a network environment. In one embodiment, the storage drives are able to execute higher level commands, such as network-level, file-system commands, with privileged access to various resources, such as the storage media, hardware, memory, firmware, etc. In one embodiment, the storage drive is configured to receive and execute network-level file-system commands, such as NFS or SMB commands. Accordingly, the storage drives can directly receive and execute the network-level, file-system commands more efficiently and without the need for executing user-space applications.

[0009] The embodiments of the invention may be implemented on any of a variety of operating systems, such as LINUX, Windows, UNIX, Mac OSX, and the like. For purposes of illustration, an exemplary embodiment for a storage device that runs the LINUX operating system is shown. One embodiment may employ a special purpose hardware or firmware component to execute the file-system commands. Alternatively, another embodiment uses kernel-mode device drivers to support network file system commands.

[0010] In addition, the embodiments may be implemented on any type of storage drive with storage media comprising a hard disk media, solid-state media, flash memory, etc. In the embodiments, the network-level file-system commands are transport agnostic, and thus, may be run over Ethernet, Wi-Fi, Infiniband, etc. The network level file-system protocols may include, but are not limited to, NFS, SMB, Apple Filing Protocol ("AFP"), file transfer protocol ("FTP"), remote file system ("RFS"), WebDAV, etc.

[0011] Certain embodiments of the inventions will now be described. These embodiments are presented by way of example only, and are not intended to limit the scope of the inventions. Indeed, the novel methods and systems described herein may be embodied in a variety of other forms. Furthermore, various omissions, substitutions and changes in the form of the methods and systems described herein may be made without departing from the spirit of the inventions. To illustrate some of the embodiments, reference will now be made to the figures.

[0012] FIGS. 1A and 1B shows a conventional storage device that provides a network share over a network. In particular, FIG. 1A shows a conventional direct attached storage drive. FIG. 1B shows a conventional network attached storage.

[0013] Referring now to FIG. 1A, a storage drive 102 may comprise a media assembly 104 having a storage media 106, a drive controller 108, a memory 110, and a drive interface 112. These components are well known to those skilled in the art.

[0014] As shown, the storage drive 102 is connected to a client device 114. As a conventional drive, the storage drive 102 is a block-level device and interfaces the client device 114 via storage interface 116. For example, the storage interface 116 and drive interface 112 may be a SCSI, IDE/ATA, SAS, or SATA interface.

[0015] In order to implement a file system, the client device 114 also comprises an operating system 118, which supports the execution of one or more applications 120. During operation, the applications 120 operate based on files and a file system, which are controlled and maintained by the operating system 118. In order to access or retrieve a file, the operating system 118 identifies which blocks correspond to a particular file and provide various block-level commands to the storage drive 102 via the drive interface 112. The controller 108 then locates the requested blocks on the storage media 106 and instructs the media assembly 104. Therefore, in this basic storage configuration, file-system commands are handled at the application level by an application running in an operating system's user space.

[0016] However, the same type of approach is used in shared files systems or distributed files systems over a network. As shown in FIG. 1B, a network attached storage (NAS) 100 is shown. The NAS 100 may comprise one or more storage drives 102, for example, in a pedestal or rack mounted chassis. In addition, the NAS 100 comprises a NAS controller 116, which executes an operating system 118 and a file server application 120. The NAS 100 further comprises a network interface 120 to connect to a network 122.

[0017] Notably, the file server application 120 is an application that runs in the user-space of the operating system 118 to support a network share. In this conventional configuration, the network interface 120 merely passes network communications from the network 120 to the operating system 118, which then forwards the appropriate communications to the file server 120. In turn, the file server 120 interprets and executes the various network file system commands by translating these file-system commands into block-level commands for the network share stored on the storage drive 102. Therefore, the conventional storage architectures require the use of a user-space application to execute network-level file-system commands.

[0018] As noted above, the embodiments overcome the need for user-space applications and software to support higher level commands on a drive, such as network-level file-system commands. In some embodiments, the drive can directly execute file-system commands, such as NFS commands, SMB commands, etc.

[0019] FIG. 2 shows an exemplary embodiment of a storage drive 200 that can execute file-system commands. As shown, the storage drive 200 comprises a media assembly 202, a storage media 204, a controller 206, a memory 208, a file-system processor 210, a network interface 212, and a file system 214. These components will now be briefly described.

[0020] The media assembly 202 and storage media 204 represent the components and physical medium used by the storage drive 200 to store and access data. For example, in one embodiment, the media assembly 202 and storage media 204 may be implemented as a hard disk medium, a solid state medium, a flash memory, etc. Any form of storage technology and medium may be used in the embodiments.

[0021] The controller 206 controls the operations of the storage drive 200. For example, the controller 206 may control the operations for reading and writing data to the media assembly 202 and storage media 204. The controller 206 may be implemented using well-known components to those skilled in the art.

[0022] The memory 208 serves as temporary memory location for the controller 206, such as for buffering data, queuing instructions, etc. As shown, the memory 208 also provides a storage location for at least a portion of a file system 214 that has been implemented on the storage drive 200.

[0023] File-system processor 210 represents the processing resources that receive and execute network-level file-system commands for the storage drive. As described further below, the file-system processor 210 may be implemented in a variety of ways. The file-system processor 210 possesses privileged access to resources of the storage drive, such as the media assembly 202 and storage media 204. Privileged access may refer to those forms of access that are restricted to an exclusive group of components or access that is treated preferentially or differently from normal access requests, such as conventional application software requests running in the user space of an operating system. For example, in some embodiments, privileged access may refer to direct access that does not require translation of commands or an intermediate component. Kernel device drivers are one example of components having privileged access in accordance with some

embodiments. In another embodiment, as further described below, the file-system processor 210 may be implemented based on hardware or firmware.

[0024] In some embodiments, the storage drive 200 is an embedded device running an embedded operating system. In other embodiments, the controller 206 in the storage drive 200 may be loaded with an operating system, such as LINUX or Windows and the file-system processor 210 comprises a set of kernel device drivers that are configured to execute network-level file-system commands, such as SMB commands or NFS commands. The file-system processor 210 may also comprise block-level storage media device driver, such as a SATA driver, to interface with the media assembly 202 and storage media 204.

[0025] In some embodiments, the file-system processor 210 may comprise device drivers that implement various security protections, for example, to prevent buffer overruns and restriction of certain operations. The device drivers may be written in known programming languages, such as C, etc.

[0026] The kernel device drivers used in the file-system processor 210 may be included in the storage device 200 in various ways. For example, in one embodiment, the kernel device drivers may be loaded on to the controller 206 as loadable kernel modules. Alternatively, the file-system processor 210 may be installed as part of the embedded operating system in controller 206.

[0027] Since the file-system processor 210 operates in the kernel space of the operating system, it may run with privileged access to the resources of the storage drive 200, such as the memory 208, the controller 206, the media assembly 202, and storage media 204 relative to other code or software running in user space on the storage device 200.

[0028] These forms of kernel device drivers in the file-system processor 210 allow the storage device 200 to execute the network file system commands in kernel space with enhanced privileges, which is faster and more efficient than using user space software.

[0029] In some embodiments, the file-system processor 210 may be implemented using a combination of hardware and/or firmware. For example, in one embodiment, the file-system processor 210 may comprise firmware, such as a field-programmable gate array, or application-specific integrated circuit, which is

configured to interpret and execute network-level file-system commands. In one embodiment, the file-system processor 210 comprises firmware that is configured to execute SMB commands. In another embodiment, the file-system processor 210 comprises firmware that is configured to execute NFS commands. As noted, the file-system processor 210 may be configured to execute any type of file system or protocol.

[0030] The network interface 122 provides a communications interface for the storage drive 100. The network interface 122 may support any type of wired or wireless communications, such as Ethernet, Gigabit Ethernet, infiniband, Wi-Fi, etc.

[0031] In one embodiment, the network interface 122 comprises one or more hardware or firmware components for sending and receiving packets. For example, the network interface 122 may be configured to encapsulate/de-encapsulate TCP/IP packets transporting various network-level file-system commands, such as SMB or NFS commands. The network interface 122 then passes these commands to the file-system processor 210 for interpretation and execution.

[0032] In one embodiment, the network interface 122 may also provide various security features, such as encryption/decryption. For example, the network interface 122 may comprise an encryption chip or system.

[0033] In some embodiments, the network interface 122 is supported by a device driver executing on the controller 206 and/or file-system processor 210. In one embodiment, the device driver for the network interface 122 runs in the kernel space provided by the operating system running on the controller 206.

[0034] FIG. 3 illustrates an exemplary process flow of one embodiment. For purposes of illustration, FIG. 3 relates to a storage drive 200 that is attached to network via a network interface, such as an Ethernet or Gigabit Ethernet interface, and providing access to its files based on NFS commands received from a client (not shown) or other host device (not shown). Of course, other types of network-level file-system commands may be processed and executed by the storage drive 200.

[0035] As shown, in stage 300, the storage drive 200 may receive one or more network file system ("NFS") commands via the network interface 212. The NFS commands may be encapsulated within one or more data packets, such as IP packets, transported over Ethernet frames on a network to which the network

interface 212 is connected. As noted, the network interface 212 may be an Ethernet or Gigabit Ethernet interface. The components for such types of interfaces are known to those skilled in the art. The network interface 212 may then perform various data receiving and transmitting functions related to de-encapsulating the NFS commands from the Ethernet frame and IP packets.

[0036] In stage 302, the network interface 212 may forward the NFS command to the file-system processor 210. File-system processor 210 may comprise hardware and/or software running on the controller 206, which has privileged access to the other components of the storage drive 200. As noted previously, for example, in one embodiment, the file-system processor 210 may be implemented as special-purpose firmware (such as a field-programmable gate array or “FPGA”, or application specific integrated circuit, or “ASIC”) on the storage drive 200. Alternatively, the file-system processor 210 may be program code, such as a collection of software and device drivers running on the controller 206.

[0037] In order to support execution of the network-level file-system commands, such as an NFS command, the file-system processor 210 may access file system data 214 stored in the memory 208. The file system data 214 may also be retrieved from storage media 204 by the file-system processor 210.

[0038] In stage 304, the file-system processor 210 executes the network-level file-system command, such as a NFS command or SMB command, by using its privileged access to the resources of the storage drive 200. Of note, in contrast to conventional storage drives, the storage drive 200 is able to execute the file-system command itself rather than relying on another device, such as a client computer running application software or application-level software running on the drive. Instead, as shown in FIGS. 2 and 3, the storage drive 200 may receive the file-system commands directly and execute them using the file-system processor 210.

[0039] The file-system processor 210 may execute the file-system commands based on privileged access. For example, the file-system processor 210 may be firmware that is connected to the controller 206 and given access to its processing logic and may have privileged access to the media assembly 202 and storage media 204. In addition, the file-system processor 210 may have privileged access to the memory 208.

[0040] Alternatively, in another embodiment, the file-system processor 210 may be software running on the controller 206, but having privileged access as a kernel-level application or kernel device drivers running within the operating system installed on controller 206.

[0041] In one embodiment, file-system processor 210 executes the file-system commands by providing file-level commands, rather than block-level commands, to the controller 206 and media assembly 202. In order to interpret the file-level commands, the controller 206 and media assembly 202 may be configured to employ a mapping or data structure that maps a file to locations on the storage media 204. For example, an exemplary table is illustrated below for executing NFS commands.

File Handle	Node	Generation	Storage Media Sector

[0042] Such a table may be maintained by the file-system processor 210 and stored in the memory 208 or in a location on the storage media 204. The above table is merely exemplary and one of ordinary skill will understand that the embodiments may be implemented for other types of file systems and file-system commands.

[0043] The features and attributes of the specific embodiments disclosed above may be combined in different ways to form additional embodiments, all of which fall within the scope of the present disclosure. Although the present disclosure provides certain embodiments and applications, other embodiments that are apparent to those of ordinary skill in the art, including embodiments, which do not provide all of the features and advantages set forth herein, are also within the scope of this disclosure. Accordingly, the scope of the present disclosure is intended to be defined only by reference to the appended claims.

WHAT IS CLAIMED IS:

1. A storage drive configured to execute file system commands, said storage drive comprising:
 - at least one storage media;
 - a controller configured to control the at least one storage media;
 - a memory, coupled to the controller, storing at least a portion of a file system for files stored on the at least one storage media;
 - a communications interface configured to communicate with a network and receive network-level, file-system commands; and
 - a processor, coupled to the communications interface and having privileged access to the controller and at least one storage media, configured by program code to execute the received network-level, file-system commands relating to files stored on the at least one storage media.
2. The storage drive of claim 1, wherein the at least one storage media comprises a hard disk.
3. The storage drive of claim 1, wherein the at least one storage media comprises a solid-state medium.
4. The storage drive of claim 1, wherein the communications interface comprises an Ethernet-protocol network communications interface.
5. The storage drive of claim 4, wherein the processor executes network-level, file-system commands.
6. The storage drive of claim 4, wherein the processor executes server message block commands.
7. The storage drive of claim 1, wherein the controller is configured by program code for an embedded operating system.

8. The storage drive of claim 1, wherein the processor comprises firmware implementing program code to execute the network level, file-system commands.
9. A storage device configured to execute commands with privileged access to storage media of the storage device, said storage device comprising:
- at least one storage media;
 - a controller configured to control the at least one storage media;
 - a memory, coupled to the controller, for storing at least a portion of a file system related to files stored on the at least one storage media;
 - a network interface configured to interface a network; and
 - a processor executing an operating system, having privileged access to the controller and the at least one storage media and coupled to the network interface, configured by program code to execute network-level, file-system commands relating to files stored on the at least one storage media.
10. The storage device of claim 9, wherein the storage media comprise at least one hard disk.
11. The storage device of claim 9, wherein the controller is configured to execute an embedded LINUX operating system.
12. The storage device of claim 9, wherein the controller is configured to execute an embedded Windows operating system.
13. The storage device of claim 9, wherein the processor comprises firmware implementing the program code for executing network-level, file-system commands.
14. The storage device of claim 9, wherein processor is configured to execute server message block protocol commands on files stored on the at least one storage media based on privileged access to the at least one storage media.

15. A method of processing network-level, file-system commands by a storage drive, wherein the storage drive comprises a storage media, a controller for controlling the storage media, a memory, coupled to the controller, storing at least a portion of a file system related to files stored on the storage media, a network communications interface for communications with a network, and a processor executing program code based on privileged access to the controller and the storage media of the storage drive, said method comprising:

- receiving, via the network communications interface, at least one network-level, file-system commands;

- forwarding, from the network interface to the processor, the received at least one network-level, file-system commands; and

- executing, by the processor on the storage media, the network-level, file-system commands based on the privileged access of the processor to the controller and the storage media.

16. The method of claim 15, wherein executing the network-level, file-system commands comprises executing, in firmware within the processor, the network-level, file-system commands.

17. The method of claim 15, wherein executing the network-level, file-system commands comprises executing, by the processor, server message block protocol commands on the storage media.

18. A network attached storage device configured to execute network-level, file-system commands, said storage device comprising:

- a storage media;

- a controller for controlling the storage media;

- a memory storing at least a portion of a file system for files stored on the storage media;

- a network interface configured to interface a network and receive network-level, file-system commands; and

a processor, coupled to the network interface, executing program code with privileged access to the controller and storage media to execute network-level, file-system commands related to files stored on the storage media.

19. The storage device of claim 18, wherein the storage media comprise at least one hard drive.

20. The storage device of claim 18, wherein the controller is configured to execute an embedded LINUX operating system.

21. The storage device of claim 18, wherein the processor is configured to execute a Windows operating system.

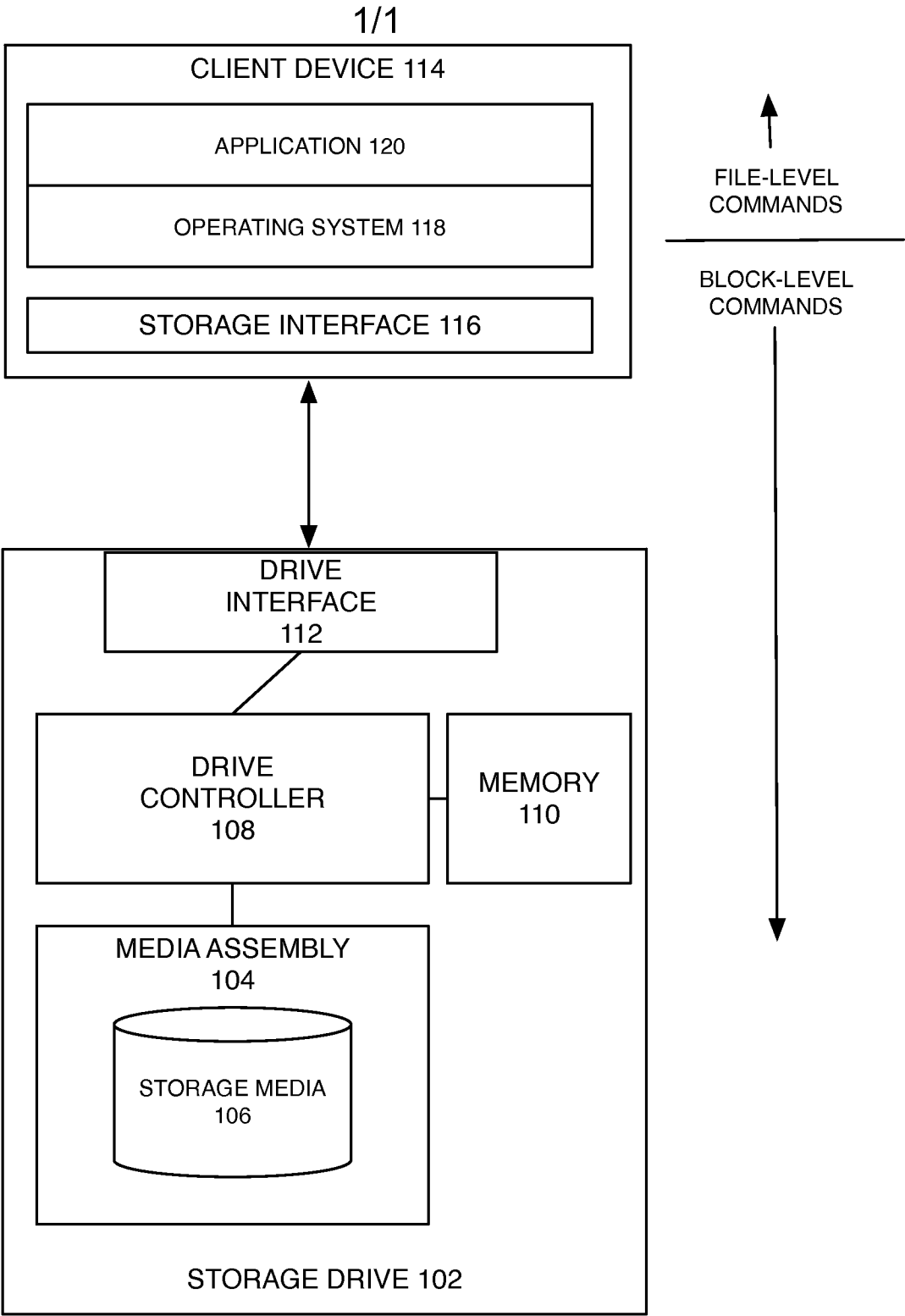
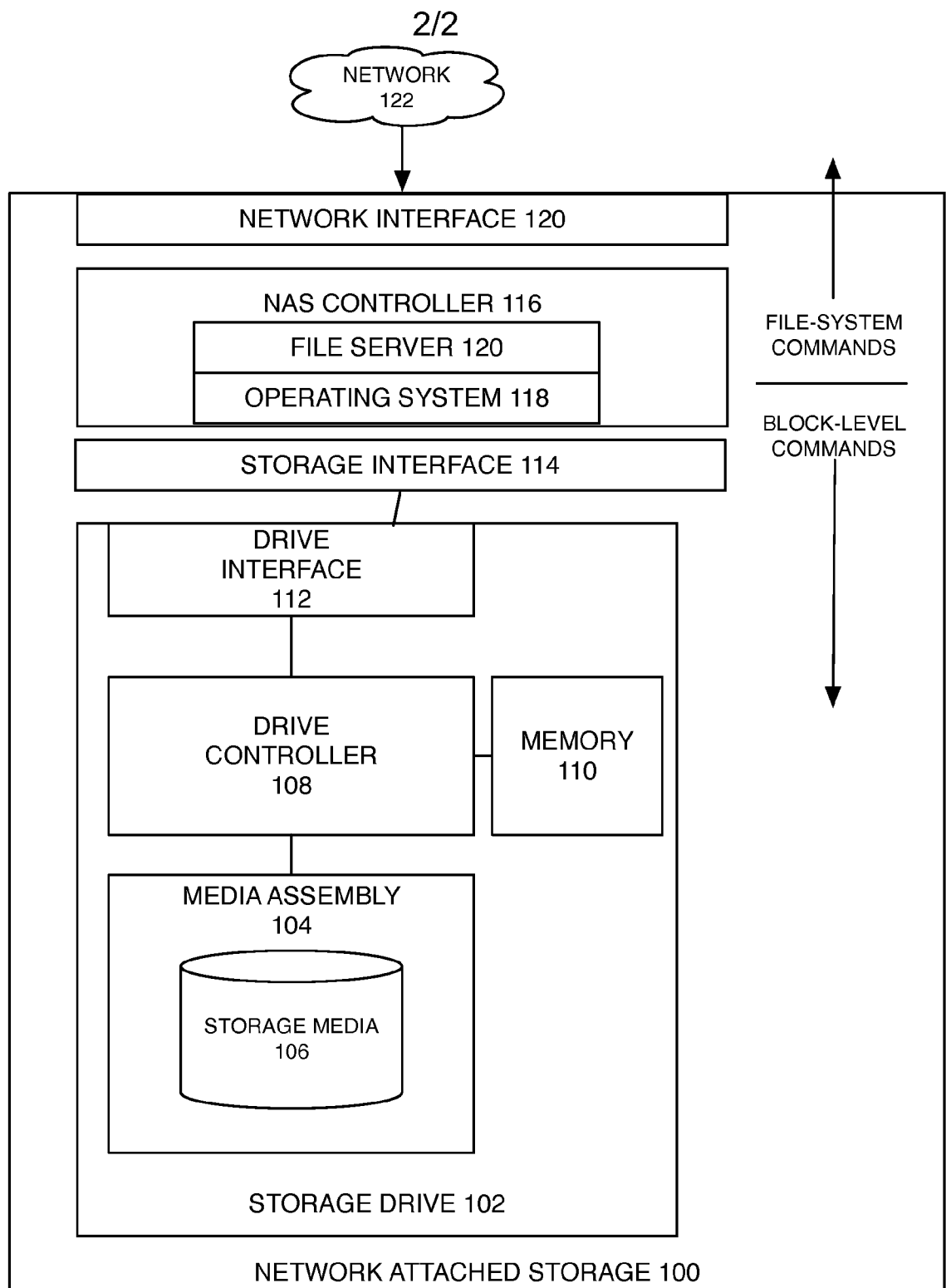


FIG. 1A
(PRIOR ART)



3/3

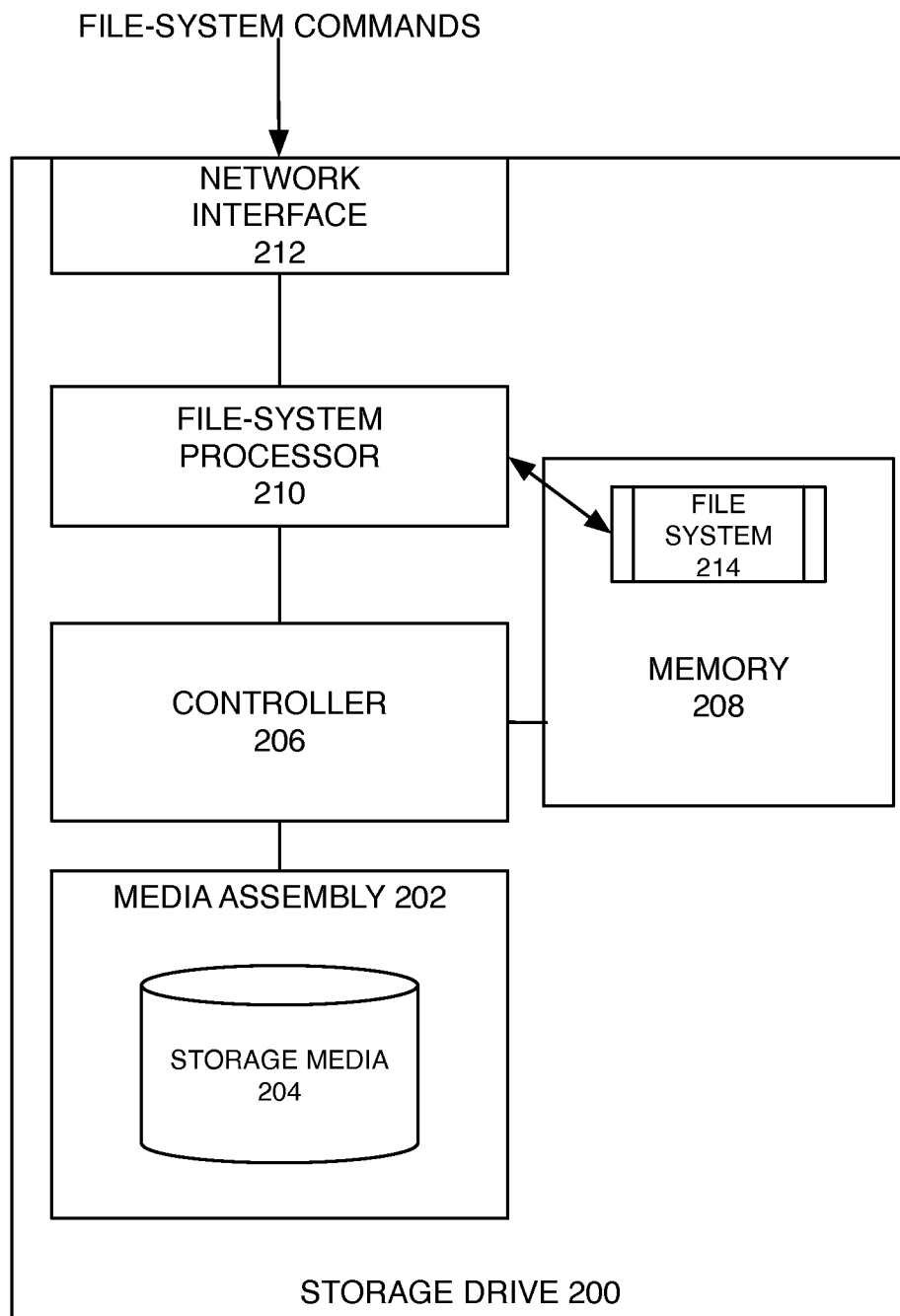
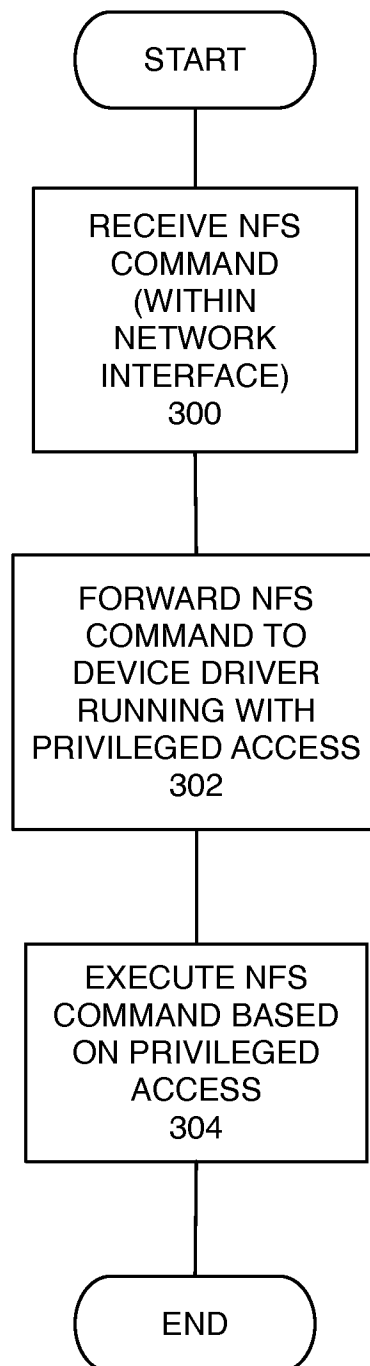


FIG. 2

4/4

FIG. 3



INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2014/032418**A. CLASSIFICATION OF SUBJECT MATTER****G06F 9/30(2006.01)i**

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F 9/30

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

Korean utility models and applications for utility models

Japanese utility models and applications for utility models

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

eKOMPASS(KIPO internal) & Keywords: storage system, network-attached storage, privileged access.

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	WO 03-017682 A2 (QUALCOMM, INCORPORATED et al.) 27 February 2003 See abstract, paragraphs [0009]-[0012], and figures 2-5.	1-21
A	US 2010-0223373 A1 (LITTLEFIELD DUNCAN et al.) 02 September 2010 See abstract, claims 1-15, and figures 4-10.	1-21
A	US 2011-0138116 A1 (LIPINSKI GREG J. et al.) 09 June 2011 See abstract, paragraphs [0025]-[0031], and figures 4-5.	1-21
A	US 2007-0067364 A1 (DOUGLAS F. BARBIAN et al.) 22 March 2007 See abstract, paragraphs [0011]-[0015], and figures 1-5.	1-21



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

31 July 2014 (31.07.2014)

Date of mailing of the international search report

31 July 2014 (31.07.2014)

Name and mailing address of the ISA/KR

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

Facsimile No. +82-42-472-7140

Authorized officer

JI, Jeong Hoon

Telephone No. +82-42-481-5688



INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2014/032418

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 03-017682 A2	27/02/2003	AR 036214A1	18/08/2004
		BR 0211917 A	28/09/2004
		CA 2457617 A1	27/02/2003
		CN 1564992 A	12/01/2005
		CN 1564992 C0	28/03/2007
		EP 1417602 A2	12/05/2004
		EP 1417602 A4	22/08/2007
		HK 1070442 A1	07/09/2007
		IL 160395D0	25/07/2004
		JP 2005-500608 A	06/01/2005
		JP 2005-500608 T	06/01/2005
		JP 2010-182319 A	19/08/2010
		KR 10-0538659 B1	26/12/2005
		KR 20040015702 A	19/02/2004
		MX PA04002494A	30/07/2004
		NZ 531200 A	31/03/2006
		PE 03752003 A1	12/04/2003
		RU 2004107492 A	27/09/2005
		RU 2307388 C2	27/09/2007
		TW 236298 B	11/07/2005
		TW I236298B	11/07/2005
		US 2003-0061504 A1	27/03/2003
		US 7921287 B2	05/04/2011
		WO 03-017682 A3	27/11/2003
		WO 0301-7682 A2	27/02/2003
		WO 0301-7682 A3	27/11/2003
US 2010-0223373 A1	02/09/2010	US 2008-0147754 A1	19/06/2008
		US 2008-0147836 A1	19/06/2008
		US 2008-0147997 A1	19/06/2008
		US 2011-0035419 A1	10/02/2011
		US 2011-0035621 A1	10/02/2011
		US 7769890 B2	03/08/2010
		US 8073969 B2	06/12/2011
		US 8677091 B2	18/03/2014
US 2011-0138116 A1	09/06/2011	CN102203743 A	28/09/2011
		EP 2340491 A1	06/07/2011
		WO 2010-047713 A1	29/04/2010
US 2007-0067364 A1	22/03/2007	EP 1710676 A2	11/10/2006
		EP 1710676 A3	19/08/2009
		JP 2006-294031 A	26/10/2006
		US 2012-254264 A1	04/10/2012
		US 8223600 B2	17/07/2012