



- (51) International Patent Classification:
G06F 13/10 (2006.01) *G06F 13/14* (2006.01)
- (21) International Application Number:
PCT/US2013/062756
- (22) International Filing Date:
30 September 2013 (30.09.2013)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:
13/714,385 13 December 2012 (13.12.2012) US
- (71) Applicant: **WESTERN DIGITAL TECHNOLOGIES, INC.** [US/US]; 3355 Michelson Drive, Suite 100, Irvine, CA 92612 (US).
- (72) Inventors: **LIN, James**; c/o Western Digital Technologies, Inc., 3355 Michelson Drive, Suite 100, Irvine, CA 92612 (US). **BARNES, Edwin**; c/o Western Digital Technologies, Inc., 3355 Michelson Drive, Suite 100, Irvine, CA 92612 (US).
- (74) Agent: **DELANEY, Karoline, A.**; Knobbe, Martens, Olson & Bear, LLP, 2040 Main Street, 14th Floor, Irvine, CA 92614 (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, 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 PROVISIONING A BOOTABLE IMAGE ON TO AN EXTERNAL DRIVE

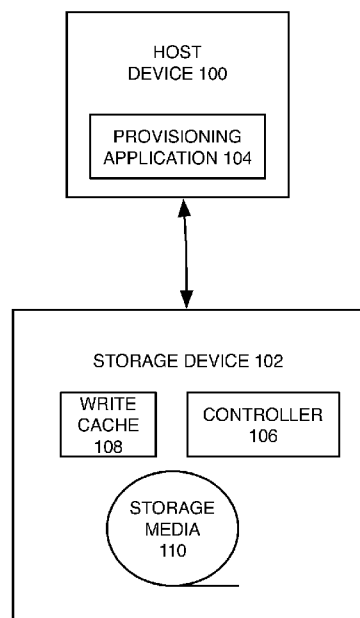


FIG. 1

(57) Abstract: The present invention relates to a method of optimizing the provisioning of a bootable image onto a storage device. In some embodiments, a host device executes a provisioning application to image a storage drive as a bootable drive. During the provisioning process, the storage device is configured to disguise its use of write caching during the provisioning process. In one embodiment, the storage device is configured to suppress forced unit access commands and cache flush commands for the provisioning application. In another embodiment, the storage device is configured to reject forced unit access commands. The storage device may disguise its use of write caching based on various criteria, such as a length of time, a counter, and the like.



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).

— *as to the applicant's entitlement to claim the priority of
the earlier application (Rule 4.17(iii))*

Declarations under Rule 4.17:

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

Published:

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

METHODS AND SYSTEMS FOR PROVISIONING A BOOTABLE IMAGE ON TO AN EXTERNAL DRIVE

BACKGROUND

[0001] Today, many operating systems can be booted from a portable external storage device, such as a direct attached storage device. For example, the Microsoft Windows 8, supports "Windows To Go." Windows To Go allows a portable storage drive to be imaged with a full bootable operating system. This allows the portable storage drive to be plugged into virtually any computer and run its image.

[0002] In order to image a portable drive, it is connected to a host device, such as a personal computer or server and imaged with an imaging application or tool, such as ImageX. Unfortunately, the provisioning process has been found to take considerable time, e.g., several hours. For an enterprise seeking to image large numbers of drives, this amount of time is a significant disadvantage. Therefore, it would be desirable to optimize or reduce the amount of time needed to create bootable storage device.

BRIEF DESCRIPTION OF THE DRAWINGS

[0003] Systems and methods which embody the various features of the invention will now be described with reference to the following drawings, in which:

[0004] FIG. 1 is a block diagram illustrating a storage device that is coupled to a host device.

[0005] FIG. 2 illustrates a conventional process for imaging a portable storage device.

[0006] FIG. 3 illustrates an exemplary process flow for imaging a portable storage device in accordance with an embodiment of the present invention.

DETAILED DESCRIPTION

[0007] The embodiments relate to creating a portable or external bootable storage device. With the known imaging processes, the writing of the bootable image to the storage device employs a conservative write process that requires frequent acknowledgements and confirmation exchanges between the host and the storage device. In particular, many of the known provisioning processes employ forced unit access commands frequently to guarantee that the write data has been written to the media of the storage device. However, the frequent use of these commands slow the performance of the storage device because it prevents the use of write caching by the storage device.

[0008] Accordingly, in the embodiments, methods and systems are provided to reduce the amount of time needed to create a bootable image on a storage device. For example, one embodiment relates to provisioning a bootable image onto an external storage device, such as a direct attached storage device. Those skilled in the art will recognize that any type of bootable external drive may be created by the embodiments. In some embodiments, in order to help reduce the amount of time needed for the imaging process, the storage device is configured to disguise its use of write caching from the imaging application running on the host device. The storage device responds as if it is following the write commands and confirmation acknowledgements of the host during provisioning, but is transparently employing write caching. The disguise feature avoids error conditions and allows the imaging application to function normally. To minimize risks of this approach, the disguise feature may be configured with a limited duration, such as a finite number of cycles or short time period. The write caching allows the storage device to optimize its performance and drastically reduces the amount of time needed to provision the bootable image on to the drive.

[0009] 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.

[0010] FIG. 1 is a block diagram illustrating a host device 100 that is coupled to a storage device 102. As will be explained further below, in some embodiments, the host device 100 images the storage device 102 as a bootable storage device, such as a Windows To Go Drive.

[0011] Host device 100 refers generally to any computer or system. For example, the host device 100 may be a personal computer, a laptop computer, a server, a tablet computer, a mobile device, etc. Host device 100 may comprise well-known hardware, firmware, and software. For example, host device 100 may comprise a processor (not shown) executing an operating system, such as the Windows™, Mac OS, Linux, UNIX, operating systems, and the like.

[0012] In addition, as shown in FIG. 1, the host device 100 may comprise a communications interface, such as a Universal Serial Bus (USB) interface, for communication with the storage device 102. As is well known, USB is an industry standard interface that defines cabling, connectors, and communications protocols for communications between a computer, such as host device 100 and a peripheral device, such as storage device 102.

[0013] As shown, a host device 100 may be coupled to a storage device 102 via one of its interfaces, such as a USB interface, a firewire interface, a Thunderbolt interface, etc.

[0014] As also shown, the host device 100 provides a hardware and software platform upon which a provisioning application 104 may execute. The provisioning application 104 is a tool or application that creates, edits, and deploys a bootable image, such as an image with a bootable operating system, to a target storage device or drive. In some embodiments, the provisioning application 104 uses the known ImageX tool provided by Microsoft™. In one embodiment, the host device 100 is running a provisioning application 104, for example, provided by the manufacturer of the storage device 102. The provisioning application 104 may be downloaded or copied onto the host device 100 from a network, such as the Internet, or the drive itself.

[0015] In some embodiments, the provisioning application 104 is configured to recognize the storage device 102 when it is coupled to the host device 100 and trigger disguising of its write caching for the provisioning process. In one embodiment, the provisioning application 104 is included as part of an application pre-installed on the storage device 102.

[0016] Storage device 102 refers to any data storage device that may be coupled to the host device 100. For example, the storage device 102 may be a hard disk drive, a flash drive, a solid-state drive, a hybrid drive, a network attached storage, etc. As noted, the storage device 102 may be a portable, external storage device that is intended to include a bootable image. The storage device 102 may connect to the host device 100 via a USB 2.0 or 3.0 connection and the like.

[0017] As will be further described below, in some embodiments, the storage device 102 is configured to disguise its use of write caching during provisioning of the bootable image by the provisioning application 104. In some embodiments, this disguise feature is implemented for a limited duration, such as a finite number of cycles, a period of time, etc. For example, in one embodiment, the storage device 102 is configured to limit its disguise

features to 3 power cycles or connection cycles. Alternatively, in other embodiments, the storage device 102 is configured to limit its disguise features for a time period, such as 1 hour, 1 day, etc.

[0018] As shown, the storage device 102 may comprise various hardware and software that is known to those skilled in the art. For example, the storage device 102 may comprise, among other things, a controller 106, a write cache 108, and a storage media 110.

[0019] The controller 106 performs various operations of the storage device 102 described herein. The controller 106 may be implemented using one or more processors for executing instructions and may further include memory, such as a volatile or non-volatile memory, for storing data (e.g., data being processed) and/or instructions. The instructions may be executed by the one or more processors to perform the various functions of the controller 106 described herein. The one or more processors may include a microcontroller, a Digital Signal Processor (DSP), an Application Specific Integrated Circuit (ASIC), a Field Programmable Gate Array (FPGA), hard-wired logic, analog circuitry and/or a combination thereof.

[0020] The write cache 108 is an embedded memory of the storage device to act as a buffer between the host device 100 and the storage media 110. When using write caching, the controller 106 signals the host device 100 that a write command is complete immediately after receiving the write data, even before the data are actually written to the storage media 110. This early signal allows the host device 100 and provisioning application 104 to continue working even though the data has not actually been written yet.

[0021] Of note, in conventional provisioning of a bootable image, the write process attempts to avoid or minimize the use of write caching, for example, through the use of Forced Unit Access commands or cache flush commands. This is done by conventional

provisioning processes, because if power is lost before the data are permanently fixed in the media, the data will be lost from the write cache 108, and the file system and boot image may be left in an inconsistent state. Accordingly, the conventional policy is to turn caching off because the integrity of data is deemed more important than write performance. In addition, the conventional provisioning processes send data to disk in a carefully managed order and to issue cache flush commands at various times. Unfortunately, this results in a provisioning process that takes significant amounts of time, e.g., several hours or more.

[0022] In some embodiments, the controller 106 is configured to disguise its use of write caching during provisioning of a bootable image by managing the vulnerable period between signaling the write complete and actual performance of writing data to the storage media 110. For example, the controller 106 may control deferral of any write commands as newly arriving requests are received. In some embodiments, the storage device 102 may also comprise a battery-backed memory system for the write cache 108 or other protective features.

[0023] The storage media 110 represents the media in which data is stored. The storage media 110 may comprise, for example, any number of magnetic media, solid-state media, etc. For example, the storage media 110 may comprise a number of magnetic disks. Alternatively, the storage media 110 may comprise a solid-state drive. Of course, in yet other embodiments, the storage media 110 may comprise different forms of media alone or in combination.

[0024] FIG. 2 illustrates a conventional imaging process that is known to those skilled in the art. In particular, first, a provisioning application may send write data for a bootable image to the storage device. Typically, the storage device will temporarily hold this write data in its write cache.

[0025] Second, as noted above, the provisioning application deems data integrity more important than performance, and thus, sends a cache flush or Forced Unit Access command to the storage device. This command is intended to limit or prevent the use of write caching by the storage device.

[0026] Third, the storage device acknowledges the command, which causes it delay writing data to its storage medium. Fourth, the storage device is commanded to write the data from the write cache in to the storage medium. This process is repeated until the bootable image has been written to the storage device. As can be seen, however, this conventional process is tightly controlled by the provisioning application and host and prevents the storage device from optimizing its performance. Accordingly, FIG. 3 is provided below to illustrate an embodiment of the present invention.

[0027] FIG. 3 illustrates an exemplary process flow in which the storage device 102 disguises its use of write caching and allows normal execution of the provisioning application 104. As shown, first, the host device 100 has installed a provisioning application 104. The provisioning application 104 may be retrieved from the storage device 102. Alternatively, the host device 100, for example may download the provisioning application 104, from the Internet.

[0028] In some embodiments, the host device 100 may run a utility application 111. In one embodiment, the utility application 111 provides a utility and user interface for a user at host device 100 to request and configure the storage device 102 as a bootable external drive. In some embodiments, the utility application 111 sends one or more commands to the storage device 102 to commence disguise of its write caching, which is described further below.

[0029] In some embodiments, the utility application 111 works in cooperation with the provisioning application 104 to provision and image a bootable image onto the storage

device 102. For example, a user of host device 100 may first start the utility application 111 to determine the configuration and available features of the storage device 102. The user may then run the provisioning application 104 to request imaging of a bootable image on to the storage device. As noted, the provisioning application 104 may then employ a tool or utility, such as ImageX, provided by the operating system of the host device 100 to image the storage device 102.

[0030] For example, when started or running, the utility application 111 may be configured to recognize the storage device 102 and whether it is capable of expediting the imaging process, for example, by disguising write caching. In some embodiments, the utility application 111 may determine the device identifier of the storage device 102 to determine the manufacturer, model, device drivers, etc. This information may be retrieved, for example, from the registry maintained by the operating system running on the host device 100. If the utility application 111 determines that the storage device 102 does not support expedited imaging, then it will image the bootable image conventionally. If the storage device 102 can support expedited imaging, then the utility application 111 may send an indicator, command, or flag to the storage device 102.

[0031] In response, the storage device 102 may initiate its disguise feature, which is described further below. In some embodiments, the storage device 102 implements the disguise feature for a limited duration. For example, the storage device 102 may only use its disguise feature for a finite number of power cycles, i.e., when the storage device 102 is connected/disconnected from the host device 100, or turned off/on, etc. In other embodiments, the storage device 102 will limit the disguise feature for a limited time, such as 30 minutes, 1 hour, etc.

[0032] After checking the storage device 102, the utility application 111 may then have the provisioning application 104 commence imaging of the storage device 102. The

write data for the bootable image may be sent in various forms known to those skilled in the art.

[0033] Second, as shown during this writing process, the operating system (not shown) of the host device 100, at various times, sends a Forced Unit Access command, cache flush command, etc. to the storage device 102. These operations are requested by the operating system in order to protect the data integrity of the bootable image. For example, the operating system may send a Forced Unit Access Command after each file has been sent to the storage device. In addition, the operating system of the host device 100 may send one or more synchronized cache commands to the storage device 100. As noted, the operating system of the host device 100 sends these commands protect the data integrity of the bootable image. However, such protective features can dramatically slow down the write performance of the storage device 102. For example, in the prior art, imaging of a bootable image and associated data can take up to several hours.

[0034] In the embodiments, however, the storage device 102 provides an alternative to the inefficient data integrity used in the prior art. In particular, third, the storage device 102 acknowledges the command from the host device 100 to indicate that the write cache 108 will be flushed and confirm when the write data has been fixed to the storage media 110. In some embodiments, however, this acknowledgement or response is a disguise of the write cache because, contrary to the acknowledgement, the storage device 102 continues caching the write data in the write cache 108. The acknowledgment provides a positive response back to the host device 100 and the operating system. Thus, host device 100 is effectively led to believe that the storage device 102 has performed the requested data integrity commands, e.g., a Forced Unit Access command or cache flush command. The storage device 102, however, may continue to use write caching and optimize the write performance of the bootable image.

[0035] Fourth, the storage device 102 employs the use of write caching to optimize its write performance. Such write acceleration is known to those skilled in the art. The process repeats or continues until the bootable image has been written to the storage device 102 and storage media 110.

[0036] 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. For example, the present disclosure is described with reference to a storage device, such as a hard disk drive, coupled to a host device via a USB interface. The invention may also be implemented in other types of devices, such as a peripheral device, a flash drive, a solid-state drive, a hybrid drive, a network attached storage that communicate via USB. In addition, the invention may be implemented for other communications protocols. 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 method of provisioning a bootable image onto a portable storage device, wherein the portable storage device comprises a write cache and a storage media, said method comprising:

receiving, from a host, write data for a bootable image;

receiving, from the host, at least one command requesting immediate writing of the write data to the media of the portable storage device;

disguising, by the portable storage device from the host, write caching of the write data in the write cache of portable storage device; and

writing, after the disguise of the write caching, the write data to the storage media of the portable storage device.

2. The method of claim 1, wherein receiving write data for a bootable image comprises receiving write data for configuring the portable storage device as a Windows To Go drive.

3. The method of claim 1, wherein receiving at least one command requesting immediate writing of the write data comprises receiving a forced unit access command from the host.

4. The method of claim 1, wherein receiving at least one command requesting immediate writing of the write data comprises receiving a cache flush command from the host.

5. The method of claim 1, wherein disguising write caching of the write data by the portable storage device comprises accepting a forced unit access command from the host, but continuing to temporarily cache the write data in the write cache of the portable storage device.

6. The method of claim 1, wherein disguising write caching of the write data by the portable storage device comprises rejecting a forced unit access command from the host.

7. The method of claim 1, wherein disguising write caching of the write data by the portable storage comprises disguising write caching based on a counter.

8. The method of claim 7, wherein disguising write caching based on a counter comprises:

- setting the counter to an initial value;
- decrementing the counter in response to a startup cycle of the portable storage device; and
- terminating disguising of the write caching when the counter reaches a value of zero.

9. The method of claim 7, wherein disguising write caching based on a counter comprises:

- setting the counter to an initial value;
- incrementing the counter in response to a startup cycle of the portable storage device; and
- terminating disguising of the write caching when the counter reaches a threshold value.

10. The method of claim 1, wherein disguising write caching of the write data comprises disguising write caching of the write data based on a time period.

11. A portable storage device comprising:

- a medium for storing data and an image comprising a bootable operating system;
- a write cache for temporarily storing data to be written to the medium;
- an interface configured for communications with a host device; and

a controller configured to cache write data from the host in the write cache, respond to requests from the host device for requesting immediate writing of data from the write cache to the media, wherein the controller is configured to selectively disguise write caching during provisioning of the boot image onto the portable storage device, and write the write data to the storage medium after disguising the write caching.

12. The portable storage device of claim 11, wherein the portable storage device is configured as a Windows To Go drive.

13. The portable storage device of claim 11, wherein the controller is configured to receive a forced unit access command from the host.

14. The portable storage device of claim 11, wherein the controller is configured to receive a cache flush command from the host.

15. The portable storage device of claim 11, wherein the controller is configured to disguise write caching of the write data by accepting a forced unit access command from the host, but continuing to temporarily cache the write data in the write cache of the portable storage device.

16. The portable storage device of claim 11, wherein the controller is configured to disguise write caching of the write data by rejecting a forced unit access command from the host.

17. The portable storage device of claim 11, wherein the controller is configured to disguise write caching of the write data based on a counter.

18. The portable storage device of claim 17, wherein the controller is configured to set the counter to an initial value, decrement the counter in response to a startup cycle of the portable storage device, and terminate disguising of the write caching when the counter reaches a value of zero.

19. The portable storage device of claim 17, wherein the controller is configured to set the counter to an initial value, increment the counter in response to a startup cycle of the portable storage device, and terminate disguising of the write caching when the counter reaches a threshold value.

20. The portable storage device of claim 11, wherein the controller is configured to disguise write caching of the write data temporarily based on a time period.

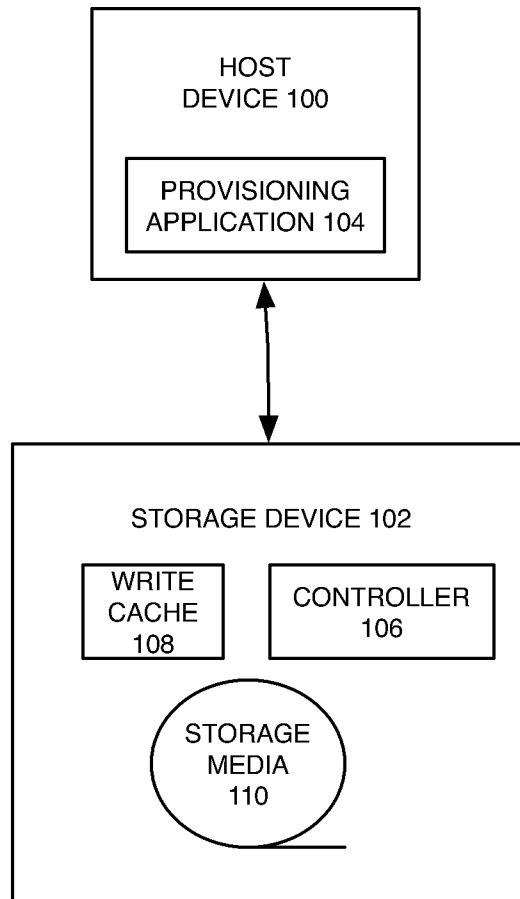


FIG. 1

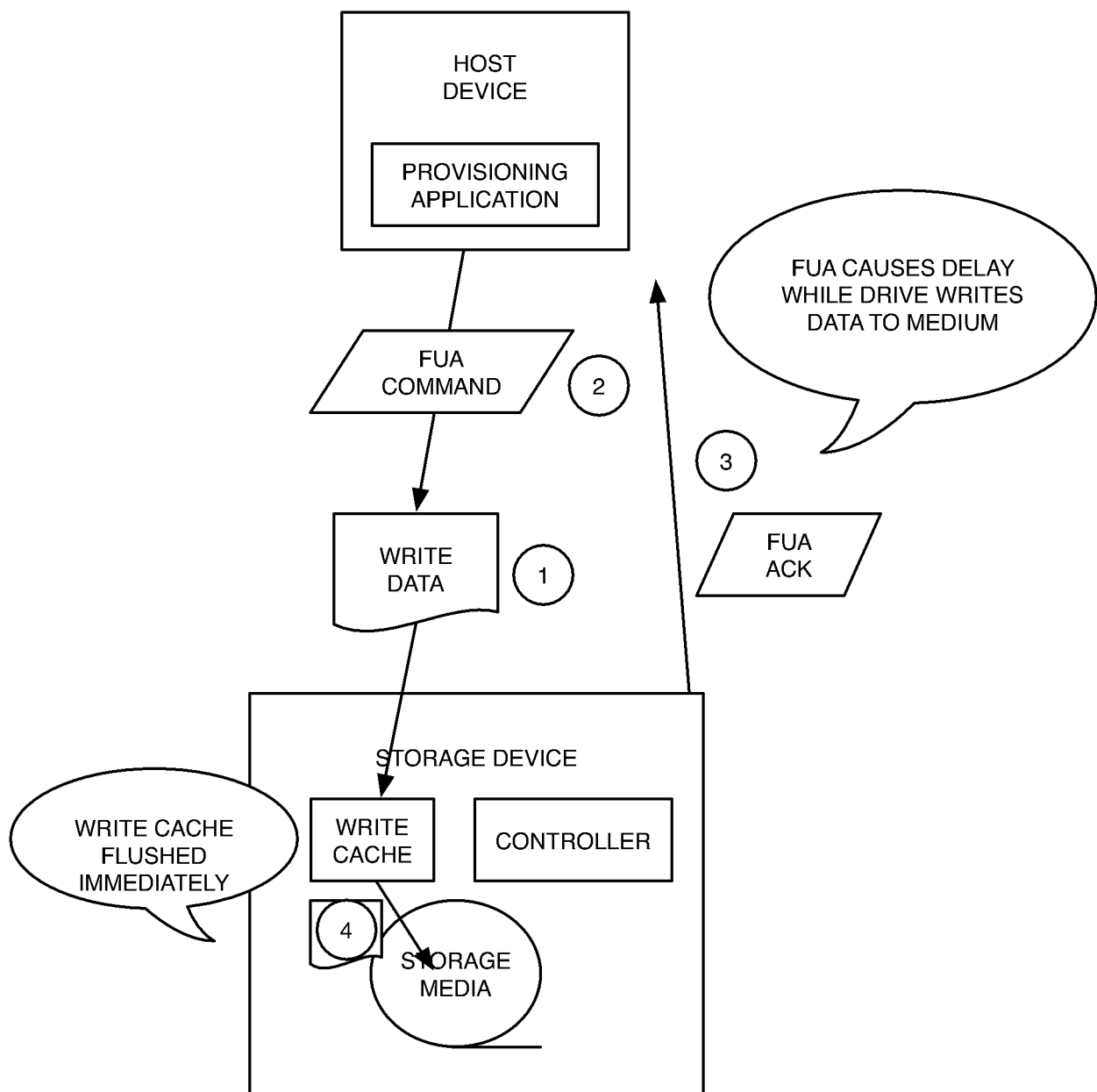


FIG. 2 (PRIOR ART)

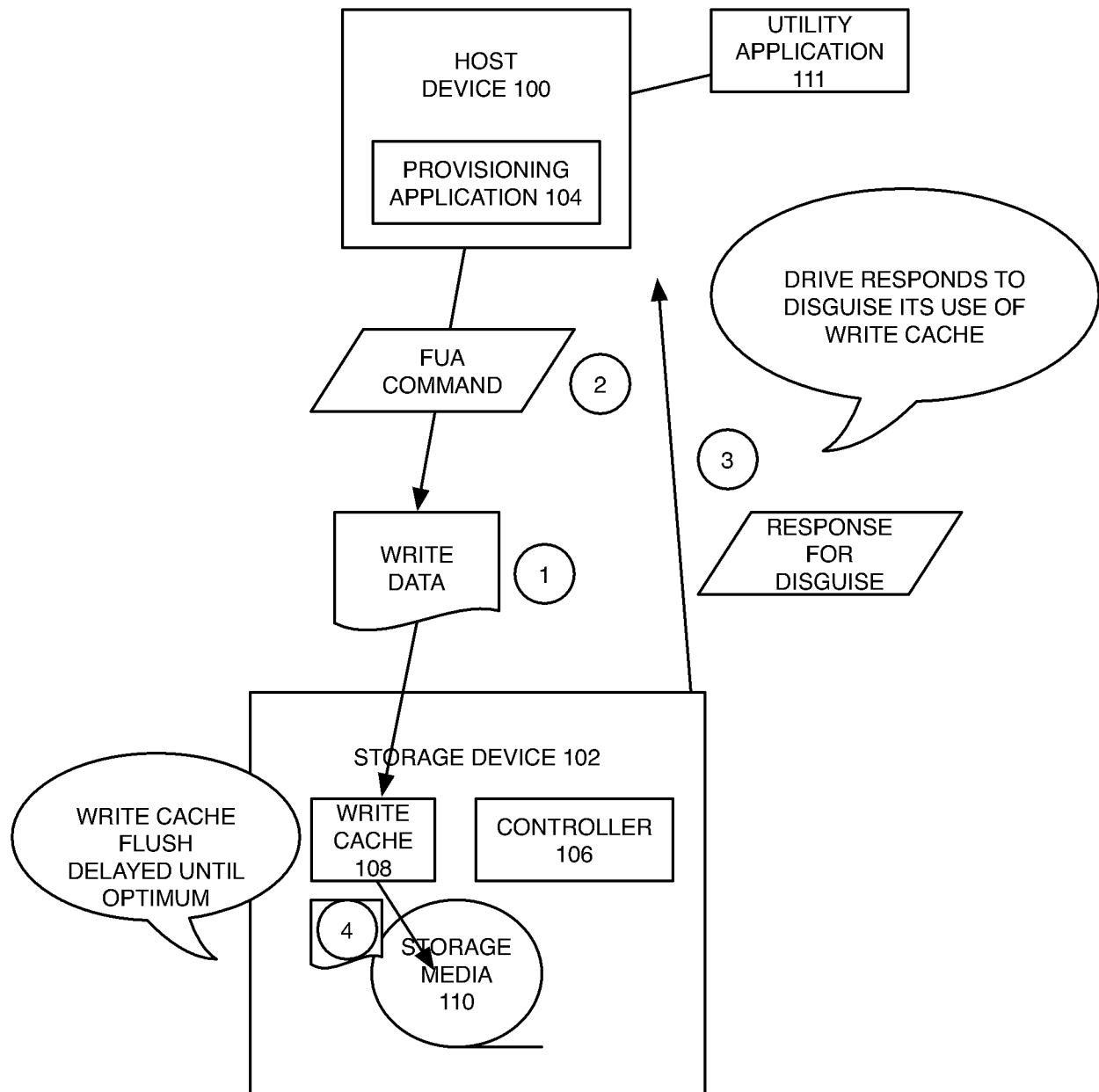


FIG. 3

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2013/062756**A. CLASSIFICATION OF SUBJECT MATTER****G06F 13/10(2006.01)i, G06F 13/14(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 13/10; G06F 9/445; G06F 12/00; G06F 009/445; G06F 15/177; G06F 13/14

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: bootable image, portable storage device, external device, write, write cache, write buffer, command, forced unit access, flush, disguise, Windows To Go

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 2012-0303942 A1 (ERIC PEACOCK) 29 November 2012 See paragraphs [0001], [0005], [0009]-[0011], [0016], [0022]-[0026], [0031]-[0035], [0046]-[0052]; and claims 1, 10.	1-2, 4, 11-12, 14
A		3, 5-10, 13, 15-20
Y	US 2003-0182500 A1 (DAVID M. RAVES et al.) 25 September 2003 See paragraphs [0001]-[0010], [0025]-[0027], [0030]-[0036]; and claims 1, 3.	1-2, 4, 11-12, 14
A	US 2006-0259754 A1 (WESLEY G. MILLER et al.) 16 November 2006 See paragraphs [0002]-[0006], [0013]-[0017], [0034]-[0045], [0050]; and claims 1, 4.	1-20
A	US 2007-0283343 A1 (YARIV ARIDOR et al.) 06 December 2007 See paragraphs [0003]-[0023], [0033], [0040]-[0045], [0049]-[0055], [0058], [0061]-[0065]; and claims 1-4.	1-20
A	US 2005-0149924 A1 (ESHWARI P. KOMARLA et al.) 07 July 2005 See paragraphs [0001]-[0008], [0025]-[0026], [0031]-[0032], [0038]-[0043], [0047]-[0049], [0072]-[0079]; and claim 1.	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

28 January 2014 (28.01.2014)

Date of mailing of the international search report

28 January 2014 (28.01.2014)

Name and mailing address of the ISA/KR

Korean Intellectual Property Office
189 Cheongsa-ro, Seo-gu, Daejeon Metropolitan City,
302-701, Republic of Korea

Facsimile No. +82-42-472-7140

Authorized officer

BYUN, Sung Cheal

Telephone No. +82-42-481-8262



INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2013/062756

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2012-0303942 A1	29/11/2012	None	
US 2003-0182500 A1	25/09/2003	US 6904496 B2	07/06/2005
US 2006-0259754 A1	16/11/2006	CN 100378648 C	02/04/2008
		CN 1839413 A	27/09/2006
		EP 1654708 A1	10/05/2006
		JP 2007-521528 A	02/08/2007
		KR 10-2007-0048638 A	09/05/2007
		US 2003-0231188 A1	18/12/2003
		US 2003-0233363 A1	18/12/2003
		US 2004-0006689 A1	08/01/2004
		US 2004-0015536 A1	22/01/2004
		US 2004-0034849 A1	19/02/2004
		US 2005-0240627 A1	27/10/2005
		US 2006-0130017 A1	15/06/2006
		US 6947954 B2	20/09/2005
		US 7017144 B2	21/03/2006
		US 7120786 B2	10/10/2006
		US 7461095 B2	02/12/2008
		US 7464176 B2	09/12/2008
		US 7664944 B2	16/02/2010
		WO 2005-020156 A1	03/03/2005
		WO 2007-020156 A8	15/02/2007
US 2007-0283343 A1	06/12/2007	US 8302091 B2	30/10/2012
US 2005-0149924 A1	07/07/2005	US 7207039 B2	17/04/2007