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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
8 August 2013 (08.08.2013)

WIPO | PCT

(10) International Publication Number
WO 2013/115806 A1

(51) International Patent Classification:
G06F 13/14 (2006.01) *G06F 9/44* (2006.01)

(21) International Application Number:
PCT/US2012/023396

(22) International Filing Date:
31 January 2012 (31.01.2012)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant *(for all designated States except US)*: **HEWLETT-PACKARD DEVELOPMENT COMPANY, L.P.** [US/US]; 11445 Compaq Center Drive W, Houston, Texas 77070 (US).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **BOPARDIKAR, Raju, C.** [IN/US]; 2580 55th St, Boulder, Colorado 80301 (US). **VOIGT, Douglas, L.** [US/US]; 11311 Chinden Blvd, Boise, Idaho 83714-0021 (US). **BARRON, Dwight, L.** [US/US]; 11445 Compaq Center Dr W, Houston, Texas 77070 (US). **PEREZ, Paul, L.** [US/US]; 11445 Compaq Center Dr W, Houston, Texas 77070 (US).

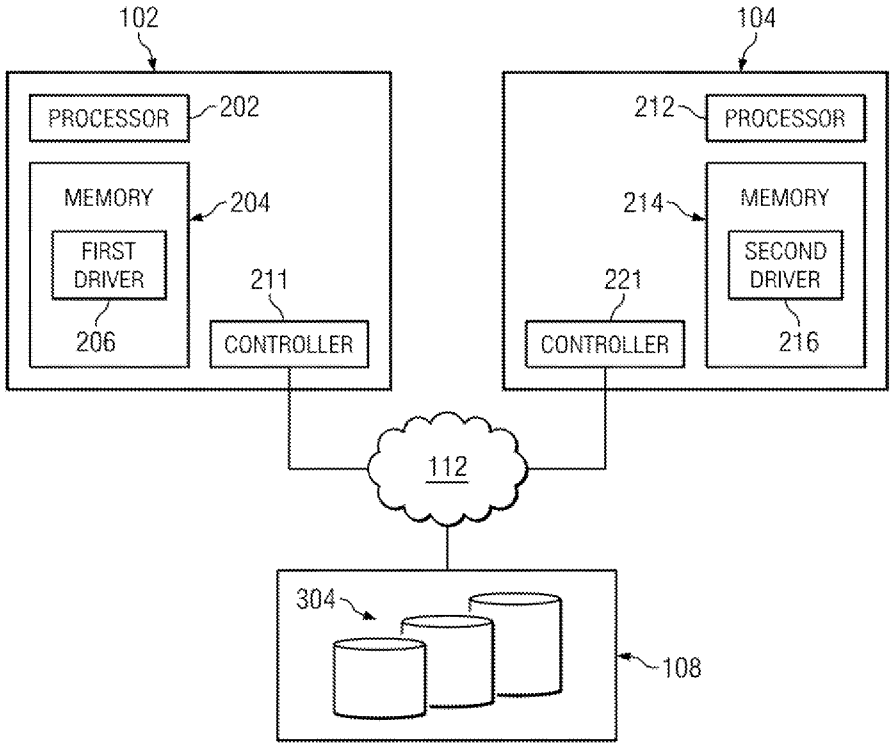
(74) Agents: **SEGARRA, Roosevelt** et al.; Hewlett-Packard Company, Intellectual Property Administration, 3404 E. Harmony Road, Mail Stop 35, Fort Collins, CO 80528 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, 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, KM, 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, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, 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, MD, 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,

[Continued on next page]

(54) Title: DRIVERS AND CONTROLLERS



(57) Abstract: Disclosed herein is a technique to transfer at least one unfinished operation from one controller to a second controller, if the first controller has ceased.

WO 2013/115806 A1

SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

— *as to the identity of the inventor (Rule 4.17(i))*

Published:

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

DRIVERS AND CONTROLLERS

BACKGROUND

[0001] Disk controllers are circuits that enable a processor to communicate with a data storage resource. Many data storage arrangements today divide and replicate data among multiple physical drives. Multiple physical drives arranged in such a way may be called a redundant array of independent disks ("RAID"). Disk array products may be equipped with a plurality of controllers to provide failover management. Such disk array controllers manage the physical disk drives and present them to a computer as logical units such that applications executing therein may perceive these disk arrays as a single drive. Failure of one controller may trigger a second controller to substitute for the first. This allows such failure to be transparent to an application. The controllers included in a disk array may be designed to be in communication with each other such that a given controller in the disk array is always aware of the state of other controllers therein.

BRIEF DESCRIPTION OF THE DRAWINGS

[0002] FIG. 1 is an example configuration of computer apparatus in accordance with aspects of the disclosure.

[0003] FIG. 2 is an example of processes in communication with a storage device in accordance with aspects of the disclosure.

[0004] FIG. 3 illustrates a flow diagram in accordance with aspects of the disclosure.

[0005] FIG. 4 is a working example of controllers executing in accordance with aspects of the disclosure.

[0006] FIG. 5 is a working example of controller failover in accordance with aspects of the disclosure.

DETAILED DESCRIPTION

[0007] As noted above, many disk arrays are equipped with a plurality of controllers designed to manage failover scenarios therein. However, many disk controllers, such as host bus adapter ("HBA") based controllers, are simple controllers that may be arranged within a computer as a peripheral component

interconnect ("PCI") expansion card or may be built into a motherboard. Such controllers may be designed to execute independently of other controllers such that there is no failover capability therein. All operations initiated after the failure of such a controller may never be executed, which may result in permanent loss of data.

[0008] In view of the foregoing, aspects of the present disclosure provide a system and method that determine whether a first controller has ceased execution such that the first controller has stopped implementing operations in a storage resource. In another aspect, if it is determined that the first controller has ceased, a configuration file associated with the first controller may be accessed. The configuration file may enable a second controller to substitute for the first controller. In a further aspect, at least one unfinished operation may be implemented in the storage resource. The aspects, features and advantages of the present disclosure will be appreciated when considered with reference to the following description of examples and accompanying figures. The following description does not limit the application; rather, the scope of the disclosure is defined by the appended claims and equivalents.

[0009] FIG. 1 presents an example of computer apparatus 102 and 104 depicting various components in accordance with aspects of the disclosure. Computers 102 and 104 may comprise any device capable of processing instructions and transmitting data to and from other computers, including a laptop, a full-sized personal computer, a high-end server, or a network computer lacking local storage capability. Computer apparatus 102 and 104 may include all the components normally used in connection with a computer. For example, they may have a keyboard, mouse, and/or various other types of input devices such as pen-inputs, joysticks, buttons, touch screens, etc., as well as a display, which could include, for instance, a CRT, LCD, plasma screen monitor, TV, projector, etc. In another example, they may have a graphics processing unit ("GPU"), redundant power supply, fans, and various input/output cards, such as Peripheral Component Interconnect ("PCI") cards.

[0010] Computer apparatus 102 and 104 may include processors 202 and 212 and memories 204 and 214 respectively. Memories 204 and 214 may store

first driver 206 and second driver 216 respectively. First driver 206 and second driver 216 may be retrieved and executed by their respective processors 202 and 212. The processors may be any number of well known processors, such as processors from Intel® Corporation. Alternatively, the processors may be dedicated controllers for executing operations, such as an application specific integrated circuit ("ASIC"). In addition to processors 202 and 212, a remote maintenance processor may be used to monitor components of computer apparatus 102 and 104 for suspect conditions.

[0011] Memories 204 and 214 may be volatile random access memory ("RAM") devices. The memories may be divided into multiple memory segments organized as dual in-line memory modules ("DIMMs"). Alternatively, memories 204 and 214 may comprise other types of devices, such as memory provided on floppy disk drives, tapes, and hard disk drives, or other storage devices that may be coupled to their respective computers directly or indirectly. Memories 204 and 214 may also include non-volatile random access memory ("NVRAM") devices, which may be any type of NVRAM, such as phase change memory ("PCM"), spin-torque transfer RAM ("STT-RAM"), or programmable permanent memory (e.g., flash memory). The memory may also include any combination of one or more of the foregoing and/or other devices as well. Although all the components of computer apparatus 102 and 104 are functionally illustrated as being within the same block, it will be understood that the components may or may not be stored within the same physical housing. Furthermore, each computer may actually comprise multiple processors and memories working in tandem.

[0012] Computer apparatus 102 and 104 of FIG. 1 may be arranged in a networked configuration. For example, each computer may be a node in a cluster of computers and may be capable of directly or indirectly communicating with each other or with other computers or devices in a cluster. While the following examples and illustrations concentrate on communications between computer apparatus 102 and 104, it should be appreciated that a cluster may include additional interconnected computers and that computers 102 and 104 are featured merely for ease of illustration. The computers disclosed in FIG. 1 may be interconnected via a network 112, which may be a local area network ("LAN"), wide area network

("WAN"), the Internet, etc. Network 112 and intervening nodes therein may also use various protocols including virtual private networks, local Ethernet networks, private networks using communication protocols proprietary to one or more companies, cellular and wireless networks, HTTP, and various combinations of the foregoing. In addition, the intervening nodes of network 112 may utilize remote direct memory access ("RDMA") to exchange information with the memory of another computer in the cluster. It should be appreciated that computer apparatus 102 and 104 may be acknowledged as an individual node in a network containing a larger number of computers. The cluster may be arranged as a load balancing network such that computers 102 and 104 exchange information with each other for the purpose of receiving, processing, and replicating data.

[0013] FIG. 1 also shows disk array controllers 211 and 221. Disk array controllers 211 and 221 may be simple HBA based controllers coupled to their respective computers via a host-side interface, such as PCI , Serial ATA (SATA) or serial attached small computer system interface ("SAS"), which allows processors 202 and 212 to transmit one or more input/output requests to disk array 304. Disk controllers 211 and 221 may communicate with disk array 304 via a drive-side interface (e.g., FC, storage area network ("SAS"), etc.). Disk array 304 may be housed in, for example, computer apparatus 108. While FIG. 1 depicts disk array controllers 211 and 221 in communication with disk array 304, it is understood that disk array controllers 211 and 221 may enable a processor to communicate with other storage resources and that FIG. 1 is merely illustrative.

[0014] First driver 206 and second driver 216 may comprise any set of machine readable instructions to be executed directly (such as machine code) or indirectly (such as scripts) by the processor(s). The instructions of the drivers may be stored in any computer language or format, such as in object code or modules of source code. The instructions may be stored in object code format for direct processing by a processor, or in any other computer language including scripts or collections of independent source code modules that are interpreted on demand or compiled in advance. However, it will be appreciated that first drivers 206 and second driver 216 may be realized in the form of software, hardware, or a combination of hardware and software.

[0015] In one example, first driver 206 or second driver 216 may be realized in any non-transitory computer-readable media for use by or in connection with an instruction execution system such as computer apparatus 102 and 104, an ASIC, or other system that can fetch or obtain the logic from non-transitory computer-readable media and execute the instructions contained therein. "Non-transitory computer-readable media" can be any media that can contain, store, or maintain programs and data for use by or in connection with the instruction execution system. Non-transitory computer readable media may comprise any one of many physical media such as, for example, electronic, magnetic, optical, electromagnetic, or semiconductor media. More specific examples of suitable non-transitory computer-readable media include, but are not limited to, a portable magnetic computer diskette such as floppy diskettes or hard drives, a read-only memory ("ROM"), an erasable programmable read-only memory, or a portable compact disc.

[0016] First driver 206 may interface processor 202 with controller 211. In turn, controller 211 may interface first driver 206 with disk array 304. Accordingly, first driver 206 may forward data operations, originating from processor 202, to disk array 304 via controller 211. As with first driver 206, second driver 216 may interface processor 212 with controller 221. In turn, controller 221 may interface second driver 216 with disk array 304. The operations forwarded by first driver 206 may be unrelated to the data operations forwarded by second driver 216. First driver 206 may also replicate an operation associated with data, such as an input/output operation, to second driver 216 or vice-versa. While first driver 206 is shown executing in a first domain and second driver 216 is shown executing in a second domain different from the first domain, it is understood that other examples may execute both drivers in the same domain.

[0017] FIG. 2 is a high level block diagram depicting an illustrative arrangement of drivers 206 and 216. Applications 402 and 404, which may be local applications or an application from a remote computer, may transmit a request for a data operation, such as an input/output operation, to first driver 206 and second driver 216 respectively. First driver 206 and second driver 216 may abstract the underlying storage resources that are utilized for data operations.

Upon receipt of a request for a data operation, such as a write operation, first driver 206 may forward the operation to controller 211. Controller 211 may then forward the same to storage resource 406, which may be disk array 304. Second driver 216 may forward data operations from application 404 to controller 221. As with controller 211, controller 221 may forward the data to the same storage resource 406. Both drivers may also replicate operations to each other at all times so as to maintain redundant copies of data at different locations within storage resource 406. Application 404 may be unrelated to application 402.

[0018] One working example of a system and method for managing controller failovers in accordance with aspects of the present disclosure is shown in FIGS. 3-5. In particular, FIG. 3 illustrates a flow diagram of a process for managing controller failovers. FIGS. 4-5 illustrate various aspects of failover management. The actions shown in FIGS. 4-5 will be discussed below with regard to the flow diagram of FIG. 3.

[0019] In block 302, it may be determined whether a first controller has ceased execution such that the first controller has stopped implementing operations in a storage resource. Referring to the example of FIG. 4, controller 211 and first driver 206 are shown functioning normally. In this example, the storage resource is disk array 304 and controllers 211 and 221 are non-redundant disk array controllers. Data operations may be replicated from first driver 206 to second driver 216 during normal operation. First driver 206 may instruct controller 211 to implement operations within selected volumes of disk array 304, while second driver 216 may instruct controller 221 to redundantly implement the same within volumes of disk array 304 that are different than those used by controller 211. Second driver 216 may determine that first controller 211 has failed, when first driver 206 no longer responds to second driver 216. Alternatively, second driver 216 may determine that controller 211 has failed, when there is a lack of communication with controller 211, such as lack of a heartbeat. Referring now to the example of FIG. 5, controller 211 is shown in a state of failure. The data operations forwarded to first driver 206 by application 402 are shown being replicated to second driver 216 and awaiting execution by second driver 216 via

second controller 221. The data operations shown in FIG. 5 are operations 502, 504, and 506.

[0020] Referring back to FIG. 3, if the first controller has ceased, a configuration file associated with first controller 211 may be accessed, as shown in block 303. The configuration file may enable controller 221 to substitute for first controller 211. In one example, the configuration file may be stored at a location in disk array 304 such that second driver 216 may upload the configuration file therefrom. The configuration file may contain information that enables controller 221 to implement the unfinished operations in the same location within disk array 304 as first controller 211, such as, for example, the same RAID volumes. The configuration file may include user volumes visible to controller 211, volume worldwide names, or other SCSI configuration data, such as RAID level configuration. Referring back to FIG. 3, at least one unfinished operation may be implemented, as shown in block 305. Referring back to figure 5, second driver 216 may obtain the at least one unfinished operation from first driver 206. Second driver 216 may initiate execution of the unfinished operations by forwarding them to controller 221. Having access to the configuration file associated with controller 211 may allow controller 221 to substitute for controller 211. This transfer of control may be transparent to application 402. First driver 206 may continue replicating data operations to second driver 216 while controller 211 is inactive. As noted above, first driver 206 may replicate data to second driver 216 or vice versa at all times notwithstanding the state of controller 211 or controller 221. However, when controller 211 fails, second driver 216 may instruct controller 221 to further implement the replicated operation in the same location in storage used by controller 211. Such location may be specified in the configuration file associated with controller 211. Referring back to FIG. 3, if the first controller has not ceased, the second driver 216 may continue executing as normal, as shown in block 306.

[0021] Advantageously, the above-described system and method provides failover capabilities to controllers that may be designed to execute independently, such as inexpensive HBA based controllers. In that regard, users of such controllers may be rest assured that their data will be maintained notwithstanding

the failure thereof. In turn, users may have transparent failover management without purchasing expensive enterprise controllers.

[0022] Although the disclosure herein has been described with reference to particular examples, it is to be understood that these examples are merely illustrative of the principles of the disclosure. It is therefore to be understood that numerous modifications may be made to the examples and that other arrangements may be devised without departing from the spirit and scope of the disclosure as defined by the appended claims. Furthermore, while particular processes are shown in a specific order in the appended drawings, such processes are not limited to any particular order unless such order is expressly set forth herein. Rather, various steps can be handled in a different order or simultaneously, and steps may be omitted or added.

CLAIMS

1. A system comprising:

a first driver to interface a first processor with a first controller, the first controller interfacing the first driver with a storage resource;

a second driver to interface a second processor with a second controller, the second controller interfacing the second driver with said storage resource, the first controller and the second controller being designed to execute independently of each other, the second driver having instructions therein which, if executed, causes the second processor to:

determine whether the first controller has ceased execution such that the first controller has stopped implementing operations in the storage resource;

if the first controller has ceased, access a configuration file associated with the first controller, the configuration file enabling the second controller to substitute for the first controller; and

implement at least one unfinished operation in the storage resource.

2. The system of claim 1, wherein the first driver executes in a first domain and the second driver executes in a second domain different than the first domain.

3. The system of claim 1, wherein the storage resource is a redundant array of independent disks.

4. The system of claim 1, wherein the second driver has instructions therein which, if executed, further causes the second processor to retrieve the configuration file associated with the first controller from the storage resource.

5. The system of claim 1, wherein the second driver, if executed, further causes the second processor to initiate the at least one unfinished operation via the second controller.

6. The system of claim 1, wherein the at least one unfinished operation originates from the first processor.

7. The system of claim 6, wherein the second driver, if executed, further causes the second processor to obtain the at least one unfinished operation from the first processor via the first driver.

8. A non-transitory computer readable medium having instructions stored therein which, if executed, causes a processor to:

determine whether a first controller has ceased execution such that the first controller has stopped implementing operations in a storage resource;

if the first controller has ceased, access a configuration file associated with the first controller, the configuration file enabling a second controller to substitute for the first controller; and

implement at least one unfinished operation in the storage resource.

9. The non-transitory computer readable medium of claim 8, wherein the storage resource is a redundant array of independent disks.

10. The non-transitory computer readable medium of claim 9, wherein the, instructions, if executed, further causes the processor to retrieve the configuration file associated with the first controller from the storage resource.

11. The non-transitory computer readable medium of claim 10, wherein the instructions, if executed, causes the processor to initiate the at least one unfinished operation via the second controller.

12. The non-transitory computer readable medium of claim 8, wherein the at least one unfinished operation originates from a different processor.

13. The non-transitory computer readable medium of claim 12, wherein the instructions, if executed, further causes the processor to obtain the at least one unfinished operation from the different processor.

14. A method comprising:

determining whether a first controller has ceased execution such that the first controller has stopped implementing operations in a storage resource;

if the first controller has ceased, accessing a configuration file associated with the first controller, the configuration file enabling a second controller to substitute for the first controller, the first controller and the second controller being host bus adapter controllers; and

implementing at least one unfinished operation in the storage resource.

15. The method of claim 14, further comprising retrieving the configuration file associated with the first controller from the storage resource.

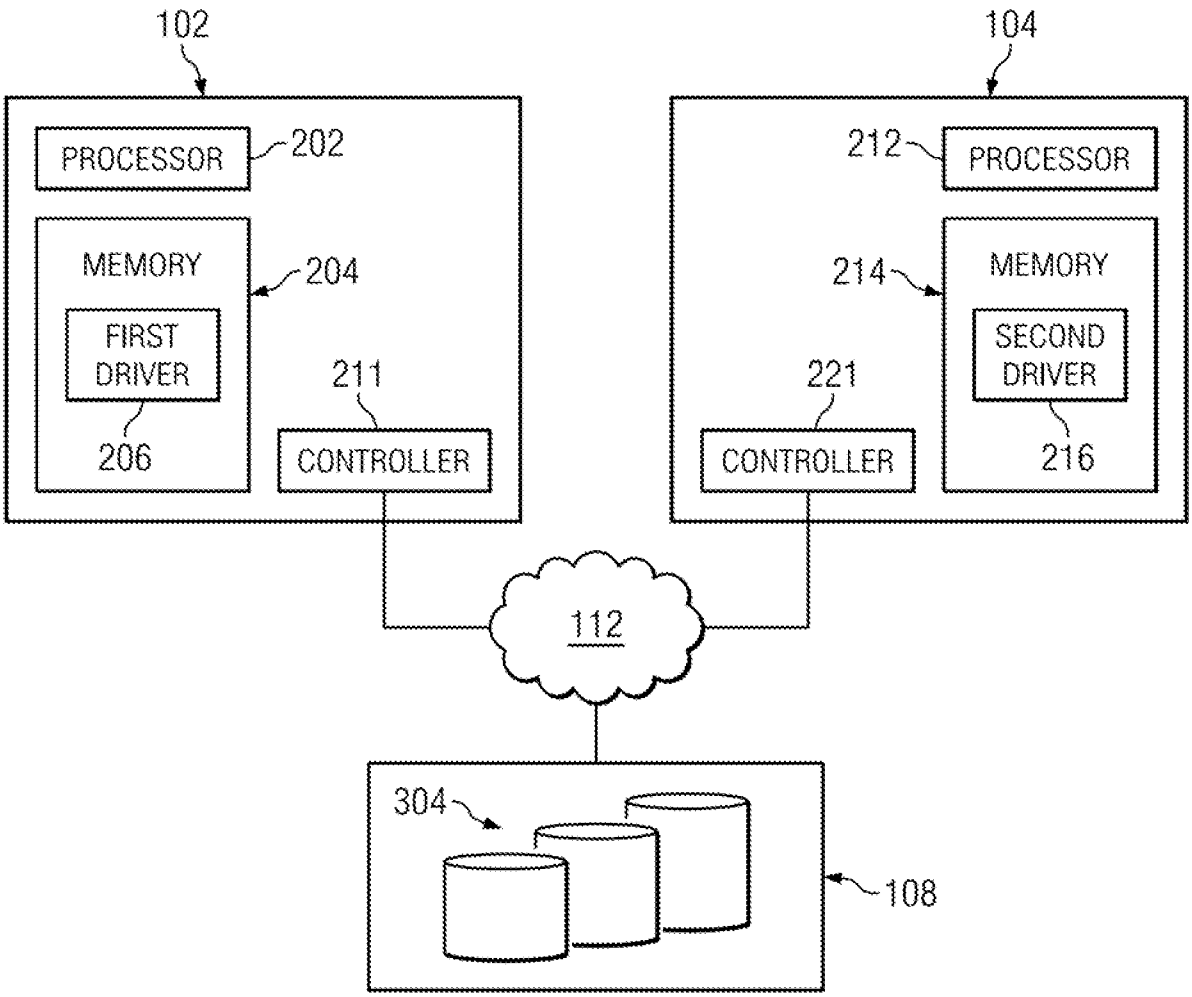

102
104
PROCESSOR
202
212
PROCESSOR
MEMORY
204
214
MEMORY
FIRST DRIVER
206
211
CONTROLLER
221
CONTROLLER
SECOND DRIVER
216
112
304
108

FIG. 1

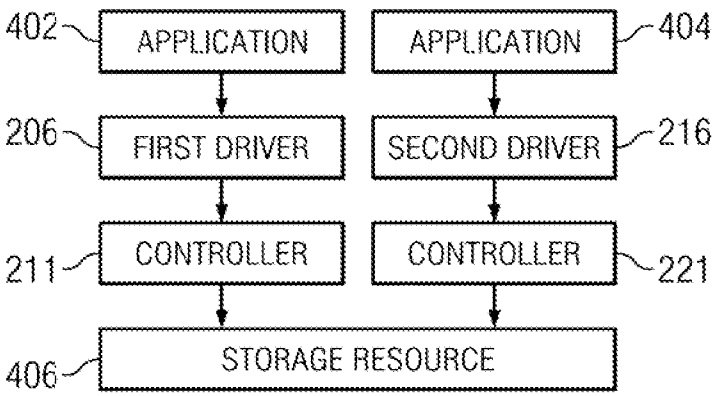

402
APPLICATION
APPLICATION
404
206
FIRST DRIVER
SECOND DRIVER
216
211
CONTROLLER
CONTROLLER
221
406
STORAGE RESOURCE

FIG. 2

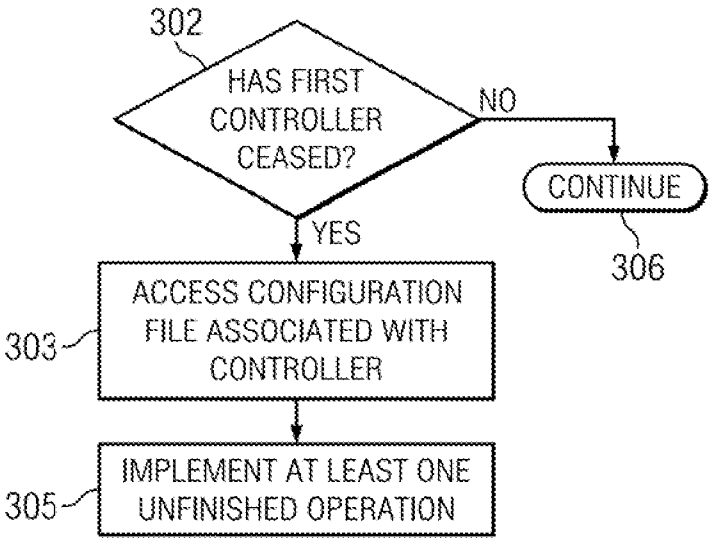

302
HAS FIRST CONTROLLER CEASED?
NO
CONTINUE
306
YES
ACCESS CONFIGURATION FILE ASSOCIATED WITH CONTROLLER
303
IMPLEMENT AT LEAST ONE UNFINISHED OPERATION
305

FIG. 3

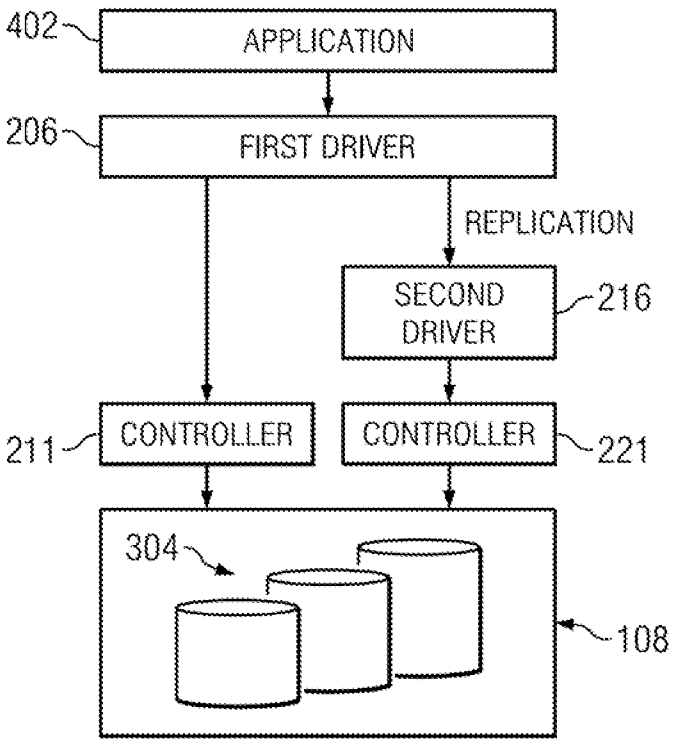

402
APPLICATION
206
FIRST DRIVER
REPLICATION
SECOND DRIVER
216
211
CONTROLLER
CONTROLLER
221
304
108

FIG. 4

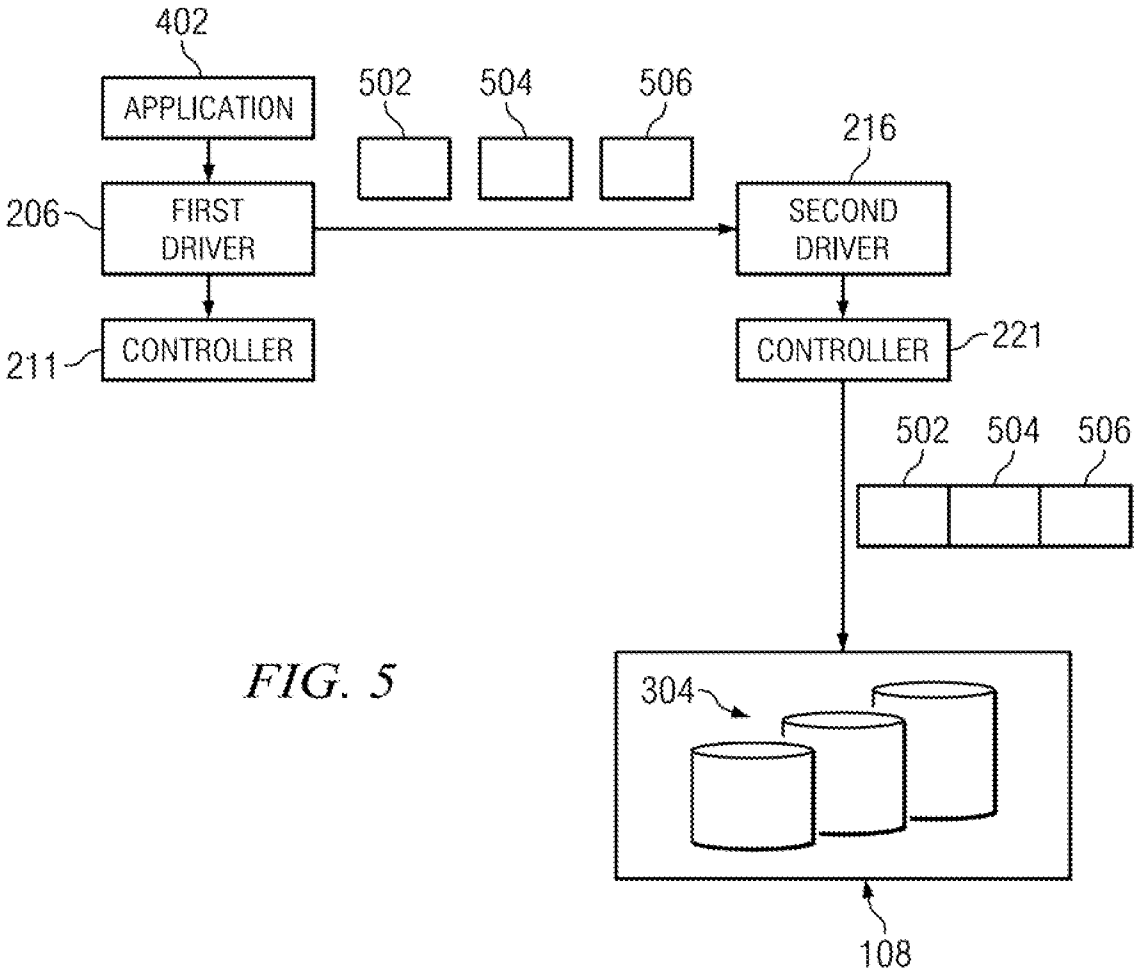

402
APPLICATION
502
504
506
216
206
FIRST DRIVER
SECOND DRIVER
211
CONTROLLER
CONTROLLER
221
502
504
506
304
108

FIG. 5

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2012/023396

A. CLASSIFICATION OF SUBJECT MATTER

G06F 13/14(2006.01)i, G06F 9/44(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)

IPC : G06F

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: RAID, driver, controller, processor

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2006-0075283 A1 (COPAN Systems, Inc.) 06 April 2006 See paragraphs [0030],[0034]; claim 14 and figures 1 and 2.	1-15
A	US 2004-0268037 A1 (BUCHANAN WILLIAM W. et al.) 30 December 2004 See abstract.	1-15
A	US 2006-0218436 A1 (DELL PRODUCTS L.P.) 28 September 2006 See abstract.	1-15
A	US 2006-0136654 A1 (Broadcom Corporation.) 22 June 2006 See abstract.	1-15

☐ Further documents are listed in the continuation of Box C.

☒ See patent family annex.

* Special categories of cited documents:

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

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

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

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

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

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

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

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

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
08 OCTOBER 2012 (08.10.2012)	**10 OCTOBER 2012 (10.10.2012)**
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 Ko Jae Yong Telephone No. 82-42-481-8212

Form PCT/ISA/210 (second sheet) (July 2009)

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2012/023396

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2006-0075283 A1	06.04.2006	US 7434090 B2	07.10.2008
US 2004-0268037 A1	30.12.2004	US 7213102 B2	01.05.2007
US 2006-0218436 A1	28.09.2006	None	
US 2006-0136654 A1	22.06.2006	US 7730257 B2	01.06.2010