



- (51) **International Patent Classification:**
G06F 13/38 (2006.01)
- (21) **International Application Number:**
PCT/US20 13/067787
- (22) **International Filing Date:**
31 October 2013 (31.10.2013)
- (25) **Filing Language:** English
- (26) **Publication Language:** English
- (71) **Applicant:** HEWLETT-PACKARD DEVELOPMENT COMPANY, L.P. [US/US]; 11445 Compaq Center Drive W., Houston, Texas 77070 (US).
- (72) **Inventors:** TAMMA, Roopesh Kumar; 4245 Technology Drive, Fremont, California 94538 (US). NAZARI, Siamak; 4245 Technology Drive, Fremont, California 94538 (US). NATARAJAN, Ajitkumar A.; 4245 Technology Drive, Fremont, California 94538 (US).
- (74) **Agents:** ORTEGA, Arthur et al; Hewlett-Packard Development Company, L.P., Intellectual Property Administration, Mail Stop 35 3404 E. Harmony Road, Fort Collins, Colorado 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, 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, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

- as to the identity of the inventor (Rule 4.1 7(I))
- as to applicant's entitlement to apply for and be granted a patent (Rule 4.1 7(H))

Published:

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



W0 2015/065436 A1

(54) **Title:** TARGET PORT PROCESSING OF A DATA TRANSFER

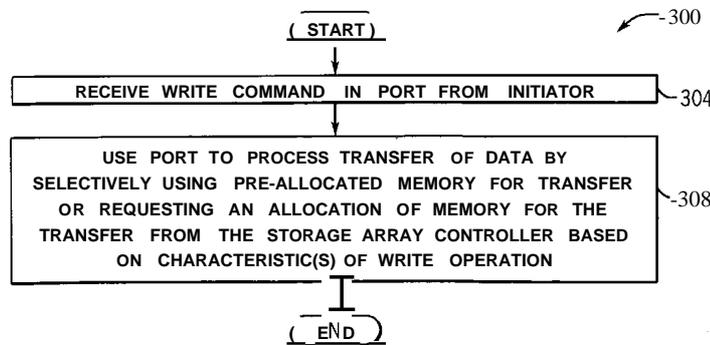


FIG. 3

(57) **Abstract:** A technique includes receiving a command in a target port, where the command is provided by an initiator and is associated with a write operation. The technique includes, in response to the command, using the target to process a data transfer for the initiator associated with the write operation. The processing includes, based on a characteristic of the command, selectively using memory for the transfer pre-allocated by a storage array controller prior to receipt of the command by the target port or requesting an allocation of memory for the transfer from the storage array controller.

TARGET PORT PROCESSING OF A DATA TRANSFER

BACKGROUND

A computer may access a storage area network (SAN) for purposes of storing and retrieving large amounts of data. The typical SAN includes a consolidated pool of mass storage devices (magnetic tape drives, hard drives, optical drives, and so forth), and the SAN typically provides relatively high speed block level storage, which may be advantageous for backup applications, archival applications, database applications and other such purposes.

BRIEF DESCRIPTION OF THE DRAWINGS

[0001] Fig. 1 is a schematic diagram of a computer system according to an example implementation.

[0002] Fig. 2 is a schematic diagram of a physical machine of Fig. 1, which contains a host adaptor and a storage array controller according to an example implementation.

[0003] Figs. 3 and 4 are flow diagrams depicting target port processing of a command according to example implementations.

DETAILED DESCRIPTION

[0005] Referring to Fig. 1, in accordance with example implementations, a computer system 100 includes a storage area network (SAN) 110, which contains a consolidated pool of physical media storage devices 130 (magnetic tape drives, optical drives, hard disk drives, combinations of these devices, and so forth), which may be used by clients 102 (desktop computers, portable computers, tablet computers, smartphones, and so forth) for data storage and retrieval purposes.

[0006] As an example, the clients 102 may communicate with various servers 170 (databases, email servers, file servers, and so forth) of the computer system 100, and as a result of these communications, the servers 170 may generate block-level access requests to store and retrieve data to and from the SAN 110. For the example of Fig. 1, the clients 102 may communicate with the servers 170 using network fabric 106, such as local area network (LAN)-based fabric, wide area network (WAN)-based fabric, Internet-based fabric, and so forth.

[0007] In general, the block-level reads and writes that are generated by the servers 170 are processed by storage array controllers 134 of storage arrays 120 (N storage arrays 120-1 . . . 120-N, being depicted in Fig. 1) of the SAN 110 for purposes of storing the data in and retrieving data from physical storage devices 130 of the arrays 120.

[0008] As depicted in Fig. 1, the servers 170 are coupled to the storage arrays 120 by SAN fabric 180, which may, as examples, contain fibre channel (FC) fabric, Internet Protocol (IP)-based fabric, switches, gateways, FC-to-SCSI bridges, and so forth, as can be appreciated by the skilled artisan.

[0009] In accordance with example implementations that are disclosed herein, the servers 170 communicate with the storage arrays 120 using a messaging protocol that complies with a Fibre Channel Protocol (FCP) or an internet Small Computer Systems Interface (iSCSI) protocol; and more specifically, a

server 170 may initiate a given read or write operation by providing a message that targets a specific host adaptor port.

[0010] In general, in accordance with example implementations, a given storage array 120 contains one or multiple host adapters 136. The host adapter 136 provides a front end interface that is constructed to communicate with the servers 170 and present the storage on the drives 130 of the storage array 120 as logical units. The storage array 120 further includes a storage array controller 134 that, among its other functions, performs logical-to-physical storage unit conversions and provides a back-end interface to communicate with the associated drives 130 of the storage array 120.

[0011] In the context of the following discussion, for a given write or read that transfers data between a server 170 and a storage array 120, an "initiator" (a host port of a server 170, for example) initiates the write or read operation with a given "target" port (a port of the host adaptor 136, for example) by providing a message that identifies the target port, contains a write/read command and specifies one or multiple characteristics of the associated write/read operation. The message may be an information unit (IU) (for FCP) or a protocol data unit (PDU) for (iSCSI protocol).

[0012] A given target and initiator may undergo a login process that sets up how data transfers occur between the two entities. For example, as a result of the login process, a given target port may be designated as being capable to receive unsolicited data (or "immediate" data) that accompanies a write command in a message from the initiator.

[0013] Alternatively, the login process may result in designating a target port to not receive unsolicited data with a write command. For this latter configuration, the target port controls the timing of when the initiator provides the data associated with the write in that the initiator does not provide the write data until the target port responds with a message indicating the target port's readiness to receive the write data. This may otherwise be referred to

as the target port providing an XFR_RDY (for FCP) or R2T (for iSCSI protocol) signal.

[0014] Turning now to a more specific example, an initiator may generate a message that contains a write command that targets a specific port of a host adapter 136. For this example, and other examples described herein, the target port is configured to not receive unsolicited data with a write command. Instead, for a write operation, the initiator waits for the target port to provide an indication of readiness to receive the write data before the initiator provides the data to the target port.

[0015] When a target port receives data as part of a write operation, the target port transfers the data into a region of main memory of the storage array controller 134, which the storage array controller 134 allocates for this purpose. At any one time, the main memory of the storage array controller 134 may be allocated to receive data from multiple ongoing write operations.

[0016] One way to handle the processing of a write command that is received at a target port is for the storage array controller to be directly involved in the data transfer phase of the write operation. In this approach, the storage array controller controls when the target port asserts (to the initiator) its readiness to receive the write data. Moreover, the storage array controller allocates regions of its main memory for receiving write data among the target ports as write commands are received and as memory becomes available.

[0017] More specifically, in the direct approach, in response to receiving a write command, the target port first notifies (via an interrupt, for example) the storage array controller about the command. The storage array controller then allocates part of its main memory to receive the associated write data and informs the target port about the allocation. After receiving the memory allocation, the target port responds to the initiator with an indication of readiness (i.e., the target port provides a message with the XFR_RDY or R2T signal), and the initiator responds by transferring the data to the target port.

[0018] SCSI write transactions may be handled in a way to reduce the number of interrupts per transaction on the target and improve CPU utilization and latency. One technique to reduce the number of interrupts per transaction on the target (and reduce the latency of SCSI write requests) is a SCSI standard-based "first burst" technique in which the target receives a burst of unsolicited data with the write command. The first burst feature is set up by the initiator and target negotiating the first burst feature in a login process, so that when the target is configured to receive first bursts, the target uses pre-allocated buffers. Therefore, when the initiator sends a write command, the write command is accompanied with write data, and the target uses the pre-allocated buffers to store the data before interrupting the array controller. However, the initiator may not be constructed or configured to implement first bursts.

[0019] In accordance with example techniques that are disclosed herein, buffers on the target host bus adapter are pre-allocated for non-first burst write transactions, which also allows a reduction in the number of interrupts without initiator involvement and does not depend on the ability of the initiator to support first bursts.

[0020] In this manner, systems and techniques are disclosed herein to optimize writes (optimize SCSI writes, for example) by pre-allocating memory for the transfer of the associated write data between an initiator and a target port. In this context, "pre-allocated" memory refers to one or more multiple regions of the storage array controller's memory that are allocated by the controller for exclusive use by a given port for future write operations. The pre-allocation means that the storage array controller 134 is not directly involved in the data phases of certain write operations. In this manner, in accordance with example implementations that are disclosed herein, a given target port is constructed to, for a certain write operation, communicate with an initiator to transfer write data to the target port and store the data in a pre-allocated memory region of the storage array controller 134, all without involvement by the storage array controller 134. Therefore, among possible

advantages, overhead may be offloaded from the storage array controller 134 to the target port, and times associated with the transfer of write data may be decreased.

[0021] More specifically, in accordance with example implementations, the storage array controller 134 programs a given port of the host adaptor 136 with one or multiple parameters that characterize a class of write operations whose data transfers are handled by the port using pre-allocated memory. For use by the port for qualifying write operations, the array controller 134 pre-allocates one or more memory buffers to the given port .

[0022] In accordance with example implementations, a given port has exclusive access to its allocated memory buffer(s) for the designated class of writes until the port releases the allocated memory buffers back to the storage array controller 134. If a given write command does not fall within the designated class, then the storage array controller 134 is directly involved in the data phase: the target port alerts the storage array controller 134 to receipt of the command; and the target port waits for the storage array controller 134 to allocate memory for the transfer of the associated write data before the target port sends a message to the initiator indicating readiness to receive the write data.

[0023] Thus, referring to Fig. 3, in accordance with example implementations, a technique 300 includes receiving (block 304) a write command in a port from an initiator. Pursuant to block 308 of the technique 300, the port processes the transfer of the data associated with the write operation by selectively using pre-allocated memory for the transfer or requesting an allocation of memory for the data transfer from the storage array controller based on one or more characteristics of the write operation (as described by the write command).

[0024] Referring to Fig. 2 in conjunction with Fig. 1, in accordance with example implementations, the host adaptor 136 and the storage array controller 134 may be part of the same physical machine 200. In this context,

the physical machine 200 is an actual machine that is formed from actual hardware (central processing units (CPUs), memory devices, bus interfaces, and the like) and actual machine executable instructions, or "software" (operating system instructions, driver instructions, application instructions, and the like).

[0025] As an example, the storage array controller 134 may be formed from a main system board of the physical machine 200, and the host adaptor 136 may be formed by a host adaptor card that is inserted into a corresponding bus slot on the motherboard. In further implementations, the storage array controller 134 and the host adaptor 136 may be implemented further on the same motherboard. Thus, many variations are contemplated, which are within the scope of the appended claims.

[0026] As depicted in Fig. 2, in general, the host adaptor 136 may contain one or multiple port processors (Q port processors 210-1 . . . 210-Q, being depicted in Fig. 2) that form respective target ports 204 (Q target ports 204-1 . . . 204-Q, being depicted in Fig. 2 as examples). In some example implementations, a given port processor 210 is configured to process read and write operations for an assigned target port 204; and in further example implementations, a given port processor 210 may process read and write operations for multiple assigned target ports 204. As part of this processing, the port processors 210 handle data transfers associated with write commands within a predefined class using pre-allocated memory buffers, as described herein.

[0027] More specifically, in accordance with example implementations, the storage array controller 134 includes one or multiple central processing units (CPUs) 214, which are coupled to a main memory 220 of the storage array controller 134 via a bridge 218.

[0028] In general, the main memory 220 may temporarily store instructions associated with the execution of machine executable instructions, as well as data involved in the preliminary, intermediate and final results associated with this processing. In accordance with some implementations, the main memory

220 may store machine executable instructions that when executed by the CPU(s) 214 cause the CPU(s) 214 to perform all or part of the techniques that are disclosed herein, such as the techniques 300 and 400 (described below).

[0029] In general, the main memory 220 is a non-transitory storage medium that may be formed from semiconductor storage devices, optical storage devices, magnetic media-based storage devices, removable media devices, and so forth, depending on the particular implementation.

[0030] In accordance with example implementations, regions of the main memory 220 are allocated to receive incoming write data. More specifically, in accordance with example implementations, the memory 220 contains buffers 221 that receive incoming write data. The buffers 221 are designated regions of the main memory 220. The buffers 221 may each have the same length, or size; or the buffers 221 may have different sizes, depending on the particular implementation.

[0031] When a port processor 210 receives write data for an associated write operation, the port processor 210 performs a direct memory access (DMA) to the main memory 220 for purposes of storing the write data in the allocated buffer(s) 221. After the data has been transferred, the CPU(s) 214 may perform such functions as logical-to-physical data unit conversions and store the data in one or more of the storage devices 130 via one or multiple input/output (I/O) processors 230. The allocated buffer(s) 221 for a given write command may be pre-allocated before receipt of the write command or may be allocated after receipt of the write command, depending on whether the associated write falls within the qualifying, or designated, class.

[0032] In accordance with example implementations, the CPU(s) 214 identify a qualifying, or designated class of write operations to be handled by a given port processor 210 for a given port 204 and programs the port processor 210 accordingly. In this regard, the CPU(s) 214 may program a given port processor 210 with one or multiple parameters that describe the class of write operations, as well as program the port processor 210 with a pre-allocated

pool of one or multiple buffers 221 to be used in the transfer of the write data to the main memory 220. In accordance with example implementations, the pre-allocated buffer(s) 221 are used exclusively by the assigned port 204.

[0033] The designated class of write operations to be handled by a given port 204 are defined by one or multiple parameters that describe one or multiple characteristics of the write operations to be covered. For example, in accordance with some implementations, a given write operation class includes write commands that are associated with a given block length, or size, of write data. In this manner, the block size may be related to a cache line size of a cache memory 222 of the storage array controller 134. For example, a given cache size may be thirty two kilobytes (kB).

[0034] In accordance with further example implementations, the port processor 210 may take into account considerations other than the block size of the write data. For example, in accordance with some example implementations, given write operation may be associated with a data offset. The class of write operations may be write operations in which the offset does not cause the write data to be stored in more than one cache line.

[0035] For example, the block size of a given write operation may be equal to the cache line size. Therefore, in accordance with an example implementation, an offset of ten may cause two cache lines to be used and therefore, may not be considered part of the write class. As such, the storage array controller 134 handles the initial stages of such a write operation by first allocating the buffer(s) 221 to the port processor 210, and then the port processor 210 signals its readiness to the initiator.

[0036] Referring to Fig. 4, as a more specific example, in accordance with example implementations, a write command may be processed by the storage array 120 pursuant to a technique 400. The technique 400 includes the target port receiving (block 404) a command from an initiator. If the target port determines (decision block 408) that the command is a write command, then the target port determines (decision block 412) whether the write command

qualifies for the port to handle communication with the initiator to transfer the data using pre-allocated memory. If so, the target port uses (block 416) buffers from its pre-allocated pool to handle the transfer and indicates its readiness to the initiator, pursuant to block 416. If, however, the write command does not qualify (decision block 412), the port forwards (block 420) the request to the storage array controller, waits for the identification of the memory buffer(s) from the controller and then indicates readiness of the port to receive the data. As depicted in Fig. 4, after indicating its readiness to the initiator, the target port waits (block 424) for the write data.

[0037] As indicated by decision block 428, the transfer of the data to the memory of the storage array controller depends on whether the write is within the class to be handled using pre-allocated memory. In this manner, if the write is a qualified write, the target port uses DMA to transfer the data to the pre-allocated memory of the storage array controller at the proper offset, pursuant to block 432. Otherwise, the target port uses DMA transfer to transfer the data to the storage array buffers allocated after receipt of the write command, pursuant to block 436.

[0038] At the conclusion of the data transfer, the target port notifies (block 440) the storage array controller about the data phase completion. The target port then waits (block 444) for the status from the storage array controller and posts (block 450) the status to the initiator, pursuant to block 450.

[0039] As depicted in Fig. 4, if the command is not a write command (decision block 408), the port uses the storage array controller to regulate the additional processing phases with the initiator, pursuant to block 446; and control proceeds to block 444.

[0040] Among the potential advantages of the techniques and systems that are disclosed herein, write operation performance may be improved without the need for the initiator to modify its standard behavior. Configuration is under the full control of the storage array. The storage array controller may experience a significant reduction in interrupt processing, and the storage

array controller may experience a significant decrease in its CPU loading. The systems and techniques are disclosed herein may be especially beneficial for storage arrays that have a relatively high target port density. Other and different advantages are contemplated, which are within the scope of the appended claims.

[0041] While a limited number of examples have been disclosed herein, those skilled in the art, having the benefit of this disclosure, will appreciate numerous modifications and variations therefrom. It is intended that the appended claims cover all such modifications and variations.

What is claimed is:

- 1 1 A method comprising:
2 receiving a command in a target port, the command being provided by an
3 initiator and being associated with a write operation; and
4 in response to receiving the command, using the target to process a data
5 transfer for the initiator associated with the write operation, wherein the processing
6 includes, based on a characteristic of the command, selectively using memory for
7 the transfer pre-allocated by a storage array controller prior to receipt of the
8 command by the target port or requesting an allocation of memory for the transfer
9 from the storage array controller.

- 1 2. The method of claim 1, wherein using the target port comprises
2 selectively using the target port to provide a ready to transfer signal to the initiator
3 without requesting the allocation of memory for the transfer from the storage array
4 controller.

- 1 3. The method of claim 1, wherein the command comprises a write
2 command, and using the target port comprises selectively indicating readiness of the
3 target port to receive the data without first requesting allocation of memory based at
4 least in part on a block size associated with the data.

- 1 4. The method of claim 3, wherein using the target port comprises
2 selectively indicating readiness of the target port to receive the data without first
3 requesting allocation of memory based at least in part on whether the block size
4 exceeds a line size of a cache memory used by the storage array controller.

- 1 5. The method of claim 1, wherein using the target port comprises
2 selectively indicating readiness of the target port to receive the data without first
3 requesting allocation of memory from the storage array controller based at least in
4 part on whether an offset associated with the data is associated with more than one
5 line of a cache memory used by the storage array controller.

1 6. The method of claim 1, further comprising:
2 using the storage array controller to program the target port with at least one
3 parameter indicative of the characteristic.

1 7. The method of claim 1, further comprising:
2 using the storage array controller to pre-allocate memory to the target port for
3 the transfer.

1 8. An apparatus comprising:
2 a storage array controller to, in response to a port indicating reception of
3 command from an initiator associated with a write operation, allocate memory to
4 receive data associated with the write operation; and
5 a port processor to process the write operation, the port processor to, based
6 on a characteristic of the write operation, selectively:
7 provide an indication to the initiator of a readiness of the port to receive
8 the data; and
9 transfer the data to a memory region pre-allocated by the storage
10 array controller before the port receives the command.

1 9. The apparatus of claim 8, wherein the port processor is programmed
2 with the pre-allocated memory region, the pre-allocated memory region being
3 associated with a least one memory buffer reserved by the storage array controller
4 for use by the port processor.

1 10. The apparatus of claim 8, wherein the port processor performs a direct
2 memory access (DMA) transfer of the data to the memory region.

1 11. The apparatus of claim 8, wherein the port processor selectively waits
2 for an association of memory from the storage array controller after the port
3 processor receives the command based on the characteristic of the write operation.

1 12. The apparatus of claim 8, further comprising:
2 media storage drives coupled to the storage array controller, wherein the
3 storage array controller stores the data in at least one of the drives,

1 13. An article comprising a non-transitory computer readable storage
2 medium to store instructions that when executed by a computer cause the computer
3 to:
4 program a port processor with at least one parameter describing at least one
5 characteristic of a write command from an initiator for which the port processor is to
6 process the write operation by indicating a readiness to the initiator to receive data
7 associated with the write command and transfer the data to a memory region
8 allocated prior to receipt of the write command by the port processor; and
9 allocate memory for the port processor to transfer data associated with
10 another write command not described by the at least one parameter after receipt of
11 the another write command by the port processor.

1 14. The article of claim 13, wherein the characteristic comprises a block
2 size associated with the data.

1 15. The article of claim 13, wherein the at least one parameter indicates
2 whether the data will be associated with one cache line or associated with multiple
3 cache lines.

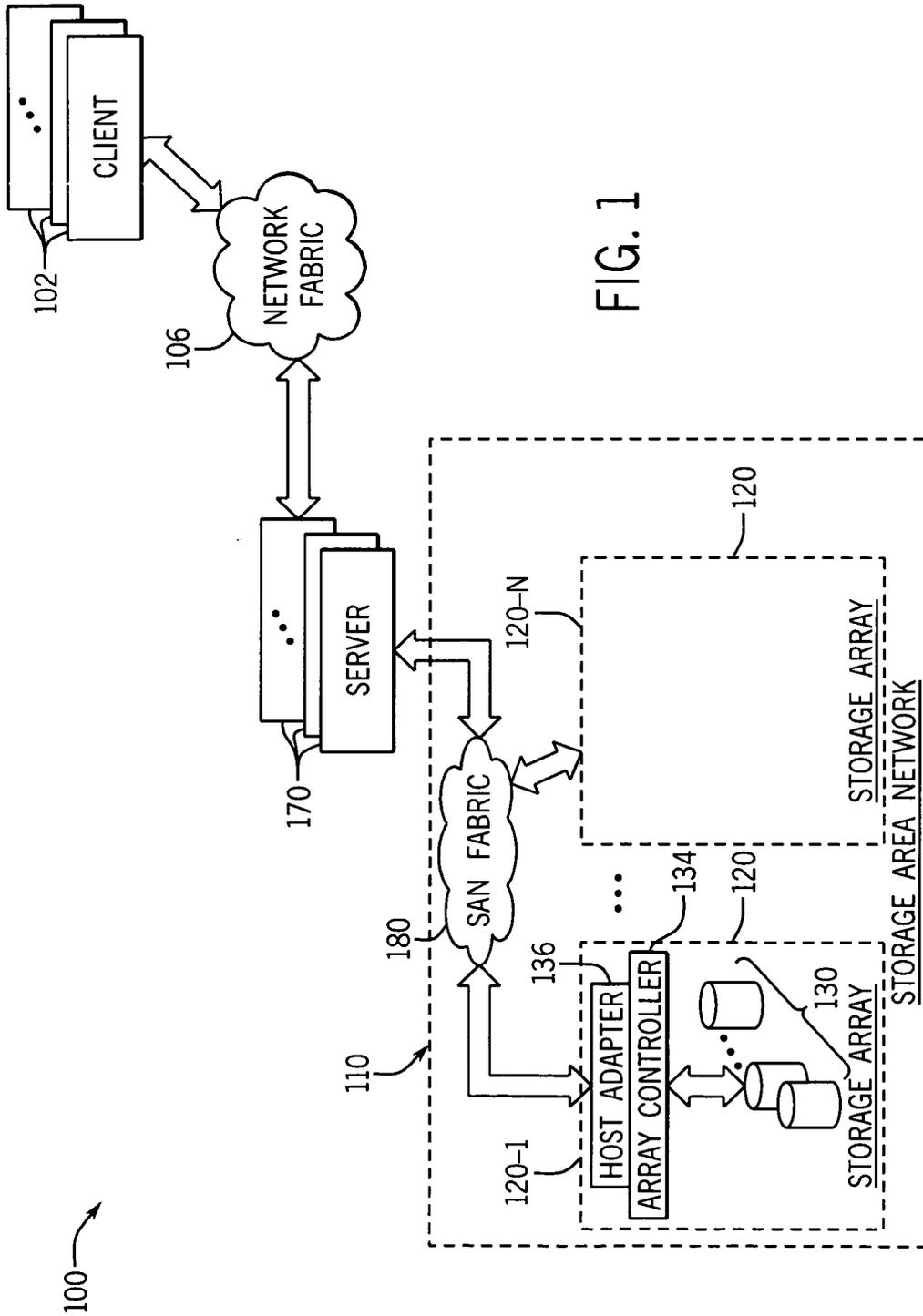


FIG. 1

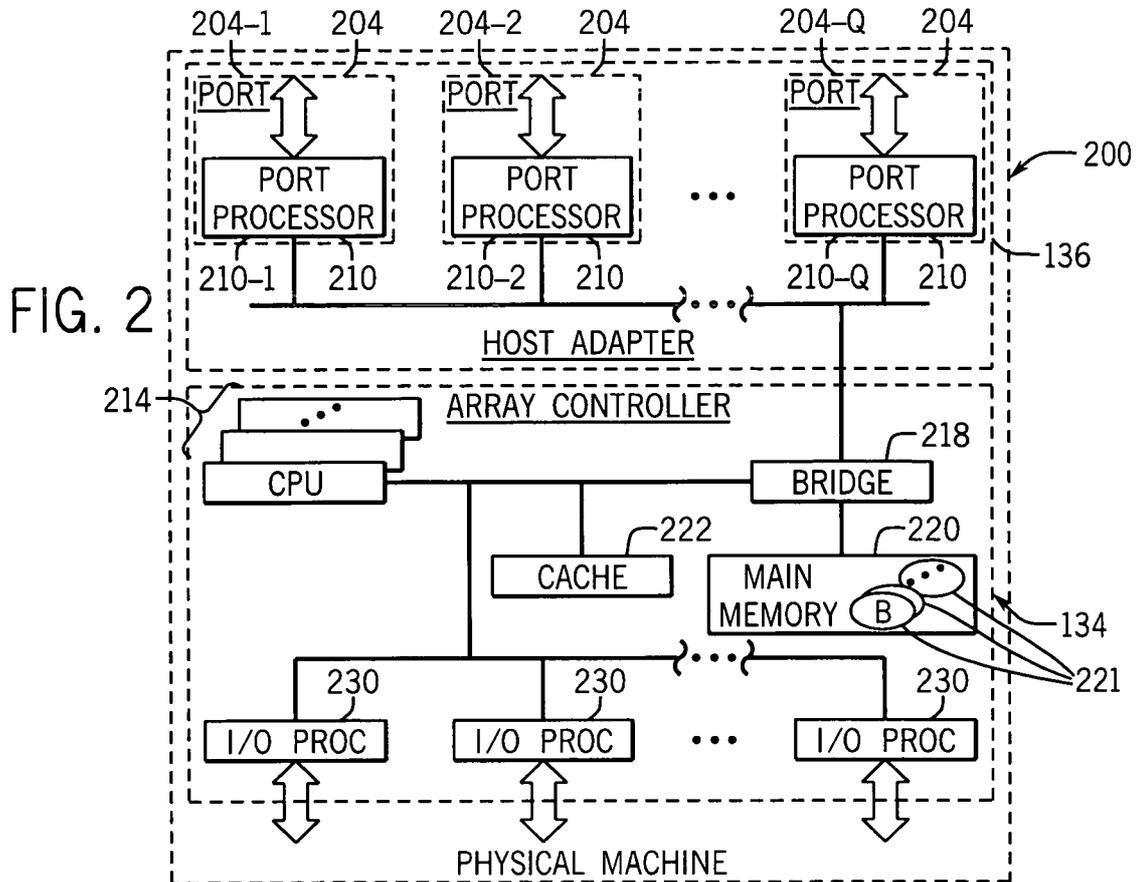


FIG. 2

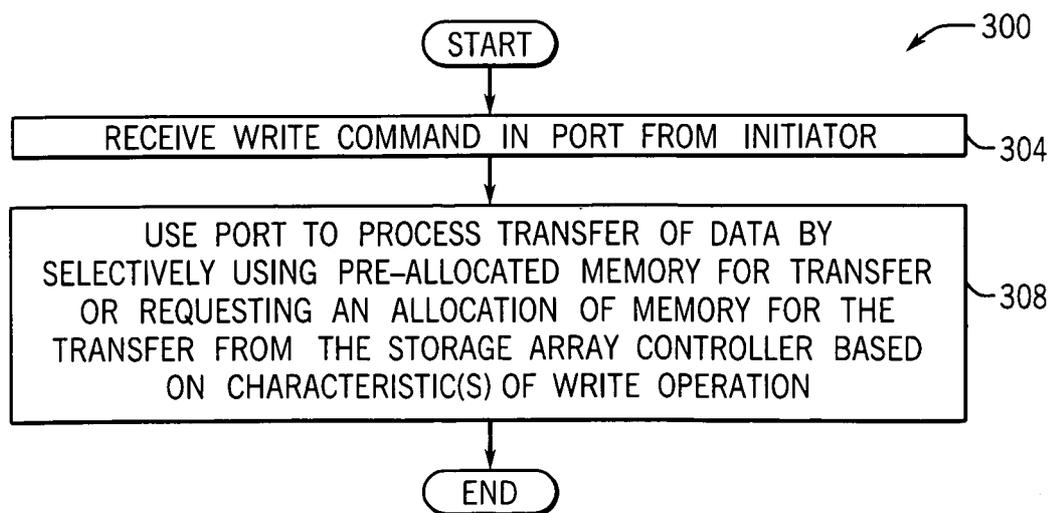


FIG. 3

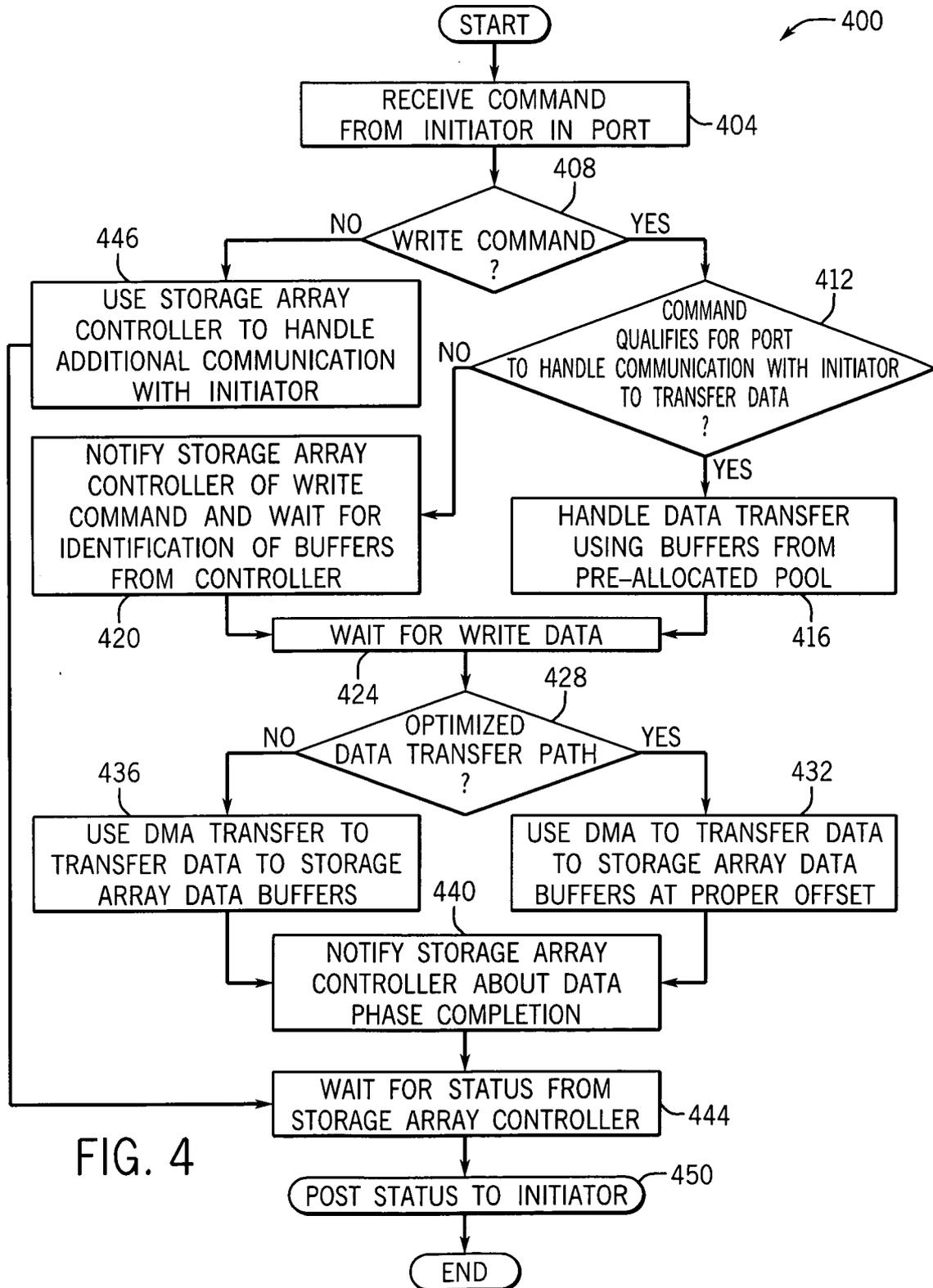


FIG. 4

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US20 13/067787**A. CLASSIFICATION OF SUBJECT MATTER****G06F 13/38(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 13/38; G06F 12/02; G06F 15/173; G06F 1/30; G06F 12/00; G06F 13/14; G06F 15/167; G06F 13/00

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, array, controller, command, port, write, memory, initiator, pre-allocate, server, SAN

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 2007-0208821 AI (JOSEPH C. PITTMAN) 06 September 2007 See paragraphs [0009] , [0013] , [0028] , [0064] ; claims 1, 4; and figures 1, 7.	1-15
Y	US 2011-0004732 AI (SHAFFI HAGOP KRAKIRIAN et al.) 06 January 2011 See paragraphs [0036] , [0084] , [0097] , [0101H0102] , [0105H0106] , [0109] ; claim 1; and figures 3, 5A-5F, 7-8 , 12-13.	1-15
A	US 2012-0089854 AI (JASON BREAKSTONE et al.) 12 April 2012 See paragraphs [0031]-[0033] ; and figure 2.	1-15
A	US 2011-0289252 AI (GEOFFREY B. HOESE et al.) 24 November 2011 See paragraphs [0026]- [0027] ; and figure 3.	1-15
A	US 2013-0151646 AI (SRIRAM CHIDAMBARAM et al.) 13 June 2013 See paragraphs [0125]-[0162] ; and figures 1A-1B .	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 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

25 July 2014 (25.07.2014)

Date of mailing of the international search report

25 July 2014 (25.07.2014)

Name and mailing address of the ISA/KR


International Application Division
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/US2013/067787

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2007-0208821 AI	06/09/2007	US 7249227 B1 us 7849274 B2	24/07/2007 07/12/2010
US 2011-0004732 AI	06/01/2011	None	
US 2012-0089854 AI	12/04/2012	US 8688926 B2	01/04/2014
US 2011-0289252 AI	24/11/2011	AT 313204 T CA 2315199 AI CA 2315199 C DE 69832818 D1 DE 69832818 T2 EP 1044414 AI EP 1044414 A4 EP 1044414 B1 EP 1696638 AI EP 2375698 AI EP 2375699 AI JP 2002-500396 A JP 4691251 B2 US 2002-010812 AI us 2002-010813 AI us 2002-052986 AI us 2002-083222 AI us 2002-083223 AI us 2002-083240 AI us 2002-161950 AI us 2003-131157 AI us 2004-054838 AI us 2005-283560 AI us 2006-143322 AI us 2006-218322 AI us 2007-299995 AI us 2008-046601 AI us 2008-046602 AI us 2008-065808 AI us 2008-307444 AI us 2009-319715 AI us 2009-319742 AI us 2009-327566 AI us 2010-121993 AI us 2011-035527 AI us 2011-035528 AI us 2011-040949 AI us 2011-106998 AI us 2013-151807 AI us 5941972 A us 6421753 B1 us 6425035 B2	15/12/2005 08/07/1999 21/01/2003 19/01/2006 05/10/2006 18/10/2000 09/05/2001 14/12/2005 30/08/2006 12/10/2011 12/10/2011 08/01/2002 01/06/2011 24/01/2002 24/01/2002 02/05/2002 27/06/2002 27/06/2002 27/06/2002 31/10/2002 10/07/2003 18/03/2004 22/12/2005 29/06/2006 28/09/2006 27/12/2007 21/02/2008 21/02/2008 13/03/2008 11/12/2008 24/12/2009 24/12/2009 31/12/2009 13/05/2010 10/02/2011 10/02/2011 17/02/2011 05/05/2011 13/06/2013 24/08/1999 16/07/2002 23/07/2002

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2013/067787

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
		US 6425036 B2	23/07/2002
		US 6738854 B2	18/05/2004
		US 6763419 B2	13/07/2004
		US 6789152 B2	07/09/2004
		US 7051147 B2	23/05/2006
		US 7340549 B2	04/03/2008
		US 7552266 B2	23/06/2009
		US 7689754 B2	30/03/2010
		US 7694058 B2	06/04/2010
		US 7934040 B2	26/04/2011
		US 7934041 B2	26/04/2011
		US 7937517 B2	03/05/2011
		US 7984221 B2	19/07/2011
		US 7984224 B2	19/07/2011
		US 7987311 B2	26/07/2011
		US 8015339 B2	06/09/2011
		US 8028117 B2	27/09/2011
		US 8046515 B2	25/10/2011
		US 8402193 B2	19/03/2013
		US 8402194 B2	19/03/2013
		wo 1999-34297 A1	08/07/1999
US 2013-0151646 A1	13/06/2013	US 7843906 B1	30/11/2010
		US 8218538 B1	10/07/2012