



- (51) **International Patent Classification:**
G06F 12/08 (2016.01) *G06F 12/06* (2006.01)
- (21) **International Application Number:**
PCT/US2015/012184
- (22) **International Filing Date:**
21 January 2015 (21.01.2015)
- (25) **Filing Language:** English
- (26) **Publication Language:** English
- (71) **Applicant:** HEWLETT PACKARD ENTERPRISE DEVELOPMENT LP [US/US]; 11445 Compaq Center Drive West, Houston, TX 77070 (US).
- (72) **Inventors:** DENEUL, Nathaniel S; 11445 Compaq Center Dr W, Houston, Texas 77070 (US). BLACK, Joseph David; 11445 Compaq Center Dr W, Houston, Texas 77070 (US).
- (74) **Agents:** ORTEGA, Arthur et al.; Hewlett Packard Enterprise, 3404 E. Harmony Road, Mail Stop 79, 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, 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, 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, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

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

[Continued on next page]

(54) **Title:** VIRTUAL STORAGE

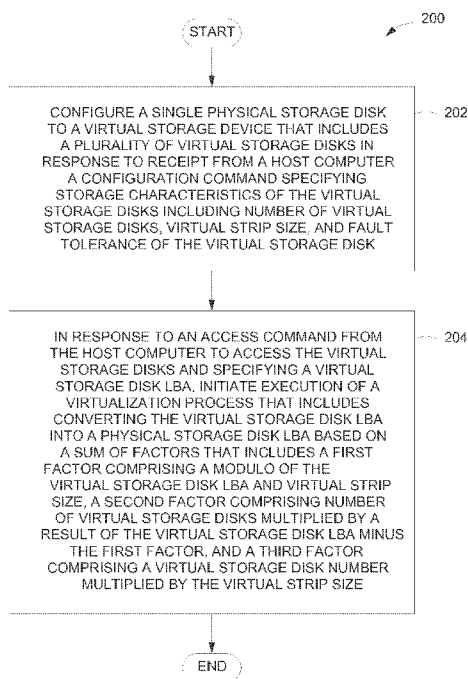


FIG. 2

(57) **Abstract:** In some examples, techniques for virtual storage includes configuring a single physical storage disk to a virtual storage device that includes a plurality of virtual storage disks from a command specifying storage characteristics of the virtual storage disks including number of virtual storage disks, virtual strip size, and fault tolerance of the virtual storage disks. In response to access the virtual storage disks and specifying a virtual storage disk Logical Block Address (LBA), converting the virtual storage disk LBA into a physical storage disk LBA based on a sum of factors that includes a first factor comprising a modulo of the virtual storage disk LBA and virtual strip size, a second factor comprising number of virtual storage disks multiplied by a result of the virtual storage disk LBA minus the first factor, and a third factor comprising a virtual storage disk number multiplied by the virtual strip size.



— *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))*

VIRTUAL STORAGE

BACKGROUND

[0001] Storage devices, such as hard disk drives and solid state disks, can be arranged in various configurations for different purposes. For example, storage devices can be configured to provide fault tolerance with different redundancy levels as part of a Redundant Array of Independent Disks (RAID) storage configuration. In such a configuration, the storage devices can be arranged to represent logical or virtual storage and to provide different performance and redundancy based on the RAID level.

BRIEF DESCRIPTION OF THE DRAWINGS

[0002] Fig. 1 is a block diagram of a computer system for virtual storage according to an example implementation.

[0003] Fig. 2 is a flow diagram for performing virtual storage according to an example implementation.

[0004] Fig. 3 is a block diagram of virtual storage according to an example implementation.

[0005] Fig. 4 is a block diagram of virtual storage according to another example implementation.

[0006] Fig. 5 is a flow diagram of virtual storage according to another example implementation.

[0007] Fig. 6 is a table of virtual storage according to an example implementation.

[0008] Fig. 7 is an example block diagram showing a non-transitory, computer-readable medium that stores instructions for a computer system for virtual storage in accordance with an example implementation.

DETAILED DESCRIPTION

[0009] Storage devices, such as hard disk drives and solid state disks, can be arranged in various configurations for different purposes. For example, storage devices can be configured to provide fault tolerance with different redundancy levels as part of a Redundant Array of Independent Disks (RAID) storage configuration. In such a configuration, the storage devices can be arranged to represent logical or virtual storage and to provide different performance and redundancy based on the RAID level. In one example, virtual storage techniques may allow for grouping of a plurality of physical storage from network storage devices to provide single storage device. Redundancy of storage devices can be based on mirroring of data, where data in a source storage device is copied to a mirror storage device (which contains a mirror copy of the data in the source storage device). In this arrangement, if an error or fault condition causes data of the source storage device to be unavailable, the mirror storage device can be accessed to retrieve the data.

[0010] Another form of redundancy is parity-based redundancy where actual data is stored across a group of storage devices, and parity information associated with the data is stored in another storage device. If data within any of the group of storage devices were to become inaccessible (due to data error or storage device fault or failure), the parity information from the other non-failed storage device can be accessed to rebuild or reconstruct the data. Examples of parity-based redundancy configurations such as RAID configurations, including RAID-5 and RAID-6 storage configurations. An example of a mirroring redundancy configurations is the RAID-1 configuration. In RAID-3 and RAID-4 configurations, parity information is stored in dedicated storage devices. In RAID-5 and RAID-6 storage configurations, parity information is distributed across all of the storage devices. Although reference is made to RAID in this description, it is noted that some embodiments of the present application can be applied to other types of redundancy configurations, or to any arrangement in which a storage volume is implemented across multiple storage devices (whether redundancy is used

or not). A storage volume may be defined as virtual storage that provides a virtual representation of storage that comprises or is associated with physical storage elements such as storage devices. For example, the system can receive host access commands or requests from a host to access data or information on storage volume where the requests include storage volume address information and then the system translates the volume address information into the actual physical address of the corresponding data on the storage devices. The system can then forward or direct the processed host requests to the appropriate storage devices.

[0011] When any portion of a particular storage device is detected as failed or exhibiting some other fault condition, the entirety of the particular storage device may be marked as unavailable for use. As a result, the storage volumes may be unable to use the particular storage device. A fault condition or failure of a storage device can include any error condition that prevents access of a portion of the storage device. The error condition can be due to a hardware or software failure that prevents access of the portion of the storage device. In such cases, the system can implement a reconstruction or rebuild process that includes generating rebuild requests comprising commands directed to the storage subsystem to read the actual user data from the storage devices that have not failed and parity data from the storage devices to rebuild or reconstruct the data from the failed storage devices. In addition to the rebuild requests, the system also can process host requests from a host to read and write data to storage volumes that have not failed as well as failed, where such host requests may be relevant to performance of the system. Storage systems may include backup management functionality to perform backup and restore operations. Backup operations may include generating a copy of data that is in use to allow the data to be recovered or restored in the event the data is lost or corrupted. Restore operations may include retrieving the copy of the data and replacing the lost or corrupted data with the retrieved copy of the data.

[0012] However, some storage systems may not be able to provide redundancy because hardware redundancy may be either too costly or limited by physical space. In some storage systems, data redundancy may be provided either external to the system or not at all. Some storage devices or media devices may occasionally encounter data loss in a non-catastrophic manner which may leads to problem with handling resulting command errors and rebuilding or regenerating the data or returning the subsequent command failures.

[0013] The techniques of the present application may help improve the performance or functionality of computer and storage systems. For example, the techniques may implement a storage stack to configure or divide a single physical storage disk or media into multiple separate virtual storage disks in accordance with a process to allow the generation of RAID level fault tolerance with reduced levels of performance loss. The storage stack can be implemented as hardware, software or a combination thereof. These techniques may enable a storage system or a storage controller of a storage system to perform data checking and data repair without the need for multiple real physical disks and with little or no performance loss to most InputOutput (IO) patterns such as from read and write access commands from hosts to access storage.

[0014] Computer systems may include striping or data striping techniques to allow the system to segment logically sequential data, such as a file, so that consecutive segments are stored on different physical storage devices. The striping techniques may be used for processing data more quickly than may be provided by a single storage device. The computer may distribute segments across devices allow data to be accessed concurrently which may increase total data throughput. Computer systems may include fault tolerance techniques to allow the system to continue to operate properly in the event of the failure of (or one or more faults within) some of its components. Computer systems may employ Logical Unit (LUN) which may be defined as a unique identifier used to designate individual or collections of storage devices for

address by a protocol associated with various network interfaces. In one example, LUNs may be employed for management of block storage arrays shared over a Storage Area Network (SAN). Computer systems may employ Logical Block Address (LBA) addressing techniques for specifying the location of blocks of data stored on computer storage devices. In one example, an LBA may be a linear addressing technique where blocks are located using an integer index, with the first block being LBA 0, the second LBA 1, and so on.

[0015] In one example, the techniques of the present application may provide a storage stack to implement a method allow configuration software applications of a computer system to provide a set of options for configuring a single physical storage device or disk (storage media) as a set of virtual storage disks. The configuration or options of the virtual storage disks may include specifying a number of virtual storage disks or devices, virtual strip size of the virtual disks or devices and fault tolerance from a RAID level configuration. The system, upon receiving a configuration command, may save the configuration and relevant information in a persistent manner. Once the system configures or establishes the single physical storage disk, the storage stack may expose a LUN to a host system allowing the host system access to the virtual disks. The total capacity of this LUN may include the original capacity of the physical storage disk less the capacity reserved to accomplish the desired fault tolerance. The host may now access the physical storage disk directly with a logical block address and a command specifying a storage access command such as to read data from or write data to the storage disk.

[0016] When a computer system or storage controller of the computer system receives an access command directed to the LUN, it may initiate execution of a virtualization process that includes converting the single command or request into separate requests to the virtual storage disks comprising the LUN. The individual requests may then be converted from the virtual storage disks specifying a virtual storage LBA to a request directed or targeted to the original physical storage disk in accordance to a sum of three

factors: a first factor comprising calculation of a modulo of the number of virtual storage disks and virtual strip size, a second factor comprising a calculation of the number of virtual storage disks multiplied by a result of the virtual storage disk LBA minus the first factor, and a third factor comprising a calculation of the virtual storage disk number multiplied by the virtual strip size. The virtual storage disk LBA includes the LBA for a virtual storage disk specified in an access command from a host. The virtual storage strip size may be specified by the configuration command from the host or other configuration application. The number of virtual storage disks may be the configured number of virtual storage disks specified by the configuration command sent from the host. The virtual disk number may be the specific virtual storage disk which is the resulting or actual storage location desired by the host in the access command.

[0017] In this manner, these techniques may allow for virtual storage disks or devices of equal LBA strip ranges to be contiguous which may help increase the overall performance or functionality of the system that may be obtained from the physical storage disk or media when sequential LBA operations are being performed. These techniques may apply to recovering data from failed devices from a fault condition from whatever fault tolerance selected.

[0018] In another example, the techniques of the present application disclose virtualization module to configure a single physical storage disk to a virtual storage device that includes a plurality of virtual storage disks in response to receipt from a host computer a configuration command specifying storage characteristics of the virtual storage disks including number of virtual storage disks, virtual strip size, and fault tolerance of the virtual storage disks. The virtualization module, in response to an access command from the host computer to access the virtual storage disks and specifying a virtual storage disk LBA, initiate a virtualization process that includes converting the virtual storage disk LBA into a physical storage disk LBA based on a sum of factors that includes: a first factor comprising a modulo of the number of virtual

storage disks and virtual strip size, a second factor comprising number of virtual storage disks multiplied by a result of the virtual storage LBA minus the first factor, and a third factor comprising a virtual storage disk number multiplied by the virtual strip size.

[0019] In some examples, the access command may include a read command from the host computer to read data from a virtual LBA of the virtual storage disks. The access command may include a write command from the host computer to write data to a virtual LBA of the virtual storage disks. The virtualization module may be further configured to configure the storage virtual disks in accordance with a fault tolerance level of RAID-4 wherein the parity information is assigned to a parity virtual disk and. In response to a storage fault condition associated with the storage virtual disks, the virtualization module may initiate execution of a rebuild process that includes employing the fault tolerance configuration of the storage virtual disks that includes reading parity information stored in the parity virtual disk to rebuild data as a result of the storage fault condition. The virtualization module may be further configured to configure the storage virtual disks in accordance with a fault tolerance level of RAID-5 wherein the parity information is distributed across parity virtual disks and. In response to a storage fault condition associated with the storage virtual disks, the virtualization module may initiate execution of a rebuild process that includes employing the fault tolerance configuration of the storage virtual disks that includes reading parity information stored across parity virtual disks to rebuild data as a result of the storage fault condition.

[0020] In this manner, the techniques of the present application may help improve the performance as well as the functionality of computer and storage systems. For example, this may allow or enable a storage stack of a computer system to build fault tolerance for the purpose of data protection against physical storage disk media errors into a single physical storage disk or media device. In another example, the techniques may allow the storage stack to correct errors on a single physical storage disk or media device

transparent to a host application. In another example, these techniques may allow a storage stack to correct errors on a scan initiated by the storage stack for the purpose of proactively finding and correcting errors. In yet another example, these techniques may help increase the performance for storage devices that have optimal performance with sequential IO by placing LBAs for the created LUN on contiguous LBAs on the underlying physical storage disk or media. In another example, these techniques may help fault tolerance operations that require multiple sub-operations (e.g., parity generation), where the virtual conversion or mapping calculations help ensure that those operations will be in close proximity, thereby helping increase performance.

[0021] Fig. 1 is a block diagram of a computer system 100 for virtual storage according to an example implementation.

[0022] In one example, computer device 102 is configured to communicate with storage device 104 over communication channel 112. The computer device 102 includes a virtualization module 106 to communicate with storage device 104 as well as other devices such as host computers (not shown). The storage device 104 includes a physical storage disk 110 which is configured by virtualization module 106 as a plurality of virtual storage disks 108 (108-1 through 108-n, where n can be any number), as discussed in detail below.

[0023] In one example, virtualization module 106 may be configured to communicate with hosts or other computer devices to configure storage device 104. For example, virtualization module 106 may be able to configure single physical storage disk 110 as a virtual storage device that includes a plurality of virtual storage disks 108. The configuration can be performed in response to receipt of a configuration command from a host computer or other computer device. The configuration command may include information about configuration of physical storage disk 110. The configuration command may specify storage characteristics of virtual storage disks 108 including number of

virtual storage disks, virtual strip size, and fault tolerance of the virtual storage disks.

[0024] The virtualization module 106 may be able to communicate with host computers or other computers to allow hosts to access storage from storage device 104. For example, virtualization module 106 may be able to respond to an access command from a host computer to access virtual storage disks 108 and specifying a virtual storage disk LBA. In response to the command, virtualization module 106 may initiate execution of a virtualization process that includes converting the virtual storage disk LBA into a physical storage disk LBA based on a sum of factors that includes: a first factor comprising a modulo of the virtual storage disk LBA and virtual strip size, a second factor comprising number of virtual storage disks multiplied by a result of the virtual storage disk LBA minus the first factor, and a third factor comprising a virtual storage disk number multiplied by the virtual strip size. The Table 1 below illustrates the virtualization process that generates a physical storage device LBA:

Table 1

$\begin{aligned} \text{Physical Storage Device LBA} = & \\ & (\text{Virtual Storage Disk LBA modulo Virtual Strip size}) + \\ & (\text{Virtual Storage Disk LBA} - (\text{Virtual Storage Disk LBA modulo Virtual Strip} \\ & \text{size})) * \text{Number Virtual Devices}) + \\ & (\text{Virtual Disk Number} * \text{Virtual Strip Size}) \end{aligned}$

[0025] In some examples, the access command may include a read command from a host computer to read data from a virtual LBA of virtual storage disks 108. In another example, the access command may include a write command from a host computer to write data to a virtual LBA of virtual storage disks 108. The virtualization module 106 may be further configured to configure virtual storage disks 108 in accordance with a fault tolerance level of RAID-4 wherein the parity information is assigned to a parity virtual disk and. In response to a storage fault condition associated with the storage virtual

disks, virtualization module 106 may initiate execution of a rebuild process. The rebuild process may include having virtualization module 106 employ the fault tolerance configuration of the storage virtual disks that includes a process of reading parity information stored in the parity virtual disk to rebuild data as a result of the storage fault condition.

[0026] In another example, virtualization module 106 may be further configured to configure virtual storage disks 108 in accordance with a fault tolerance level of RAID-5 wherein the parity information is distributed across parity virtual disks. In response to a storage fault condition associated with the storage virtual disks, virtualization module 106 may initiate execution of a rebuild process. The rebuild process may include having virtualization module 106 employ the fault tolerance configuration of the storage virtual disks that includes reading parity information stored across parity virtual disks to rebuild data as a result of the storage fault condition.

[0027] The computer device 102 may be any electronic device capable of data processing such as a server computer, mobile device, notebook computer and the like. The functionality of the components of computer device 102 may be implemented in hardware, software or a combination thereof. In one example, computer device 102 may include functionality to manage the operation of the computer device. For example, computer device 102 may include functionality to communicate with other computer devices such as host computers to receive access commands from the host computer to access storage from storage device 104.

[0028] The storage device 104 may include a single storage disk 110 as shown configured to present logical storage devices to computer device 102 or other electronic devices such as hosts. The storage device 104 is shown to include a single physical storage disk 110 but may include a plurality of storage devices (not shown) configured to practice the techniques of the present application. In one example, computer device 102 may be coupled to other computer devices such as hosts which may access the logical

configuration of storage array as LUNS. The storage device 104 may include any means to store data for later access or retrieval. The storage device 104 may include non-volatile memory, volatile memory or a combination thereof. Examples of non-volatile memory include, but are not limited to, Electrically Erasable Programmable Read Only Memory (EEPROM) and Read Only Memory (ROM). Examples of volatile memory include, but are not limited to, Static Random Access Memory (SRAM), and Dynamic Random Access Memory (DRAM). Examples of storage devices may include, but are not limited to, Hard Disk Drives (HDDs), Compact Disks (CDs), Solid State Drives (SSDs), optical drives, flash memory devices and other like devices.

[0029] The communication channel 112 may include any electronic communication means of communication including wired, wireless, network based such SAN, Ethernet, FC (Fibre Channel) and the like.

[0030] In this manner, the techniques of the present application may help improve the performance as well as the functionality of computer and storage systems such as system 100. For example, these techniques may allow or enable a storage stack of a computer system to generate or build fault tolerance for the purpose of data protection against physical storage disk media errors into a single physical storage disk or media device. In another example, the techniques may allow storage stack to correct errors on a single physical storage disk or media device transparent to a host application. In another example, these techniques may allow a storage stack to correct errors on a scan initiated by the storage stack for the purpose of proactively finding and correcting errors. In yet another example, these techniques may help increase the performance for storage devices that have optimal performance with sequential IO by placing LBAs for the created LUN on contiguous LBAs on the underlying physical storage disk or media. In another example, these techniques may help fault tolerance operations that require multiple sub-operations (e.g., parity generation), where the virtual conversion or mapping calculations help ensure that those operations will be in close proximity, thereby helping increase performance.

[0031] It should be understood that the description of system 100 herein is for illustrative purposes and other implementations of the system may be employed to practice the techniques of the present application. For example, computer device 102 is shown as a single component but the functionality of the computer device may be distributed among a plurality of computer devices to practice the techniques of the present application.

[0032] Fig. 2 is a flow diagram for performing virtual storage according to an example implementation.

[0033] In one example, to illustrate operation, it may be assumed that computer device 102 is configured to communicate with storage device 104 and another device such as a host computer device.

[0034] Processing may begin at block 202, wherein virtualization module 106 configures a single physical storage disk 110 to a virtual storage device that includes a plurality of virtual storage disks 108. The configuration can be performed in response to receipt of a configuration command from a host computer or other computer device. The configuration command may include information about configuration of physical storage disk 110. The configuration command may specify storage characteristics of virtual storage disks 108 including number of virtual storage disks, virtual strip size, and fault tolerance of the virtual storage disks. Processing then proceeds to block 204.

[0035] At block 204, virtualization module 106 responds to an access command from a host computer to access virtual storage disks 108 and specifying a virtual storage disk LBA. In response to the command, virtualization module 106 initiates execution of a virtualization process that includes converting the virtual storage disk LBA into a physical storage disk LBA based on a sum of factors that includes: a first factor comprising a modulo of the virtual storage disk LBA and virtual strip size, a second factor comprising number of virtual storage disks multiplied by a result of the virtual storage disk LBA minus the first factor, and a third factor comprising a virtual

storage disk number multiplied by the virtual strip size. In one example, once processing of block 204 is completed, processing may then proceed back to block 202 to process further commands or requests. In another example, processing may terminate at the End block.

[0036] In another example, processing may include having virtualization module 106 configure virtual storage disks 108 in accordance with a fault tolerance level of RAID-4 wherein the parity information is assigned to a parity virtual disk and. In response to a storage fault condition associated with virtual storage disks 108, virtualization module 106 may initiate execution of a rebuild process. The rebuild process may include having virtualization module 106 employ the fault tolerance configuration of virtual storage disks 108 that includes reading parity information stored in the parity virtual disk to rebuild data as a result of the storage fault condition.

[0037] In another example, processing may include having virtualization module 106 configure virtual storage disks 108 in accordance with a fault tolerance level of RAID-5 wherein the parity information is distributed across parity virtual disks and. In response to a storage fault condition associated with virtual storage disks 108, virtualization module 106 may initiate execution of a rebuild process. The rebuild process may include having virtualization module 106 employ the fault tolerance configuration of virtual storage disks 108 that includes reading parity information stored across parity virtual disks to rebuild data as a result of the storage fault condition.

[0038] It should be understood that the above process 200 is for illustrative purposes and that other implementations may be employed to practice the techniques of the present application. For example, computer device 102 is shown as a single component but the functionality of the computer device may be distributed among a plurality of computer devices to practice the techniques of the present application. In another example, a storage controller may be employed with computer device 102 and/or storage device 104 to practice the techniques of the present application.

[0039] Fig. 3 is a diagram of virtual storage 300 according to an example implementation. As explained above, virtualization module 106 may configure single physical storage disk to a virtual storage device that includes a plurality of virtual storage disks.

[0040] For example, virtualization module 106 configures a single physical storage disk 302 to a virtual storage device that includes four virtual storage disks 304 (Virtual Disk 1 through Virtual Disk 4). As explained above, in one example, virtualization module 106 may be configured to execute a virtualization process that includes converting a virtual storage disk LBA into a physical storage disk LBA based on a sum of factors. In this case, to illustrate operation, the virtualization process executed by virtualization module 106 present to a host a plurality of virtual storage disks 304 instead of single physical storage disk 302. In another example, virtualization module 106 configures four virtual storage disks 306 (Virtual Disk 1 through Virtual Disk 4) with a data layout of virtual storage that includes a mapping to physical storage disk 302 without striping. In another example, on the other hand, virtualization module 106 configures four virtual storage disks 308 (Virtual Disk 1 through Virtual Disk 4) with a data layout that includes a virtual storage mapping to physical storage disk 302 with data striping. In addition, the data layout of the four virtual storage disks 308 (Virtual Disk 1 through Virtual Disk 4) are shown to include physical LBA ranges from 0x0000 (hex value) to 0x7FFFF (hex value) in an increasing manner.

[0041] Fig. 4 is a block diagram of virtual storage 400 according to another example implementation. As explained above, in one example, virtualization module 106 may be configured to execute a virtualization process that includes converting the virtual storage disk LBA into a physical storage disk LBA based on a sum of factors. In one example, to illustrate operation, virtualization module 106 configures a single physical storage disk 402 to a virtual storage device that includes four virtual storage disks 404 (Virtual Disk 1 through Virtual Disk 4). The virtualization module 106 presents to a host virtual storage disks 404 instead of single physical storage disk 402. In this

case, virtualization module 106 provides RAID fault tolerance where the storage capacity of raw physical storage disk 402 is shown relative to virtual disk mappings 404 and RAID logical device 406 configured in RAID-4 or RAID-5 fault tolerance.

[0042] Fig. 5 is a flow diagram of virtual storage 500 according to another example implementation. In one example, to illustrate operation, shown is a data flow of a host read access command for a single block of logical data. It may be assumed, to illustrate operation, virtualization module 106 configures a single physical storage disk 110 to a virtual storage device that includes a plurality of virtual storage disks 108. The configuration may be performed in response to receipt of a configuration command from a host computer or other computer device. The configuration command may include information about configuration of physical storage disk 110. The configuration command may specify storage characteristics of virtual storage disks 108 including number of virtual storage disks, virtual strip size, and fault tolerance of the virtual storage disks. In addition, as explained above, in one example, virtualization module 106 may be configured to execute a virtualization process that includes converting the virtual storage disk LBA into a physical storage disk LBA based on a sum of factors.

[0043] Processing may begin at block 502, wherein virtualization module 106 receives a read access command from a host to read a single block of logical data from storage device 104. The read command may include a request to read data for a virtual LBA for a logical volume of the virtual storage disks. Processing then proceeds to block 504.

[0044] At block 504, virtualization module 106 maps logical LBA to virtual storage disk and LBA range. In one example, virtualization module 106 converts or maps the virtual or logical LBA received from the host to virtual storage disk and LBA range of virtual storage disks 108 of storage device 104. Processing then proceeds to block 506.

[0045] At block 506, virtualization module 106 maps virtual storage disk and LBA to physical storage disk LBA. In one example, virtualization module 106 converts or maps the virtual or logical LBA received from the host to virtual storage disk and LBA range of physical storage disk 110 of storage device 104. Processing then proceeds to block 508.

[0046] At block 508, virtualization module 106 sends a read request or command to storage device 104. The storage device 104 may include a storage controller to manage the read request from virtualization module 106. In another example, computer device 102 may include a storage controller to manage the read request from virtualization module 106. Processing then proceeds to block 510.

[0047] At block 510, virtualization module 106 checks whether the read request or command was successful. If the read request was successful, then processing proceeds to block 512. On the other hand, if the read request was not successful, then processing proceeds to block 516.

[0048] At block 512, virtualization module 106 receives the requested data from storage device 104 and forwards the data to the host that requested the data. Processing proceeds to block 514.

[0049] At block 514, virtualization module 106 sends a message or notification to the host indicating the read request was successful. Processing may then terminate or return to block 502 to continue to receive read requests from a host.

[0050] At block 516, virtualization module 106 initiates execution of a rebuild process. In one example, the rebuild process may include a data regeneration process where one or more virtual storage disks are read depending on RAID fault tolerance configuration.

[0051] In another example, processing may include having virtualization module 106 configure virtual storage disks 108 in accordance with a fault tolerance level of RAID-4 wherein the parity information is assigned to a parity

virtual disk. In response to a storage fault condition associated with virtual storage disks 108, virtualization module 106 may initiate execution of a rebuild process. The rebuild process may include having virtualization module 106 employ the fault tolerance configuration of virtual storage disks 108 that includes reading parity information stored in the parity virtual disk to rebuild data as a result of the storage fault condition.

[0052] In another example, processing may include having virtualization module 106 configure virtual storage disks 108 in accordance with a fault tolerance level of RAID-5 wherein the parity information is distributed across parity virtual disks. In response to a storage fault condition associated with virtual storage disks 108, virtualization module 106 may initiate execution of a rebuild process. The rebuild process may include having virtualization module 106 employ the fault tolerance configuration of virtual storage disks 108 that includes reading parity information stored across parity virtual disks to rebuild data as a result of the storage fault condition. Processing proceeds to block 518.

[0053] At block 518, virtualization module 106 executes a conversion or translation process that includes mapping virtual storage disk and LBA's to physical storage disk LBA's. Processing proceeds to block 520.

[0054] At block 520, virtualization module 106 send rebuild read requests or commands to the disk. In one example, the read requests are based on the RAID level of the virtual storage disks. Processing proceeds to block 522.

[0055] At block 522, virtualization module 106 checks whether the rebuild read requests were successful. If the read requests were successful, then processing proceeds to block 524. On the other hand, if the read requests were not successful, then processing proceeds to block 526.

[0056] At block 524, virtualization module 106 perform any data transformation associated with rebuilding logical data. Processing proceeds to block 512.

[0057] At block 526, virtualization module 106 determines that logical data cannot be regenerated from the rebuild process. Processing proceeds to block 528.

[0058] At block 528, virtualization module 106 sends to the host a message indicating that the read request resulted or completed with a failure. In one example, the message can indicate a failure status indicating "Unrecoverable Read Error". Processing may then terminate or return to block 502 to continue to receive read requests from a host.

[0059] It should be understood that the above process 500 is for illustrative purposes and that other implementations may be employed to the practice the techniques of the present application. For example, virtualization module 106 may handle multiple read commands from a host to access storage device 104.

[0060] Fig. 6 is a table of virtual storage 600 according to an example implementation. For example, to illustrate, it may be assumed that virtualization module 106 configured storage device 104 to have four virtual storage disks (Virtual Drive 0 through Virtual Drive 3) with a RAID-4 fault tolerance configuration with a 128 block strip.

[0061] In one example, virtualization module 106 may receive a read access command from a host to read data from an address specified by LUN LBA 1536 for 16 blocks. The virtualization module 106 converts the read request address information (LUN LBA 1536 for 16 blocks) to Virtual LBA using RAID mapping: Virtual Disk 0, Virtual LBA 512-527. Then virtualization module 106 converts the virtual address information (Virtual Disk 0, Virtual LBA 512-527) to Physical Disk LBA using the above virtualization process: Virtual Disk 0, Virtual LBA 512-527 = Physical Disk LBA 2048-2063.

[0062] In another example, virtualization module 106 may receive a read access command from a host to read data from an address specified by LUN LBA 2000 for 60 blocks. The virtualization module 106 converts the read

request address information (LUN LBA 2000 for 60 blocks) to Virtual LBA via RAID mapping: Virtual Disk 0, Virtual LBA 720-767 and Virtual Disk 1, Virtual LBA 640-651. Then virtualization module 106 converts the virtual address information (Virtual Disk 0, Virtual LBA 720-767 and Virtual Disk 1, Virtual LBA 640-651) to Physical Disk LBA using the virtualization process: Virtual Disk 0, Virtual LBA 720-767 = Physical Disk LBA 2640-2687 and Virtual Disk 1 640-651 = Physical Disk LBA 2688-2699.

[0063] In another example, virtualization module 106 may receive a write access command to write data to an address specified by LUN LBA 1536 for 16 blocks. The virtualization module 106 converts the write request address information to virtual LBA via RAID mapping: Virtual Disk 0, Virtual LBA 512-527 (write), Virtual Disk 1, Virtual LBA 512-527 (read for parity), Virtual Disk 2, Virtual LBA 512-527 (read for parity) and Virtual Disk 3, Virtual LBA 512-527 (for parity write). The virtualization module 106 converts this virtual address information to Physical Disk LBA using the above virtualization process: Virtual Disk 0, Virtual LBA 512-527 = Physical Disk LBA 2048-2063, Virtual Disk 1, Virtual LBA 512-527 = Physical Disk LBA 2176-2191, Virtual Disk 2, Virtual LBA 512-527 = Physical Disk LBA 2304-2319, Virtual Disk 3, Virtual LBA 512-527 = Physical Disk LBA 2432-2447.

[0064] In this manner, these techniques may provide a LUN with parity fault tolerance, without hardware tolerance, and with adjacent LBA location without exposing any mapping. In one example, the virtual to physical mapping process may be optimized using hardware techniques.

[0065] Fig. 7 is an example block diagram showing a non-transitory, computer-readable medium that stores instructions for a computer system for backup operations in accordance with an example implementation. The non-transitory, computer-readable medium is generally referred to by the reference number 700 and may be included in components of system 100 as described herein. The non-transitory, computer-readable medium 700 may correspond to any typical storage device that stores computer-implemented

instructions, such as programming code or the like. For example, the non-transitory, computer-readable medium 700 may include one or more of a non-volatile memory, a volatile memory, and/or one or more storage devices. Examples of non-volatile memory include, but are not limited to, EEPROM and ROM. Examples of volatile memory include, but are not limited to, SRAM, and DRAM. Examples of storage devices include, but are not limited to, hard disk drives, compact disc drives, digital versatile disc drives, optical drives, and flash memory devices.

[0066] A processor 702 generally retrieves and executes the instructions stored in the non-transitory, computer-readable medium 700 to operate the components of system 100 in accordance with an example. In an example, the tangible, machine-readable medium 700 may be accessed by the processor 702 over a bus 704. A first region 706 of the non-transitory, computer-readable medium 700 may include virtualization module 106 functionality as described herein.

[0067] Although shown as contiguous blocks, the software components may be stored in any order or configuration. For example, if the non-transitory, computer-readable medium 700 is a hard drive, the software components may be stored in non-contiguous, or even overlapping, sectors.

What is claimed is:

1. A method comprising:
 - configuring a single physical storage disk to a virtual storage device that includes a plurality of virtual storage disks in response to receipt from a host computer a configuration command specifying storage characteristics of the virtual storage disks including number of virtual storage disks, virtual strip size, and fault tolerance of the virtual storage disks; and
 - in response to an access command from the host computer to access the virtual storage disks and specifying a virtual storage disk Logical Block Address (LBA), initiate execution of a virtualization process that includes converting the virtual storage disk LBA into a physical storage disk LBA based on a sum of factors that includes:
 - a first factor comprising a modulo of the virtual storage disk LBA and virtual strip size,
 - a second factor comprising number of virtual storage disks multiplied by a result of the virtual storage LBA minus the first factor, and
 - a third factor comprising a virtual storage disk number multiplied by the virtual strip size.
2. The method of claim 1, wherein the access command includes a read command from the host computer to read data from a virtual storage disk LBA of the virtual storage disks.
3. The method of claim 1, wherein the access command includes a write command from the host computer to write data to a virtual storage disk LBA of the virtual storage disks.
4. The method of claim 1, further comprising configuring the storage virtual disks in accordance with a fault tolerance level of Redundant Array Of Independent Disks (RAID-4) wherein the parity information is assigned to a parity virtual disk and, in response to a storage fault condition associated with

the storage virtual disks, initiating execution of a rebuild process that includes employing the fault tolerance configuration of the storage virtual disks that includes reading parity information stored in the parity virtual disk to rebuild data as a result of the storage fault condition.

5. The method of claim 1, further comprising configuring the storage virtual disks in accordance with a fault tolerance level of RAID-5 wherein the parity information is distributed across parity virtual disks and, in response to a storage fault condition associated with the storage virtual disks, initiating execution of a rebuild process that includes employing the fault tolerance configuration of the storage virtual disks that includes reading parity information stored across parity virtual disks to rebuild data as a result of the storage fault condition.

6. A computer comprising:
a virtualization module to:

configure a single physical storage disk to a virtual storage device that includes a plurality of virtual storage disks in response to receipt from a host computer a configuration command specifying storage characteristics of the virtual storage disks including number of virtual storage disks, virtual strip size, and fault tolerance of the virtual storage disks, and

in response to an access command from the host computer to access the virtual storage disks and specifying a virtual storage disk Logical Block Address (LBA), initiate execution of a virtualization process that includes conversion of the virtual storage disk LBA into a physical storage disk LBA based on a sum of factors that includes:

a first factor comprising a modulo of the virtual storage disk LBA and virtual strip size,

a second factor comprising number of virtual storage disks multiplied by a result of the virtual storage disk LBA minus the first factor, and

a third factor comprising a virtual storage disk number multiplied by the virtual strip size.

7. The computer of claim 6, wherein the access command includes a read command from the host computer to read data from a virtual storage disk LBA of the virtual storage disks.

8. The computer of claim 6, wherein the access command includes a write command from the host computer to write data to a virtual storage disk LBA of the virtual storage disks.

9. The computer of claim 6, wherein the virtualization module further comprising to configure the storage virtual disks in accordance with a fault tolerance level of Redundant Array Of Independent Disks (RAID-4) wherein the parity information is assigned to a parity virtual disk and, in response to a storage fault condition associated with the storage virtual disks, initiate execution of a rebuild process that includes employing the fault tolerance configuration of the storage virtual disks that includes reading parity information stored in the parity virtual disk to rebuild data as a result of the storage fault condition.

10. The computer of claim 6, wherein the virtualization module further comprising to configure the storage virtual disks in accordance with a fault tolerance level of RAID-5 wherein the parity information is distributed across parity virtual disks and, in response to a storage fault condition associated with the storage virtual disks, initiating execution of a rebuild process that includes employing the fault tolerance configuration of the storage virtual disks that includes reading parity information stored across parity virtual disks to rebuild data as a result of the storage fault condition.

11. An article comprising a non-transitory computer readable storage medium to store instructions that when executed by a computer to cause the computer to:

configure a single physical storage disk to a virtual storage device that includes a plurality of virtual storage disks in response to receipt from a host computer a configuration command specifying storage characteristics of the virtual storage disks including number of virtual storage disks, virtual strip size, and fault tolerance of the virtual storage disks; and

in response to an access command from the host computer to access the virtual storage disks and specifying a virtual storage disk Logical Block Address (LBA), initiate execution of a virtualization process that includes conversion of the virtual storage disk LBA into a physical storage disk LBA based on a sum of factors that includes:

a first factor comprising a modulo of the virtual storage disk LBA and virtual strip size,

a second factor comprising number of virtual storage disks multiplied by a result of the virtual storage disk LBA minus the first factor, and

a third factor comprising a virtual storage disk number multiplied by the virtual strip size.

12. The article of claim 11, wherein the access command includes a read command from the host computer to read data from a virtual storage disk LBA of the virtual storage disks.

13. The article of claim 11, wherein the access command includes a write command from the host computer to write data to a virtual storage disk LBA of the virtual storage disks.

14. The article of claim 11, further comprising instructions that if executed cause a computer to configure the storage virtual disks in accordance with a fault tolerance level of Redundant Array Of Independent Disks (RAID-4) wherein the parity information is assigned to a parity virtual disk and, in

response to a storage fault condition associated with the storage virtual disks, initiate execution of a rebuild process that includes employing the fault tolerance configuration of the storage virtual disks that includes reading parity information stored in the parity virtual disk to rebuild data as a result of the storage fault condition.

15. The article of claim 11, further comprising instructions that if executed cause a computer to configure the storage virtual disks in accordance with a fault tolerance level of RAID-5 wherein the parity information is distributed across parity virtual disks and, in response to a storage fault condition associated with the storage virtual disks, initiating execution of a rebuild process that includes employing the fault tolerance configuration of the storage virtual disks that includes reading parity information stored across parity virtual disks to rebuild data as a result of the storage fault condition.

1/7

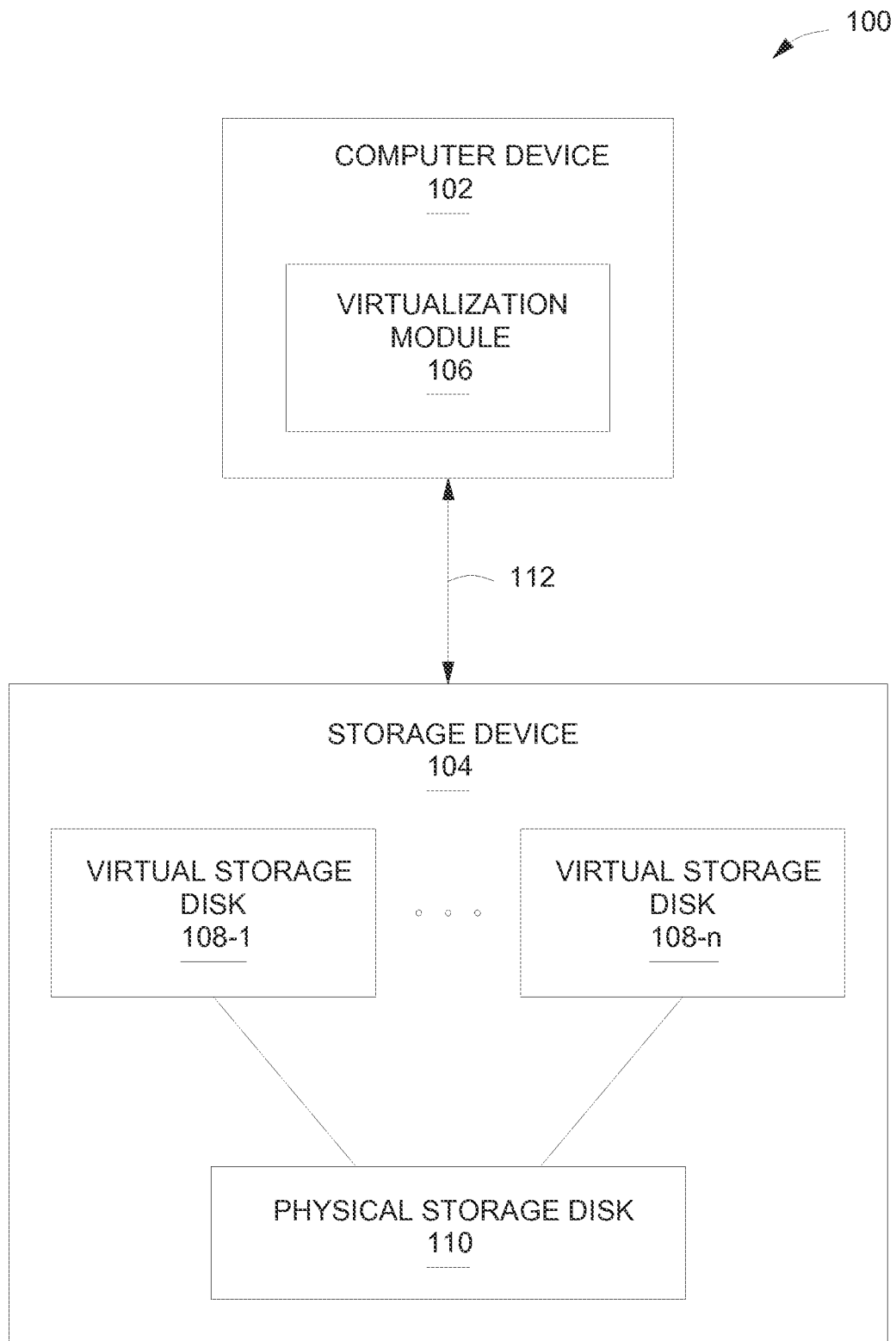


FIG. 1

2/7

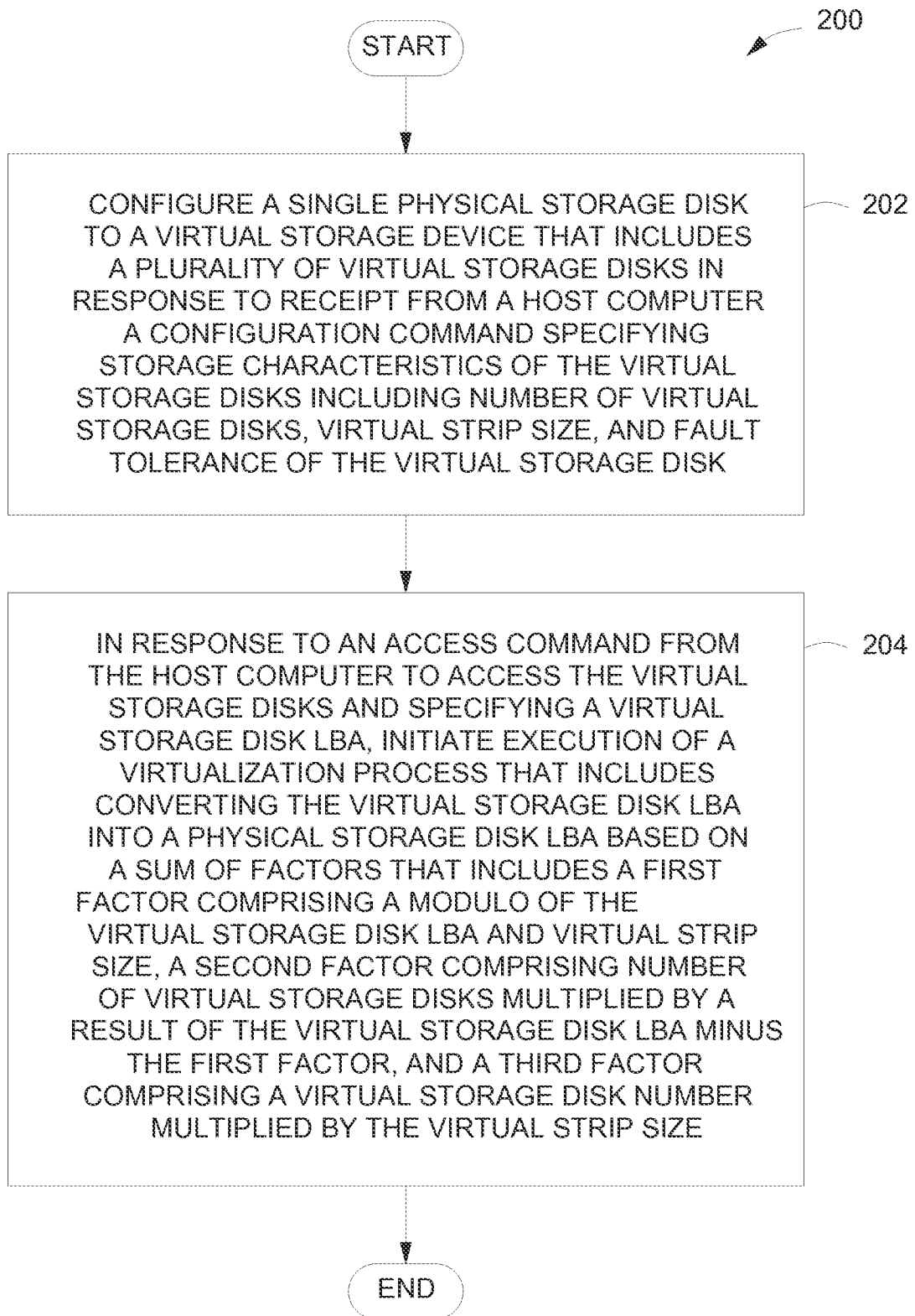


FIG. 2

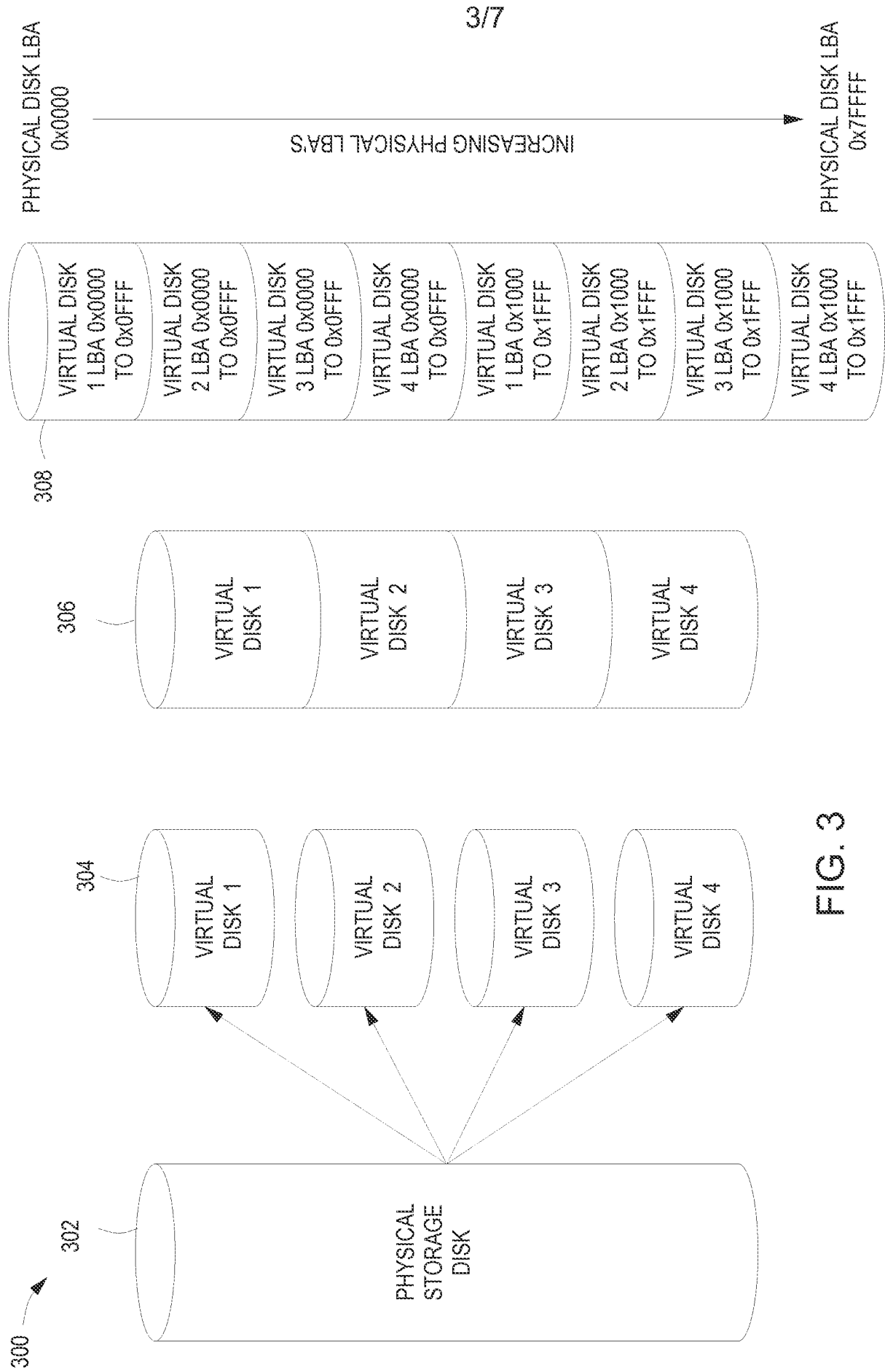


FIG. 3

4/7

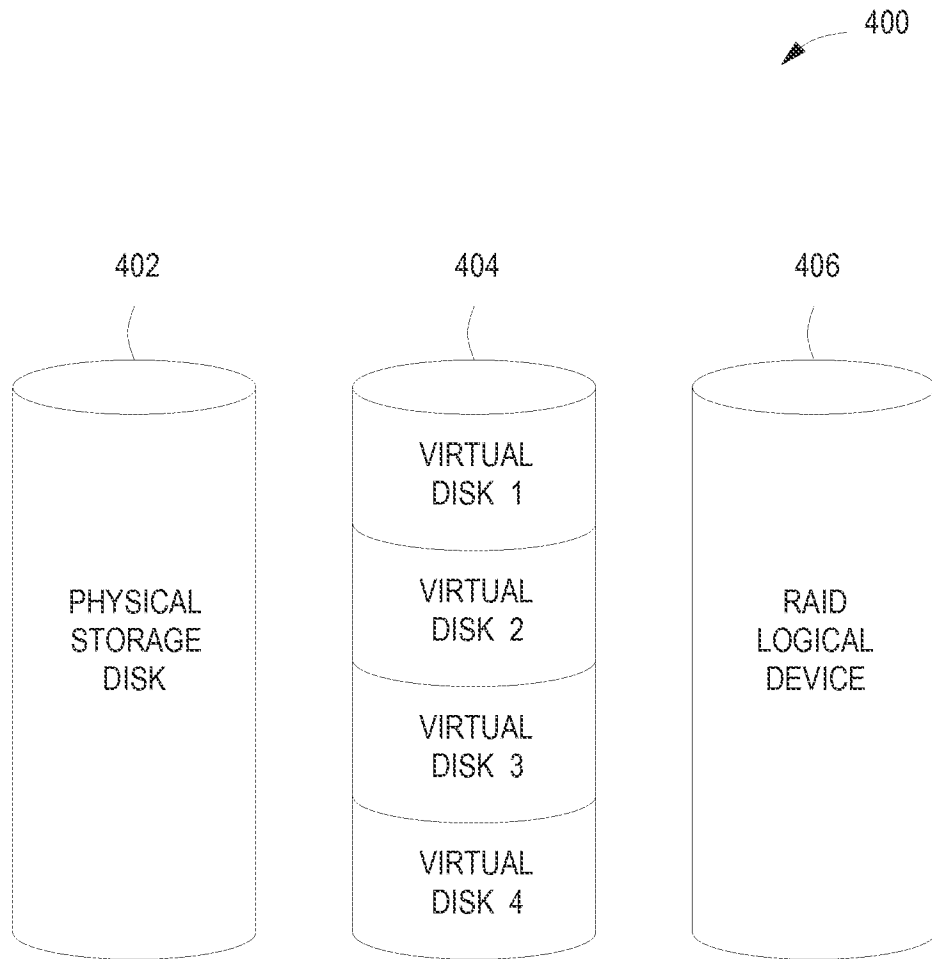


FIG. 4

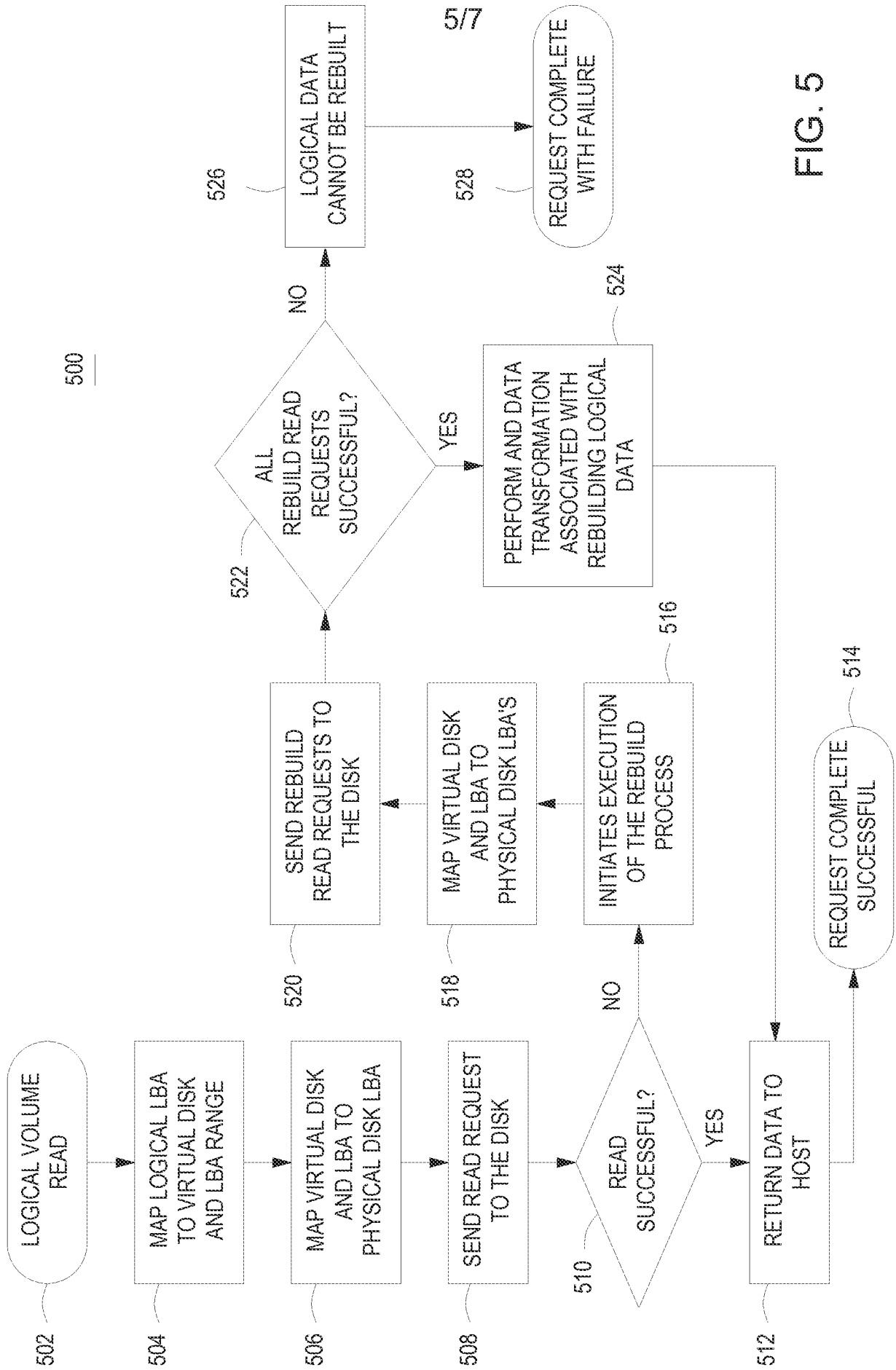


FIG. 5

6/7

600

RAID LBA	RAID HEX	MAPS TO VIRTUAL DISK LBA	VIRTUAL DRIVE	TYPE	VIRTUAL LBA	HEX	MAPS TO PHYSICAL DISK LBA	PHYSICAL DISK LBA	HEX
0	0	• • •	0	DATA	0	0	• • •	0	0
128	80	• • •	1	DATA	0	0	• • •	128	80
256	100	• • •	2	DATA	0	0	• • •	256	100
		• • •	3	PARITY	0	0	• • •	384	180
384	180	• • •	0	DATA	128	80	• • •	512	200
512	200	• • •	1	DATA	128	80	• • •	640	280
640	280	• • •	2	DATA	128	80	• • •	768	300
		• • •	3	PARITY	128	80	• • •	896	380
768	300	• • •	0	DATA	256	100	• • •	1024	400
896	380	• • •	1	DATA	256	100	• • •	1152	480
1024	400	• • •	2	DATA	256	100	• • •	1280	500
		• • •	3	PARITY	256	100	• • •	1408	580
1152	480	• • •	0	DATA	384	180	• • •	1536	600
1280	500	• • •	1	DATA	384	180	• • •	1664	680
1408	580	• • •	2	DATA	384	180	• • •	1792	700
		• • •	3	PARITY	384	180	• • •	1920	780
1536	600	• • •	0	DATA	512	200	• • •	2048	800
1664	680	• • •	1	DATA	512	200	• • •	2176	880
1792	700	• • •	2	DATA	512	200	• • •	2304	900
		• • •	3	PARITY	512	200	• • •	2432	980
1920	780	• • •	0	DATA	640	280	• • •	2560	A00
2048	800	• • •	1	DATA	640	280	• • •	2688	A80
2176	880	• • •	2	DATA	640	280	• • •	2816	B00
		• • •	3	PARITY	640	280	• • •	2944	B80

FIG. 6

7/7

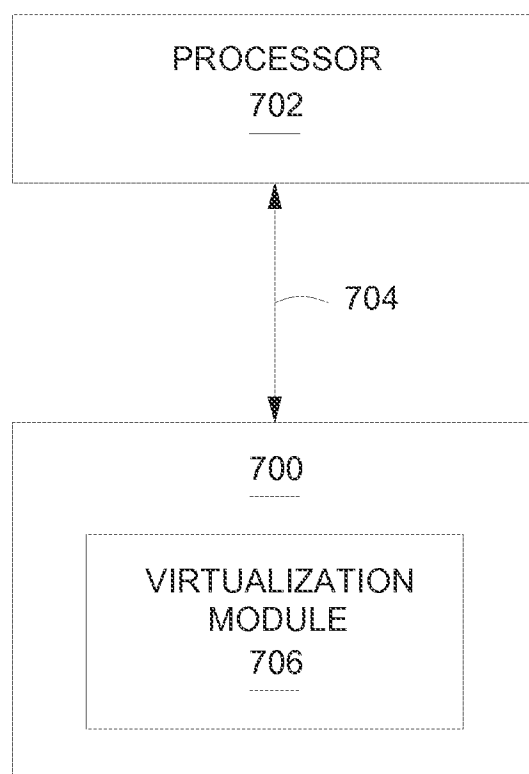


FIG. 7

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2015/012184**A. CLASSIFICATION OF SUBJECT MATTER****G06F 12/08(2006.01)i, G06F 12/06(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 12/08; G06F 15/16; G06F 12/00; G06F 9/455; H04L 12/28; G06F 12/06

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: single, physical, multiple, virtual, storage, logical block address (LBA), redundant array of independent disk (RAID)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2008-0163207 A1 (JOHN REUMANN et al.) 03 July 2008 See paragraphs [0002], [0007], [0031]; and figure 4.	1-15
A	US 2007-0233946 A1 (DOUGLAS E. JEWETT et al.) 04 October 2007 See paragraphs [0013], [0036], [0065]; and figure 1.	1-15
A	WO 2007-024740 A2 (SILICON IMAGE, INC.) 01 March 2007 See paragraphs [0004], [0018]; and figure 3.	1-15
A	US 2003-0167303 A1 (MIN ZHU et al.) 04 September 2003 See paragraphs [0009], [0034]; and figure 3.	1-15
A	US 2006-0272015 A1 (CHARLES WILLIAM FRANK et al.) 30 November 2006 See paragraphs [0018], [0074]; and figure 5.	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

14 September 2015 (14.09.2015)

Date of mailing of the international search report

05 October 2015 (05.10.2015)

Name and mailing address of the ISA/KR

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

Facsimile No. +82-42-472-7140

Authorized officer

BYUN, Sung Cheal

Telephone No. +82-42-481-8262



INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2015/012184

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2008-0163207 A1	03/07/2008	US 08381209 B2 US 08776059 B2 US 2013-061224 A1	19/02/2013 08/07/2014 07/03/2013
US 2007-0233946 A1	04/10/2007	AU 8644901 A EP 1317711 A1 JP 2004-506980 A US 07392291 B2 US 07428581 B2 US 08271606 B2 US 2002-0049825 A1 US 2008-0313187 A1 US 2008-0313301 A1 WO 02-15018 A1	25/02/2002 11/06/2003 04/03/2004 24/06/2008 23/09/2008 18/09/2012 25/04/2002 18/12/2008 18/12/2008 21/02/2002
WO 2007-024740 A2	01/03/2007	CN 101008934 A CN 101008934 B CN 101008934 C CN 101356506 A CN 101356506 B CN 103744790 A EP 1811396 A2 EP 1811396 A3 EP 1934751 A2 EP 1934751 A4 JP 05155861 B2 JP 05376756 B2 JP 05581360 B2 JP 2007-179549 A JP 2009-508192 A JP 2013-020635 A KR 10-1340176 B1 KR 10-1409927 B1 KR 10-1421480 B1 KR 10-2008-0038436 A TW 200801964 A TW I350970 B TW I352908 B US 07571269 B2 US 08595434 B2 US 2007-0180172 A1 US 2007-050538 A1 US 2014-059295 A1 WO 2007-024740 A3	01/08/2007 17/11/2010 01/08/2007 28/01/2009 08/01/2014 23/04/2014 25/07/2007 29/07/2009 25/06/2008 04/08/2010 06/03/2013 25/12/2013 27/08/2014 12/07/2007 26/02/2009 31/01/2013 10/12/2013 19/06/2014 23/07/2014 06/05/2008 01/01/2008 21/10/2011 21/11/2011 04/08/2009 26/11/2013 02/08/2007 01/03/2007 27/02/2014 21/12/2007
US 2003-0167303 A1	04/09/2003	US 07069298 B2 US 2006-0161651 A1	27/06/2006 20/07/2006
US 2006-0272015 A1	30/11/2006	TW I292530 B	11/01/2008

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2015/012184Patent document
cited in search reportPublication
datePatent family
member(s)Publication
date

US 07620981 B2	17/11/2009
US 08387132 B2	26/02/2013
US 08726363 B2	13/05/2014
US 2010-0095023 A1	15/04/2010
US 2012-195315 A1	02/08/2012