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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
4 December 2014 (04.12.2014)

WIPO | PCT

(10) International Publication Number
WO 2014/193770 A1

(51) International Patent Classification:
*G06F 3/06* (2006.01)

(21) International Application Number:
PCT/US2014/039480

(22) International Filing Date:
27 May 2014 (27.05.2014)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
13/904,989 29 May 2013 (29.05.2013) US

(71) Applicant: **MICROSOFT CORPORATION** [US/US]; One Microsoft Way, Redmond, Washington 98052-6399 (US).

(72) Inventors: **KUZNETSOV, Vyacheslav**; c/o Microsoft Corporation, LCA - International Patents (8/1172), One Microsoft Way, Redmond, Washington 98052-6399 (US). **SHANKAR, Vinod R.**; c/o Microsoft Corporation, LCA - International Patents (8/1172), One Microsoft Way, Redmond, Washington 98052-6399 (US). **D'AMATO, Andrea**; c/o Microsoft Corporation, LCA - International Patents (8/1172), One Microsoft Way, Redmond, Washington 98052-6399 (US). **DION, David Allen**; c/o Microsoft Corporation, LCA - International Patents (8/1172), One Microsoft Way, Redmond, Washington 98052-6399 (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,

*[Continued on next page]*

(54) Title: DISTRIBUTED STORAGE DEFENSE IN A CLUSTER

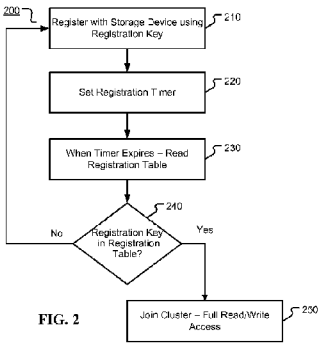


FIG. 2

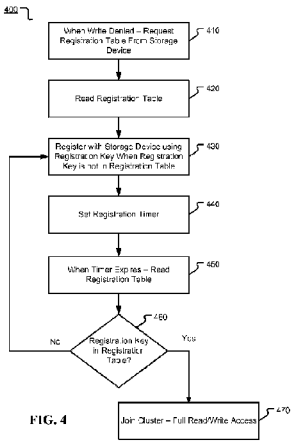


FIG. 4

(57) Abstract: Embodiments provide a method and system for enabling access to a storage device. Specifically, a node may request admittance to a cluster that has read and write access to a storage device. The node seeking access to the storage device must be first be approved by other nodes in the cluster. As part of the request, the node seeking access to the storage device sends a registration key to a storage device. Upon expiration of a registration timer, the node seeking access to the storage device receives a registration table from the storage device and determines whether its registration key is stored in the registration table. If the registration key is stored in the registration table the node has been accepted in the cluster and as a result, has been granted read and write access to the storage device.

WO 2014/193770 A1

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

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

**Published**:

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

— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))*

# DISTRIBUTED STORAGE DEFENSE IN A CLUSTER

## BACKGROUND

**[0001]** In typical shared storage situations in which a cluster of nodes has access to a storage device, at least one node in the cluster is connected to the storage device. As a result, the node that is connected to the storage device is responsible for handling the defense of the storage device. However, in situations where the cluster has access to multiple storage devices, a single node in the cluster may not be connected to each storage device. As a result, some of the storage devices may be unprotected.

**[0002]** It is with respect to these and other general considerations that embodiments have been made. Also, although relatively specific problems have been discussed, it should be understood that the embodiments should not be limited to solving the specific problems identified in the background.

## SUMMARY

**[0003]** This summary is provided to introduce a selection of concepts in a simplified form that are further described below in the Detail Description section. This summary is not intended to identify key features or essential features of the claimed subject matter, nor is it intended to be used as an aid in determining the scope of the claimed subject matter.

**[0004]** Embodiments of the present disclosure provide a method and system for enabling access to a storage device and for protecting one or more storage devices that are accessible by various nodes in a cluster. Specifically, one or more embodiments describe how a node may be admitted to a cluster and thereby obtain read and write access to a storage device that is connected to at least one node in a cluster of nodes. Additionally, one or more embodiments provide that nodes may monitor a registration table associated with a cluster and remove entries from non-recognized nodes. For those nodes that were removed, the node can attempt to re-register with the registration table to seek re-admission to the cluster.

**[0005]** As will be explained below, the node seeking access to the storage device uses cluster communication protocols to get into the cluster. Once admitted to the cluster, the node can be eligible to gain access to one or more storage devices utilized by the cluster. To gain access to the storage device, the node seeking access to the storage device sends a registration key to the storage device. After registering with the storage device, the node sets a registration timer. In embodiments, the registration timer is equivalent to a time period during which each node in the cluster has an opportunity to determine whether the

node seeking access to the storage device should be granted the requested access. Upon expiration of the registration timer, the node seeking access to the storage device receives a registration table from the storage device. Once the registration table is received, the node determines whether its registration key is stored in the registration table. If the registration key is stored in the registration table, the node is permitted access to the storage device. More specifically, the node is granted write access to the storage device.

**[0006]** Embodiments may be implemented as a computer process, a computing system or as an article of manufacture such as a computer program product or computer readable media. The computer program product may be computer storage media readable by a computer system and encoding a computer program of instructions for executing a computer process. The computer program product may also be a propagated signal on a carrier readable by a computing system and encoding a computer program of instructions for executing a computer process.

## BRIEF DESCRIPTION OF THE DRAWINGS

**[0007]** Non-limiting and non-exhaustive embodiments are described with reference to the following Figures in which:

**[0008]** Figure 1 illustrates a system in which a plurality of nodes in a cluster are connected to respective storage devices according to one or more embodiments of the present disclosure;

**[0009]** Figure 2 illustrates a method for requesting membership in a cluster according to one or more embodiments of the present disclosure;

**[0010]** Figure 3 illustrates a method for determining access to a storage device associated with a cluster of nodes according to one or more embodiments of the present disclosure;

**[0011]** Figure 4 illustrates a method for requesting re-admittance in a cluster according to one or more embodiments of the present disclosure;

**[0012]** Figure 5 is a block diagram illustrating how various nodes in a cluster may be connected to a physical storage device according to one or more embodiments of the present disclosure;

**[0013]** Figure 6 is a block diagram illustrating example physical components of a computing device that may be used with one or more embodiments of the present disclosure;

**[0014]** Figures 7A and 7B are simplified block diagrams of a mobile computing device that may be used with one or more embodiments of the present disclosure; and

**[0015]** Figure 8 is a simplified block diagram of a distributed computing system that may be used with one or more embodiments of the present disclosure.

## DETAILED DESCRIPTION

**[0016]** Various embodiments are described more fully below with reference to the accompanying drawings, which form a part hereof, and which show specific exemplary embodiments. However, embodiments may be implemented in many different forms and should not be construed as limited to the embodiments set forth herein; rather, these embodiments are provided so that this disclosure will be thorough and complete, and will fully convey the scope of the embodiments to those skilled in the art. Embodiments may be practiced as methods, systems or devices. Accordingly, embodiments may take the form of a hardware implementation, an entirely software implementation or an implementation combining software and hardware aspects. The following detailed description is, therefore, not to be taken in a limiting sense.

**[0017]** Figure 1 illustrates a system 100 in which a plurality of nodes in a cluster 102 are connected to respective storage devices according to one or more embodiments of the present disclosure. As shown in Figure 1, a cluster 102 may include a plurality of nodes 102A-102D. Although four nodes are shown, it is contemplated that the cluster 102 may have more than four nodes or fewer than four nodes. In certain embodiments, a node may be a computing device, such as, for example, a personal computer, tablet, laptop, smartphone, personal digital assistant and the like. In other embodiments, a node may be a server computing device.

**[0018]** Figure 1 also shows that each node in the cluster 102 is connected to one or more storage devices. In certain embodiments, the storage device may be a direct attached storage device (i.e., a storage device that is directly connected to a host system or device). It is also contemplated that the storage device may be accessed by a number of nodes in the cluster using one or more paths. For example, one or more nodes may be physically connected to the storage device while other nodes in the cluster may use a remote path to connect to the storage device. In addition, a single node may have multiple physical connections to various storage devices and one or more remote connections to the various storage devices. It is also contemplated that each node in the cluster may be able to view the activity and connections of each of the other nodes in the cluster. In sum, the system 100 may be asymmetrical in that some storage devices are available to some nodes while other storage devices are not available to those nodes.

**[0019]** For example, as shown in Figure 1, nodes 102A and 102B are connected to storage device 104, node 102C is connected to storage device 104 and storage device 106, and node 102D is connected to storage device 106 and storage device 108. In certain embodiments, the storage devices 104-108 comprise a storage pool. As there is not a single node in the cluster 102 that has access to each storage device in the storage pool, each node in the cluster 102 is responsible for running a defense algorithm to ensure that only nodes that are part of the cluster have read and write access to the storage devices. Thus, each node in the cluster 102 concurrently protects the storage devices that they are connected to in the storage pool.

**[0020]** Referring back to Figure 1, each of nodes 102A, 102B and 102C are connected to storage device 104. As discussed, each of the nodes may have a physical connection to the storage device 104 or a remote connection (i.e., a connection to the storage device 104 through a node having a physical connection to the storage device 104). Because nodes 102A, 102B and 102C are connected to storage device 104, each node has read and write access to the storage device 104. Further, each of nodes 102A, 102B and 102C in the cluster 102 can detect the presence of the other nodes in the cluster 102 and determine the activity of each of the other nodes in the cluster.

**[0021]** In embodiments, rights to a particular storage device are determined by persistent reservation. That is, the storage device, such as, for example, storage device 104, maintains a reservation of a particular node even when the storage device is offline or has been rebooted. For discussion purposes, a reservation of a particular node occurs when a node reserves a particular storage device and prevents another, unauthorized, node from accessing the storage device.

**[0022]** Referring back to the example above, each of nodes 102A, 102B and 102C has read and write access to the storage device 104 because each of the nodes 102A, 102B and 102C are part of the cluster 102. As will be explained in detail below, each node in the cluster 102 runs a defense algorithm at a time *t* to determine whether any other node in the cluster 102 has lost connectivity to either: (i) the other nodes in the cluster, or (ii) the storage device 104.

**[0023]** If, for example, node 102A loses connectivity to the nodes 102B and 102C or to the storage device 104, node 102B or node102C independently determine that node 102A should no longer have (at least) write access to the storage device 104 and thus be prohibited from accessing the storage device. When a connection has been lost, nodes 102B and 102C take the workload of node 102A and also take steps to ensure that node

102A can no longer write to the storage device 104 as permitting node 102A to write to the storage device may corrupt the data on the storage device 104. It is contemplated that although node 102A may have lost connectivity to nodes 102B and 102C, node 102A may still have connectivity to the storage device 104. Likewise, it is contemplated that if the node 102A lost connectivity to the storage device 104, node 102A may still be connected to node 102B and/or node 102C.

**[0024]** Returning to the example above, in order to prohibit node 102A from writing to the storage device 104, nodes 102B or 102C send a request to the storage device 104 to scratch node 102A from a node registration table. More specifically, a request is sent to the storage device to scratch a registration key associated with the node 102A from the node registration table. As a result, the storage device 104 will no longer accept write commands from a physical path or a remote path associated with node 102A. In certain embodiments, although node 102A will not have write access to the storage device 104, node 102A will still have read access to the storage device 104.

**[0025]** In certain embodiments, either of nodes 102B or 102C may send the scratch request independently from each other. In another embodiment, the nodes of a cluster may be required to reach a consensus on whether a particular node should be scratched. In still yet another embodiment, the node itself may determine that it should be removed from the cluster. For example, if node 102A determines that it has lost a connection to one of the other nodes or to the storage device 104, node 102A can remove one or more paths from itself to the storage device or instruct the storage device 104 to remove its registration key from the node registration table.

**[0026]** Referring to the node registration table, in certain embodiments, the node registration table is maintained by the storage device 104 and lists the nodes that have write access to the storage device 104. In embodiments, the node registration table includes a registration key for each node that has write access to the storage device. In certain embodiments, the registration key comprises a 64 bit integer having the following format: (i) a 32 bit hash of a cluster global unique identifier (that is the same for all nodes in the cluster); (ii) an 8 bit key revision; (iii) an 8 bit node number; and (iv) a 16 bit signature. Although a specific size and configuration of the registration key is set forth, it is contemplated that the registration key may have any number of bits and also have various configurations so long as the registration key is unique to each node.

**[0027]** As will be explained in greater detail below, once a node is scratched, the node may request re-admittance to the cluster. To request re-admittance, the node that was

scratched may send an updated registration key to the storage device. Once the node has re-registered with the storage device, each of the other nodes in the cluster make a determination as to whether the node should re-admitted to the cluster. Based on the determination of the nodes in the cluster, the node seeking re-admittance may be granted re-admittance or denied re-admittance. In embodiments, the nodes in the cluster may base their determination on any number of factors including, connectivity speed of the node seeking re-admittance, reliability of the node seeking re-admittance, access that the node seeking re-admittance has to other storage devices in a storage pool and the like.

**[0028]** Figure 2 illustrates a method 200 for requesting membership in a cluster according to one or more embodiments of the present disclosure. In certain embodiments, the method 200 may be used by a node to request membership in a cluster such as cluster 102 of Figure 1. As discussed above, once the node has been admitted to the cluster, the node may have read and write access to that particular storage device.

**[0029]** Specifically, a joining node may attempt to use cluster communication protocols to communicate with other nodes in order to gain admittance to an active cluster. In this case, once the joining node enters the active state, it will execute method 200 to gain access to the storage. If the joining node fails to communicate with other joining or active nodes via cluster protocols, and the joining node believes that there might not be an active cluster, the joining node may execute the method 200 to gain access to storage and thus become the first active node. For purposes of discussion, when the node requests access to and gains admittance to a cluster, the node is seen as an active node or entering the active state. For example, all nodes that are running the cluster communication protocol and are currently participating in cluster membership are considered active nodes. Additionally, nodes that have access to the one or more storage devices associated with a cluster are considered storage nodes. In embodiments, storage nodes are subsets of active node sets.

**[0030]** As shown in Figure 2, method 200 begins after a node has been admitted to a cluster using a cluster communication protocol. Once admitted to the cluster, one or more embodiments provide that the node seeks to access one or more storage devices associated with the cluster, such as, for example, storage device 104 (Figure 1) by registering 210 with the storage device using a registration key. As discussed, the registration key may comprise a 64 bit integer having various components discussed above.

**[0031]** Once the registration key has been sent to the storage device, flow proceeds to operation 220 in which a registration timer is set. In certain embodiments, this registration timer may be maintained by the node that is requesting access to the storage device or to a

storage pool. However, it is contemplated that the storage device or another node in the cluster could also maintain the registration timer. In embodiments, the length of the registration timer is equivalent to a time period *t*. In certain embodiments the time period *t* is 3 seconds. More specifically, the time period *t* is equivalent to the time it any other node in the cluster to perform a scrub that should occur every 3 seconds (taking into account any delays that may occur due to CPU loads, I/O latency and the like.

**[0032]** Upon expiration of the timer, flow proceeds to operation 230 in which the node registration table is read from the storage device. As discussed above, the node registration table is maintained by the storage device (or by at least one storage device in a storage pool) and contains a listing of each registration key associated with every node in the cluster.

**[0033]** Once the registration table is received and read, flow proceeds to operation 240 in which it is determined whether the node's registration key is contained in the node registration table. If the node's registration key is contained in the node registration table, each of the nodes in the cluster have run the defense algorithm and the storage device has accepted the node's request for access to the storage device. As a result, flow proceeds to operation 250 in which the node is permitted to access the storage device, and more specifically, to have write access to the storage device.

**[0034]** However, if it is determined in operation 240 that the requesting node's registration key is not in the node registration table, flow proceeds back to operation 210 and the node attempts to register with the storage device a second time. The method repeats and the node requesting access to the storage device again requests and reads the registration table to determine whether its registration key is stored in the node registration table.

**[0035]** Figure 3 illustrates a method 300 for determining access to a storage device associated with a cluster of nodes according to one or more embodiments of the present disclosure. In certain embodiments, method 300 is performed by each node in the cluster that is seen as an active node (i.e., all nodes that are running the cluster communication protocol and are currently participating in the cluster membership). Method 300 may also be performed by a node that is considered a storage node (i.e., any node that has access to one or more storage devices associated with the cluster).

**[0036]** Method 300 begins when a node "scrubs" 310 registration keys from the node registration table. Specifically, a node looks for other nodes that are not part of an active cluster. When a registration key is scrubbed from the disk registration table, the node

associated with the scrubbed registration key no longer has (at least) write access to a particular storage device or storage pool associated with the cluster. In certain embodiments, a registration key is scrubbed when one node in a cluster believes that another node in the cluster should no longer have write access to the particular storage device or storage pool. This may be the result of one of the nodes losing a connection to another node in the cluster, a node removing itself from the cluster or from a node losing a connection to the storage device. In situations in which a node has removed itself from the cluster, that node may send a request to the storage device indicating that its registration key should be removed from the node registration table. In another embodiment, one of the other nodes in the cluster may request that the node be scratched (i.e., removed) from the cluster during the scrubbing process. In certain embodiments, when a node is scratched from the cluster, the other nodes may be configured to prevent the commands from the removed node from reaching the storage device.

**[0037]** As shown in Figure 3, the scrubbing process has various sub-operations. The scrubbing process begins in sub-operation 311 in which a node reads the node registration table maintained by the storage device. As discussed above, the node registration table contains a list of all the registration keys associated with each node in the cluster.

**[0038]** Flow then proceeds to sub-operation 312 in which one or more nodes that do not have active membership in the cluster are scratched from the cluster. In embodiments, each node in the cluster is viewable by every other node in the cluster and may be connected to one or more storage devices either by a remote connection or a physical connection. As each node in the cluster has a view of every other node in the cluster, the node that is reading the node registration table can determine which nodes in the cluster have associated registration keys in the node registration table. Thus, a node scrubs a registration table received from a storage device. If a registration key is in the table but the node is not active, the node is scratched.

**[0039]** In certain embodiments, a node may not be scratched until multiple nodes in the cluster reach a similar determination (i.e., that the node being scratched does not have a registration key in the node registration table). In other embodiments, a node may be scratched when a single node reaches a determination that the node to be scratched does not have an associated registration key in the node registration table.

**[0040]** Once a request has been sent for the node to be scratched, the node running the algorithm determines 313 whether the node was successfully scratched. If the node was not successfully scratched, the node running the algorithm performs a self-check to

determine whether its own registration key is present in the node registration table. If its own registration key is not present in the node registration table, flow proceeds to sub-operation 314 and the node enters the "guest state" and seeks re-admittance to the cluster using one or more cluster protocols. If the node was successfully scratched, flow proceeds to operation 315 and the node reserves the storage device if the storage device has not already been reserved. That is, the node takes ownership of the storage device and then share access to this storage device with other nodes that are in the same cluster.

**[0041]** When the registration keys in the node registration table have been scrubbed, flow proceeds to operation 320 in which a second timer is set. In embodiments, the second timer is proportional to the registration timer discussed in Figure 2. For example, the scrubbing algorithm runs in parallel on all the nodes (e.g., each node scrubs keys every 3 seconds independently of all the other nodes. In certain embodiments, the scrubbing algorithm runs in parallel because one node may not be able to reliably tell if another node is also scrubbing the registration table or if the one of the nodes has lost its connection to the storage and therefore can't scrub the registration table.

**[0042]** When a node joins the cluster, the node registers with the cluster and waits to get admitted. If the node is admitted, one embodiment provides that the node scrubs and surfaces the disk as soon as it can. The three second scrubbing timer is set and the node can validate its own registration on the storage.

**[0043]** Upon expiration of the timer, flow proceeds to operation 330 in which the node surfaces one or more paths to the storage device. That is, the node may determine which paths, either physical or remote, need to be connected to other nodes and/or storage devices in the cluster. For example, referring to Figure 1, if node 102B joined the cluster 102, and node 102A established or has a physical connection to the storage device 104, the physical path is advertised to other nodes in the cluster and the other nodes in the cluster, for example, 102D, may establish a remote path to node 102A and take advantage of the physical connection between node 102A and the storage device 104.

**[0044]** Figure 4 illustrates a method 400 for requesting re-admittance to a cluster according to one or more embodiments of the present disclosure. In certain embodiments, a node seeking re-admittance to the cluster may have removed itself from the cluster or may have been removed by another node in the cluster as discussed above with respect to Figure 3. In embodiments, method 400 begins when a node in a cluster sends a write command to a storage device and is notified that the write was not successful. If a write was unsuccessful, the node may request that the command be sent down a different path.

Additionally, or alternatively, the node may request that all other pending commands that are in progress be canceled. Upon receipt of the notification, the node requests 410 the node registration table from the storage device. As discussed above, the node registration table is maintained by the storage device and includes various registration keys that are associated with respective nodes in the cluster.

**[0045]** When the node registration table has been received by the requesting node, the node reads 420 the registration table to determine whether its own registration key is contained in the registration table. When it is determined that the node's registration key is not in the node registration table, the node registers 430 with the storage device using another registration key. As discussed above, the registration key may be a 64 bit integer having a 32 bit hash of a cluster global unique identifier, an 8 bit key revision, an 8 bit node number and a 16 bit signature. In certain embodiments, as the node has to re-register, the node may generate a new registration key where at least a portion of the registration key is incremented or changed. Thus, the node, the storage device or other nodes in the cluster may track the number of times a new path has to be set up for the node. In some embodiments, the number of times the node has requested re-admittance may affect the node obtaining re-admittance to the cluster. In addition, the change to the registration key helps ensure that write commands coming from the node and/or from paths associated with the node (when the node is re-admitted to the cluster) are distinguishable from stale write commands from the node that may still be waiting to be written to the storage device.

**[0046]** Once the node has registered its newly generated registration key with the storage device, a registration timer is set 440. As previously discussed, the length of the registration timer is equivalent to about 2.5 times the time period *t*. That is, the time period *t* is equivalent to the time it takes each node in the cluster to run the defense algorithm disclosed herein.

**[0047]** Upon expiration of the timer, flow proceeds to operation 450 and the node seeking write access to the storage device reads the registration table maintained by the storage device. If it is determined 460 that the newly generated registration key is in the node registration table, the node is granted write access to the storage device.

**[0048]** However, if it is determined in operation 460 that the node's newly generated registration key is not in the node registration table, flow proceeds back to operation 430 node re-registers and the registration time is reset. In certain embodiments, if the node has been rejected a determined number of times, the node will no longer seek write access to

the storage device. In certain embodiments, the node may again seek admittance to the cluster or access to the storage device after a determined period of time has elapsed.

**[0049]** Figure 5 is a block diagram illustrating a system 500 by which two nodes in a cluster may access a physical disk using remote paths and/or physical paths according to one or more embodiments of the present disclosure. As discussed above with reference to Figures 1 through 4, various nodes in a cluster may be connected to one or more storage devices in a storage pool. Those connections (or spindles) can be physical connections or remote connections. As will be explained below, a node may utilize various paths to connect to one or more storage devices.

**[0050]** In the exemplary embodiment shown in Figure 5, the cluster may have two nodes, Node A 510 and Node B 520. Each node may have a physical connection to a physical Disk 530. Although only two nodes are shown, it is contemplated that a cluster may consist of more than two nodes. Additionally, it is contemplated that each node may have a physical connection to one or more storage devices. As shown in Figure 5, a node may have access to a physical disk through a number of different paths. For example, Node A 510 has a physical path to the physical disk 530 and also has a remote path to the physical disk 530 through the target 524 of Node B 520. In certain embodiments, a single node may have multiple physical paths to the same disk. In such embodiments, the node will expose all these physical paths to all of the other nodes in the cluster.

**[0051]** As also shown in Figure 5, Node A has access to a virtual disk 511, a multi path object 512 that aggregates multiple physical paths and remote paths to a physical disk 530, a remote path object 513that instantiates one or more remote paths to the physical disk 530 through another node, such as, for example, Node B 520, a target object 514 which serves to advertise one or more physical connections between Node A 510 and the physical disk 530 and which allows other nodes in the cluster to gain connectivity (e.g., through a remote path) to the physical disk 530 via Node A 510, and a physical path object 515 instantiates one or more physical connections or paths to the physical disk 530.

**[0052]** Likewise, Node B 520 has access to a virtual disk 521, a multi-path object 522 that aggregates multiple physical paths and remote paths from Node B 520 to the physical disk 530, a remote path object 523that 513that instantiates one or more remote paths to the physical disk 530 through another node, such as, for example, Node A 510, a target object 524 that advertises the physical path to the physical disk 530 to other nodes in the cluster, and a physical path object 525 that instantiates one or more physical connections or paths to the physical disk 530. Although one remote path is shown for both Node A 510 and

Node B 520, it is contemplated that a single node may have multiple remote paths. It is also contemplated that each node may have multiple physical paths.

**[0053]** In embodiments, the preferred path through which various commands are sent to the physical disk 530 is the physical path. For example, when a new disk is detected, one or more nodes of the cluster will register or reserve the disk. As discussed above, this process includes running the defense algorithm discussed above with respect to Figure 2 and subsequently creating a physical path from the node to the physical disk. In certain embodiments, each physical path, or each new instance of a physical path has a registration key that includes a cluster identifier, node identifier and a reincarnation identifier (a unique number for the physical path that is incremented each time a physical path is instantiated). In embodiments, the registration key of the path may be equivalent to the registration key of the associated node. Once the physical connection is established and the node registers with the disk using the registration key, the multi path object and the target object of the node is notified of the newly established physical path. That information is then transmitted to other nodes in the cluster so the other nodes may establish a remote path via the target of the node having the physical connection to the physical disk.

**[0054]** As discussed above, it is contemplated that one or more nodes may lose a connection to one or more other nodes in the cluster or to the physical disk. In such an event, one of the connected nodes in the cluster will request that one or more paths from the disconnected node be removed and also request that the storage device stop taking write request from one or more paths (e.g., physical paths or remote paths) associated with the disconnected nodes. Likewise, the targets associated with each node that have a remote connection to the disconnected node may also stop receiving commands from the disconnected node. Such actions prevent the disconnected node from sending additional and/or duplicate writes to the storage device that may be on the wire but not yet completed. That is, removing registration keys from the storage device and blocking write command through the target helps to ensure that disconnected nodes can't write to a disk using physical paths or remote paths.

**[0055]** For example, referring to Figure 5, Node A 510 may have lost its physical connection via its physical path object 515 to the physical disk 530. However, as shown, Node A 510 also has a remote path 513 to the physical disk 530 through the target object 524 of Node B 520. Additionally, prior to losing the connection to the physical disk 530, Node A 510 may have sent write commands to the physical disk 530 that have not yet

completed. As Node A 510 has lost connectivity to the physical disk 530, Node A 510 may have no knowledge of whether its write commands were executed or rejected.

**[0056]** However, if Node A 510 were allowed to immediately reconnect with the physical disk 530 and either re-submit the commands that may or may not have been executed, or if Node A 510 were permitted to send additional commands to the physical disk 530 (which may be out of order due to Node A 510 losing its connection), such actions may cause the data in the physical disk 530 to become corrupt. To prevent such corruption, Node B 520 preempts the physical path and/or all remote paths associated with Node A 510.

**[0057]** Once the physical and/or remote paths associated with Node A 510 are preempted, the physical disk 530 will not accept commands from the paths associated with Node A 510. As each path for each node has an associated identifier, the physical disk 530 can determine which commands are associated with the node based on the identifier of each of the paths. In certain embodiments, physical disk 530 distinguishes between physical paths. Thus, if the I/O came through a remote path, from the physical disk's perspective, the I/O would appear as if it came from the node that hosts the target to which the remote path is connected. In short, remote path I/O fencing is performed at the target while physical path I/O fencing is performed at the physical disk 530 level.

**[0058]** To further the example, each node in the cluster can see each spindle or path of every other node. Thus, Node B 520 can see that Node A 510 lost a connection to the physical disk 530. As a result, Node B 520 will discard remote path 523. However, if Node A 510 cannot communicate to the other nodes in the cluster, Node B 520 may instruct the physical disk 530 to reject write commands from Node A 510.

**[0059]** In certain embodiments, once the physical disk 530 starts rejecting the commands from the physical path of Node A 510, the multi path object 512 of Node A 510 detects the rejection of the commands. As a result, the multi path object 512 may query all other existing physical paths to determine if any are valid. If one physical path is still valid, the valid physical path is added to the multi path object 512. However, if there are no valid physical path objects, a new multi path object is created and the physical path object 515 instantiates a new physical path with a new registration key. When generated, the new physical path and its associated registration key will have a new reincarnation identifier that sets it apart from the now defunct identifier associated with the old physical path.

**[0060]** Additionally, when the node, such as, for example, Node A 510 requests re-admittance to the cluster using the new identifier, the new identifier is advertised to other nodes in the cluster. Thus, the remote path objects of other nodes can use the new identifier of the physical path of Node A 510 to connect to the physical disk 530. As discussed above, as the physical disk 530 knows not to accept commands from the old physical path, the physical disk accepts commands from the new physical path and it associated identifier when Node A 510 seeks re-admittance to the cluster through the methods described above with references to Figures 2-4.

**[0061]** Referring back to Figure 5, if Node A 510 and Node B 520 lose connectivity to each other, there may be data in a cache of an application that has not been written to the virtual disk or there may be data in the virtual disk that has not been written to the physical disk. Accordingly embodiments provide that all remaining commands on a path from a disconnected node be drained and that no further commands be accepted from the paths associated with the disconnected node.

**[0062]** The embodiments and functionalities described herein may operate via a multitude of computing systems including, without limitation, desktop computer systems, wired and wireless computing systems, mobile computing systems (e.g., mobile telephones, netbooks, tablet or slate type computers, notebook computers, and laptop computers), hand-held devices, multiprocessor systems, microprocessor-based or programmable consumer electronics, minicomputers, and mainframe computers.

**[0063]** In addition, the embodiments and functionalities described herein may operate over distributed systems (e.g., cloud-based computing systems), where application functionality, memory, data storage and retrieval and various processing functions may be operated remotely from each other over a distributed computing network, such as the Internet or an intranet. User interfaces and information of various types may be displayed via on-board computing device displays or via remote display units associated with one or more computing devices. For example user interfaces and information of various types may be displayed and interacted with on a wall surface onto which user interfaces and information of various types are projected. Interaction with the multitude of computing systems with which embodiments of the invention may be practiced include, keystroke entry, touch screen entry, voice or other audio entry, gesture entry where an associated computing device is equipped with detection (e.g., camera) functionality for capturing and interpreting user gestures for controlling the functionality of the computing device, and the like.

**[0064]** Figures 6-8 and the associated descriptions provide a discussion of a variety of operating environments in which embodiments of the invention may be practiced. However, the devices and systems illustrated and discussed with respect to Figures 6-8 are for purposes of example and illustration and are not limiting of a vast number of computing device configurations that may be utilized for practicing embodiments of the invention, described herein.

**[0065]** Figure 6 is a block diagram illustrating physical components (i.e., hardware) of a computing device 105 with which embodiments of the invention may be practiced. The computing device components described below may be suitable for the nodes or computing devices described above. In a basic configuration, the computing device 105 may include at least one processing unit 602 and a system memory 604. Depending on the configuration and type of computing device, the system memory 604 may comprise, but is not limited to, volatile storage (e.g., random access memory), non-volatile storage (e.g., read-only memory), flash memory, or any combination of such memories. The system memory 604 may include an operating system 605 and one or more program modules 606 suitable for running software various applications 620. The operating system 605, for example, may be suitable for controlling the operation of the computing device 105. Furthermore, embodiments of the invention may be practiced in conjunction with a graphics library, other operating systems, or any other application program and is not limited to any particular application or system. This basic configuration is illustrated in Figure 6 by those components within a dashed line 608. The computing device 105 may have additional features or functionality. For example, the computing device 105 may also include additional data storage devices (removable and/or non-removable) such as, for example, magnetic disks, optical disks, or tape. Such additional storage is illustrated in Figure 6 by a removable storage device 609 and a non-removable storage device 610.

**[0066]** As stated above, a number of program modules and data files may be stored in the system memory 604. While executing on the processing unit 602, the program modules 606 may perform processes including, but not limited to, one or more of the stages of the methods illustrated in Figures 1-4. Other program modules that may be used in accordance with embodiments of the present invention may include electronic mail and contacts applications, word processing applications, spreadsheet applications, database applications, slide presentation applications, drawing or computer-aided application programs, etc.

**[0067]** Furthermore, embodiments of the invention may be practiced in an electrical circuit comprising discrete electronic elements, packaged or integrated electronic chips containing logic gates, a circuit utilizing a microprocessor, or on a single chip containing electronic elements or microprocessors. For example, embodiments of the invention may be practiced via a system-on-a-chip (SOC) where each or many of the components illustrated in Figure 6 may be integrated onto a single integrated circuit. Such an SOC device may include one or more processing units, graphics units, communications units, system virtualization units and various application functionality all of which are integrated (or "burned") onto the chip substrate as a single integrated circuit. When operating via an SOC, the functionality, described herein may be operated via application-specific logic integrated with other components of the computing device 105 on the single integrated circuit (chip). Embodiments of the invention may also be practiced using other technologies capable of performing logical operations such as, for example, AND, OR, and NOT, including but not limited to mechanical, optical, fluidic, and quantum technologies. In addition, embodiments of the invention may be practiced within a general purpose computer or in any other circuits or systems.

**[0068]** The computing device 105 may also have one or more input device(s) 612 such as a keyboard, a mouse, a pen, a sound input device, a touch input device, etc. The output device(s) 614 such as a display, speakers, a printer, etc. may also be included. The aforementioned devices are examples and others may be used. The computing device 104 may include one or more communication connections 616 allowing communications with other computing devices 618. Examples of suitable communication connections 616 include, but are not limited to, RF transmitter, receiver, and/or transceiver circuitry; universal serial bus (USB), parallel, and/or serial ports.

**[0069]** The term computer readable media as used herein may include computer storage media. Computer storage media may include volatile and nonvolatile, removable and non-removable media implemented in any method or technology for storage of information, such as computer readable instructions, data structures, or program modules. The system memory 604, the removable storage device 609, and the non-removable storage device 610 are all computer storage media examples (i.e., memory storage.) Computer storage media may include RAM, ROM, electrically erasable read-only memory (EEPROM), flash memory or other memory technology, CD-ROM, digital versatile disks (DVD) or other optical storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other article of manufacture which can be used to store

information and which can be accessed by the computing device 105. Any such computer storage media may be part of the computing device 105. Computer storage media does not include a carrier wave or other propagated or modulated data signal.

**[0070]** Communication media may be embodied by computer readable instructions, data structures, program modules, or other data in a modulated data signal, such as a carrier wave or other transport mechanism, and includes any information delivery media. The term "modulated data signal" may describe a signal that has one or more characteristics set or changed in such a manner as to encode information in the signal. By way of example, and not limitation, communication media may include wired media such as a wired network or direct-wired connection, and wireless media such as acoustic, radio frequency (RF), infrared, and other wireless media.

**[0071]** Figures 7A and 7B illustrate a mobile computing device 700, for example, a mobile telephone, a smart phone, a tablet personal computer, a laptop computer, and the like, with which embodiments of the invention may be practiced. With reference to Figure 7A, one embodiment of a mobile computing device 700 for implementing the embodiments is illustrated. In a basic configuration, the mobile computing device 700 is a handheld computer having both input elements and output elements. The mobile computing device 700 typically includes a display 705 and one or more input buttons 710 that allow the user to enter information into the mobile computing device 700. The display 705 of the mobile computing device 700 may also function as an input device (e.g., a touch screen display). If included, an optional side input element 715 allows further user input. The side input element 715 may be a rotary switch, a button, or any other type of manual input element. In alternative embodiments, mobile computing device 700 may incorporate more or less input elements. For example, the display 705 may not be a touch screen in some embodiments. In yet another alternative embodiment, the mobile computing device 700 is a portable phone system, such as a cellular phone. The mobile computing device 700 may also include an optional keypad 735. Optional keypad 735 may be a physical keypad or a "soft" keypad generated on the touch screen display. In various embodiments, the output elements include the display 705 for showing a graphical user interface (GUI), a visual indicator 720 (e.g., a light emitting diode), and/or an audio transducer 725 (e.g., a speaker). In some embodiments, the mobile computing device 700 incorporates a vibration transducer for providing the user with tactile feedback. In yet another embodiment, the mobile computing device 700 incorporates input and/or output ports, such as an audio input (e.g., a microphone jack), an audio output (e.g., a headphone

jack), and a video output (e.g., a HDMI port) for sending signals to or receiving signals from an external device.

**[0072]** Figure 7B is a block diagram illustrating the architecture of one embodiment of a mobile computing device. That is, the mobile computing device 700 can incorporate a system (i.e., an architecture) 702 to implement some embodiments. In one embodiment, the system 702 is implemented as a "smart phone" capable of running one or more applications (e.g., browser, e-mail, calendaring, contact managers, messaging clients, games, and media clients/players). In some embodiments, the system 702 is integrated as a computing device, such as an integrated personal digital assistant (PDA) and wireless phone.

**[0073]** One or more application programs 766 may be loaded into the memory 762 and run on or in association with the operating system 764. Examples of the application programs include phone dialer programs, e-mail programs, personal information management (PIM) programs, word processing programs, spreadsheet programs, Internet browser programs, messaging programs, and so forth. The system 702 also includes a non-volatile storage area 768 within the memory 762. The non-volatile storage area 768 may be used to store persistent information that should not be lost if the system 702 is powered down. The application programs 766 may use and store information in the non-volatile storage area 768, such as e-mail or other messages used by an e-mail application, and the like. A synchronization application (not shown) also resides on the system 702 and is programmed to interact with a corresponding synchronization application resident on a host computer to keep the information stored in the non-volatile storage area 768 synchronized with corresponding information stored at the host computer. As should be appreciated, other applications may be loaded into the memory 762 and run on the mobile computing device 700.

**[0074]** The system 702 has a power supply 770, which may be implemented as one or more batteries. The power supply 770 might further include an external power source, such as an AC adapter or a powered docking cradle that supplements or recharges the batteries.

**[0075]** The system 702 may also include a radio 772 that performs the function of transmitting and receiving radio frequency communications. The radio 772 facilitates wireless connectivity between the system 702 and the "outside world", via a communications carrier or service provider. Transmissions to and from the radio 772 are conducted under control of the operating system 764. In other words, communications

received by the radio 772 may be disseminated to the application programs 766 via the operating system 764, and vice versa.

**[0076]** The visual indicator 720 may be used to provide visual notifications, and/or an audio interface 774 may be used for producing audible notifications via the audio transducer 725. In the illustrated embodiment, the visual indicator 720 is a light emitting diode (LED) and the audio transducer 725 is a speaker. These devices may be directly coupled to the power supply 770 so that when activated, they remain on for a duration dictated by the notification mechanism even though the processor 760 and other components might shut down for conserving battery power. The LED may be programmed to remain on indefinitely until the user takes action to indicate the powered-on status of the device. The audio interface 774 is used to provide audible signals to and receive audible signals from the user. For example, in addition to being coupled to the audio transducer 725, the audio interface 774 may also be coupled to a microphone to receive audible input, such as to facilitate a telephone conversation. In accordance with embodiments of the present invention, the microphone may also serve as an audio sensor to facilitate control of notifications, as will be described below. The system 702 may further include a video interface 776 that enables an operation of an on-board camera 730 to record still images, video stream, and the like.

**[0077]** A mobile computing device 700 implementing the system 702 may have additional features or functionality. For example, the mobile computing device 700 may also include additional data storage devices (removable and/or non-removable) such as, magnetic disks, optical disks, or tape. Such additional storage is illustrated in Figure 7B by the non-volatile storage area 768.

**[0078]** Data/information generated or captured by the mobile computing device 700 and stored via the system 702 may be stored locally on the mobile computing device 700, as described above, or the data may be stored on any number of storage media that may be accessed by the device via the radio 772 or via a wired connection between the mobile computing device 700 and a separate computing device associated with the mobile computing device 700, for example, a server computer in a distributed computing network, such as the Internet. As should be appreciated such data/information may be accessed via the mobile computing device 700 via the radio 772 or via a distributed computing network. Similarly, such data/information may be readily transferred between computing devices for storage and use according to well-known data/information transfer and storage means, including electronic mail and collaborative data/information sharing systems.

**[0079]** Figure 8 illustrates one embodiment of the architecture of a system for providing and maintaining membership in a cluster as described above. For example, the node registration table, identifiers, and the various paths between nodes and between nodes and the physical disk may be stored in different communication channels or other storage types. For example, various identifiers may be stored using a directory service 822, a web portal 824, a mailbox service 826, an instant messaging store 828, or a social networking site 830. A server 820 may provide data and/or connection types to one or more other servers or nodes in the cluster. As one example, the server 820 may be a web server that provides data over the web to clients through a network 815. By way of example, the client computing device may be implemented as the computing device 105 and embodied in a personal computer, a tablet computing device 610 and/or a mobile computing device 700 (e.g., a smart phone). Any of these embodiments of the client computing device 105, 610, 700 may obtain content from the store 816.

**[0080]** Embodiments of the present invention, for example, are described above with reference to block diagrams and/or operational illustrations of methods, systems, and computer program products according to embodiments of the invention. The functions/acts noted in the blocks may occur out of the order as shown in any flowchart. For example, two blocks shown in succession may in fact be executed substantially concurrently or the blocks may sometimes be executed in the reverse order, depending upon the functionality/acts involved.

**[0081]** The description and illustration of one or more embodiments provided in this application are not intended to limit or restrict the scope of the invention as claimed in any way. The embodiments, examples, and details provided in this application are considered sufficient to convey possession and enable others to make and use the best mode of claimed invention. The claimed invention should not be construed as being limited to any embodiment, example, or detail provided in this application. Regardless of whether shown and described in combination or separately, the various features (both structural and methodological) are intended to be selectively included or omitted to produce an embodiment with a particular set of features. Having been provided with the description and illustration of the present application, one skilled in the art may envision variations, modifications, and alternate embodiments falling within the spirit of the broader aspects of the general inventive concept embodied in this application that do not depart from the broader scope of the claimed invention.

## CLAIMS

1. A method for enabling access to a storage device, the method comprising:

sending a registration key to a storage device, wherein the storage device is connected to at least one node in a cluster of nodes;

setting a first registration timer;

upon expiration of the first registration timer, receiving a registration table from the storage device;

determining whether the registration key is stored in the registration table; and

when the registration key is stored in the registration table, joining the cluster, wherein joining the cluster enables write access to the storage device.

2. The method of claim 1, further comprising scrubbing the registration table after joining the cluster.

3. The method of claim 2, wherein scrubbing the registration table comprises:

receiving the registration table from the storage device;

reading the registration table;

scratching one or more registration keys from the registration table, wherein each of the one or more registration keys is associated with a node in the cluster.

4. The method of claim 1, wherein the registration key comprises a cluster identifier and a node identifier.

5. A computer-readable storage device encoding computer executable instructions which, when executed by one or more processors, performs a method for enabling access to a storage device, the method comprising:

sending a registration key to a storage device, wherein the storage device is connected to at least one node in a cluster of nodes;

setting a first registration timer;

upon expiration of the first registration timer, receiving a registration table from the storage device;

determining whether the registration key is stored in the registration table; and

when the registration key is stored in the registration table, joining the cluster, wherein joining the cluster enables write access to the storage device.

6. The computer-readable storage device of claim 5, further comprising instructions for scrubbing the registration table after joining the cluster.

7. The computer-readable storage device of claim 6, wherein scrubbing the registration table comprises:

receiving the registration table from the storage device;

reading the registration table;

scratching one or more registration keys from the registration table, wherein each of the one or more registration keys is associated with a node in the cluster.

8. The computer-readable storage device of claim 5, wherein the registration key comprises a cluster identifier and a node identifier.

9. A method for enabling access to a storage device, the method comprising:

sending, from a node in a cluster, a write command to the storage device, wherein the node has an associated registration key; and

upon receiving a notification that the write command was rejected:

requesting a registration table from the storage device;

determining whether the registration key associated with the node is present in the registration table; and

when the registration key associated with the node is not present in the registration table:

sending a new registration key to the storage device;

setting a first registration timer;

upon expiration of the first registration timer, receiving the registration table from the storage device;

determining whether the new registration key is stored in the registration table; and

when the new registration key is stored in the registration table, joining the cluster, wherein joining the cluster enables the node to write to the storage device.

10. The method of claim 9, further comprising scrubbing the registration table after joining the cluster.

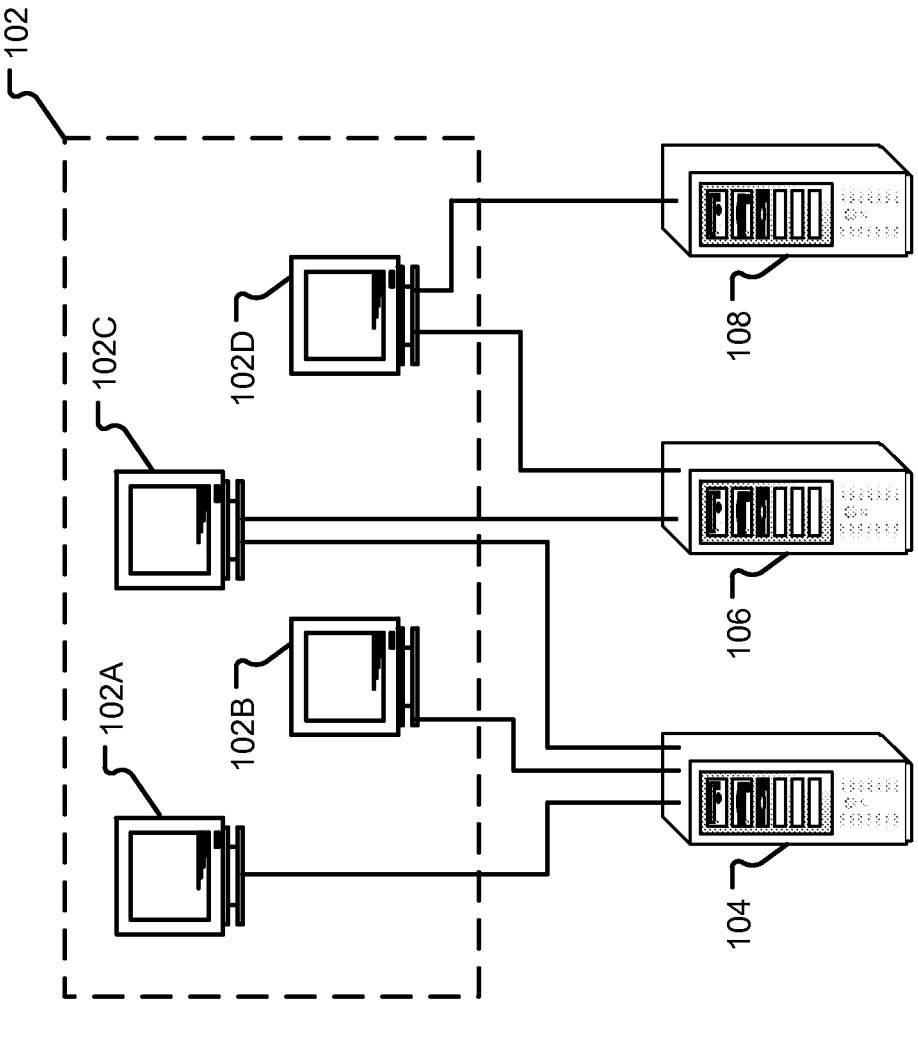


FIG. 1

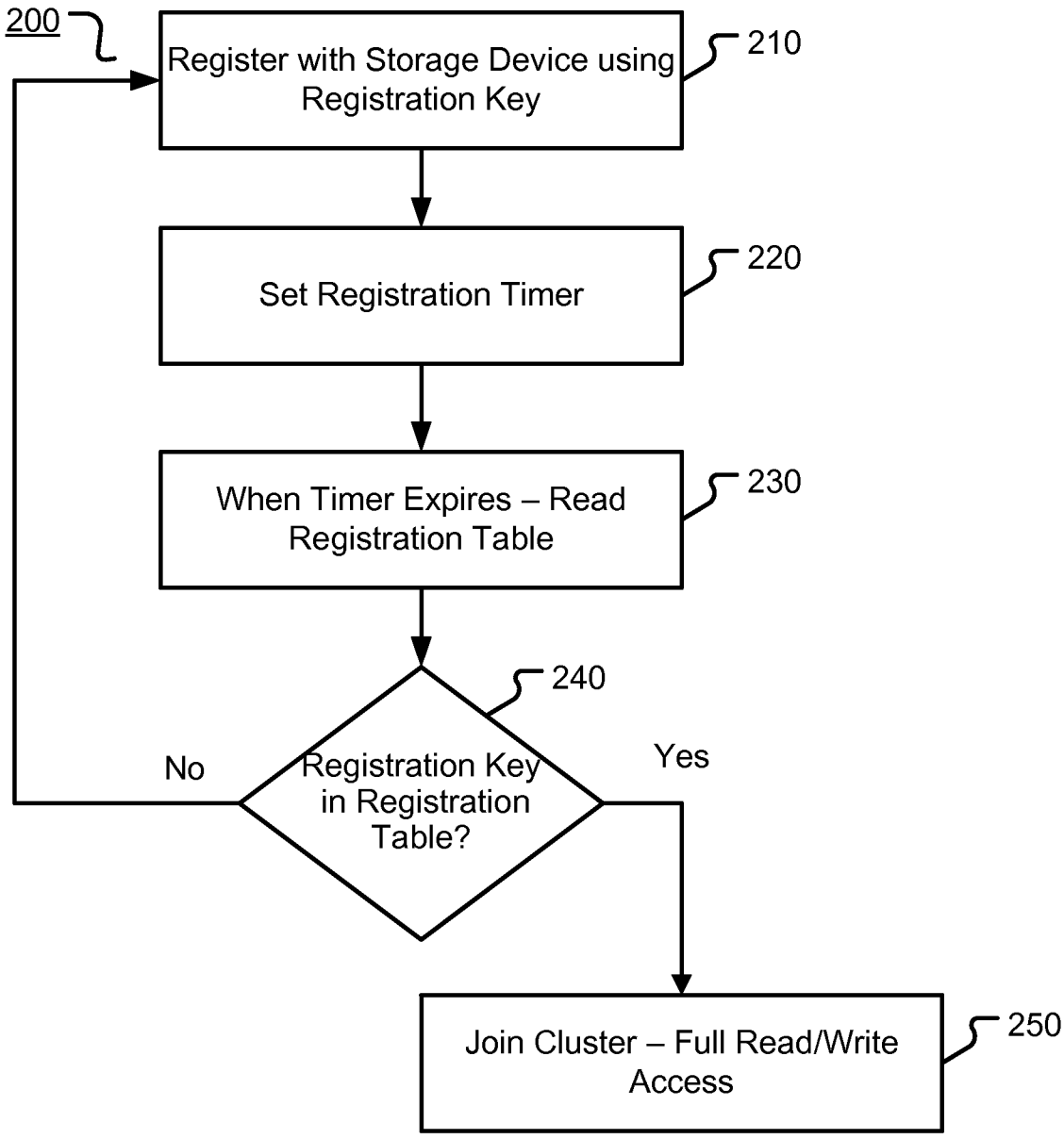
200
Register with Storage Device using Registration Key
210
Set Registration Timer
220
When Timer Expires – Read Registration Table
230
240
Registration Key in Registration Table?
No
Yes
Join Cluster – Full Read/Write Access
250

FIG. 2

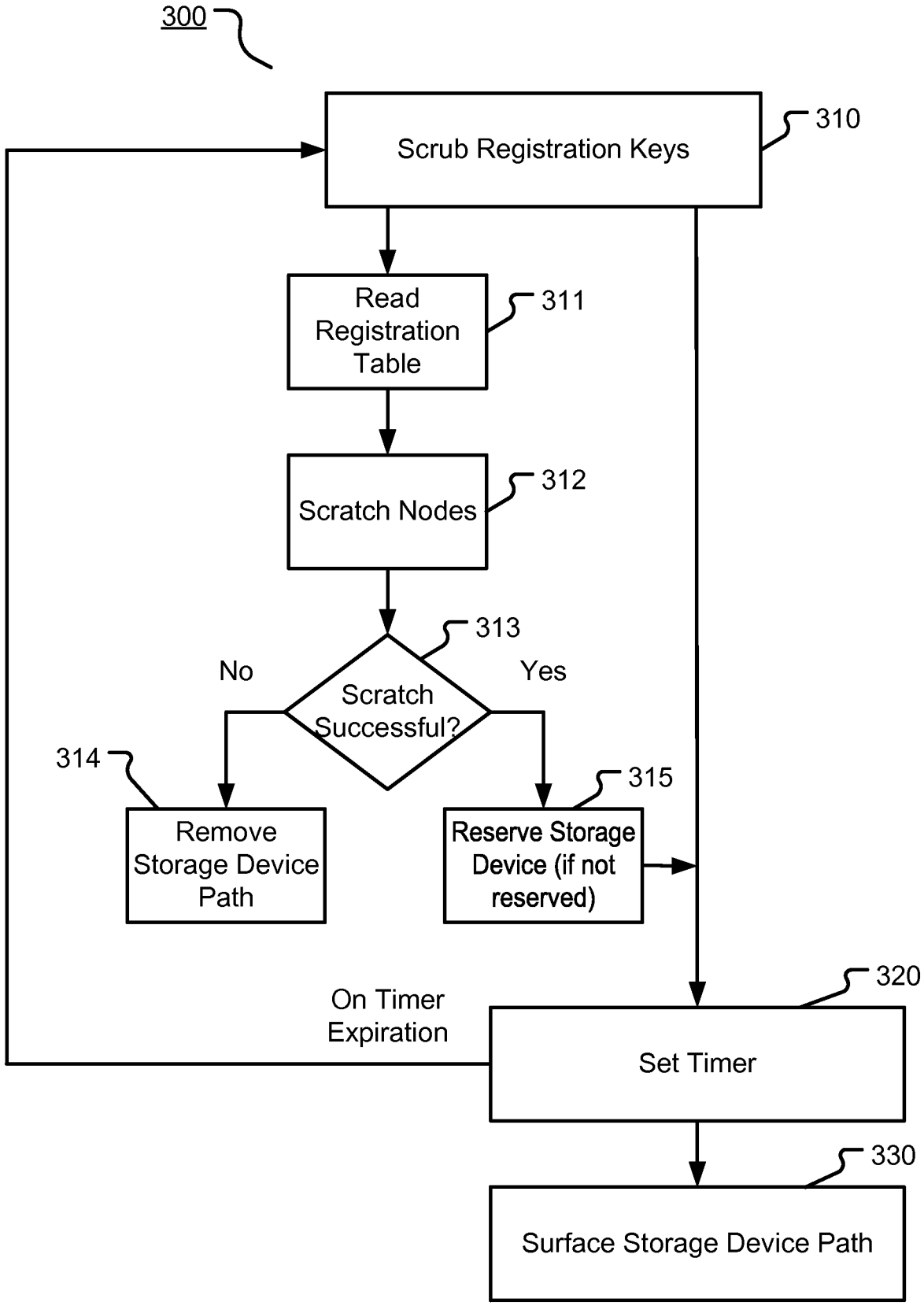
300
Scrub Registration Keys
310
Read Registration Table
311
Scratch Nodes
312
Scratch Successful?
313
No
Yes
314
Remove Storage Device Path
315
Reserve Storage Device (if not reserved)
On Timer Expiration
320
Set Timer
330
Surface Storage Device Path

FIG. 3

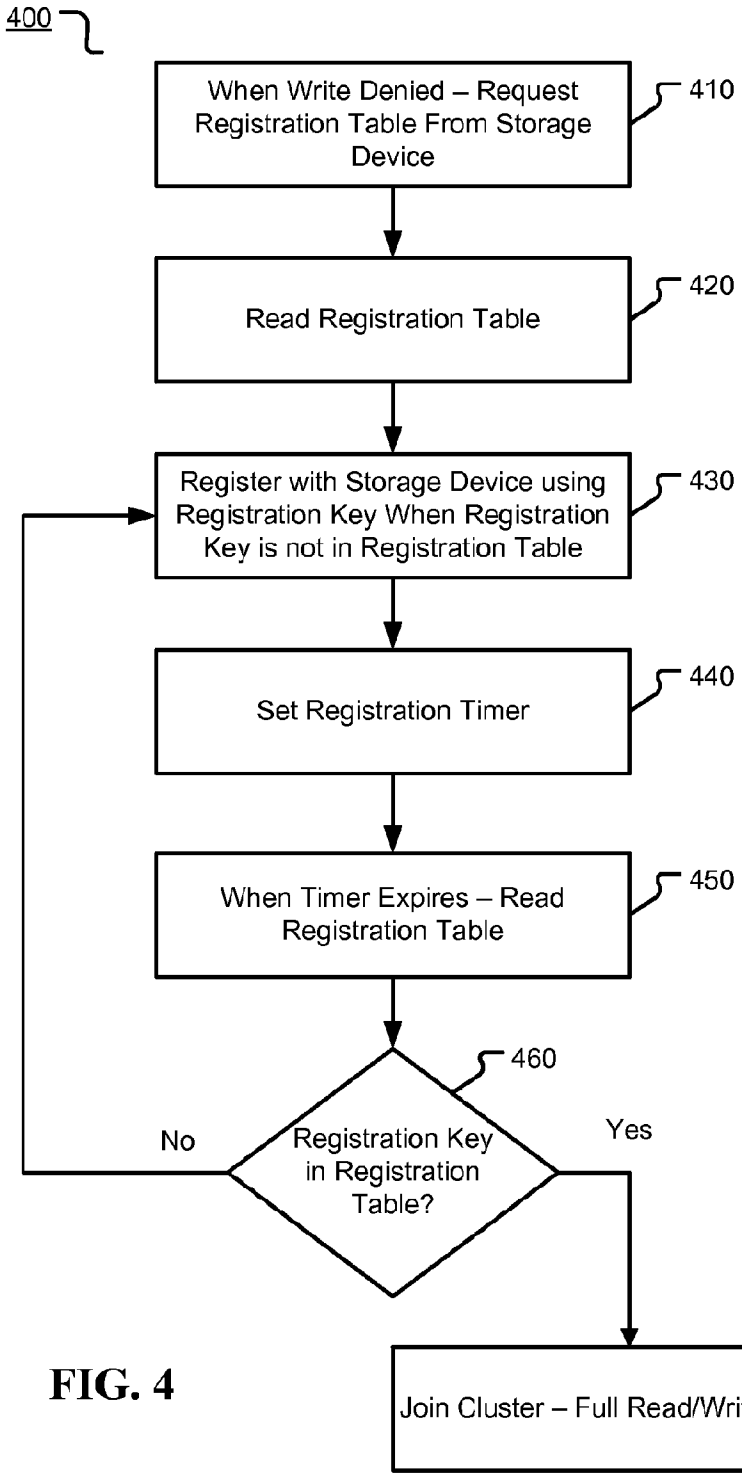
400
When Write Denied – Request Registration Table From Storage Device
410
Read Registration Table
420
Register with Storage Device using Registration Key When Registration Key is not in Registration Table
430
Set Registration Timer
440
When Timer Expires – Read Registration Table
450
460
Registration Key in Registration Table?
No
Yes
Join Cluster – Full Read/Write Access
470

FIG. 4

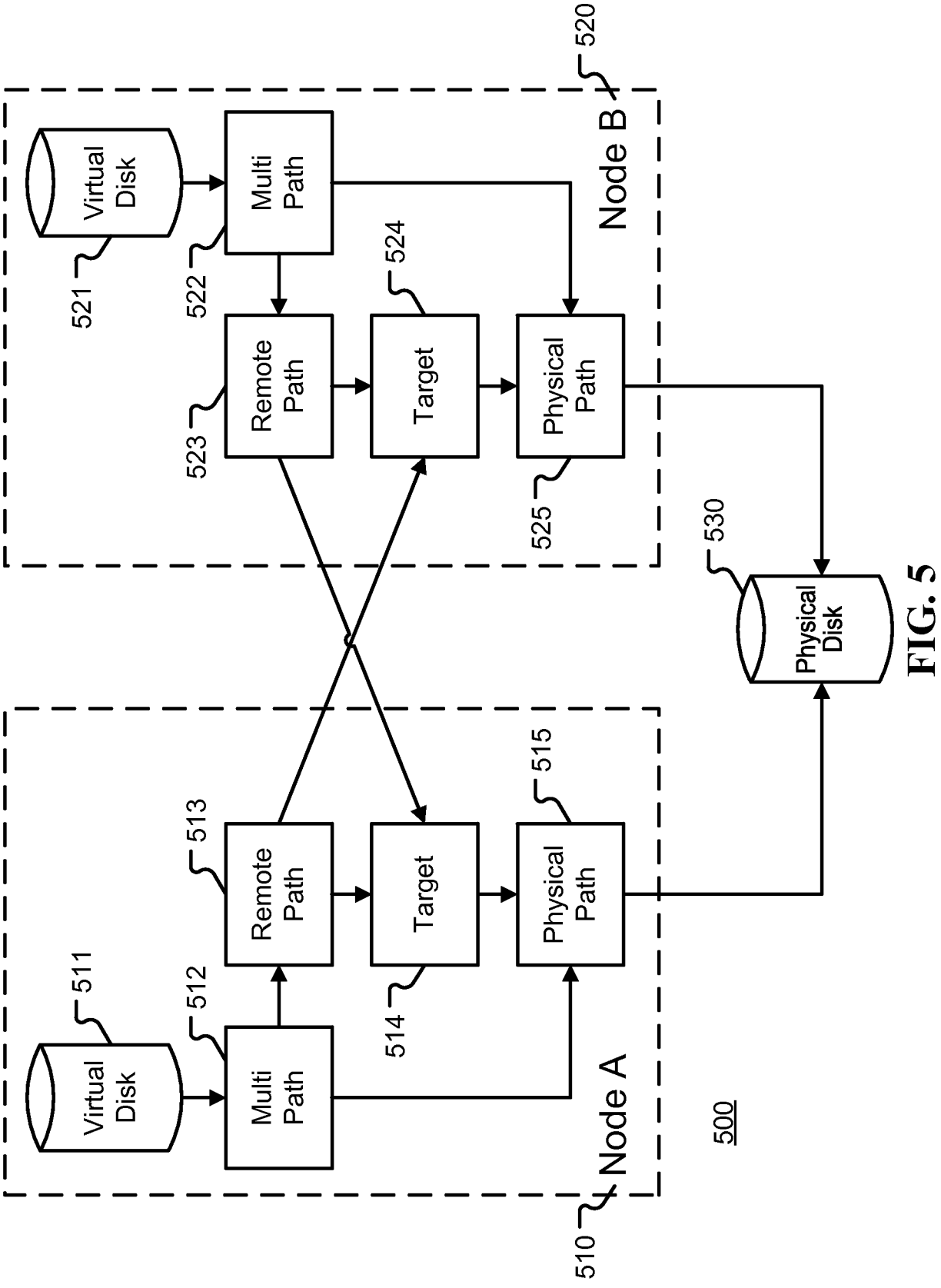
Node A
510
Virtual Disk
511
Multi Path
512
Remote Path
513
Target
514
Physical Path
515
Node B
520
Virtual Disk
521
Multi Path
522
Remote Path
523
Target
524
Physical Path
525
Physical Disk
530
500

FIG. 5

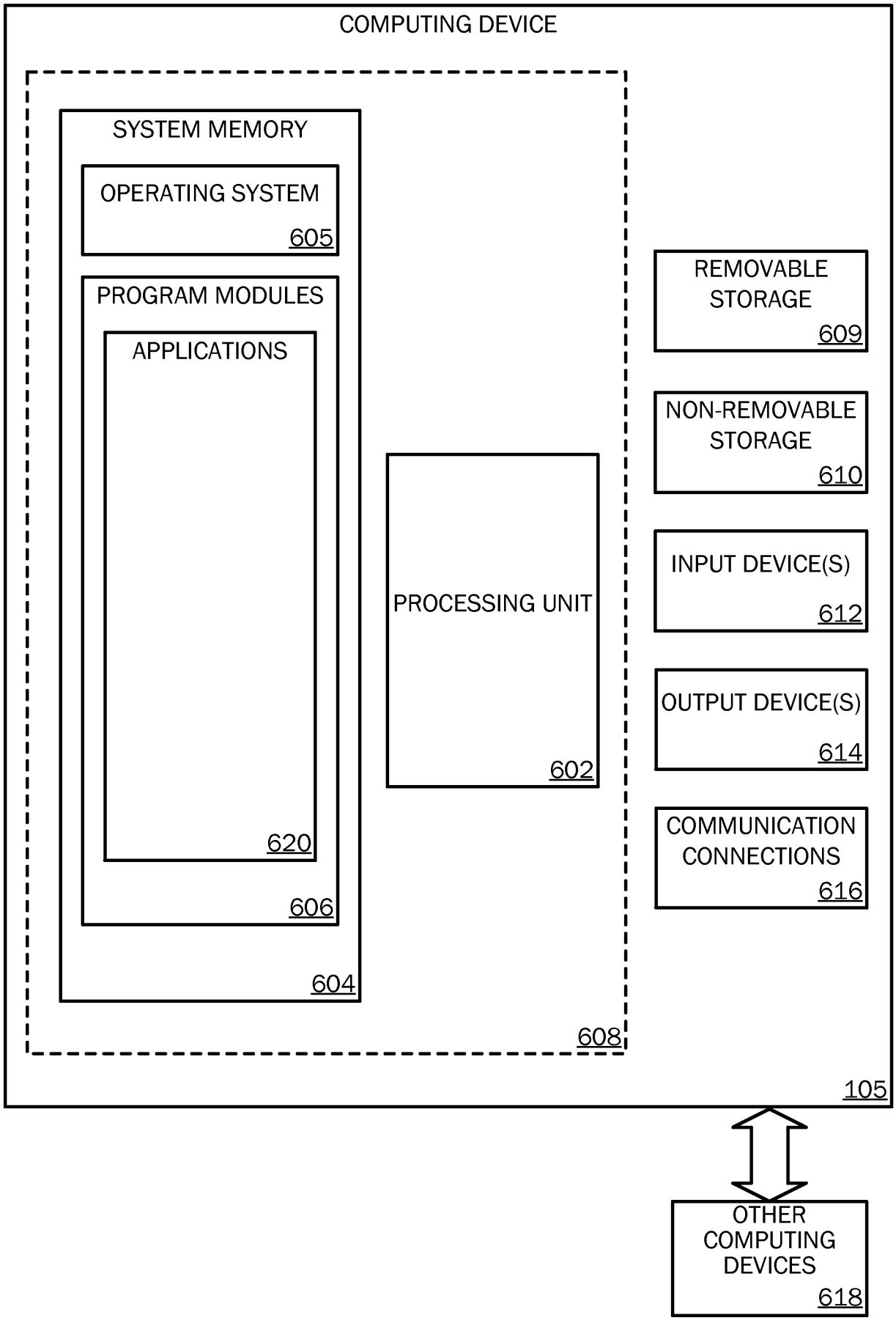
COMPUTING DEVICE
SYSTEM MEMORY
OPERATING SYSTEM
605
PROGRAM MODULES
APPLICATIONS
620
606
604
PROCESSING UNIT
602
608
REMOVABLE STORAGE
609
NON-REMOVABLE STORAGE
610
INPUT DEVICE(S)
612
OUTPUT DEVICE(S)
614
COMMUNICATION CONNECTIONS
616
105
OTHER COMPUTING DEVICES
618

FIG. 6

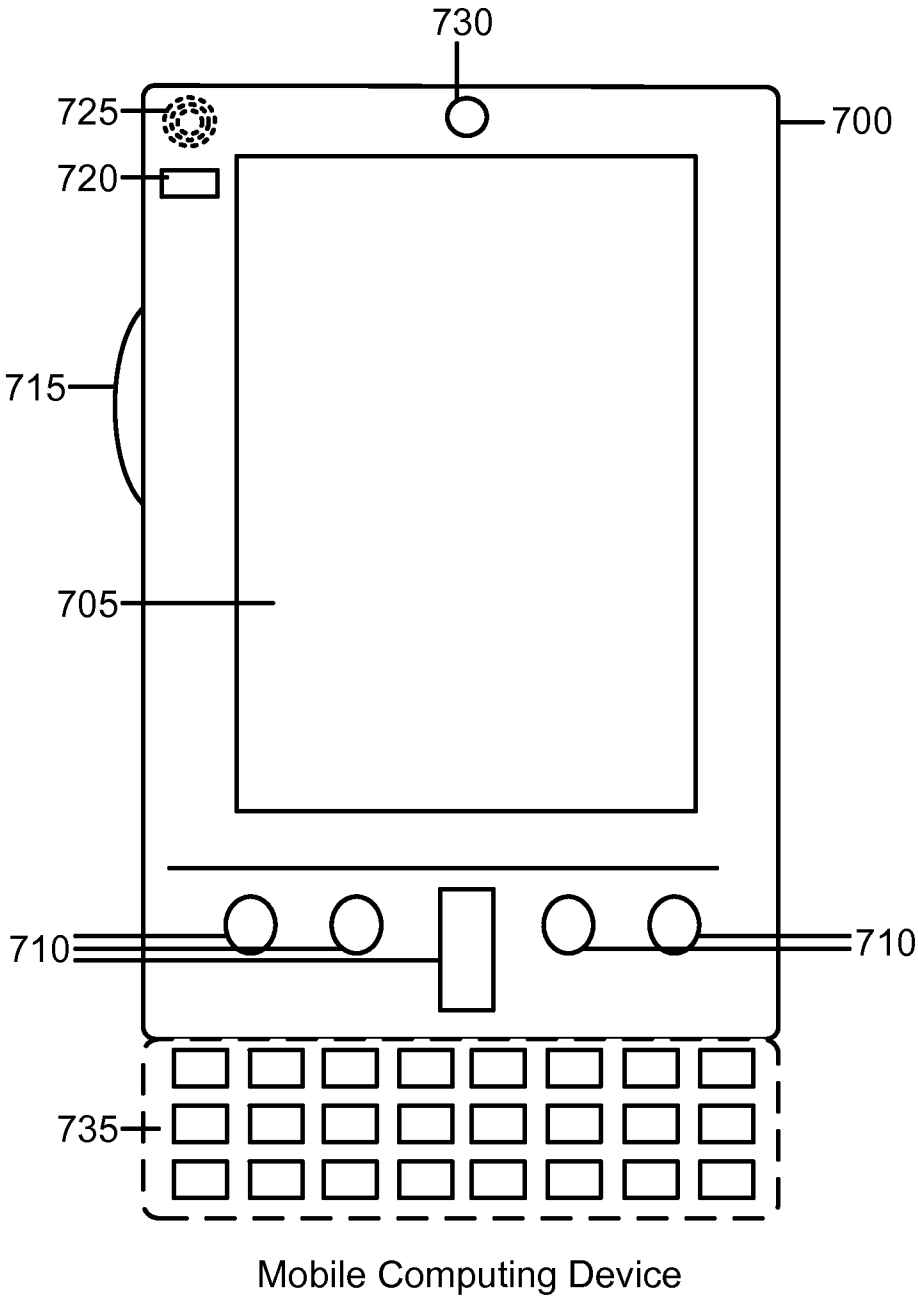


FIG. 7A

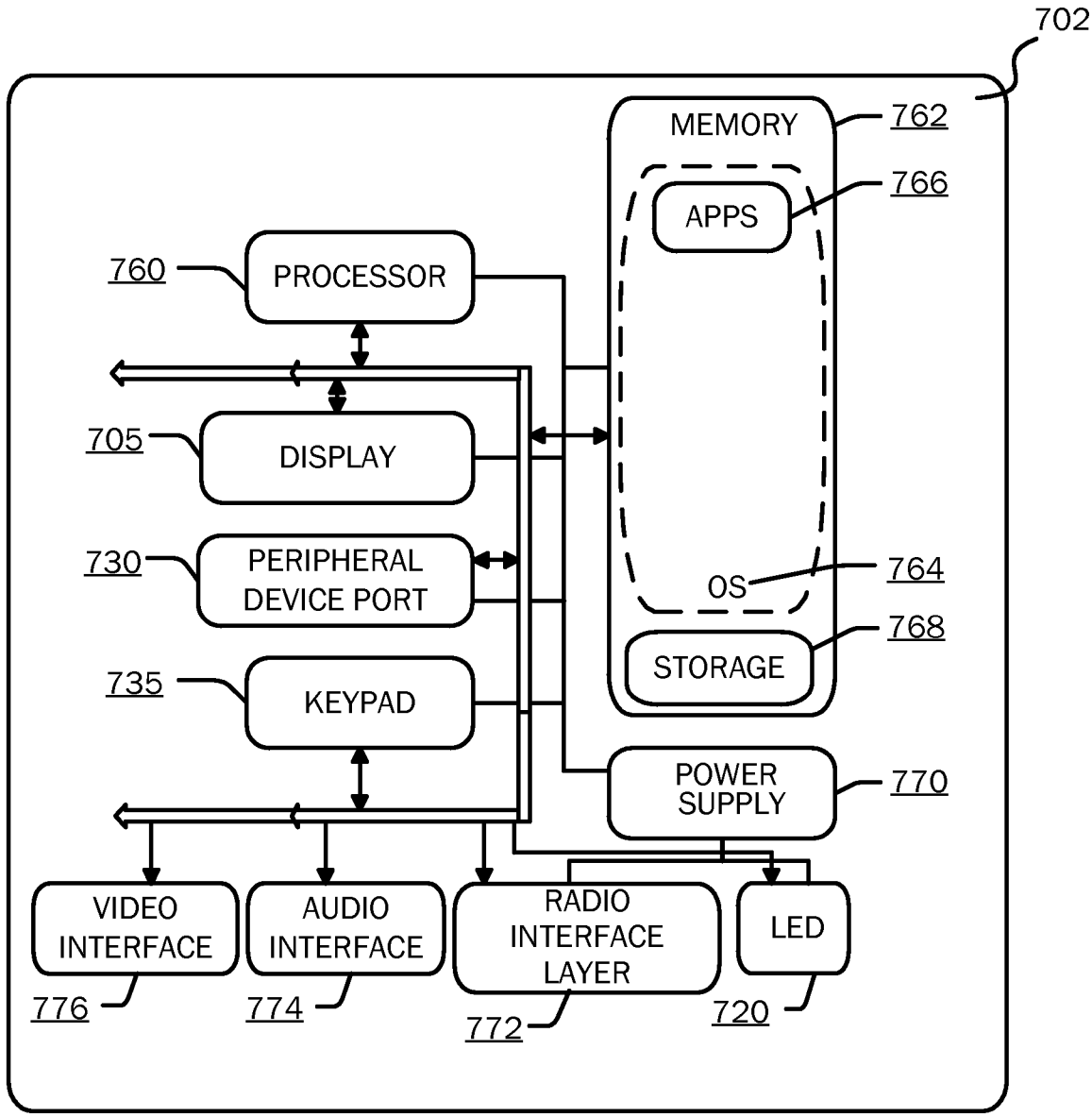
702
MEMORY
762
APPS
766
760
PROCESSOR
705
DISPLAY
730
PERIPHERAL DEVICE PORT
OS
764
768
STORAGE
735
KEYPAD
POWER SUPPLY
770
VIDEO INTERFACE
776
AUDIO INTERFACE
774
RADIO INTERFACE LAYER
772
LED
720

FIG. 7B

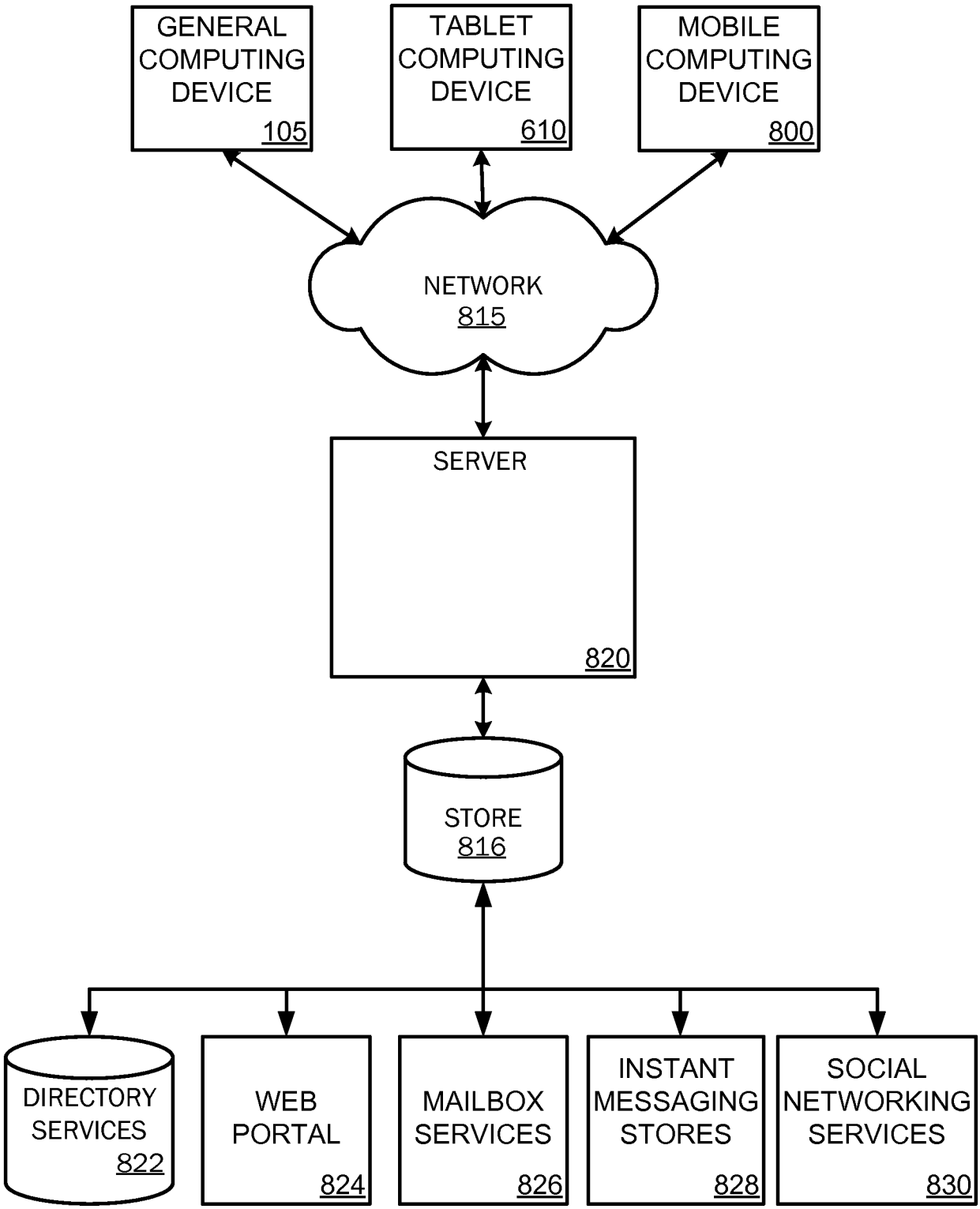
GENERAL COMPUTING DEVICE 105
TABLET COMPUTING DEVICE 610
MOBILE COMPUTING DEVICE 800
NETWORK 815
SERVER 820
STORE 816
DIRECTORY SERVICES 822
WEB PORTAL 824
MAILBOX SERVICES 826
INSTANT MESSAGING STORES 828
SOCIAL NETWORKING SERVICES 830

FIG. 8

# INTERNATIONAL SEARCH REPORT

International application No
PCT/US2014/039480

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F3/06
ADD.

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

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

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

| Category* | Citation of document, with indication, where appropriate, of the relevant passages | Relevant to claim No. |
|---|---|---|
| X | US 2010/077249 A1 (DAS RAJSEKHAR [US] ET AL) 25 March 2010 (2010-03-25) paragraphs [0004], [0017], [0022] - [0039]; figures 3-6 | 1-10 |
| X | US 2003/065782 A1 (NISHANOV GOR [US] ET AL) 3 April 2003 (2003-04-03) paragraphs [0050] - [0058] | 1-10 |
| X | WO 2009/158217 A2 (MICROSOFT CORP [US]) 30 December 2009 (2009-12-30) paragraphs [0031] - [0036] | 1-10 |

-/--

[X] Further documents are listed in the continuation of Box C.

[X] 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 | Date of mailing of the international search report |
|---|---|
| 23 September 2014 | 06/10/2014 |

| Name and mailing address of the ISA/ | Authorized officer |
|---|---|
| European Patent Office, P.B. 5818 Patentlaan 2<br>NL - 2280 HV Rijswijk<br>Tel. (+31-70) 340-2040,<br>Fax: (+31-70) 340-3016 | Alecu, Mihail |

1

Form PCT/ISA/210 (second sheet) (April 2005)

# INTERNATIONAL SEARCH REPORT

International application No
PCT/US2014/039480

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

| Category* | Citation of document, with indication, where appropriate, of the relevant passages | Relevant to claim No. |
|---|---|---|
| A | Microsoft: "How cluster reserves a disk and brings a disk online", , 11 February 2005 (2005-02-11), XP055142011, Retrieved from the Internet: URL:https://web.archive.org/web/20050211054942/http://support.microsoft.com/kb/309186 [retrieved on 2014-09-23] the whole document<br>----- | 1-10 |
| A | T10 TECHNICAL COMMITTEE ET AL: "SCSI Primary Commands - 3 Revision 23", , vol. dpANS, no. T10/1416-D 4 May 2005 (2005-05-04), pages I-XXV,1, XP002563347, Retrieved from the Internet: URL:http://www.t10.org [retrieved on 2006-10-13] the whole document<br>----- | 1-10 |

1

Form PCT/ISA/210 (continuation of second sheet) (April 2005)

# INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/US2014/039480

| Patent document cited in search report | Publication date | Patent family member(s) | Publication date |
|---|---|---|---|
| US 2010077249 A1 | 25-03-2010 | CN 102160047 A | 17-08-2011 |
| | | EP 2350849 A2 | 03-08-2011 |
| | | JP 2012503249 A | 02-02-2012 |
| | | US 2010077249 A1 | 25-03-2010 |
| | | US 2014229565 A1 | 14-08-2014 |
| | | WO 2010033335 A2 | 25-03-2010 |
| US 2003065782 A1 | 03-04-2003 | NONE | |
| WO 2009158217 A2 | 30-12-2009 | CN 102077193 A | 25-05-2011 |
| | | EP 2316077 A2 | 04-05-2011 |
| | | JP 5059974 B2 | 31-10-2012 |
| | | JP 2011526038 A | 29-09-2011 |
| | | US 2009327798 A1 | 31-12-2009 |
| | | WO 2009158217 A2 | 30-12-2009 |