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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
4 February 2016 (04.02.2016)

WIPO | PCT

(10) International Publication Number
WO 2016/018446 A1

(51) International Patent Classification:
G06F 15/16 (2006.01)

(21) International Application Number:
PCT/US2014/066017

(22) International Filing Date:
18 November 2014 (18.11.2014)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
3695/CHE/2014 29 July 2014 (29.07.2014) IN

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

(72) Inventors: **BONDURANT, Matthew David**; 2580 55th Street, Address2, Street, District, Boulder, Colorado 80301 (US). **VENKATESH, Dinesh**; 20 Dunstable Road, Westford, Massachusetts 01886 (US). **MUPPIRALA, Kishore Kumar**; Sy.No.192, Whitefield Road, Mahadevapura Post, Karnataka, 560048 Bangalore (IN). **EPSTEIN, Sasha Alexander**; 165 Dascomb Road, Andover, Massachusetts 01810-5897 (US). **LUMAN, Ronald John**; 2604 Roe Dr., Lewis Center, Ohio 43035 (US).

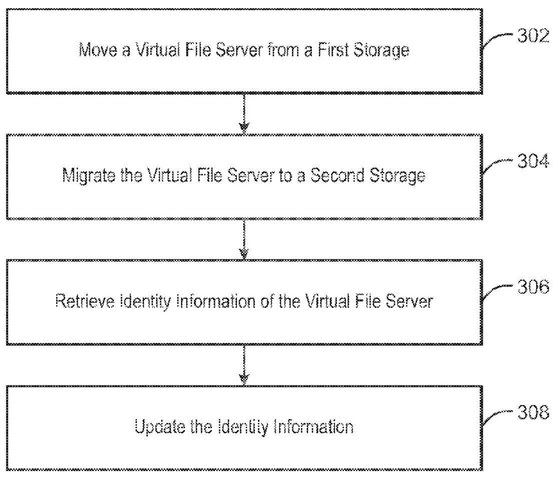
(74) Agents: **ORTEGA, Arthur** et al.; Hewlett-Packard Company, Intellectual Property Administration, 3404 East Harmony Road, Mail Stop 35, Fort Collins, Colorado 80528 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, 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,

[Continued on next page]

(54) Title: VIRTUAL FILE SERVER


Move a Virtual File Server from a First Storage
302
Migrate the Virtual File Server to a Second Storage
304
Retrieve Identity Information of the Virtual File Server
306
Update the Identity Information
308

300
FIG. 3

(57) Abstract: The present disclosure provides a method and a system for migrating a virtual file server. In an example of a method, a virtual file server is migrated from a first storage to a second storage, wherein the virtual file server comprises a server layer and a data layer. Identity information is retrieved from the server layer from the server layer for the second storage. The identity information is updated so instantiation of the virtual file server on the second storage appears the same as on the first storage.

WO 2016/018446 A1

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))*

— *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 FILE SERVER

BACKGROUND

[0001] A file system is used in computing for controlling the storage and retrieval of data. In particular, file systems are used to identify and separate data into individual units, and each unit is considered a "file" of the system. The overall file system is the logic and structure used to locate, store, retrieve, and manage groups of information or files. A file server can be a computer used to provide the location for shared disk access, which is the shared storage of computer files, and to enable the retrieval of data while computations are made by computers on the network. A virtual file server consists of a virtualized device configured to store any means of computer files and information. These computer systems and data identification and retrieval technologies are used throughout many different businesses and for a myriad of applications.

BRIEF DESCRIPTION OF THE DRAWINGS

[0002] Certain examples are described in the following detailed description and in reference to the drawings, in which:

[0003] Fig. 1 is an example block diagram of a computer system for block replication, virtualization, and migration of a file server;

[0004] Fig. 2 is a block diagram of a storage area on an example file system;

[0005] Fig. 3 is a process flow diagram of an example method for creating a detachable file server;

[0006] Fig. 4 is a process flow diagram of an example method for migrating a detachable file server between storage arrays; and

[0007] Fig. 5 is an example block diagram showing a non-transitory, computer-readable media that holds code that enables the migration of a virtual file server.

DETAILED DESCRIPTION OF SPECIFIC EXAMPLES

[0008] In computing, block replication is a technique used to ensure consistency between resources like hardware and software, and to improve the sharing and accessibility of information and the reliability those processes. Server virtualization for computing systems allows software to convert a physical server into multiple virtual machines or virtual servers. Through server virtualization, a virtual server acts as a physical computing device that can run its own operating system.

[0009] The concept of multi-tenancy is the use of a single object of software that is run on a server and serves multiple client entities or tenants. The architecture of multitenant applications is configured to partition data and configuration information virtually, with each tenant receiving its own virtualized application. A detachable self-contained file server is described herein that adds to these existing computing techniques with new methods. The technology herein combines block and file replication techniques with concepts from server virtualization to create a self-contained file server that can easily migrate from one storage array to another.

[0010] Virtualization of servers and a multitenant architecture saves physical computing space, and can provide more widespread access to information. File servers may provide the location for shared disk access. A challenge is to provide a file server solution that could provide multi-tenancy features in order to facilitate use, for example, in larger businesses with centralized information technology resources serving many departments, or a in a service provider environment for a cloud file server solution. A virtualization for a layered solution leveraging server can be used to provide a tenant container where each tenant is supported by one or more virtual machines (VMs) serving as a gateway to the underlying storage. However, scaling to a larger number of tenants requires significant computing resources in order to support the VMs, especially when there are hundreds or thousands of supported VMs.

[0011] Examples discussed herein provide a low cost and effective disaster recovery (DR) solution for a file server product. Generally, the file system is tightly coupled to the clustering software running on the file server gateways, making it necessary to boot the gateways from volumes on the storage arrays and replicate those boot volumes along with the user data to the second site. In the case of a failover event, a second set of physical file server gateways is booted off the replicated boot volumes from the first site, with ownership of the replicated user data transferred to the second site as well. Some scripts would be run on the servers to adjust for changes in hardware between sites. This leverages a cluster of physical file server gateways at one site layered on block storage, and replicates the data to a second site using block replication features of the array. This setup is difficult to maintain and does not practicably scale in a many-to-one replication scenario that might be found in a remote-office/branch-office use case.

[0012] In examples disclosed herein, file services technology are used in combination with block storage technology of storage arrays. A multi-protocol array can be used to deliver block storage, file storage, and object storage. For block storage, Fibre Channel (FC)/Internet Small Computer System Interface (iSCSI)/Fibre Channel Over Internet (FCoE) attachment can be utilized. For file storage, Server Message Block (SMB)/Network File System (NFS)/Hypertext Transfer Protocol (HTTP)/File Transfer Protocol (FTP) over Ethernet can be utilized. The examples herein provide a DR feature set similar to a remote copy for a storage array for block storage volumes, but applied to both the file and object data of the combined product. Examples provide a solution for file server consolidation, file server multi-tenancy, and file server business continuity.

[0013] Some examples described herein implement a method for transparently migrating a virtual file server, including replicating blocks of the virtual file server between a first storage and a second storage, wherein the virtual file server comprises a server layer and a data layer. The method includes migrating the virtual file server across a storage cluster boundary from the first storage to the second storage. The method includes retrieving identity information from the server layer for the second storage. The method also includes updating the identity information so the instantiation of the virtual file server on the second storage appears the same as on the first storage.

[0014] An example described herein uses a detached data store, including a storage region, to house a virtual file server. The virtual file server includes a server layer and a data store layer. The server layer includes server identity information. The data store layer includes a drive of the storage region. The virtual file server is detachable from the storage region, and the virtual file server is migrated to a second storage region.

[0015] Fig. 1 is an example block diagram of a computer system 100 for block replication, virtualization, and migration of a file server. A computer device 102 may process and store a file system. The computer device 102 may be, for example, a laptop computer, a desktop computer, a smartphone, or a computing tablet, among others. The computer device 102 may include a processor 104 that is configured to execute stored instructions, as well as a memory device 106 that stores instructions that are executable by the processor 104. The processor 104 can be a single core processor, a dual-core processor, a multi-core processor, a computing cluster, or the like. The processor 104 may be coupled to the memory device 106 by a bus 108 where the bus 108 may be a communication system that transfers data between various components of the computer device 102. In embodiments, the bus 108 may be a PCI, ISA, PCI-Express, HyperTransport®, NuBus, or the like.

[0016] The memory device 106 can include random access memory (RAM), e.g., SRAM, DRAM, zero capacitor RAM, eDRAM, EDO RAM, DDR RAM, RRAM, PRAM, read only memory (ROM), e.g., Mask ROM, PROM, EPROM, EEPROM, flash memory, or any other suitable memory systems. The computer device 100 may also include a storage device 110. The storage device 110 may include non-volatile storage devices, such as a solid-state drive, a hard drive, an optical drive, a flash drive, an array of drives, or any combinations thereof. The storage device 110 may include a number of modules configured to provide the computer device 102 with block replication and migration functionality. The storage device 110 includes a physical file system 112. The physical file system 112 includes file system information used to identify and separate data into individual units, and includes the logic and structure used to locate, store, retrieve, and manage files. The physical file system 112 includes virtual file server (VFS) 114. The VFS 114 will configure a name, network address, and authentication mechanism for access to the VFS 114 or physical file system 112. The VFS 114 includes server layer 116 and data layer 118. The server layer 116 is where the concepts of server identity are bound, such as name, IP address, and authentication mechanism, for example. The data layer 118 corresponds to a storage drive, for example, the C and D drive in a Windows file server. The configuration of file sharing is bound to the data layer 118, and a directory can be shared at the root or below within the data layer 118.

[0017] A network interface controller (NIC) 120 may also be linked to the processor 104. The NIC 126 may link the computer device 102 to the network 122 through a physical connection, or through a wireless connection. The network 122 can connect to a separate computing device 124 through another NIC 126, or a similar network connection. The network 122 allows the computer device 124 to network with resources, such as the Internet, printers, fax machines, email, instant messaging applications, and with files located on the other computing device 102, or storage servers 125, for example. Computer device 124 includes processor 128, memory 130, and bus 132 that are described as above with respect to processor 104, memory 106, and bus 108. Computer device 124 includes storage 134 to store physical file system 136. The physical file system 136 includes a virtual file server (VFS) 138 that is a replica of the VFS 114 on physical file system 112. The VFS 114 can be virtually detached from computer device 104, migrated to computer device 124, and configured as VFS 138. In some examples, the physical file system 112 can be disconnected from computer device 102 and moved to another computer device, e.g., computer device 124 to migrate the VFS 114 as VFS 138. The sever layer 140 and data layer 142 of VFS 138 correspond to the server layer 116 and data layer 118 of VFS 114. The virtualization of the physical file system 112 on VFS 114 and the replication of the VFS 114 to the VFS 138 on physical file system 136 provide a mechanism for disaster recovery (DR) without using dedicated hardware on standby at the target site.

[0018] Another computer device 144 can be connected to computer devices 102 and 124 through the network 122 through NIC 146. Computer device 144 can include processor 148, memory 150, and bus 152 that are described as above with respect to processor 104, memory 106, and bus 108. The processor 148 may be connected through the bus 152 to an input/output (I/O) device interface 154 configured to connect the computer device 144 to one or more I/O devices 156. The I/O devices 156 may include, for example, a keyboard, a mouse, or a pointing device, wherein the pointing device may include a touchpad or a touchscreen, among others. The I/O devices 156 may be built-in components of the computer device 144, or located externally to the computer device 144. The processor 148 may also be linked through the bus 152 to a display interface 158 configured to connect the computer device 144 to display devices 160. A display device 160 may be a built-in component of the computer device 144, or connected externally to the computer device 144. The display device 160 may also include a display screen of a smartphone, a computing tablet, a computer monitor, a television, or a projector, among others. Computer device 144 also includes storage 162 to store data and instructions. A user may interact with computer device 144 and configure or initiate the block replication and virtualization of a file server located on another computer device, for example VFS 114 on a first storage 110 associated with computer device 102 is transparently migrated to VFS 138 on a second storage 134 associated with computer device 124.

[0019] The block diagram of Fig. 1 is not intended to indicate that the computer system 100 is to include all of the components shown in Fig. 1. Further, any number of additional components may be included within the computer system 100, depending on the details of the specific implementation of the described herein. For example, a virtual machine (VM) can be utilized instead of computer device 144 by a user to configure or initiate file server migration. Further, the components discussed are not limited to the functionalities mentioned, but the functions could be done in different places, or by different modules, if at all.

[0020] Fig. 2 is a block diagram of a storage region in an example file system 200. The file system 200 is stored by some form of storage, for example, a storage device or storage array. The file system 200 is used to manage access to the content of files on the storage, as well as the metadata about those files. The file system 200 can automatically provision a set of volumes to meet a requested size and create a virtual file server (VFS) 202. A file store (not shown) within the VFS 202 can be created, and one or more shares can be created using SMB/NFS/HTTP of folders within the file store.

[0021] In some examples, a file system 200 is created based on porting of an ADVFS file system from HP-UX to Linux, for example. This can include a volume manager that can aggregate multiple underlying block devices into a domain. The process of domain discovery allows information about the contained file systems to be determined and imported into a cluster configuration. In some examples, a logical file system can be layered on top of the physical file system 200, and can be used to implement a scale-out file system. Scale-out storage is a networked-attached storage architecture where disk space can be expanded, even if new target drives exist on a separate storage array. The logical file system provides policy containers leveraged to store relevant policy information that described the identity of the VFS 202. Virtualization of file servers presents a logical space for data storage, and controls the process of mapping the logical space to the actual physical location of the data.

[0022] The VFS 202 includes plugins to implement different pieces of functionality to manage the system. These plugins include a data policy layer 204, and a server policy layer 206, which are the two primary layers of policy containers within the file system 200. The VFS 202 can include a platform management layer 209 for clustering and manageability. The platform management layer 209 can provide an infrastructure for implementing policy and manageability, in addition to management of cluster formation and maintenance. The platform management layer 209 is instantiated per file services cluster and is not replicated across sites, for example, from one array to another.

[0023] Instead of capturing the networking, authentication, and file sharing policy in a cluster level file store, these policies are now stored within the file system itself using the policy containers. Additional plugins (not shown) can be used to implement value added services such as, for example, anti-virus, quotas, and snapshots. The services will have associated policies stored in the VFS 202 on either the data policy layer 204, or the server policy layer 206, and not in a platform management store, or the like. The data policy later 204 corresponds to a C or D drive in a Windows file server, or the like, and is where the configuration of file sharing is bound.

[0024] The server policy layer 206 includes metadata 208 and authentication data 210, and is where concepts of server identity are bound, such as name, IP address, and particular authentication mechanism. By storing the file sharing information within the server layer 206 rather than in a cluster level file store, the sharing information is automatically recovered when the virtual file server 202 is migrated to a separate storage cluster.

[0025] A feature of the file system 200 of Fig. 2 is to provide a file server solution that incorporates multi-tenancy features. The inclusion of efficient multi-tenancy features facilitates use of the current technology in larger companies with centralized IT serving many departments, as well as in a service provider environment for a cloud file server solution, for example. A technology like a 3PAR™ array, for example, including a VM running on each 3PAR™ controller node can be used to implement the techniques described herein.

[0026] A feature leveraged by the file system 200 is the block replication of a VFS 202 between arrays, where options for both synchronous and asynchronous replication are available. Block replication is for replicating the VFS 202 making up a file system from one array to another so the file system 200 can be presented to a file services cluster (not shown) on the remote array without any a priori knowledge of the file system 200. Another included feature is the creation of block snapshots to provide a protection mechanism against corruption of the file system 200 that may not be recoverable through file based snapshots. Another included feature is technology that allows for migration of VFS 202 transparently from one array to another. Such file server migration techniques can be used for cases such as, for example, load balancing, or retirement of an array.

[0027] The block diagram of Fig. 2 is not intended to indicate that the file system 200 is to include all of the components shown in Fig. 2. Further, any number of additional components may be included within the file system 200, depending on the details of the specific implementation of the described herein. For example, the components discussed are not limited to the functionalities mentioned, but the functions could be done in different places, or by different components, if at all.

[0028] Fig. 3 is a process flow diagram of an example method 300 for creating a migratable file server. The method 300 may be implemented, for example, by the computer devices 102, 124 described with respect to Fig. 1. In particular, the computer device can be configured to permit the block replication, file server virtualization and migration techniques that are disclosed herein. The method 300 begins at block 302.

[0029] At block 302, a virtual file server is moved from a first storage array. The movement of the VFS can be physically performed, for example, by disconnecting a drive from the first storage array. In some examples, the VFS can be virtually moved by copying of the storage blocks that make up the VFS.

[0030] A typical file server is a computer attached to a network that is to provide a location for shared disk access that can be accessed by other computers that are attached to the same network. File servers are related to storage and management of data, and storage virtualization is to enable better functionality and advanced features within and across storage systems, including a number of disk drives or a disk array. Block virtualization refers to the separation of logical storage from physical storage so that logical storage may be accessed without respect to the physical storage or a heterogeneous structure. Virtualization of storage can achieve location independence by separating or abstracting the physical location of the data. For storage that is virtualized, replication is implemented above the software or device that is performing the virtualization.

[0031] At block 304, the virtual file server is migrated to a second storage array. The migration may be performed by physically connecting a drive that has been disconnected from a first storage array to a second storage array. The migration, as well as the replication, of the virtual file server contained in this physical file system is transparent so files are able to be moved without another client having access to the files knowing. The replication and migration of the virtual file server supports the scalability and multi-tenancy features described herein. In another example, the blocks that have been copied from a first storage array are written into the second storage array.

[0032] At block 306, identity information for the virtual file server is retrieved at the second storage array. After being moved or replicated and before receiving the identity information, the file system is unmounted from the first storage array using the virtual file server and an ownership change for the virtual file server is initiated to give the second storage array priority.

[0033] At block 308, the identity information is updated at the second storage array. VMs on storage clusters of the second storage array will detect the new virtual file servers and read the metadata from the virtual file servers as part of a domain discovery. Through this domain discovery, basic information about the file system contained on the virtual file server will be provided, and a mount operation can be performed on the virtual file server and file system to the second storage array.

[0034] Implementation of method 300 results in a multiprotocol array, delivering FC/iSCSI/FCoE attachment for block storage, SMB/NFS/HTTP/FTP over Ethernet for file storage, and, for example, OpenStack Swift™ (over HTTP), or the like, for object storage. The techniques described herein deliver a DR feature set similar to that used for block storage volumes, but applied to the file and object data that are being added into the combined product. The techniques disclosed simplify the DR scenario by decoupling the file server configuration from the storage cluster that is hosting the file server. Furthermore, this method 300 and method 400 below efficiently allows multi-tenancy without requiring dedicated containers, like a VM, for each tenant. Finally, the techniques herein facilitate balancing load across hardware resources by allowing a file server to migrate across physical cluster boundaries.

[0035] The process flow diagram in Fig. 3 is not intended to indicate that the method 300 is to include all of the components shown in Fig. 3. Further, the method 300 may include fewer or more steps than what is shown, depending on the details of the specific implementation.

[0036] Fig. 4 is a process flow diagram of an example method 400 for migrating a detachable file server between storage arrays. Method 400 embeds file server identity into a policy container within a virtual file server within the file system, and makes the file system easily detachable from the original host environment for the file system. Block replication techniques are able to effectively move, or replicate and migrate virtual file servers between physical hosting environments without needing to worry about replicating an entire operating system, as is typical for previous solutions. The method 400 begins at block 402.

[0037] At block 402, a virtual volume set is created and comprised of virtual file servers within a physical file system. The virtual volume set will automatically provision to meet a requested size virtual file server or number of virtual file servers. At block 404, a share is created using a protocol within the file system. The protocol is typically related to file storage and can be, for example, Server Message Block (SMB), (Network File System) NFS, or HTTP. At block 406, the virtual volume set is either physically moved or is replicated from a first storage array to a second storage array. Method 400 does not rely on a file system being tightly coupled to clustering software running on gateways. This method 400 avoids the possibility of some scripts needing to be run on servers to adjust for issues where the OS was expecting to have one set of hardware attached and instead different hardware was present. It is therefore not necessary to boot the gateways from volumes on the first storage array and replicate those boot volumes along with the user data to the second storage array, thus permitting the techniques discussed herein to be more efficient and cost effective.

[0038] At block 408, the file system is unmounted from the first storage array. The unmount operation of the file system is triggered by the initiation of an ownership change for the virtual volume set so that an array other than the host array becomes designated as the primary array. The unmount operation includes disabling the network addresses owned by the virtual file server. After the unmount operation completes, an application programming interface (API) call will be made to remove the record of the file server from the platform management store of the first array. This will be done without actually performing any operation on the virtual volumes. At block 410, the virtual file server is unpresented from a VM running on the first storage array. At this point, the first storage array is ready to relinquish ownership of the virtual file server.

[0039] At block 412, the physical file system that spans the virtual file server is brought-up on the second storage array. The second storage array now owns the virtual file server and a storage discovery operation can be requested and performed. The VMs will detect the new physical file system and the virtual file server associated with it and the virtual volumes therein and read related metadata as part of domain discovery, thereby providing basic information about the file system contained on the virtual volumes. A request is then made to import the file system into the cluster, at which point a record of the file system is made in the platform management store on the second storage array. At block 414, a mount operation can now be performed for the file system on the second storage array.

[0040] At block 416, policy information about the virtual file server is retrieved from a policy container in the file system. The policy information is retrieved during the mount of the file system on the second storage array. At block 418, the network addresses associated with the virtual file server on the second storage array are enabled. At block 420, the shares are presented over a network interface for user access to connected devices throughout the network. A user may configure the file server replication and migration from a computer device that is unrelated to the first or second storage arrays, and thereby remotely initiate the method 400 described herein. From the perspective of the client of the file server, the instantiation of the virtual file server on the second storage array looks the same as it did on the first storage array.

[0041] The process flow diagram in Fig. 4 is not intended to indicate that the method 400 is to include all of the components shown in Fig. 4. Further, the method 400 may include fewer or more steps than what is shown, depending on the details of the specific implementation.

[0042] Fig. 5 is an example block diagram showing a non-transitory, computer-readable media 500 that holds code to enable file server virtualization and migration. The computer-readable media 500 may be accessed by a processor 502 over a system bus 504. The code may direct the processor 502 to perform the steps of the current method as described with respect to Figs. 3 and 4. For example, the VFS 202, including data layer 204 and server layer 206, described as above with respect to Fig. 2, can be stored on the computer-readable media 500. The VFS 202 is configured to be migrated from computer-readable media 500 to some other storage, such as a storage array or storage cluster.

[0043] A block replication module 506 is configured to provide synchronous or asynchronous replication between storage arrays for virtual volumes, which make up a file system, and ultimately VFS 202. The block replication module 506 allows the file system to be presented to a file service cluster on a remote storage array without the array having a priori knowledge of the file system. The concept of the file server identity is embedded into a policy container of the VFS 202, specifically the server layer 206 within the file system and the file system itself, is then easily separable from the host environment. The block replication module 506, along with the migration module 508, allow a set of VFS 202 to migrate between physical hosting environments without the need to replicate an entire boot drive or operating system, and while still providing an effective DR solution for a file server product.

[0044] The migration module 508 transparently migrates the VFS 202 so files are able to be moved over the network without another client device, of a plurality of client devices, knowing. The replication and migration of the virtual file server supports multi-tenancy without the need for each tenant being supported by its own VM. The migration module 508 is to initiate an ownership change for a VFS 202, designating a second storage as primary. The migration module 506 allows the VFS 202 to unmount from a host storage. The migration module 506 can remove the record of the file system from the host storage without modification of virtual volumes or the VFS 202. The migration module 508 is also to present the newly owned VFS 202 to VMs on the second storage. The VMs will detect the new VFS 202 and read the metadata from the VFS 202, thereby allowing the VMs to access the underlying file system associated with the VFS 202. The migration module 508 is also to import the file system to the second storage, and make a record of the file system on a platform management store of the second storage. The migration module 508 ultimately mounts the file system through the virtual file server to the second storage.

[0045] An identity update module 510 is configured to complete the mount of the file system on the second storage by, for example, retrieving policy information from the server layer 206 of the VFS 202. Network addresses associated with the file system are enabled by the identity update module 510. Shares are then presented over various network interfaces for user data access throughout a network. The identity update module 510, as well as the block replication module 506 and migration module 508, are to operate transparently to client devices on the network, such that instantiation of the VFS 202 on the second storage appears the same as on the host storage environment.

[0046] The block diagram of Fig. 5 is not intended to indicate that the computer-readable media 500 is to include all of the components or modules shown in Fig. 5. Further, any number of additional components may be included within the computer-readable media 500, depending on the details of the specific implementation of the file server virtualization and migration techniques described herein.

[0047] While the present techniques may be susceptible to various modifications and alternative forms, the exemplary examples discussed above have been shown only by way of example. It is to be understood that the technique is not intended to be limited to the particular examples disclosed herein. Indeed, the present techniques include all alternatives, modifications, and equivalents falling within the true spirit and scope of the present techniques.

CLAIMS

What is claimed is:

1. A method for transparently migrating a virtual file server, comprising:
migrating a virtual file server between a first storage and a second storage, wherein the virtual file server comprises a server layer and a data layer;
retrieving identity information from the server layer for the second storage; and
updating the identity information so instantiation of the virtual file server on the second storage appears the same as on the first storage.

2. The method of claim 1, wherein migrating the virtual file server is performed by:
detaching a physical file system that comprises the virtual file server from the first storage; and
attaching the physical file system to the second storage.

3. The method of claim 1, wherein migrating the file server is performed by replicating blocks of the virtual file server from the first storage to the second storage.

4. The method of claim 1, wherein migrating the file server is performed transparently to a client of a plurality of clients on a network.

5. The method of claim 1, comprising retrieving metadata and authentication information from the virtual file server.

6. The method of claim 1, comprising binding file sharing to a drive level.

7. The method of claim 1, wherein migrating the file server comprises configuring for load balancing between storage arrays.

8. The method of claim 1, comprising creating block snapshots.

9. A detachable data store, comprising a storage region, comprising:
a virtual file server, wherein the virtual file server is comprised of a server layer and a data store layer;
wherein the server layer comprises server identity information;
wherein the data store layer comprises a data level of the storage region; and
wherein the virtual file server comprised code that automatically instantiates the virtual file server when the detachable data store is coupled to a storage system.

10. The detached data store of claim 9, wherein the detachable data store is configured to be detached from a first storage array and attached to a second storage array, and wherein the detachable data store is configured to automatically instantiate on the second storage array.

11. The detachable data store of claim 9, wherein the server identity information from the server layer comprises metadata and authentication information.

12. The detachable data store of claim 11, wherein the metadata comprises the information about the location of files and access information, and wherein the access information includes a type of drive, a list of drives, a list of users with access, or any combination thereof.

13. The detachable data store of claim 9, wherein a configuration for file sharing is bound to the drive level of the storage region.

14. A non-transitory, machine-readable medium comprising instructions that when executed by a processor cause the processor to:
activate a virtual file server that has been disconnected from a first storage array and connected to a second storage array;
retrieve identity information from a server layer in the virtual file server; and

update the identity information to make instantiation of the virtual file server on the second storage array appears the same as on the first storage array.

15. The non-transitory, machine-readable medium of claim 14, further comprising instructions that when executed by a processor cause the processor to migrate the virtual file server by replicating blocks associated with the virtual file server from the first storage array to the second storage array.

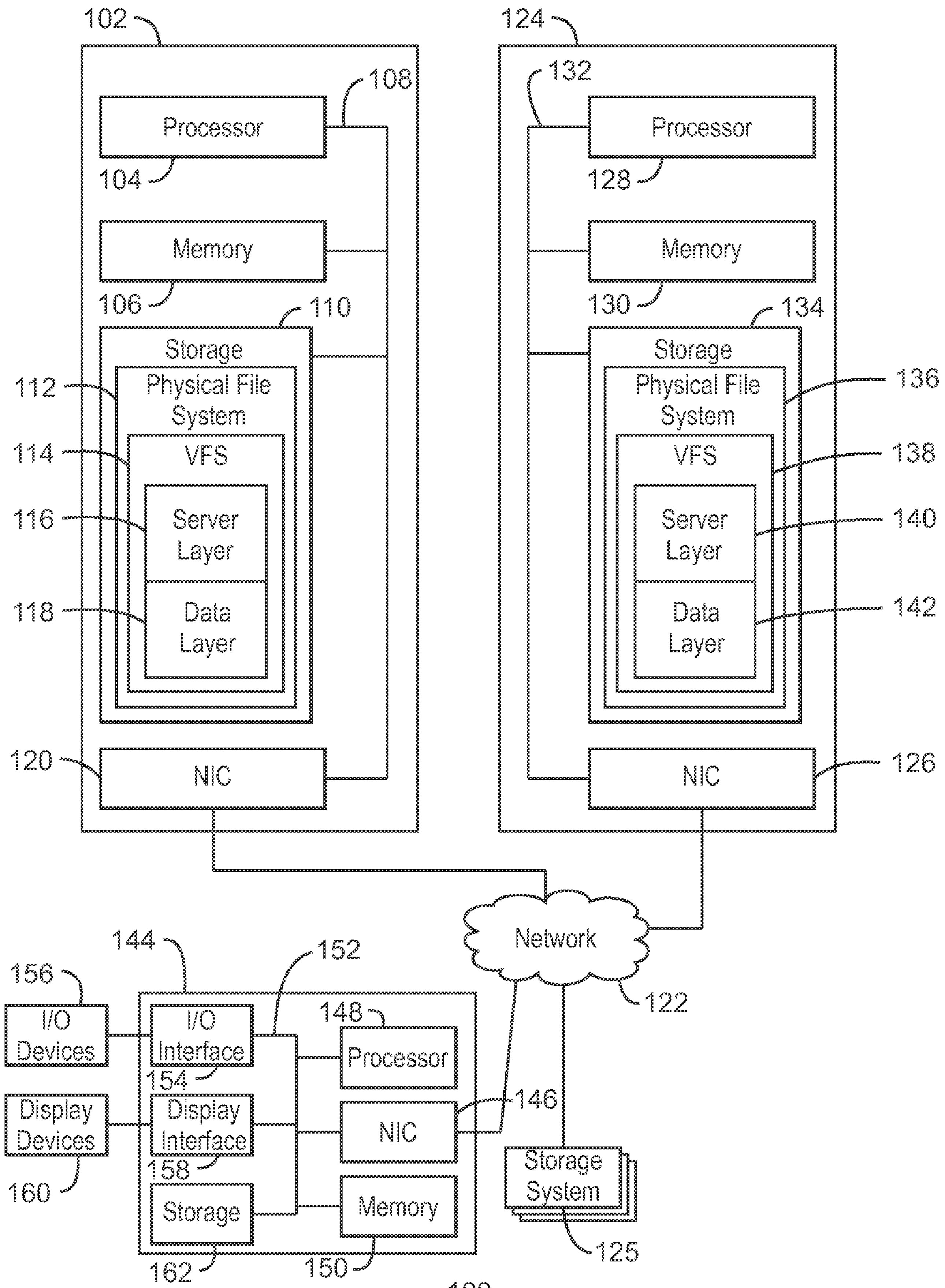

102
Processor
104
108
Memory
106
110
Storage
Physical File System
VFS
Server Layer
Data Layer
112
114
116
118
NIC
120
124
132
Processor
128
Memory
130
134
Storage
Physical File System
VFS
Server Layer
Data Layer
136
138
140
142
NIC
126
Network
122
144
156
I/O Devices
I/O Interface
154
152
148
Processor
Display Devices
160
Display Interface
158
NIC
146
Storage
162
Memory
150
Storage System
125
100

FIG. 1

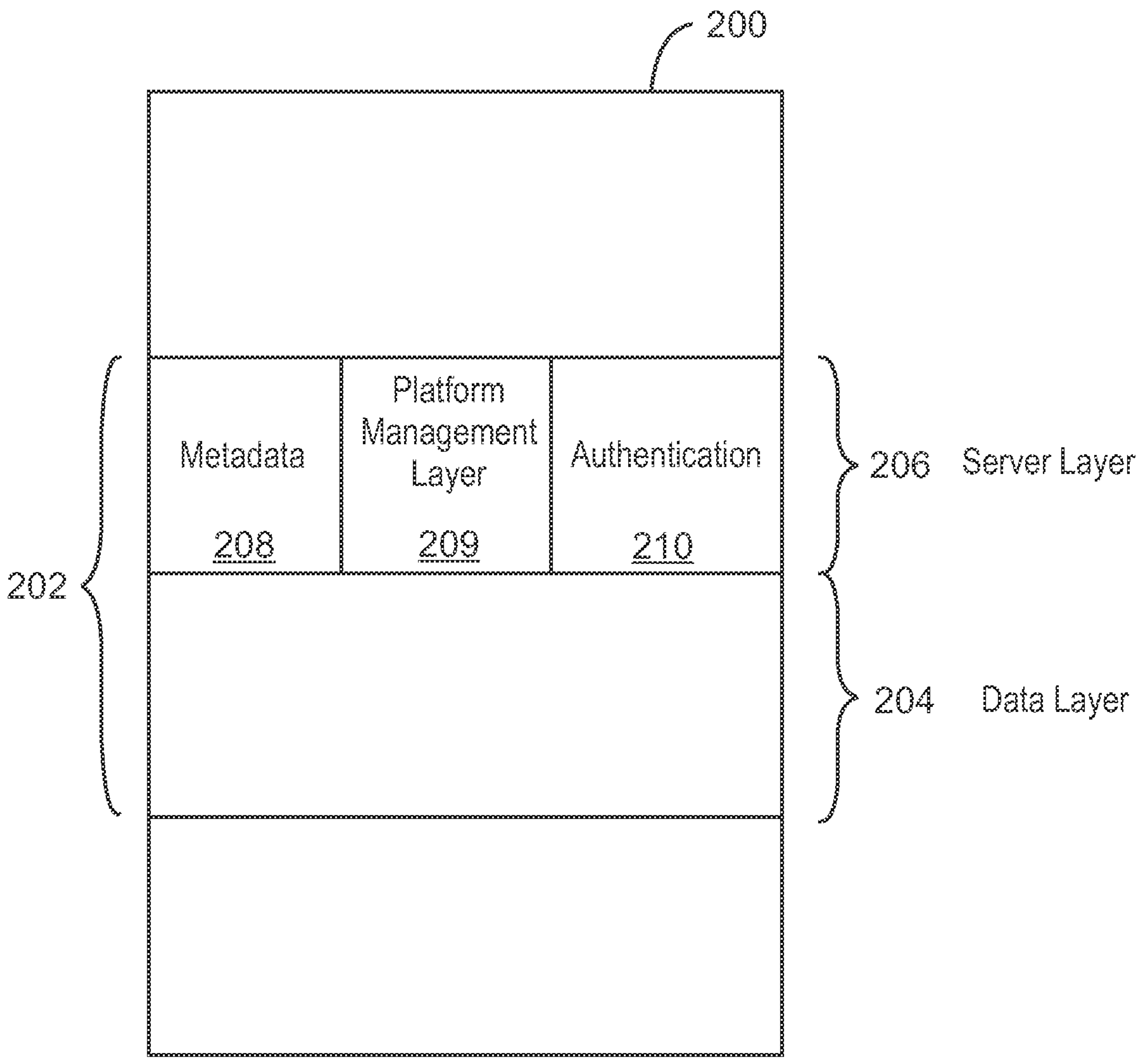
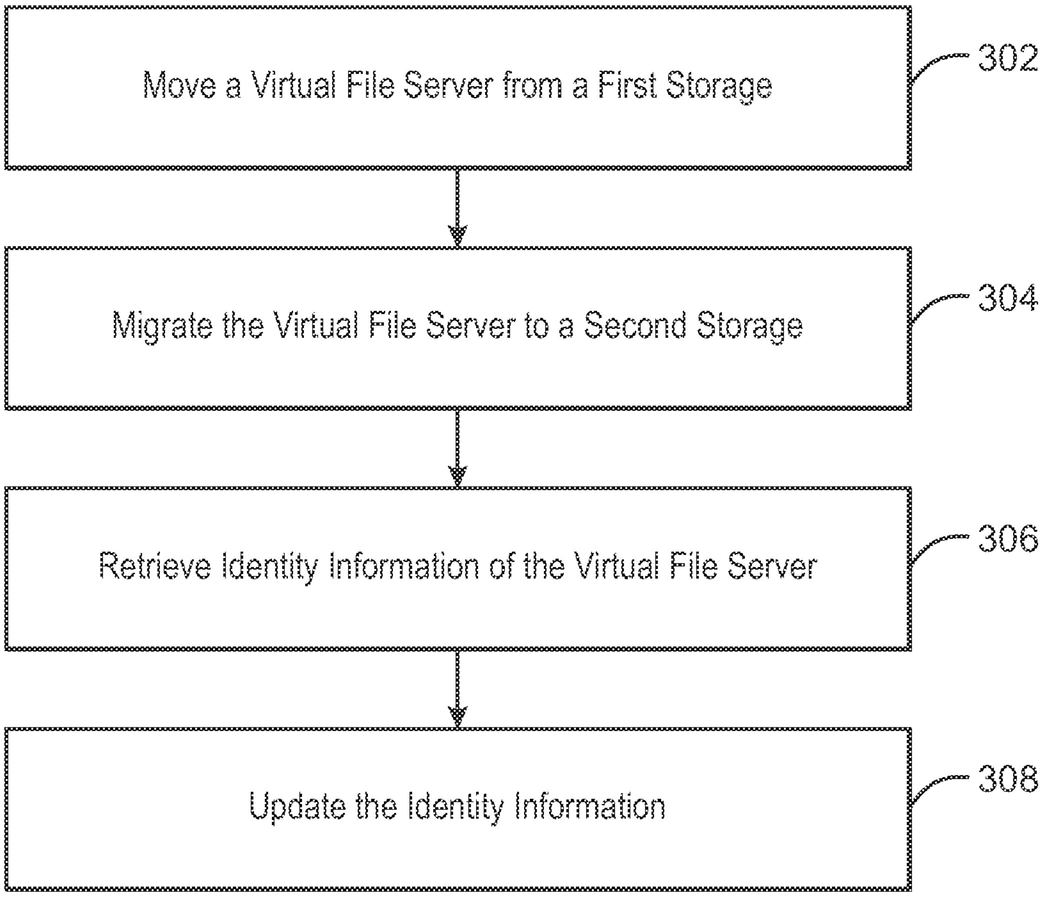
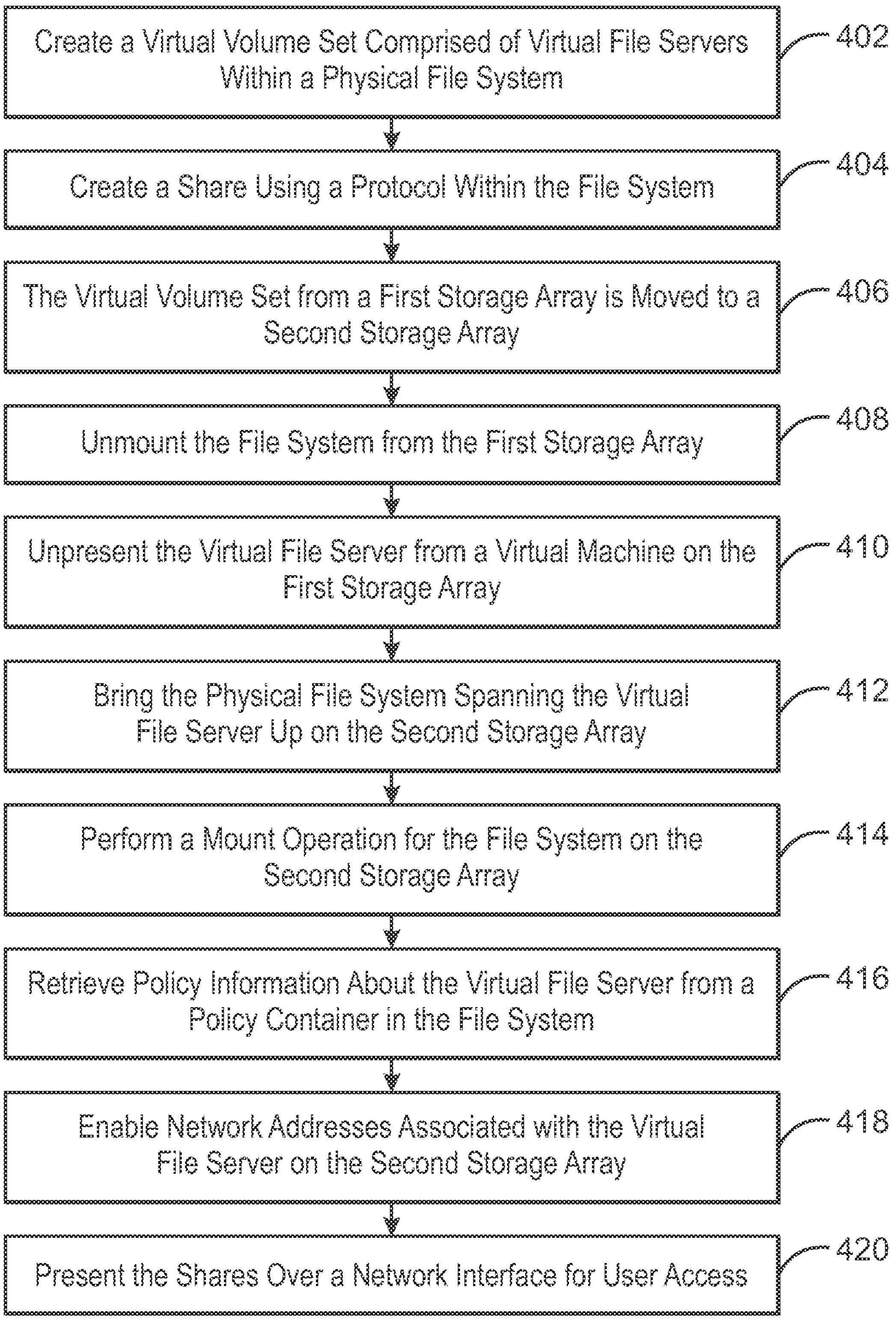

200
Metadata
208
Platform Management Layer
209
Authentication
210
206
Server Layer
202
204
Data Layer

FIG. 2


Move a Virtual File Server from a First Storage
302
Migrate the Virtual File Server to a Second Storage
304
Retrieve Identity Information of the Virtual File Server
306
Update the Identity Information
308

300
FIG. 3


Create a Virtual Volume Set Comprised of Virtual File Servers Within a Physical File System
402
Create a Share Using a Protocol Within the File System
404
The Virtual Volume Set from a First Storage Array is Moved to a Second Storage Array
406
Unmount the File System from the First Storage Array
408
Unpresent the Virtual File Server from a Virtual Machine on the First Storage Array
410
Bring the Physical File System Spanning the Virtual File Server Up on the Second Storage Array
412
Perform a Mount Operation for the File System on the Second Storage Array
414
Retrieve Policy Information About the Virtual File Server from a Policy Container in the File System
416
Enable Network Addresses Associated with the Virtual File Server on the Second Storage Array
418
Present the Shares Over a Network Interface for User Access
420

400

FIG. 4

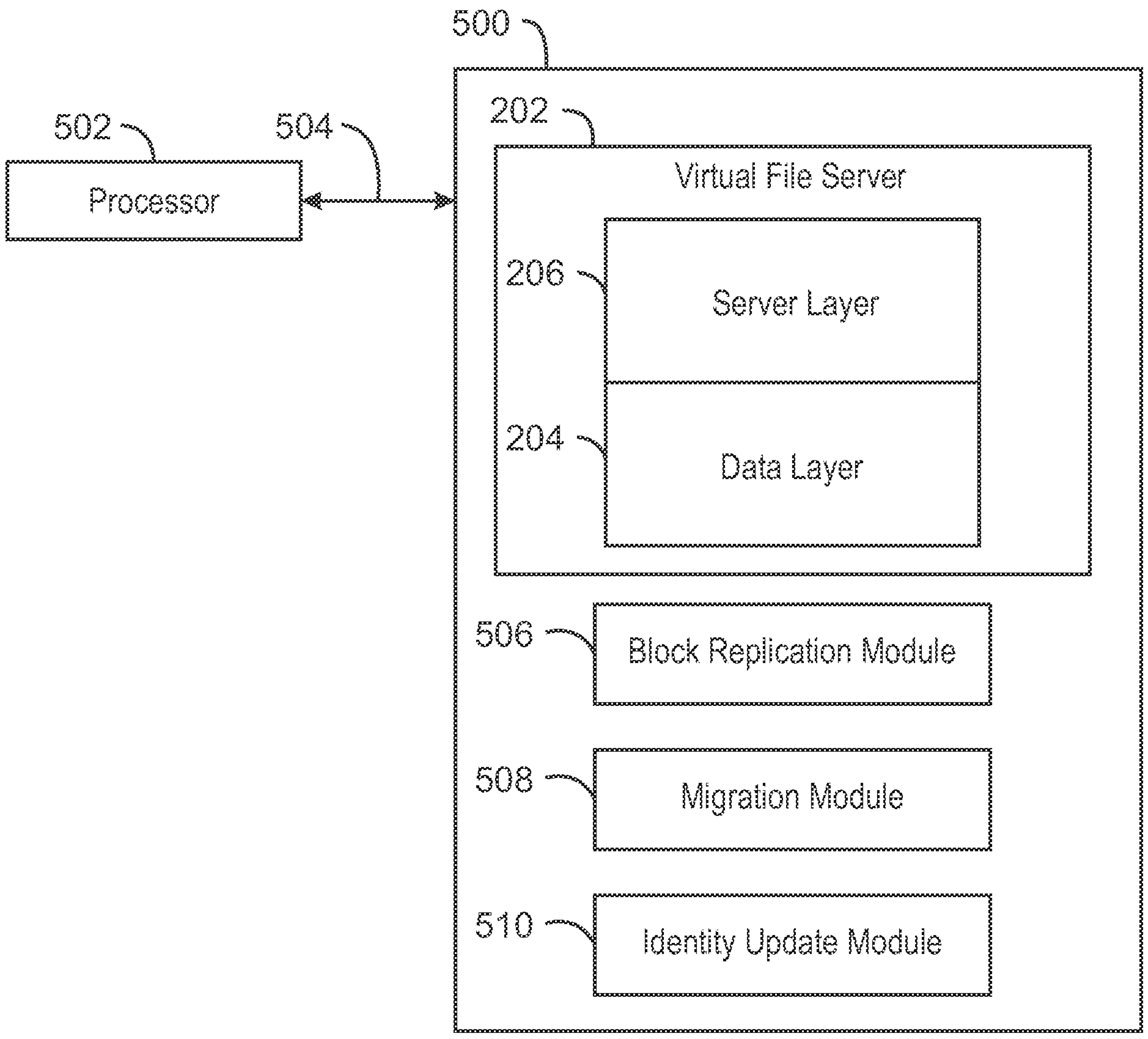

500
502
Processor
504
202
Virtual File Server
206
Server Layer
204
Data Layer
506
Block Replication Module
508
Migration Module
510
Identity Update Module

FIG. 5

INTERNATIONAL SEARCH REPORT	International application No. **PCT/US2014/066017**

A. CLASSIFICATION OF SUBJECT MATTER

G06F 15/16(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 15/16; G06F 9/44; H04L 29/08; G06F 17/30; G06F 9/46; G06F 15/177

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: migration, virtual file server, metadata, authentication, server layer, data layer

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X Y	US 2014-0143391 A1 (KEIKO TANIGAWA et al.) 22 May 2014 See abstract; paragraphs [0009]-[0013], [0073]-[0093], [0109]-[0116], [0148]-[0161]; claims 5-16 and figures 2-6, 17-19.	1,3-8,14-15 2,9-13
Y	US 2008-0148295 A1 (FREIMUTH DOUGLAS M. et al.) 19 June 2008 See abstract; paragraphs [0025]-[0031], [0169]-[0179]; claims 1-11 and figures 21A-24.	2,9-13
A	US 2007-0220121 A1 (IGNATIA SUWARNA) 20 September 2007 See abstract; paragraphs [0013]-[0019], [0033]-[0041]; claims 1-13 and figure 3.	1-15
A	US 7769720 B2 (ARMINGTON JOHN P.) 03 August 2010 See col.3, line 48 - col. 6, line 27, col. 7, line 31 - col. 8, line 11; claims 1-11 and figures 2-3, 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	Date of mailing of the international search report
29 April 2015 (29.04.2015)	**29 April 2015 (29.04.2015)**

Name and mailing address of the ISA/KR	Authorized officer
International Application Division Korean Intellectual Property Office 189 Cheongsa-ro, Seo-gu, Daejeon Metropolitan City, 302-701, Republic of Korea	HONG, Kyoung Ah
Facsimile No. ++82 42 472 7140	Telephone No. +82-42-481-5668

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2014/066017

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2014-0143391 A1	22/05/2014	WO 2014-080437 A1	30/05/2014
US 2008-0148295 A1	19/06/2008	CN 100583046 C	20/01/2010
		CN 101206587 A	25/06/2008
		JP 2008-152786 A	03/07/2008
		US 7984454 B2	19/07/2011
US 2007-0220121 A1	20/09/2007	US 8042108 B2	18/10/2011
US 7769720 B2	03/08/2010	US 2006-010176 A1	12/01/2006