



- (51) **International Patent Classification:**
G06F 12/14 (2006.01) *G06F 12/16* (2006.01)
- (21) **International Application Number:**
PCT/US2013/070046
- (22) **International Filing Date:**
14 November 2013 (14.11.2013)
- (25) **Filing Language:** English
- (26) **Publication Language:** English
- (30) **Priority Data:**
13/681,096 19 November 2012 (19.11.2012) US
- (71) **Applicant:** NETAPP, INC. [US/US]; 495 East Java Drive, Sunnyvale, California 94089 (US).
- (72) **Inventors:** KNIGHT, Frederick E.; 495 East Java Drive, Sunnyvale, California 94089 (US). MENEGHINI, John; 495 East Java Drive, Sunnyvale, California 94089 (US). RAJENDRAN, Arun Kumar; 495 East Java Drive, Sunnyvale, California 94089 (US). VIJAYA KUMARI RAJENDRAN NAIR, Saji Kumar; 495 East Java Drive, Sunnyvale, California 94089 (US).
- (74) **Agents:** SARATHY, Rajiv P. et al.; PERKINS COIE LLP, P.O. Box 1208, Seattle, Washington 98111-1208 (US).

(81) **Designated States** (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) **Designated States** (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

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

(54) **Title:** INTER-PROTOCOL COPY OFFLOAD

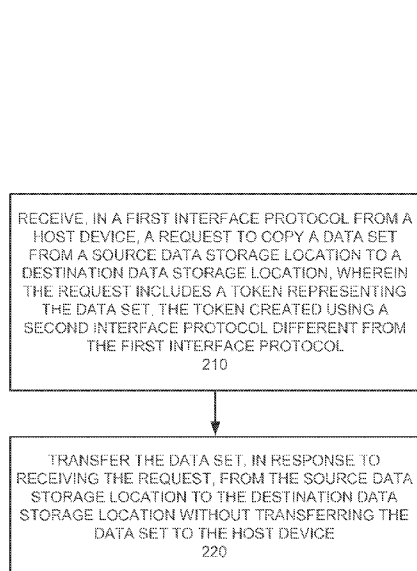


FIG. 2

(57) **Abstract:** Methods and apparatuses for performing inter-protocol copy offload operations are provided. In one embodiment, a method includes receiving a request in a first interface protocol from a host device. The request is a request to copy a data set from a source data storage location to a destination data storage location. The request includes a token, representing the data set, to be copied that was created using a second interface protocol that is different from the first interface protocol. The method also includes transferring the data set, in response to receiving the request, from the source data storage location to the destination data storage location without transferring the data set to the host device.

INTER-PROTOCOL COPY OFFLOAD

CROSS-REFERENCE TO RELATED APPLICATION

5

[0001] This application claims priority to U.S. Patent Application No. 13/681,096 filed 19 November 2012, which is hereby incorporated by reference in its entirety.

TECHNICAL FIELD

10 **[0002]** Various embodiments of the present application generally relate to the field of managing data on data storage devices. More specifically, various embodiments of the present application relate to methods and systems for performing inter-protocol copy offload operations.

15

BACKGROUND

[0003] The proliferation of computers and computing systems has resulted in a continually growing need for efficient and reliable storage of data. Host computing devices make use of data storage devices of many types and complexities to satisfy the growing data storage needs. The types of data storage devices commonly used
20 range from individual flash memory devices and hard drives to storage servers and clusters of storage servers. A storage server is a specialized computer that provides storage services related to the organization and storage of data, to one or more clients. The data is typically stored on writable persistent storage media, such as non-volatile memories and disks. A storage server is configured to operate
25 according to a client/server model of information delivery and may enable many clients or applications to access the data served by the system. A storage server can employ a storage architecture that serves the data with both random and streaming access patterns at either a file level, as in network attached storage (NAS) environments, or at the block level, as in a storage area network (SAN).

30 **[0004]** A traditional method for a host to make a copy of a selected data set is for the host to read the data set from the source storage device into the memory of the host and then transfer the data set to the destination storage location, under the control of the host. This may be accomplished by the host performing a series of buffered read/write processes on smaller chunks of the data set that is being copied.

In addition, data sets are sometimes copied from one location in a storage device or system to another location within that same storage device or system. The host controlled or host-centric copy process described above, in which the copied data flows through the host, is typically used even in cases where the source and the destination for the copy operation are within the same storage device or system.

[0005] While the host-centric copy process described above may be acceptable if the host has available bandwidth and/or the quantity of data is relatively small, the burdens associated with copying data in this manner can become significant in some cases. In addition to consuming central processing unit (CPU) resources of the host, if the data is transferred over a network, network bandwidth is also unnecessarily consumed because the data is first transferred to the host and then from the host to the destination. Processes called "copy offload" processes allow these types of data transfers to occur in a more efficient manner by transferring data directly between storage devices or systems, for example from one disk drive to another.

[0006] In the copy offload processes described above, a token is created and later returned to the storage device or system to initiate copying of the data from one location to another. Between the time the token is created and returned to the storage device, the token may also be transferred or exchanged among various hosts. Once a host returns the token to the destination storage device, the copy process occurs through communication between the source and the destination storage locations without the data having to flow through any of the hosts and without the hosts managing the data transfer process.

[0007] In addition to reducing the use of host computing resources, and potentially network bandwidth, the use of a token allows the copy or data transfer process to be separated from other operations of the host. Once the host interacts with the source storage device to create the token, the actual copying or transfer of the data can occur at a later point in time. In some cases the token may be transferred to a second host device. The copy process is started when the second host provides the token to the storage device or system. However, current implementations only allow copy offload processes involving more than one host to occur when both hosts are using the same interface protocol to communicate with the storage device(s) or system(s). Similarly, current copy offload implementations involving more than one storage device or system are only supported when the

storage devices or systems utilize the same interface protocol. For example, when a token is created for a host that communicates with a storage system using a SCSI interface, that token can only be used by other SCSI hosts and can only be used for copying the associated data to other SCSI storage devices or systems.

5

SUMMARY

[0008] Methods and apparatuses for inter-protocol copy offload processes are introduced here. These methods and apparatuses resolve the problems described above by providing support for inter-protocol token-based copy offload processes that are compatible with hosts, storage devices, and storage systems using different interface protocols.

[0009] In one embodiment, a method includes receiving a request in a first interface protocol from a host device. The request is a request to copy a data set from a source data storage location to a destination data storage location. The request includes a token, representing the data set, to be copied that was created using a second interface protocol that is different from the first interface protocol. The method also includes transferring the data set, in response to receiving the request, from the source data storage location to the destination data storage location without transferring the data set to the host device.

[0010] In another embodiment, a data storage system is provided. The data storage system includes a first communication interface using a first interface protocol, a second communication interface using a second interface protocol, and a copy system processor. The copy system processor is configured to receive a request at the first communication interface from a host device. The request is a request to copy a data set from a source storage location in a data storage device to a destination storage location. The request includes a token that represents the data set and was created using the second communication interface. The copy system processor is also configured to validate the token and transfer the data set from the source storage location to the destination storage location without transferring the data set through the host device.

[0011] Using the techniques introduced herein, token-based copy offload processes can be performed among devices that use different interface protocols.

This allows the benefits of copy offload processes, such as host computing resource savings and network bandwidth savings, to be realized when copying data in multi-protocol data storage environments. For example, a host communicating with a data storage system using a SCSI interface may request and received a copy offload
5 token. That token may be transferred to a second host that communicates with the data storage system using a Common Internet File System (CIFS) interface. The token is transferred to the data storage system, using CIFS, and is used by the data storage system to identify the data associated with the token and copy the data even though the token was created for a SCSI host and returned for use by a CIFS client
10 using a CIFS interface.

[0012] Embodiments introduced here also include other methods, systems with various components, and non-transitory machine-readable storage media storing instructions that, when executed by one or more processors, direct the one or more
15 processors to perform the methods, variations of the methods, or other operations described herein. While multiple embodiments are disclosed, still other embodiments will become apparent to those skilled in the art from the following detailed description, which shows and describes illustrative embodiments of the invention. As will be realized, the invention is capable of modifications in various
20 aspects, all without departing from the scope of the present invention. Accordingly, the drawings and detailed description are to be regarded as illustrative in nature and not restrictive.

BRIEF DESCRIPTION OF THE DRAWINGS

[0013] Embodiments of the present invention will be described and explained
25 through the use of the accompanying drawings in which:

[0014] Figure 1 illustrates an operating environment in which some embodiments of the disclosed techniques may be utilized;

[0015] Figure 2 illustrates a method of performing an inter-protocol copy offload operation;

30 **[0016]** Figure 3 illustrates an operating environment in which some embodiments of the disclosed techniques may be utilized;

[0017] Figure 4 illustrates a storage server that may be used in an embodiment of the disclosed techniques;

[0018] Figure 5 illustrates an operating environment in which the disclosed techniques may be used with two storage server systems;

5 **[0019]** Figure 6 illustrates a storage system with inter-protocol copy offload capability;

[0020] Figure 7 illustrates communication between two hosts and a storage device for performing an inter-protocol copy offload operation; and

[0021] Figure 8 illustrates communication between two hosts and two storage
10 devices for performing an inter-protocol copy offload operation.

[0022] The drawings have not necessarily been drawn to scale. For example, the dimensions of some of the elements in the figures may be expanded or reduced to help improve the understanding of the embodiments of the present invention.

Similarly, some components and/or operations may be separated into different
15 blocks or combined into a single block for the purposes of discussion of some of the embodiments of the present invention. Moreover, while the invention is amenable to various modifications and alternative forms, specific embodiments are shown by way of example in the drawings and are described in detail below. The intention, however, is not to limit the invention to the particular embodiments described. On
20 the contrary, the invention is intended to cover all modifications, equivalents, and alternatives falling within the scope of the invention as defined by the appended claims.

DETAILED DESCRIPTION

25 **[0023]** Traditional methods of copying data from one data storage location to another require a host to read data from a storage device and then transfer that data to a destination storage location. This approach has typically been used even when the data is being copied to another location within the same storage device. This approach lacks efficiency because it uses host computing resources and
30 communication resources unnecessarily. Copy offload processes were developed in order to allow data copy and data transfer to occur directly between storage devices

or systems without the data having to flow through one or more hosts. In a tokenized copy offload process, a host that seeks to copy or transfer data sends a request to a source storage device identifying the data to be copied or transferred. In the storage device, the data is located and a token is created that is a
5 representation of the data. The token does not contain the data but acts as a unique identifier and/or locator for the data and is later used to transfer the associated data between the storage devices or systems.

[0024] In existing systems, these tokenized copy offload processes can only take place among hosts and data storage devices that use the same interface protocol.
10 This limits the use of tokenized copy offload processes to environments using only a single interface protocol. The techniques introduced here address this problem. The techniques provide a method in which a token for a copy offload process can be created for a first host interacting with a source storage device in one interface protocol and can be used by a second host interacting with a destination storage
15 device in a different interface protocol. In some cases, the source storage device and the destination storage device may be one storage device or system that is capable of interfacing with multiple hosts using different interface protocols. In other cases, the source storage device and the destination storage device are not the same storage device and each communicates with its respective host(s) using a
20 different interface protocol.

[0025] In either case, an inter-protocol copy offload token can be provided to a destination storage device using a different interface protocol than was used to communicate with the host storage device when the token was created. Once the token is received by the destination storage device, the token is transferred from the
25 destination storage device to the source storage device and the data is transferred from the source storage device to the destination storage device in response. The two storage devices are both able to make use of the token even though they interact with hosts using different interface protocols.

[0026] Figure 1 illustrates operating environment 100 in which some
30 embodiments of the solution introduced here may be utilized. Operating environment 100 includes host 110, host 120, and storage device 130. Each of host 110, host 120, and storage device 130 are configured to communicate with the other

devices. Communication may occur through a direct link between the devices or may occur through one or more networks or other devices, such as a router.

[0027] Host 110 and host 120 are any type of computing devices that interface with data storage devices. For example, each of host 110 and host 120 may be a server, a personal computer, a laptop computer, a smartphone, a tablet, or a microprocessor. Host 110 and host 120 may each also be a specialized computing device for controlling or interfacing to storage devices or systems such as a storage controller, a data access component, or a file server. Host 110 and host 120 are each users and/or providers of data that is stored on storage device 130. In some cases, one or both of host 110 and host 120 may also be referred to as clients.

[0028] Storage device 130 is any type of device for storing data. Storage device 130 may comprise a physical storage medium or media capable of storing data, for example, electronically, magnetically, and/or optically. Storage device 130 may include, for example, a hard disk drive (HDD), a solid state storage device, a tape drive, an optical drive, or other type of non-volatile data storage device, including combinations thereof. Storage device 130 may also contain volatile memory such as random access memory (RAM). In some cases, storage device 130 may be a storage server or other storage management device that interfaces to a physical storage medium, but does not include the physical storage medium.

[0029] In some cases, storage device 130 may be a single hard disk drive with an integrated controller. In other cases, storage device 130 may be a much more complex storage device or system. For example, storage device 130 may be a network-attached storage (NAS) system, a storage area network (SAN), or a component thereof, such as a storage server. The various elements used to implement storage device 130 may be located physically together or distributed in various geographical locations. If storage device 130 comprises multiple physical storage mediums, the storage mediums may be arranged in a tiered storage arrangement or in a non-tiered storage arrangement.

[0030] Storage device 130 may also be implemented as a logical data container or as a storage abstraction rather than as actual physical storage. A storage abstraction may include, for example, a file, volume, disk, virtual disk, logical unit, data stream, alternate data stream, metadata stream, or the like. Any abstraction

may be repeated to any arbitrary depth. Storage device 130 may also be implemented using combinations of the implementations described above.

[0031] For purposes of simplifying the description, only two hosts and one storage device are shown in operating environment 100; however, additional hosts
5 and/or storage devices can be present in practice.

[0032] Figure 2 illustrates method 200 of performing an inter-protocol copy offload operation in one embodiment of the solution introduced here. Figure 2 is described with respect to operating environment 100 although use in other operating environments is possible.

10 **[0033]** In method 200, storage device 130 receives a request in a first interface protocol from host device 110 (step 210). The request is to copy a data set from a source data storage location in storage device 130 to a destination data storage location in storage device 130. The request includes a token that represents the data set to be copied. The token was previously created based on a request from
15 host 120 to storage device 130 using a second interface protocol that is different from the first interface protocol. The method also includes storage device 130 transferring the data set in response to receiving the request from host 110 (step 220). The transfer is from the source data storage location in storage device 130 to the destination data storage location in storage device 130. The data transfer occurs
20 without transferring the data set to host device 110 or host device 120.

[0034] Advantageously, storage device 130 is able to process a token for a copy offload operation that is received from a host using one interface protocol even though the token was created for a different host communicating with storage device 130 using a different interface protocol. Storage device 130 is capable of producing
25 and utilizing a token that is compatible with multiple interface protocols. For example, host 120 may communicate with storage device 130 using a SCSI interface and host 110 may communicate with storage device 130 using a Network File System (NFS) interface. The token is created in the SCSI environment, but is ultimately used to conduct the copy operation in the NFS environment. Other
30 combinations of interface protocols are possible and the invention is not to be limited to any particular interface protocol or combination of interface protocols.

[0035] The disclosed techniques enable inter-protocol copy offload processes to be performed. Previously, copy offload processes were only possible when the involved hosts and storage devices(s) used the same interface protocol. Previous inter-protocol copy operations required the copied data to be transferred through one or more of the hosts using traditional buffered read/write operations and could not be performed as copy offload operations in which the data is not transferred through one or more of the hosts.

[0036] The inter-protocol token described in the examples herein is a representation of the data that will be copied and allows a host to reference the data without actually obtaining a copy of the data. After storage device 130 creates the token, it maintains information about the data such that the data can be identified and located when the token is returned. In some cases, a token may contain a random or pseudorandom number that is difficult to guess or arbitrarily generate. The token is typically much smaller than the data it represents. In some cases, the token is 512 bytes long.

[0037] Figure 3 illustrates operating environment 300 in which some embodiments of the techniques introduced here may be utilized. Operating environment 300 includes storage server system 330, clients 380A and 380B, and network 390.

[0038] Storage server system 330 includes storage server 340, HDD 350A, and HDD 350B. Storage server system 330 may also include other devices or storage components of different types that are used to manage, contain, or provide access to data or data storage resources. Storage server 340 is a computing device that includes a storage operating system that implements one or more file systems. Storage server 340 may be a server-class computer that provides storage services relating to the organization of information on writable, persistent storage media such as HDD 350A and HDD 350B. HDD 350A and HDD 350B are hard disk drives. Storage server system 330 may also include other types of storage devices or memory including solid state drives (SSDs), tape drives, optical drives, and flash memory. A typical storage server system will include many more HDDs, or other storage devices, than are illustrated in Figure 3. Storage server system 330 is illustrated as a monolithic system, but could include systems or devices which are distributed among various geographic locations. Storage server system 330 may

also include additional storage servers which operate using storage operating systems which are the same or different from storage server 340.

[0039] Storage server 340 provides data storage services for clients such as client 380A and client 380B. Each of client 380A and 380B may communicate with storage server 340 using an interface protocol such as SCSI, CIFS, NFS, or another interface protocol, including combinations thereof. The techniques and the teachings of this description can be adapted to a variety of storage architectures including, but not limited to, NAS, SAN, a disk assembly directly-attached to a client or host computer, an individual disk with a disk controller, or to another type of storage device. The term “storage server” should therefore be taken broadly to include such arrangements.

[0040] In one embodiment of the techniques introduced here, an inter-protocol copy offload operation is performed on data stored in storage server system 330. The data may initially be stored on a single storage device of storage server system 330 or may be spread across a number of storage devices which make up storage server system 330 using one or more known methods or architectures such as Redundant Array of Independent Disks (RAID). Client 380A initiates the inter-protocol copy offload operation by sending a request to storage server system 330 in a first protocol. The request includes an identification of data that client 380A wishes to copy. The data may be a single file, a group of files, a directory, a group of directories, an entire volume, or any other group or subset of data stored by storage server system 330.

[0041] Upon receiving the request, storage server 340 identifies the data of interest and creates an inter-protocol copy offload token that is a representation of or a unique identifier for the data. Storage server 340 then returns the token, or a copy of the token, to client 380A using the first interface protocol. The token has been created for use in and communicated using the first interface protocol. In some cases, the token is compliant with a standard associated with the first interface protocol specifying a format for copy offload tokens. However, the token also contains information that allows it to be used with and interpreted by devices using at least one other interface protocol.

[0042] In this example, the inter-protocol token is then transferred from client 380A to client 380B. Client 380B communicates with storage server system 330 using a second interface protocol. Client 380B uses the token to initiate the second stage of the inter-protocol copy offload operation. The second stage involves the
5 actual copying or transfer of the data associated with the token from the source storage location to a destination storage location. The copy stage of the copy offload operation may be performed immediately after the creation of the token or may be performed at a later time. In order to initiate the copying, the token is transmitted to storage server 340 by client 380B along with an identification of a destination storage
10 location for the copy offload operation. In this example, the destination is another location within storage server system 330.

[0043] Although the token was created for client 380A in response to communications conducted using the first interface protocol and may contain a format compatible with the first interface protocol, it is transferred to storage server
15 system 330 by client 380B using the second interface protocol. Storage server 340 receives the token using the second interface protocol, but is able to resolve, based on information in the token, that it was created in an interaction with a client using the first interface protocol and may have a format associated with the first interface protocol. In other words, storage server 340 is capable of producing and processing
20 tokens that are compatible with multiple interface protocols. Storage server 340 uses the received token to identify and locate the data associated with the token. This process may include verifying or validating the token.

[0044] In the preceding example, both the source data storage location and the destination data storage location are within storage server system 330. In some
25 cases, the source and the destination storage locations may be in the same storage device, but represent different logical locations within that storage device.

[0045] Copy offload processes are also sometimes referred to as offloaded data transfer (ODX) processes and are defined for Microsoft Windows. In the Windows ODX operation, a host transmits a request to a source storage device or system
30 identifying data to be copied. The host receives a token representing the data of interest from the storage device. The token does not contain the data but acts as a unique identifier and/or locator for the data. The host then uses an offload write command, including the token, to request data movement from the source to a

destination storage location. Windows ODX is designed to work with storage devices which implement the Small Computer System Interface (SCSI) standard. Specifically, Windows ODX features are supported in devices that implement SCSI Primary Commands – 4 (SPC-4) and SCSI Block Commands – 3 (SBC-3). These
5 commands are defined by T10 technical committee of the International Committee on Information Technology Standards (INCITS) which is responsible for SCSI architectural standards and storage interfaces. Copy offload processes are not limited to Windows or SCSI devices.

[0046] In some examples of the techniques disclosed herein, the inter-protocol
10 token is created using a single token format that is vendor specific but also compliant with one or more industry standards. For example, the SCSI T10 SPC-4 standard defines a 512 byte copy offload token. According to the standard, the first four bytes, numbered zero through three, are used to define a Representation of Data (ROD) type. Although the standard defines how all 512 bytes are to be used, the standard
15 also specifies that the four ROD type bytes may be used to indicate that the token has a vendor specific format. In the case of a vendor specific format, the bytes numbered 8 through 511, are used to implement the vendor specific token format. This inter-protocol token may be compliant with the SCSI copy offload standard but also contain information enabling a properly configured storage server to use the
20 token in conjunction with other interface protocols (i.e., CIFS, NFS). In this way, a storage server providing services using multiple protocols can implement and process tokens in a manner that allows a host using one of the protocols to exchange tokens with another host that uses another of the protocols without either host having any information about the other's protocol and without taking any steps
25 to accommodate the other's protocol.

[0047] In some cases, the inter-protocol token format may be proprietary or vendor specific. Storage devices, storage systems, and/or storage servers that are configured for processing vendor specific inter-protocol tokens can support inter-protocol copy offload processes among hosts using different interface protocols,
30 while devices, systems or servers that are not configured to support these inter-protocol tokens may not be able to support these operations. A vendor specific token format that supports specified interface protocols may also be subsequently extended to accommodate additional interface protocols.

[0048] Figure 4 illustrates storage server 400 in which at least some embodiments of the disclosed techniques may be utilized. Storage server 400 is one example of storage server 340. Storage server 400 includes processor(s) 410, a memory 420, a network adapter 440, and a storage adapter 450. Processor(s) 410 may implement storage management functions to logically organize data as a hierarchical structure of named directories, files, and/or data blocks.

[0049] Network adapter 440 includes one or more ports to couple storage server 400 with one or more clients, hosts, or other storage servers, over point-to-point links, wide area networks, virtual private networks implemented over a public network (Internet), or a shared local area network. Network adapter 440 can include the mechanical components as well as the electrical and signaling circuitry needed to connect storage server 400 to a network. Clients and hosts can communicate with storage server 400 over the network by exchanging packets or frames of data according to pre-defined protocols, such as Transmission Control Protocol/Internet Protocol (TCP/IP).

[0050] Storage adapter 450 provides an interface to information storage devices. The information storage devices may be any type of attached array of writable storage media, such as magnetic disk or tape, optical disk (e.g., CD-ROM or DVD), flash memory, solid-state drive (SSD), electronic random access memory (RAM), micro-electro mechanical, and/or any other similar media adapted to store information, including data and parity information. Storage adapter 450 includes a plurality of ports having I/O interface circuitry that couples with the storage devices over an I/O interconnect arrangement, such as a conventional high-performance, Fibre Channel link topology.

[0051] Memory 420 comprises storage locations that are addressable by processor(s) 410 and adapters 440 and 450 for storing software program code and data associated with the techniques introduced here.

[0052] In one example of the techniques introduced here, processor(s) 410 performs an inter-protocol copy offload process to copy data from one storage location accessed using storage adapter 450 to another storage location accessed using storage adapter 450. In response to a request received from a host using a first interface protocol, processor(s) 410 creates an inter-protocol copy offload token

that represents data to be copied and returns it to the requesting host using network adapter 440. The inter-protocol token is later received by storage server 400 from a different host that uses a different interface protocol. Because the token has been formatted such that processor(s) 410 can use it to accommodate requests
5 associated with both interface protocols, storage server 400 is able to process the token and perform the associated copy offload operation even though the token was created for a host using one interface protocol and returned by a host using a different interface protocol.

[0053] Figure 5 illustrates operating environment 500 in which some
10 embodiments of the disclosed solution may be implemented. Operating environment 500 illustrates a case in which the source storage location and the destination storage location are in two different storage server systems. Operating environment 500 includes SCSI host 510, NFS host 520, SCSI storage server system 580, and NFS storage server system 590. SCSI storage server system 580 is an example of
15 storage server system 330 and is accessed using a SCSI interface. NFS storage server system 590 is also an example of storage server system 330 and is accessed using an NFS interface. SCSI host 510 is an example of host 110 or host 120 that uses SCSI for storing data to or retrieving data from SCSI storage server system 580. NFS host 520 is an example of host 110 or host 120 that uses NFS for storing
20 data to or retrieving data from NFS storage server system 590.

[0054] In one embodiment of the disclosed techniques, NFS storage server system 590 is the source data storage location and SCSI storage server system 580 is the destination data storage location for a copy offload operation. If a data set needs to be copied from NFS storage server system 590 to SCSI storage server
25 system 580, conventional copy offload processes cannot be used because an NFS token would not be understood by the SCSI devices and a SCSI token would not be understood by the NFS devices.

[0055] In the inter-protocol copy offload techniques disclosed here, NFS host 520 transmits a message to NFS storage server system 590 identifying the data to be
30 copied and requesting a token for the data. NFS storage server system 590 identifies the data, creates the inter-protocol token, and transmits the token to NFS host 520. NFS host 520 transfers the token to SCSI host 510 in order to allow SCSI host 510 to have the data copied to its destination of choice. Although the token is

created in an NFS environment, it includes information allowing it to be used by devices of other protocols, such as SCSI, that are properly configured to process inter-protocol tokens of this type.

[0056] In this example, SCSI host 510 makes a determination that SCSI storage server system 580 will be the destination location for the data associated with the token. To initiate the copy portion of the process, SCSI host 510 transmits the token to SCSI storage server system 580. Although SCSI storage server system 580 is a SCSI device and the token is an NFS compatible token, SCSI storage server system 580 is configured to interpret the token. SCSI storage server system 580 identifies NFS storage server system 590 as the location of the source data and transfers the token to NFS storage server system 590. In response to receiving the token, NFS storage server system 590 validates the token and begins transferring the associated data directly to SCSI storage server system 580 without transferring the data through NFS host 520 or SCSI host 510. The token and the data may be transferred between NFS storage server system 590 and SCSI storage server system 580 using SCSI, NFS, or another interface protocol. In some cases, SCSI Storage Server System 580 and NFS Storage Server System 590 may communicate with each other using a proprietary interface protocol.

[0057] It should be understood that the copy offload process described in the example above may also occur in the reverse direction. In this case, SCSI storage server system 580 is the source data storage location and NFS storage server system 590 is the destination storage location. SCSI storage server system 580 generates a token in response to a request from SCSI host 510. After receiving the token, SCSI host 510 transfers the token to NFS host 520. Then, NFS host 520 transfers the token to NFS storage server system 590. Upon receiving the token from NFS storage server system 590, SCSI storage server system 580 transfers the data associated with the token to NFS storage server system 590 without transferring the data to SCSI host 510 or to NFS host 520. In addition to use with the hosts and the storage server systems illustrated in Figure 5, the techniques described herein may also be implemented with host, clients, storage server systems, and/or storage devices using other interface protocols, such as CIFS.

[0058] When an inter-protocol copy offload token is created, the data associated with the token may be subject to change at any time. If the associated data changes

before the token is utilized, the token may no longer be valid because it no longer represents the data that existed at the time the token was created. There are several possible solutions to this problem.

[0059] One solution is to make a copy of the associated data in the source storage device when the token is created. If the original instance of the data changes in the interim, the copy can remain unchanged and can be used to perform the copy offload process. Another possible solution is to invalidate a token whenever the associated data changes. This approach is sometimes called write tracking and requires continuous monitoring of the data for any potential write activities. Although this approach introduces only a moderate amount of additional host computational overhead, it eliminates some of the usefulness of the token approach because the lifetime of the token is unknown and the token may be invalidated before it is used.

[0060] Another possible solution to the problem is referred to as copy-on-write. A copy of the data associated with the token is not created at the time the token is created, but all writes to the device or system are monitored to determine if a write will affect or change any data for which a token has been created. If such a change is detected, a copy of the data is then made, before the writing, such that a copy is maintained that is representative of the state of the data at the time the token was created.

[0061] Yet another solution is to create a persistent point in time image of the data associated with the token when the token is created. In systems where new and changed data is written to new physical storage locations, a persistent point-in-time image remains unchanged even when the underlying data is changed. In some known storage systems, when data is changed, the system does not modify the changed data blocks in place. Rather, it writes each modified data block to a new physical storage location and changes the associated pointers. The original data blocks therefore are retained and remain unchanged. A persistent point-in-time image can be created in such a system such that the image is not truly a "copy" because no actual copies of the underlying data blocks are made when the persistent point-in-time image is created. Instead, the persistent point-in-time image is created by simply creating a new metadata block and storing in it pointers to the

existing data blocks. The result is a read-only view of the underlying data that represents a frozen-time view of the underlying data at the point in time when the persistent point-in-time image was created that does not require copies to be made of any of the underlying data blocks. Systems implementing persistent point-in-time
5 images enable access to older versions of the underlying data. The data associated with the inter-protocol copy offload processes and inter-protocol tokens discussed herein may be maintained using persistent point-in-time images.

[0062] Figure 6 illustrates multi-protocol storage system 690 having inter-protocol copy offload capability. Storage server system 690 includes SCSI protocol interface
10 632, CIFS protocol interface 642, copy system processor 650, and data. SCSI protocol interface 632 and CIFS protocol interface 642 are each communication interfaces used to manage communications with hosts and clients that use the corresponding interface protocols. In Figure 6, SCSI protocol interface 632 is used for communication with SCSI host 610, as well as with other SCSI hosts or SCSI
15 devices. CIFS protocol interface 642 is used for communication with CIFS client 620, as well as with other CIFS clients or devices.

[0063] Copy system processor 650 of storage system 690 includes copy manager 652, token manager 654, and copy engine 658. Copy system processor 650 may be implemented using one or more or a variety of hardware computing devices such as
20 a microprocessor, a CPU, a special-purpose processor programmed with the instructions to perform the steps, or a combination thereof. A variety of the functions associated with copy system processor 650 may also be embodied in machine-executable instructions, which may be used to cause one or more general-purpose computing devices to perform the functions. Alternatively, the functions of copy
25 system processor 650, including the functions of copy manager 652, token manager, 654, and copy engine 658, may be performed by a combination of hardware, software, and/or firmware.

[0064] Storage system 690 provides data storage services to SCSI host 610 and CIFS client 620 and may also provide data storage services to other host or clients,
30 including other hosts or clients using other interface protocols, such as NFS. In order to initiate a copy offload process, SCSI host 610 transmits a token request to storage server system 690. When SCSI protocol interface 632 receives the token request, it communicates the request to copy system processor 650. Copy system

processor 650 invokes token manager 654 to create an inter-protocol token associated with the appropriate data. In this example, the data associated with the token is data A. The inter-protocol token is compatible with a SCSI copy offload standard and also contains information that enables copy system processor 650 to support use of the token in the CIFS environment.

[0065] If SCSI protocol interface 632 receives a request to write data based on the inter-protocol token, it again communicates with copy system processor 650. Copy system processor 650 invokes token manager 654 to determine which data is associated with the token and invokes copy engine 658 to copy the data. Copy engine 658 is able to copy data A to any location within storage system 690 regardless of the protocol used to create the token and regardless of the protocol used for receiving the token.

[0066] If CIFS protocol interface 642 receives a request for a token, it communicates the request to copy system processor 650. Copy system processor 650 invokes token manager 654 to create the inter-protocol token and associate the appropriate data with the token. When CIFS protocol interface 642 receives a request to write the data associated with the token, it again communicates with copy system processor 650. Copy system processor 650 invokes token manager 654 to determine which data is associated with the token and invokes copy engine 658 to copy the data. Copy engine 658 copies the data from any location to any other location within storage system 690 regardless of which protocol is used to create the token and regardless of which protocol is used to receive the request to copy the data (i.e., receive the token).

[0067] The data transfers that result from the inter-protocol copy offload operations flow through copy engine 658 but do not flow through SCSI host 610 or CIFS client 620. As illustrated in some of the other examples, the destination location for the data being copied may also be in another storage device or system.

[0068] Figure 7 illustrates communication between two hosts and a storage device for performing an inter-protocol copy offload process in one embodiment of the techniques introduced here. In some cases, the storage device of Figure 7 may include the physical storage medium that data is stored on. In other cases, the

storage device of Figure 7 may be a storage server, storage management device, or storage control device that does not include the physical storage media.

[0069] In the example of Figure 7, host A transmits a tokenized read message to the storage device using a first interface protocol. The tokenized read message includes an identification of the data that is to be copied. The storage device identifies the data to be copied and generates an inter-protocol copy offload token. In some cases the storage device also returns a status message to host A. In some cases, the storage device may transmit the token back to host A in response to the request to create the token. In some cases, to obtain the token, host A may transmit a fetch token message to the storage device. In response, the storage device transmits the token to host A using the first interface protocol. In some cases, this may also be followed by a status message.

[0070] At a later point in time, host A transfers the token to host B and host B transfers the token to the storage device in a tokenized write message using a second interface protocol that is different than the first interface protocol. In some cases, the tokenized write message includes a destination location for the copy of the data. Even though the token is received in a communication associated with the second interface protocol, the storage device is programmed to determine that the token was created for a host using the first interface protocol and is able to decode and process the token accordingly. After performing any necessary verification or validation processes, the storage device then copies the data associated with the token to the destination. Finally, the storage device transmits a status message back to host B in the second interface protocol when the process is complete.

[0071] Figure 8 illustrates a variation of the example of Figure 7 in which the source data location and the destination storage location for the inter-protocol copy offload process are in different storage devices. Similar to the communication process illustrated in Figure 7, host A transmits a tokenized read message to storage device A in a first interface protocol and receives an inter-protocol token in response to a fetch token request message. Then, host A transfers the inter-protocol token to host B.

[0072] In order to complete the inter-protocol copy offload process, host B transmits the token to a storage device B that is the destination for the copy offload

process. The token is transmitted in a tokenized write message using a second interface protocol. The tokenized write message includes the inter-protocol token. Even though storage device B uses the second interface protocol and the token was created in a format associated with the first interface protocol, storage device B is
5 configured to decode the token and determine that it is associated with storage device A. This is due to additional information encoded in the token and the ability of storage device B to interpret that additional information. Storage device B transmits the received token to storage device A. Storage device A uses the token to identify and locate the data and transmits the associated data blocks to storage device B in
10 response to receiving the token.

[0073] In some cases, additional messages may be exchanged between the storage devices in conjunction with the data transfer. For example, storage device B may first send the token to device A to have the data prepared and then send additional commands to extract the prepared data from device A. Once the transfer
15 is complete, storage device B may transmit a status message to the host. The transfer of the data from the source to the destination may be conducted using a push model or a pull model. In a push model, the source storage device, after receiving and verifying the token, controls the process through which the data is transferred to the destination storage device. In a pull model, the destination storage
20 controls the transfer process by retrieving or reading the data from the source storage device after the token has been verified. As illustrated, the data is transferred between the storage devices to complete the copy offload process and is not transmitted to or through either of the hosts.

[0074] Embodiments of the techniques introduced here may be provided as a
25 computer program product, which may include a machine-readable medium having stored thereon non-transitory instructions which may be used to program a computer or other electronic device to perform some or all of the operations described herein. The machine-readable medium may include, but is not limited to optical disks, compact disc read-only memories (CD-ROMs), magneto-optical disks, floppy disks,
30 ROMs, random access memories (RAMs), erasable programmable read-only memories (EPROMs), electrically erasable programmable read-only memories (EEPROMs), magnetic or optical cards, flash memory, or other type of machine-readable medium suitable for storing electronic instructions. Moreover,

embodiments of the solution introduced here may also be downloaded as a computer program product, wherein the program may be transferred from a remote computer to a requesting computer by way of data signals embodied in a carrier wave or other propagation medium via a communication link.

5 **[0075]** The phrases “in some embodiments,” “according to some embodiments,” “in the embodiments shown,” “in other embodiments,” “in some cases,” “in some examples,” and the like generally mean the particular feature, structure, or characteristic following the phrase is included in at least one embodiment of the solution introduced here, and may be included in more than one embodiment. In
10 addition, such phrases do not necessarily refer to the same embodiments or different embodiments.

[0076] While detailed descriptions of one or more embodiments of the invention have been given above, various alternatives, modifications, and equivalents will be apparent to those skilled in the art without varying from the spirit of the invention.
15 For example, while the embodiments described above refer to particular features, the scope of this invention also includes embodiments having different combinations of features and embodiments that do not include all of the described features. Accordingly, the scope of the present invention is intended to embrace all such alternatives, modifications, and variations as fall within the scope of the claims,
20 together with all equivalents thereof. Therefore, the above description should not be taken as limiting the scope of the invention, which is defined by the claims.

CLAIMS

What is claimed is:

- 5 1. A method comprising:
- receiving, in a first interface protocol from a host device, a request to copy a data set from a source data storage location to a destination data storage location, wherein the request includes a token representing the data set, the token having been created using a second interface protocol different from the first interface
- 10 protocol; and
- transferring the data set, in response to receiving the request, from the source data storage location to the destination data storage location without transferring the data set to the host device.
- 15 2. The method of claim 1 wherein the source data storage location and the destination data storage location are in the same storage device and transferring the data set includes making a copy of the data set and storing the copy in the destination storage location.
- 20 3. The method of claim 1 wherein the source data storage location and the destination data storage location are in different storage devices and the data is transferred from the source data storage location to the destination data storage location using a third interface protocol.
- 25 4. The method of claim 1 further comprising, prior to receiving the request to copy the data set:
- receiving, from a requesting device using the second interface protocol, a request to generate the token;
- generating the token based on information in the request to generate the
- 30 token that identifies the data set; and
- transmitting the token to the requesting device using the second interface protocol.

5. The method of claim 4 wherein generating the token includes creating a persistent point-in-time image of the data set in the source data storage location by generating pointers to data blocks of the data set and without duplicating the data blocks.

5

6. The method of claim 5 wherein transferring the data set includes transferring the persistent point-in-time image.

7. The method of claim 1 wherein the first interface protocol is one of: Small
10 Computer System Interface (SCSI), Common Interface File System (CIFS), or Network File System (NFS).

8. The method of claim 1 wherein the second interface protocol is one of: SCSI, CIFS, or NFS.

15

9. The method of claim 1 wherein the token has a vendor-specific format and contains information indicating the vendor-specific format.

10. The method of claim 1 wherein transferring the data set from the source data
20 storage location to the destination data storage location comprises a copy offload operation.

11. The method of claim 10 wherein the copy offload operation is a SCSI T10 compliant copy offload operation and the token is compliant with the SCSI T10 SCSI
25 Primary Commands 4 (SPC-4) standard.

12. The method of claim 4 wherein the request to generate the token is a tokenized read message and the request to copy the data set is a tokenized write message.

30

13. A data storage system comprising:
a first communication interface using a first interface protocol;
a second communication interface using a second interface protocol; and
a copy system processor configured to:
- 5 receive, at the first communication interface from a host device, a request to copy a data set from a source storage location in a data storage device to a destination storage location, wherein the request includes a token representing the data set, the token having been created using the second communication interface;
- 10 validate the token; and
transfer the data set, in response to validating the token, from the source storage location to the destination storage location without transferring the data set to the host device.
- 15 14. The data storage system of claim 13 wherein the destination storage location is in the data storage device and to transfer the data set includes to store a copy of the data set in the destination storage location.
- 20 15. The data storage system of claim 13 wherein the request to copy the data set is received in a Small Computer System Interface (SCSI) tokenized write message.
16. The data storage system of claim 13 wherein the copy system processor is further configured to, prior to receiving the request to copy the data set:
receive, from a requesting device through the second communication
25 interface, a request to generate the token;
generate the token based on information in the request to generate the token that identifies the data set; and
transmit the token to the requesting device using the second communication interface.
- 30 17. The data storage system of claim 16 wherein the request to generate the token is a SCSI tokenized read message.

18. The data storage system of claim 16 wherein the host device is the requesting device.
19. The data storage system of claim 16 wherein to generate the token includes
5 to create a persistent point-in-time image of the data set in the source data storage device by generating pointers to data blocks of the data set without duplicating the data blocks.
20. The data storage system of claim 19 wherein to transfer the data set includes
10 to transfer the persistent point-in-time image.
21. The data storage system of claim 13 wherein the data set is transferred from the data storage device to the destination storage location using a proprietary interface protocol that is different than the first interface protocol and different than
15 the second interface protocol.
22. The data storage system of claim 13 wherein the first interface protocol is one of: SCSI, Common Interface File System (CIFS), or Network File System (NFS).
- 20 23. The data storage system of claim 13 wherein the second interface protocol is one of: SCSI, CIFS, or NFS.
24. The data storage system of claim 13 wherein the token contains information indicating that the token has a vendor-specific format.
25
25. The data storage system of claim 13 wherein the data storage system is operated in a Network Attached Storage (NAS) environment.
26. The data storage system of claim 13 wherein the data storage system is
30 operated in a Storage Area Network (SAN).

27. A method comprising:
- receiving, at a token manager in a storage device from a first host device in a first protocol, a token request for a copy offload process;
 - generating a token in response to the token request, wherein the token is a representation of a data set in a source storage location;
 - transmitting the token to the first host device using the first protocol;
 - receiving the token at the token manager from a second host device in a second protocol that is not the first protocol; and
 - transferring the data set, in response to receiving the token, from the source storage location to a destination storage location without transferring the data set to the first host device and without transferring the data set to the second host device.
28. The method of claim 27 wherein the first interface protocol and the second protocol are each one of: Small Computer System Interface (SCSI), Common Interface File System (CIFS), or Network File System (NFS).
29. The method of claim 27 wherein the copy offload process is a SCSI T10 compliant process.

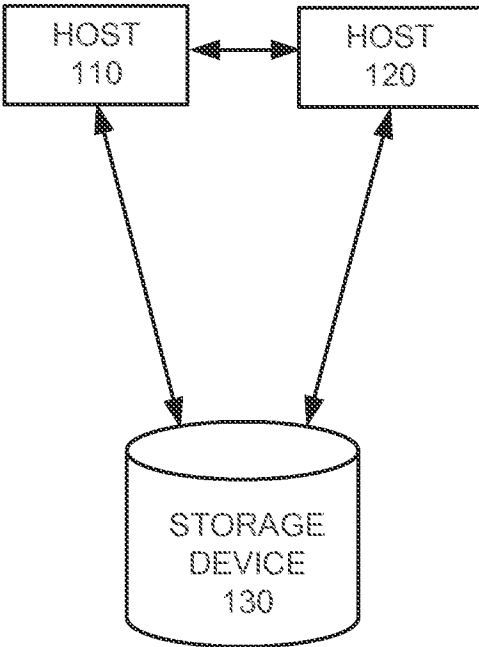


FIG. 1

2/8

200

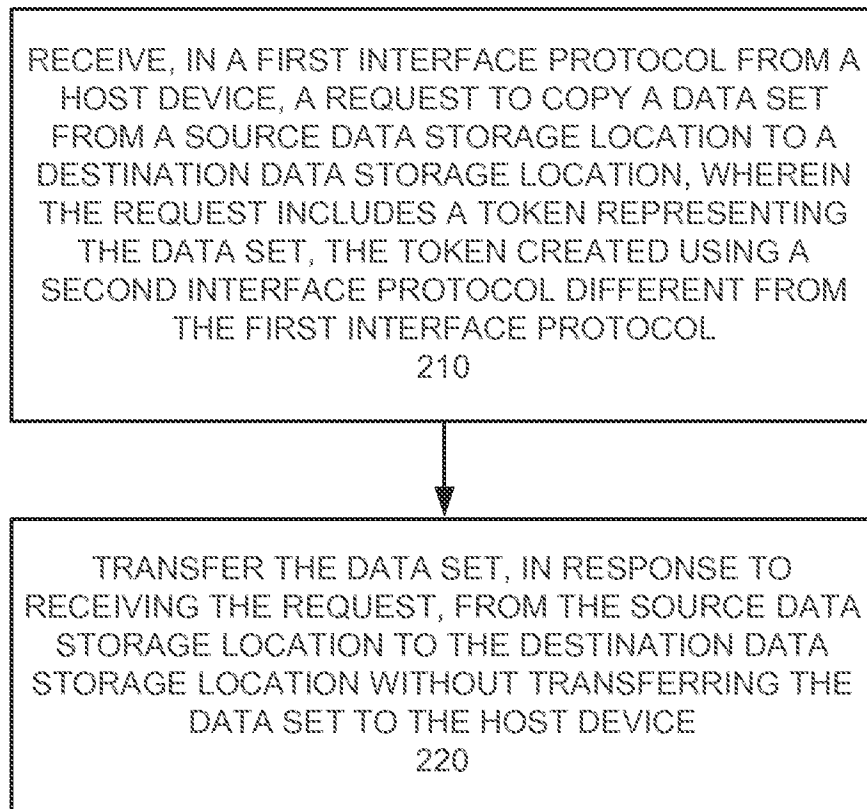


FIG. 2

3/8

300

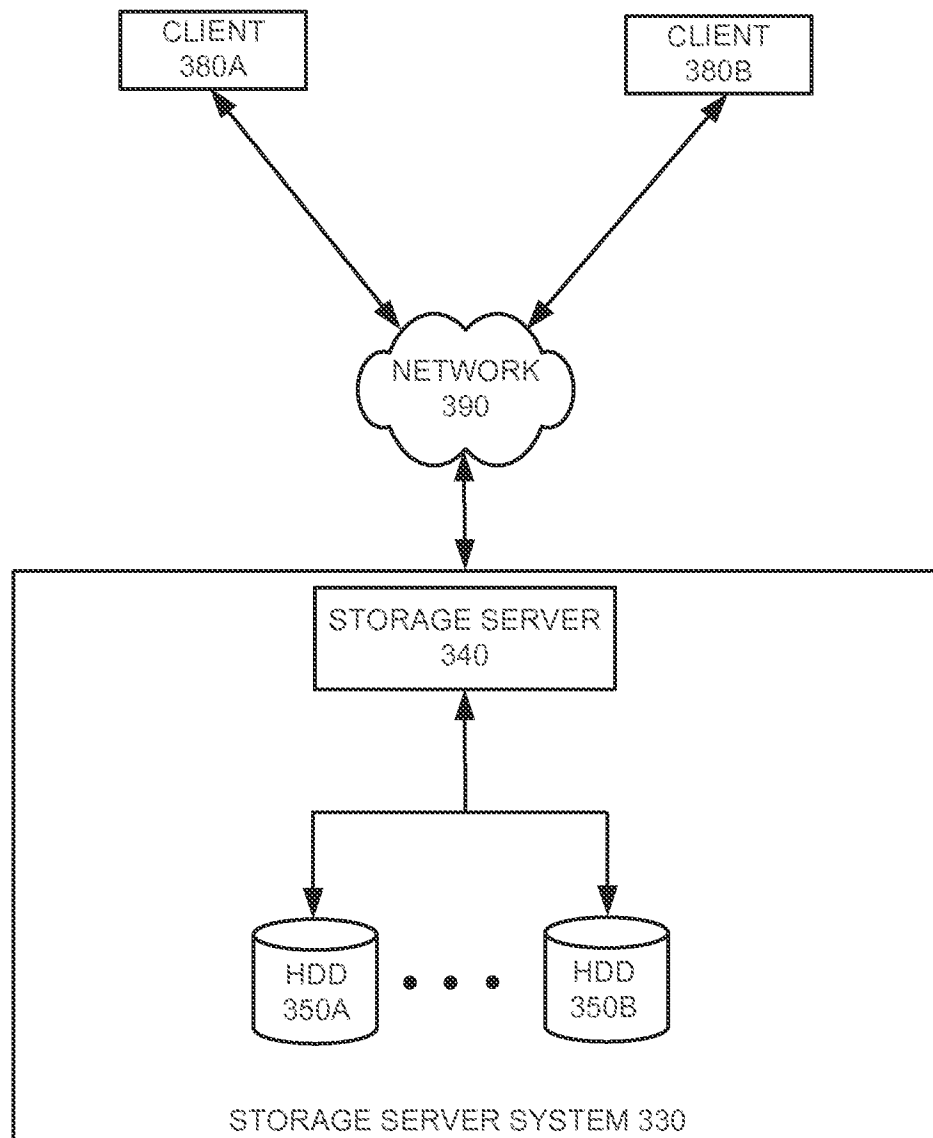


FIG. 3

4/8

400

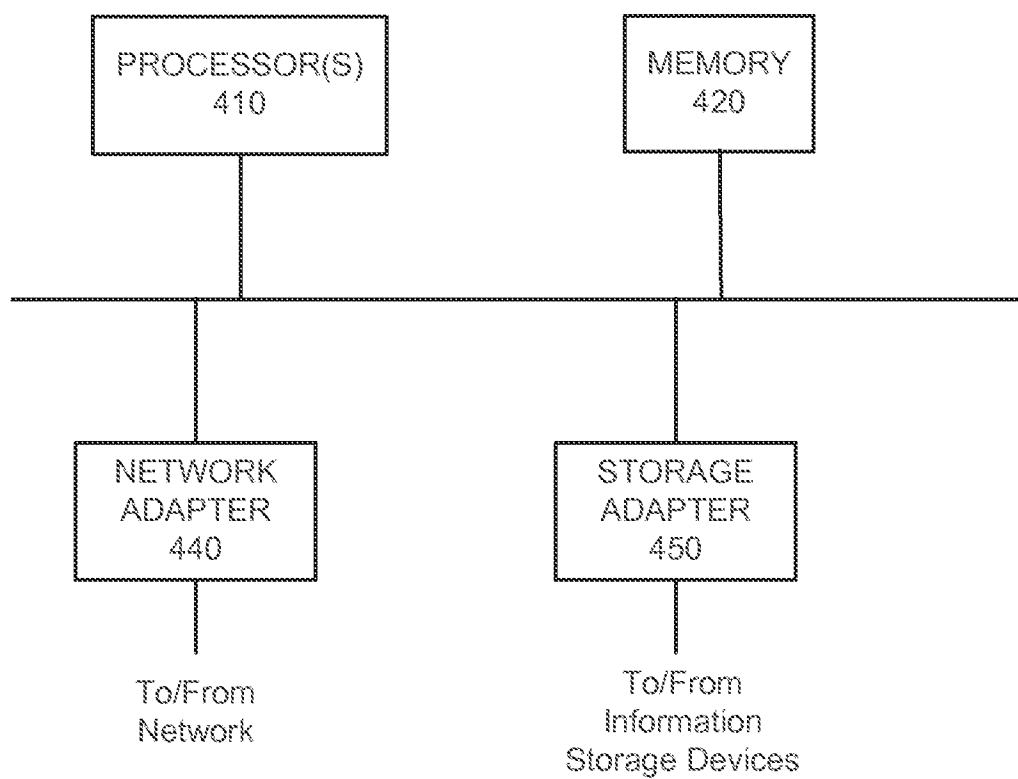


FIG. 4

5/8

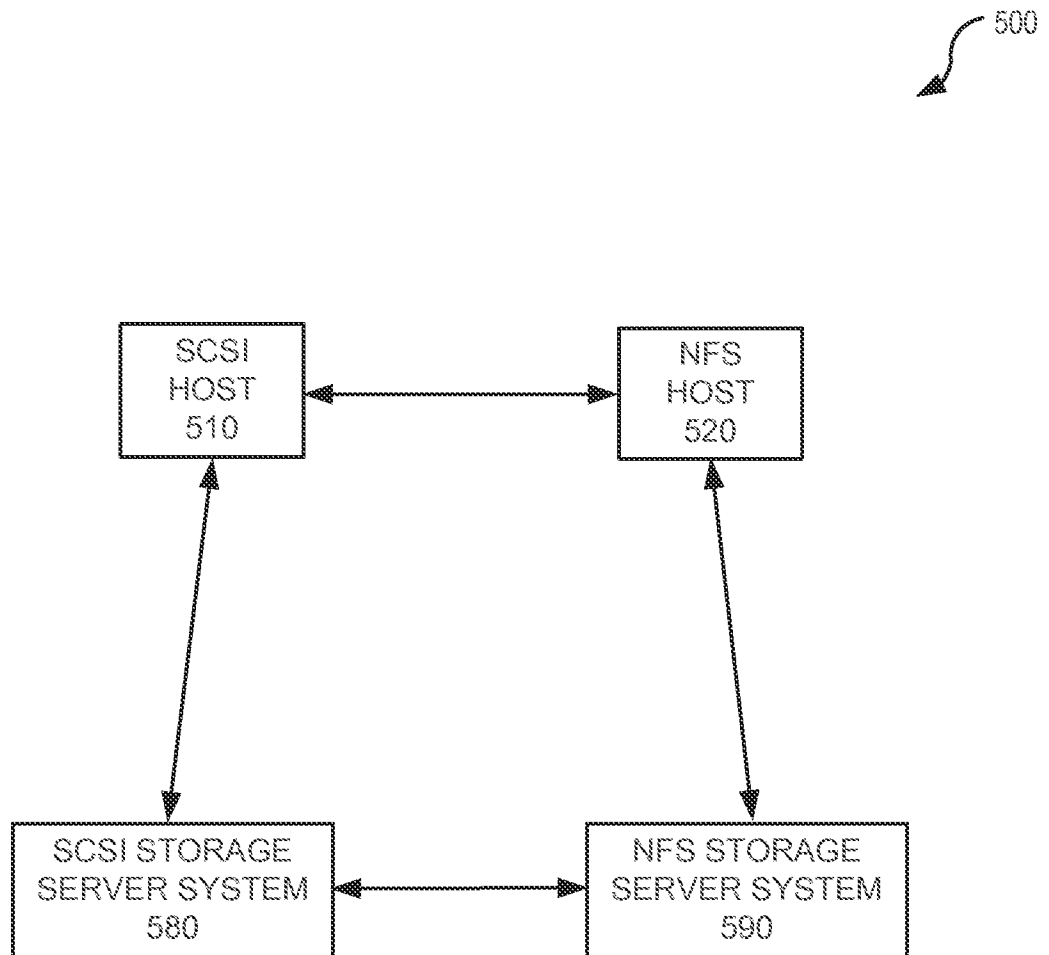


FIG. 5

6/8

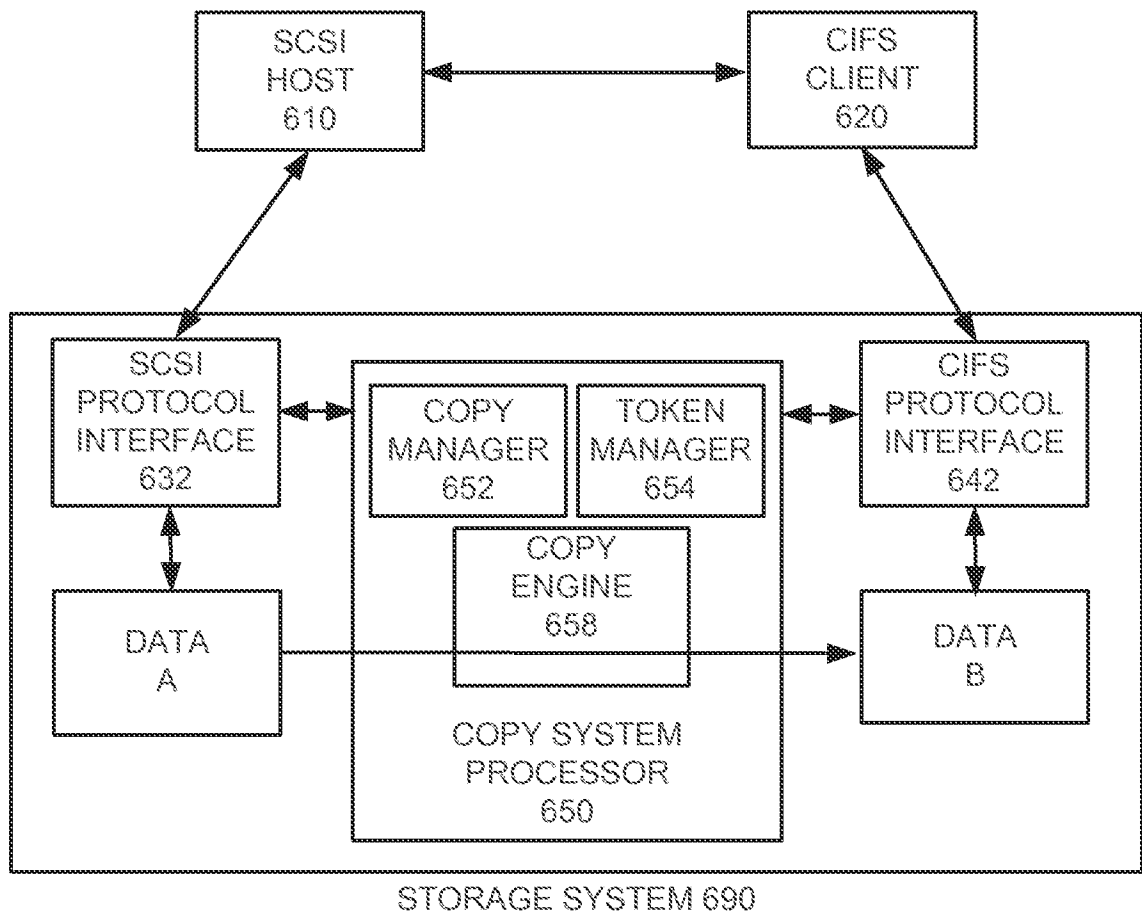


FIG. 6

7/8

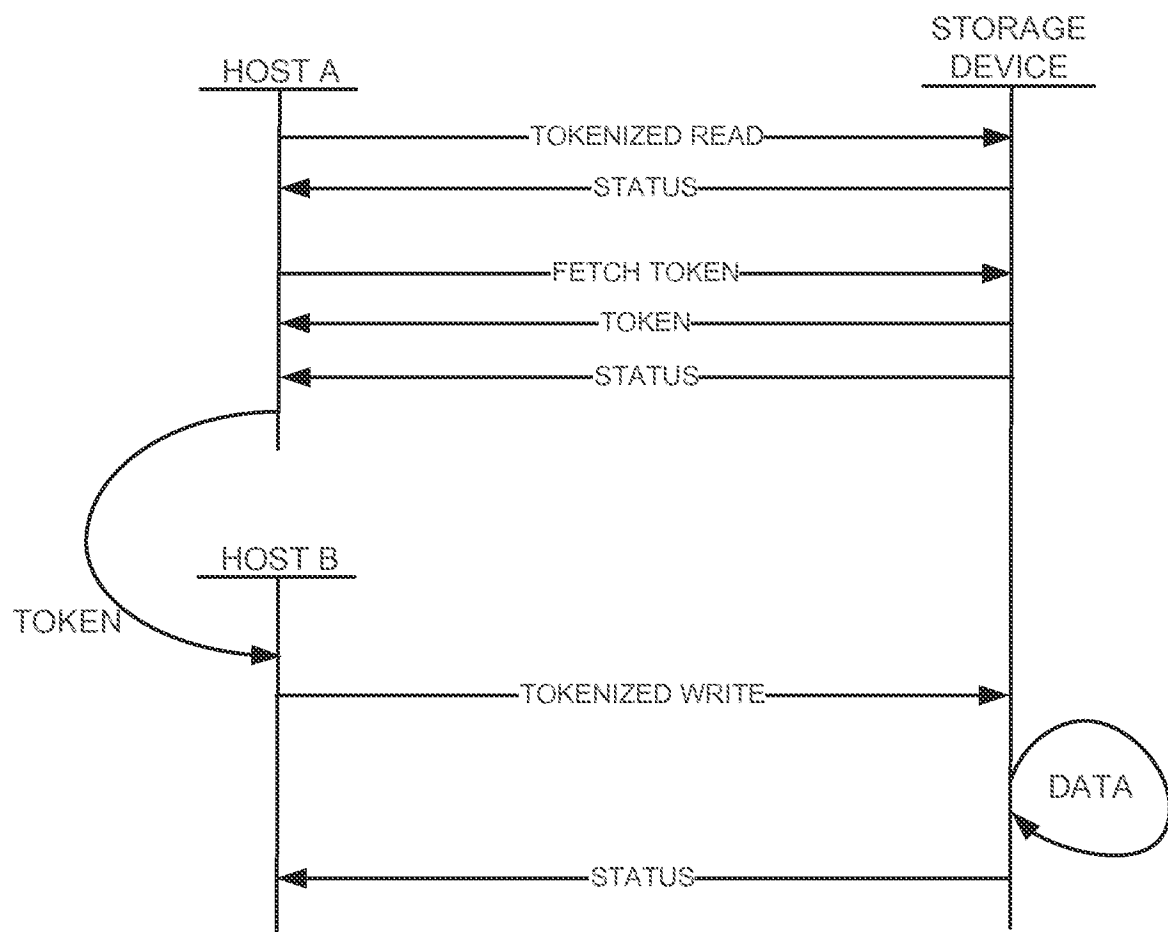


FIG. 7

8/8

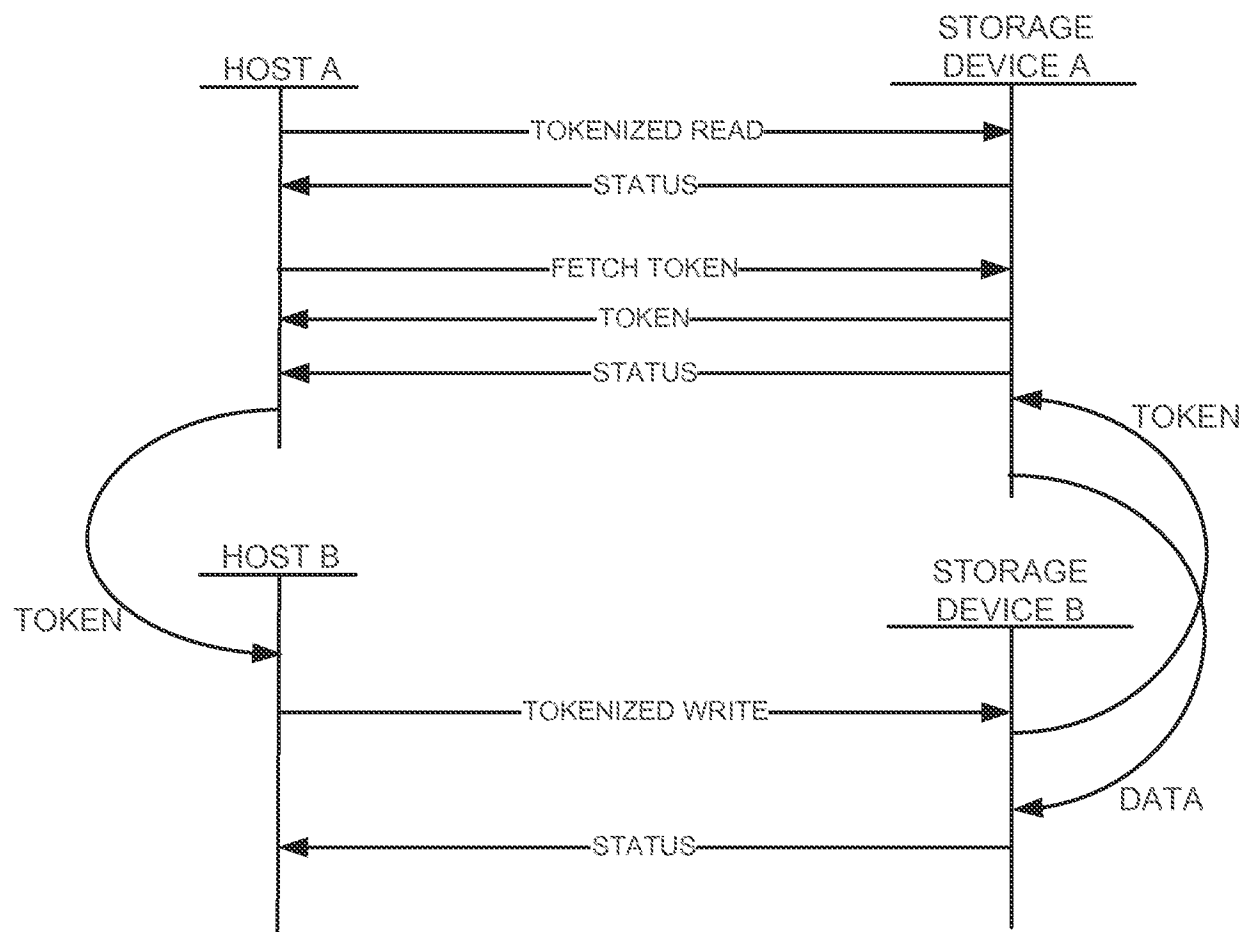


FIG. 8

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2013/070046**A. CLASSIFICATION OF SUBJECT MATTER****G06F 12/14(2006.01)i, G06F 12/16(2006.01)i**

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHEDMinimum documentation searched (classification system followed by classification symbols)
G06F 12/14; G06F 7/00; H04L 12/54; H04L 9/32; H04L 12/28; H04L 12/56; G06F 12/16Documentation 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 modelsElectronic data base consulted during the international search (name of data base and, where practicable, search terms used)
eKOMPASS(KIPO internal) & Keywords: interface protocol, host, copy offload, duplicate, multiple, different, token, identifier, storage, SCSI, CIFS, NFS, NAS, SAN, and similar terms.**C. DOCUMENTS CONSIDERED TO BE RELEVANT**

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2006-0072580 A1 (FRANK DROPPS R. et al.) 06 April 2006 See paragraphs [0006]–[0016] and [0018]–[0020]; claims 8–9; and figure 1A.	1–29
A	US 2012-0079583 A1 (NEAL R. CHRISTIANSEN et al.) 29 March 2012 See paragraphs [0003]–[0004], [0029]–[0080], [0119]–[0127], and [0139]–[0150]; and figures 2, 4–5, and 7–8.	1–29
A	WO 2007-042954 A2 (KONINKLUKE PHILIPS ELECTRONICS N.V.) 19 April 2007 See page 2, line 10 – page 3, line 6; page 6, line 26 – page 10, line 10; claim 1; and figures 3–4.	1–29
A	US 2012-0059797 A1 (ANAND PRAHLAD et al.) 08 March 2012 See paragraphs [0029]–[0049] and figures 2–4.	1–29
A	US 2004-0233910 A1 (WEN-SHYEN CHEN et al.) 25 November 2004 See paragraphs [0038]–[0056] and figures 3–4.	1–29

☐ 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

26 February 2014 (26.02.2014)

Date of mailing of the international search report

26 February 2014 (26.02.2014)

Name and mailing address of the ISA/KR

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

Facsimile No. +82-42-472-7140

Authorized officer

NHO, Ji Myong

Telephone No. +82-42-481-8528



INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2013/070046

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2006-0072580 A1	06/04/2006	AT 438135 T CN 100485598 C CN 101040251 A DE 602005015743 D1 EP 1807753 A2 EP 1807753 B1 US 7411958 B2 WO 2006-039347 A2 WO 2006-039347 A3	15/08/2009 06/05/2009 19/09/2007 10/09/2009 18/07/2007 29/07/2009 12/08/2008 13/04/2006 17/08/2006
US 2012-0079583 A1	29/03/2012	AR 083102 A1 AU 2011-305839 A1 CA 2810833 A1 CN 102520877 A EP 2619652 A2 JP 2013-539119 A TW 201224914 A WO 2012-039939 A2 WO 2012-039939 A3	30/01/2013 21/03/2013 29/03/2012 27/06/2012 31/07/2013 17/10/2013 16/06/2012 29/03/2012 31/05/2012
WO 2007-042954 A2	19/04/2007	WO 2007-042954 A3	11/10/2007
US 2012-0059797 A1	08/03/2012	US 2006-010154 A1 US 2009-248762 A1 US 2013-006944 A1 US 7546324 B2 US 8078583 B2 US 8266106 B2 US 8577844 B2 WO 2005-050381 A2 WO 2005-050381 A3 WO 2005-050381 B1	12/01/2006 01/10/2009 03/01/2013 09/06/2009 13/12/2011 11/09/2012 05/11/2013 02/06/2005 06/04/2006 11/05/2006
US 2004-0233910 A1	25/11/2004	US 2002-0133746 A1 US 2003-0033523 A1 US 2003-0135709 A1 US 2006-0236064 A1 US 2006-0277419 A1 US 6715098 B2 US 7058788 B2 US 7093127 B2 US 7330960 B2 US 7469337 B2 WO 2002-069153 A1 WO 2002-069159 A1 WO 2002-075543 A1	19/09/2002 13/02/2003 17/07/2003 19/10/2006 07/12/2006 30/03/2004 06/06/2006 15/08/2006 12/02/2008 23/12/2008 06/09/2002 06/09/2002 26/09/2002