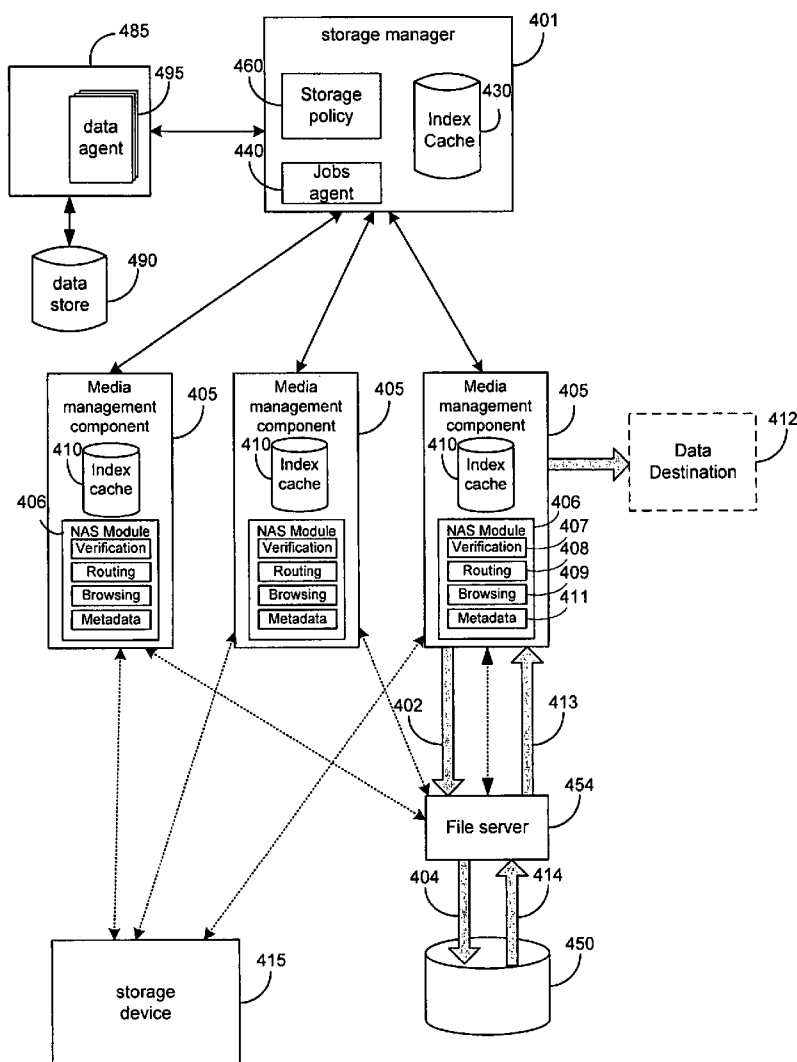


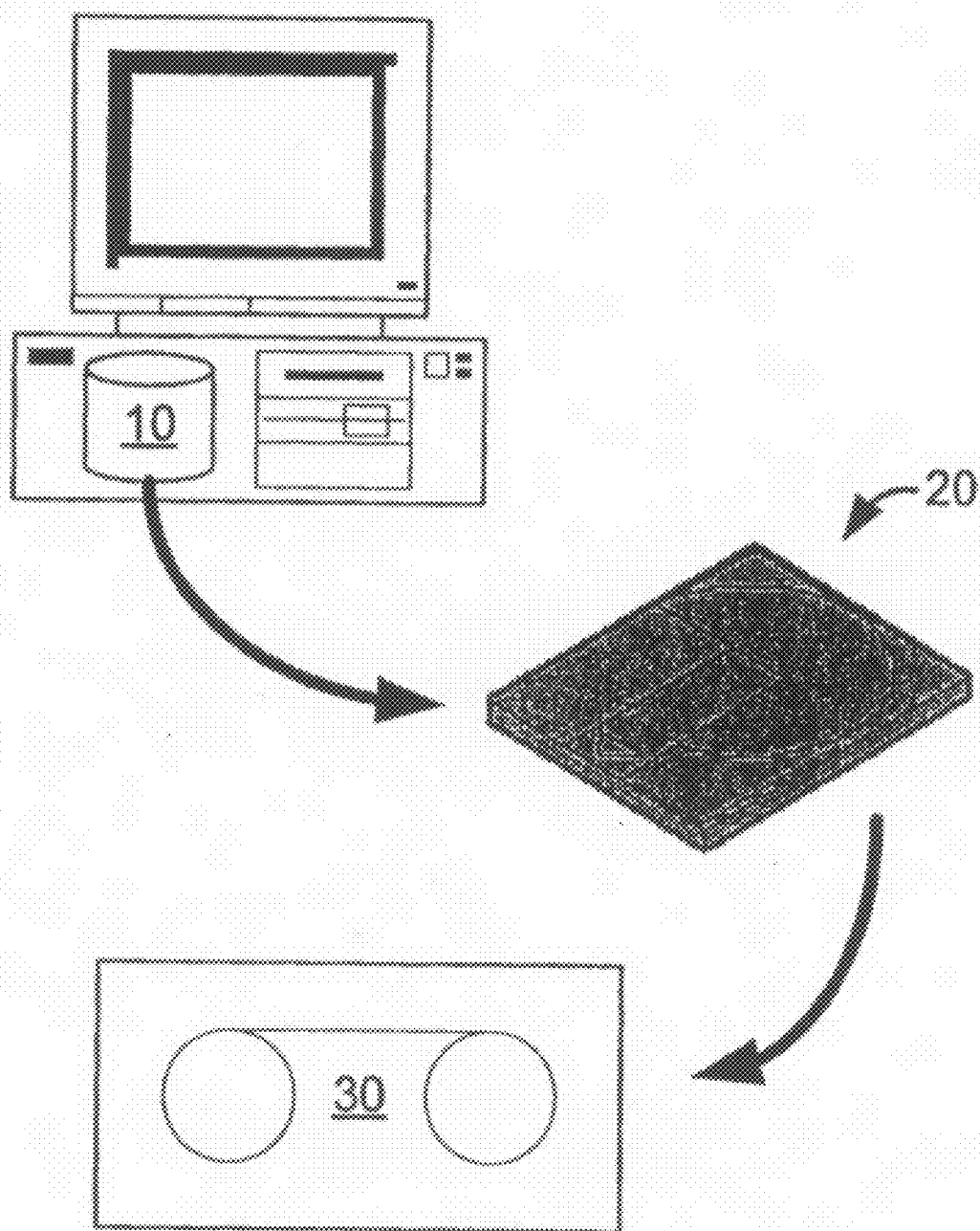


US 20080147754A1

(19) **United States**(12) **Patent Application Publication**
Littlefield et al.(10) **Pub. No.: US 2008/0147754 A1**(43) **Pub. Date: Jun. 19, 2008**(54) **SYSTEMS AND METHODS FOR
FACILITATING STORAGE OPERATIONS
USING NETWORK ATTACHED STORAGE
DEVICES****Related U.S. Application Data**(63) Continuation of application No. 11/640,730, filed on
Dec. 18, 2006.**Publication Classification**(51) **Int. Cl.**
G06F 17/30 (2006.01)
(52) **U.S. Cl.** **707/204; 707/E17.005**
(57) **ABSTRACT**Correspondence Address:
MCDERMOTT WILL & EMERY LLP
600 13TH STREET, N.W.
WASHINGTON, DC 20005-3096

A system and method for communicating, browsing, verifying and routing data in storage operation systems using network attached storage devices is provided. In some embodiments, the system may include a management module and a media management component connected to the management server, which interoperate with network attached storage devices to provide the communicating, browsing, verifying and routing functions.

(21) Appl. No.: **11/825,204**(22) Filed: **Jul. 5, 2007**400



(PRIOR ART)

Fig. 1

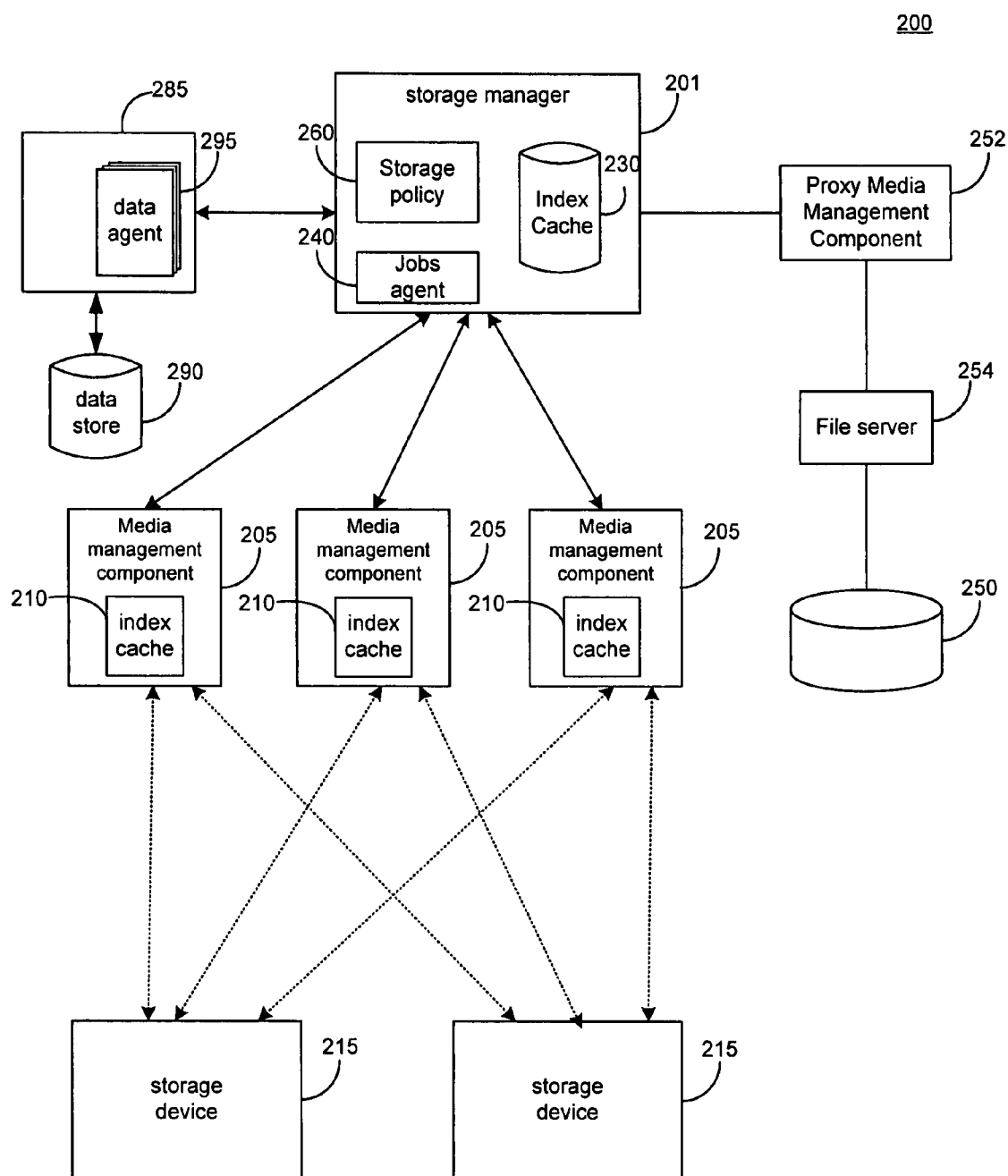


FIG. 2

Prior Art

310

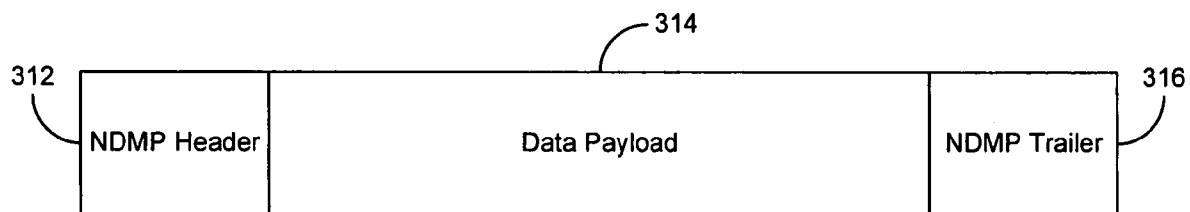


FIG. 3

Prior Art

400

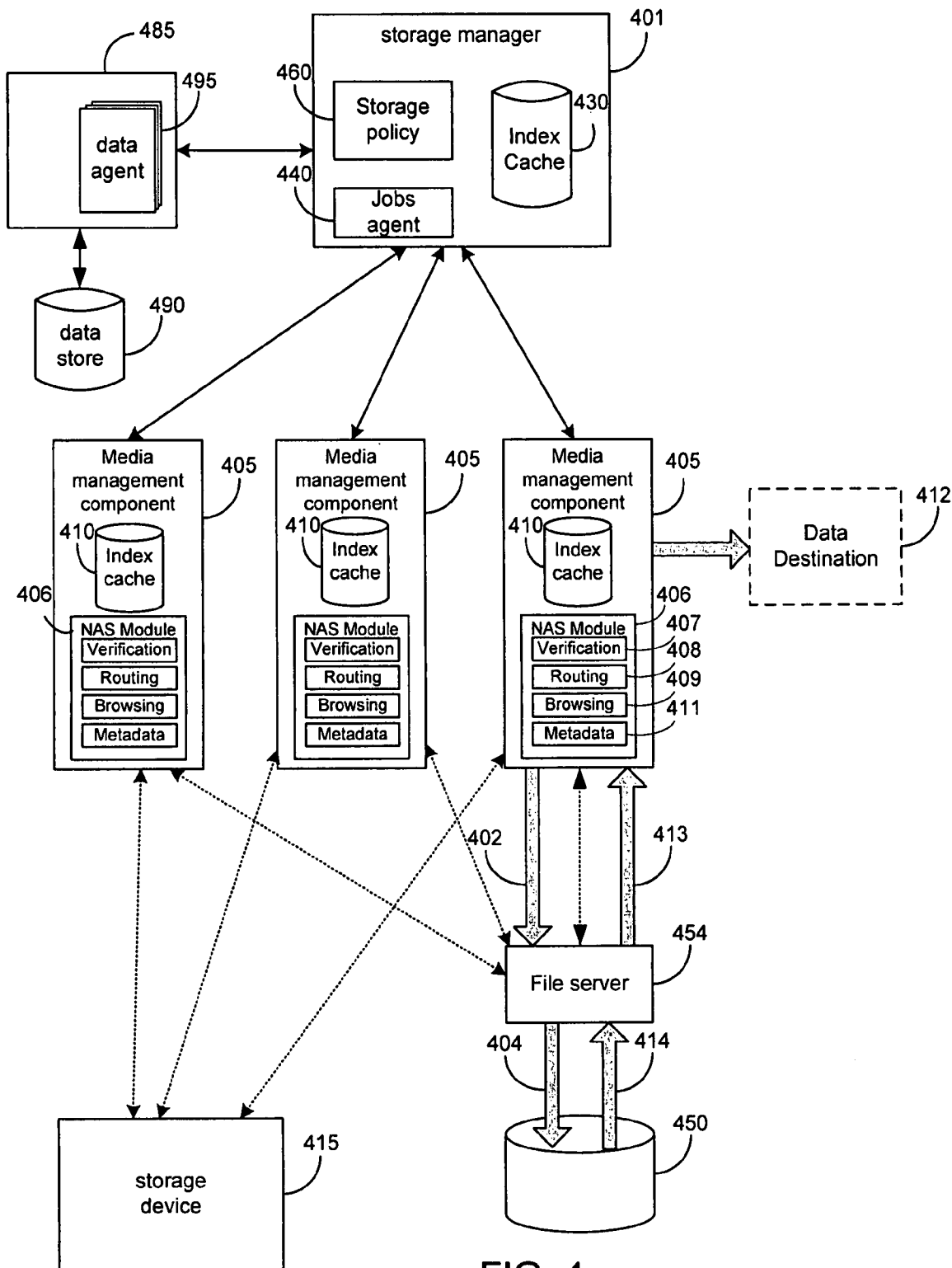


FIG. 4

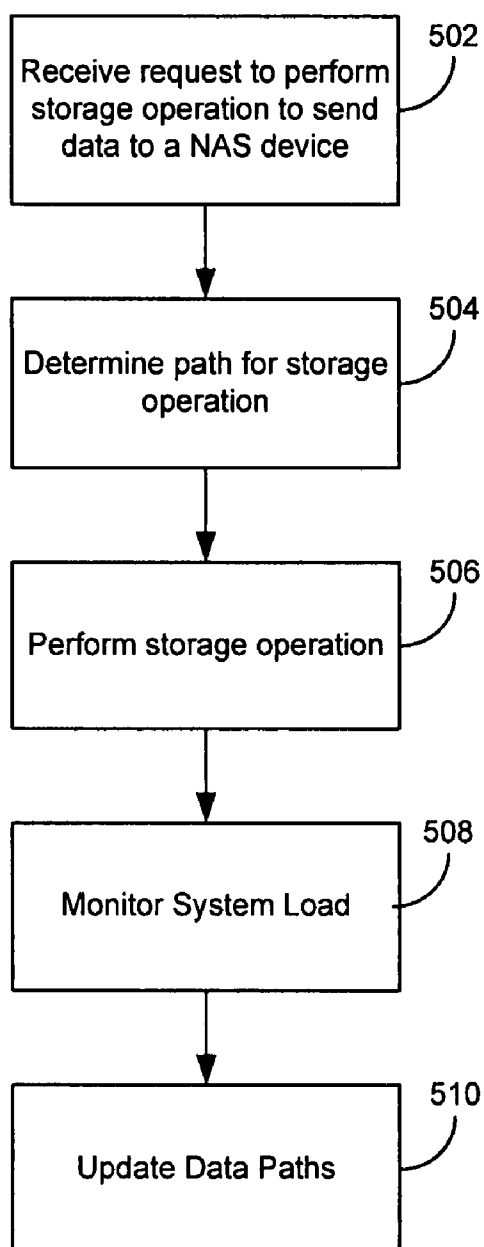


FIG. 5

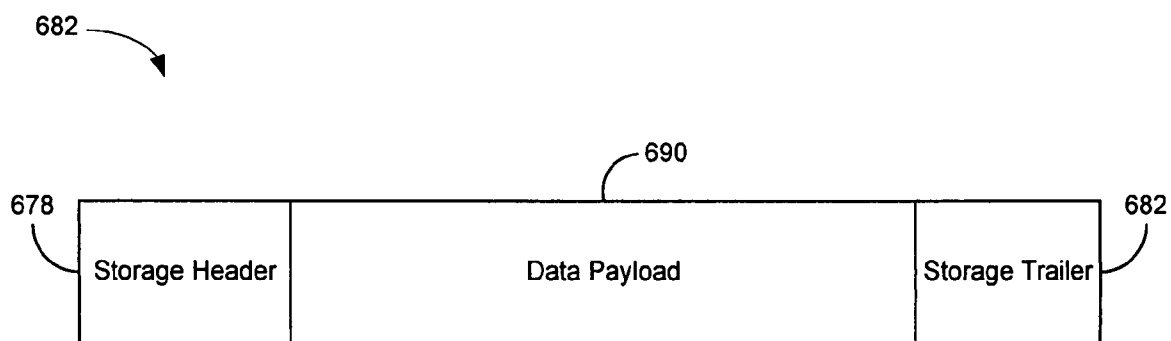


FIG. 6

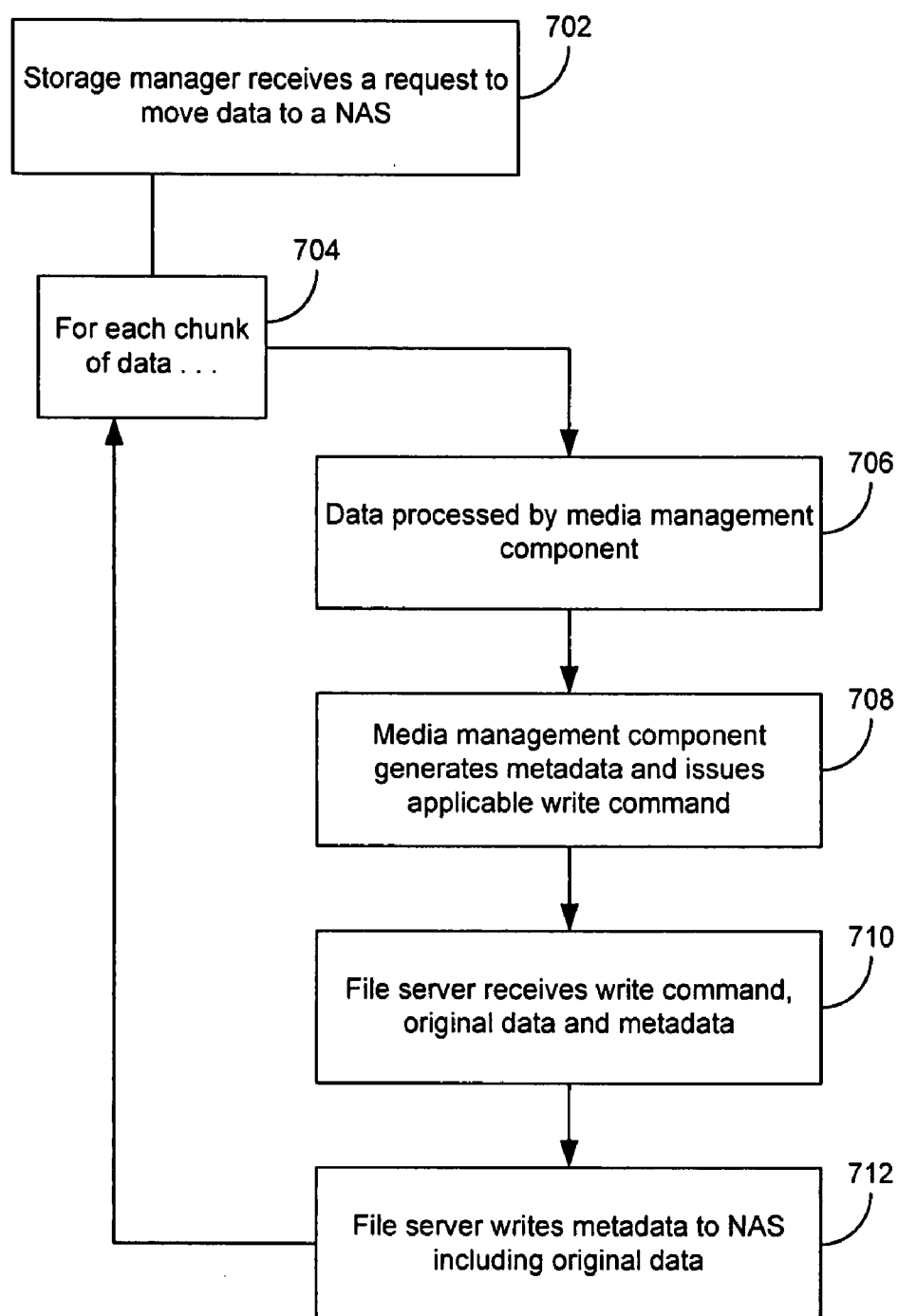


FIG. 7

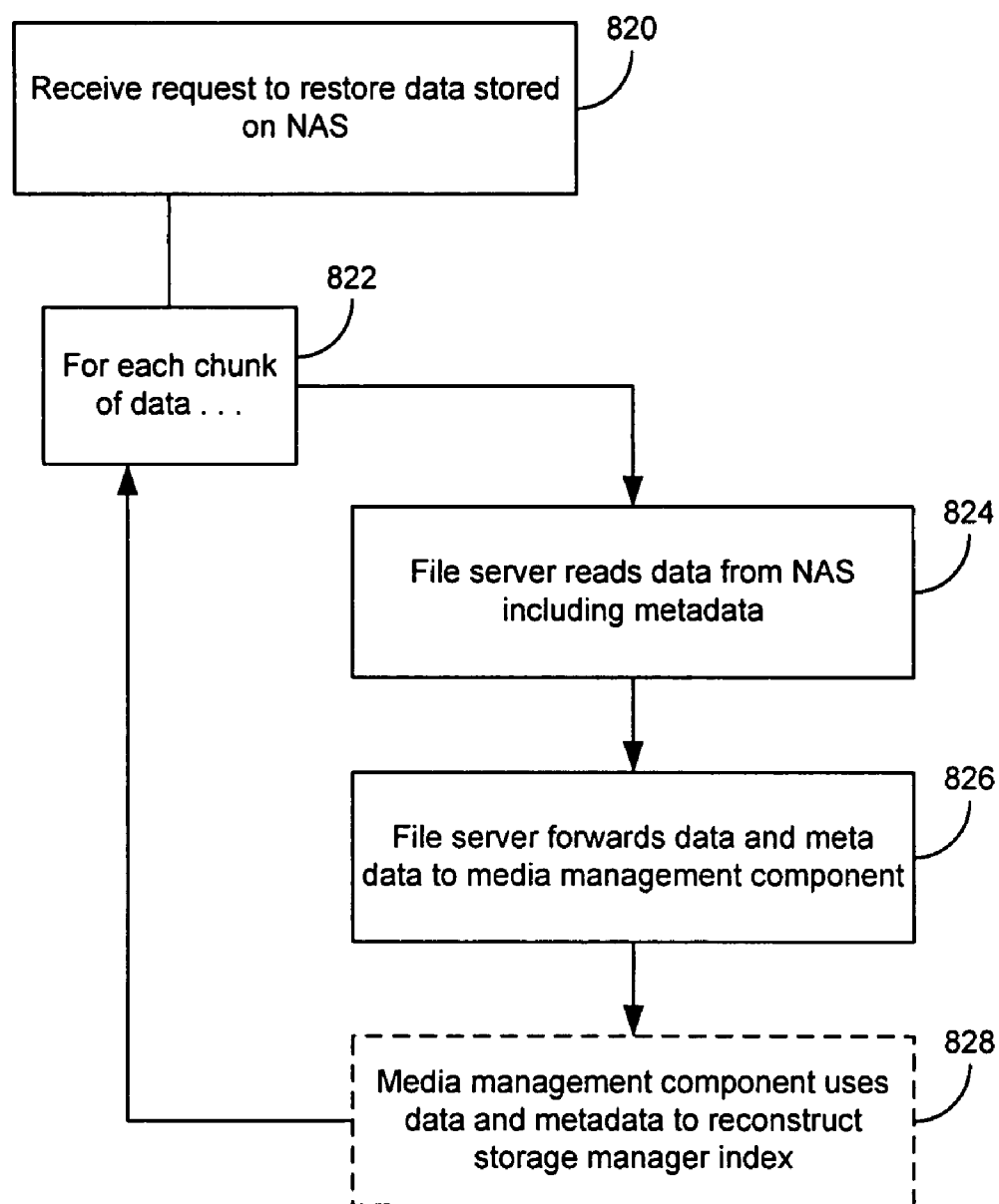


FIG. 8

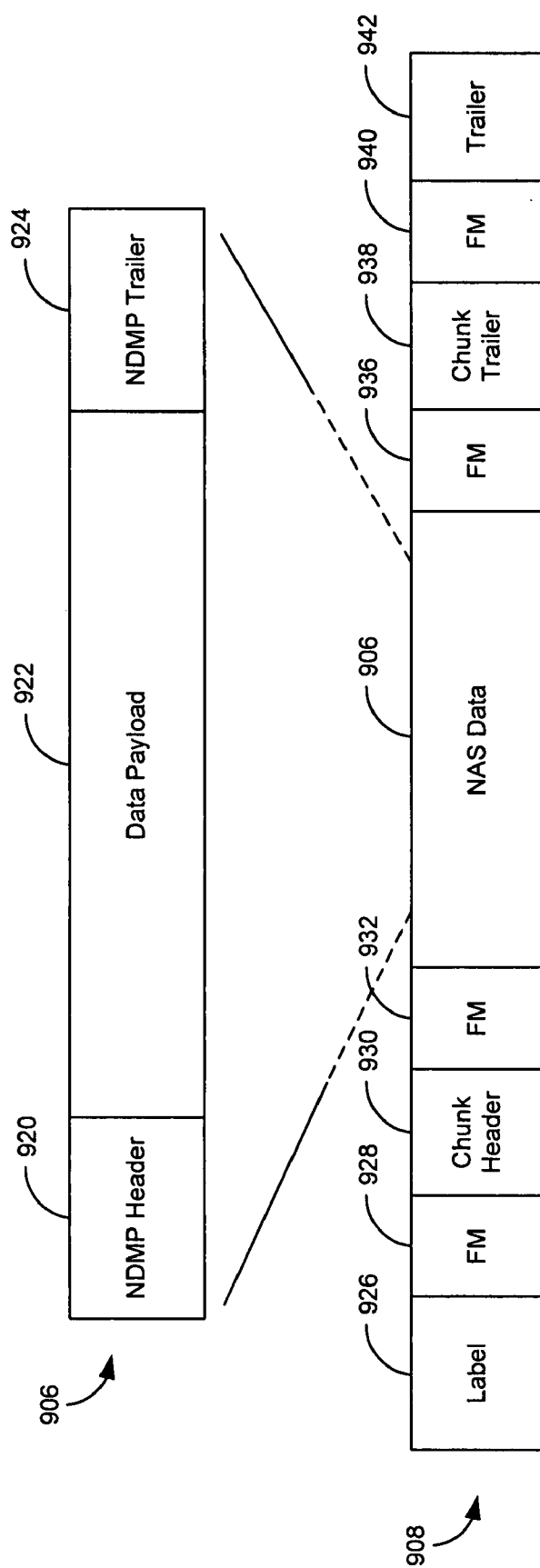


FIG. 9

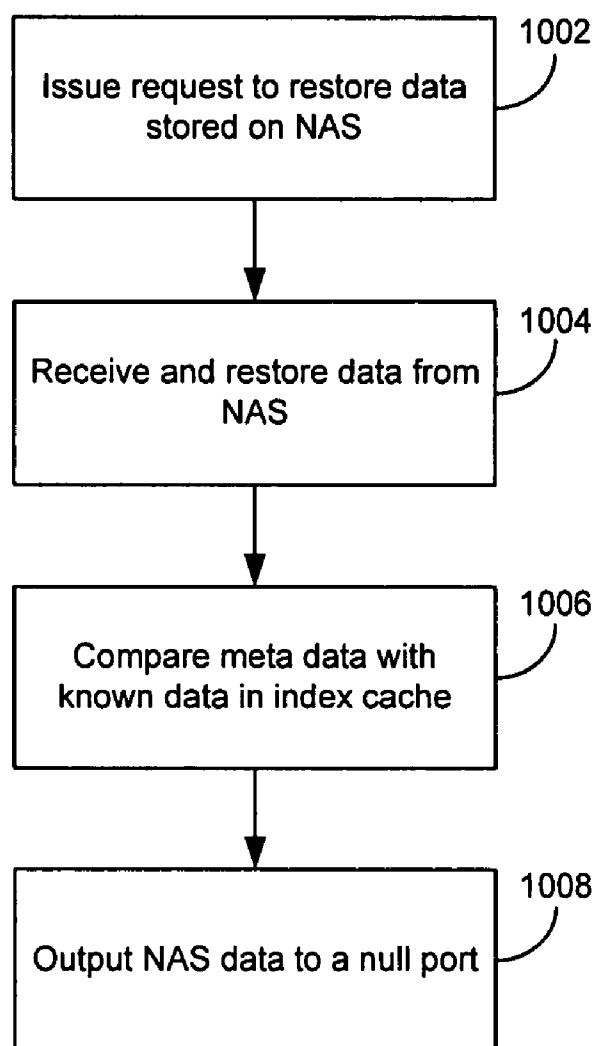


FIG. 10

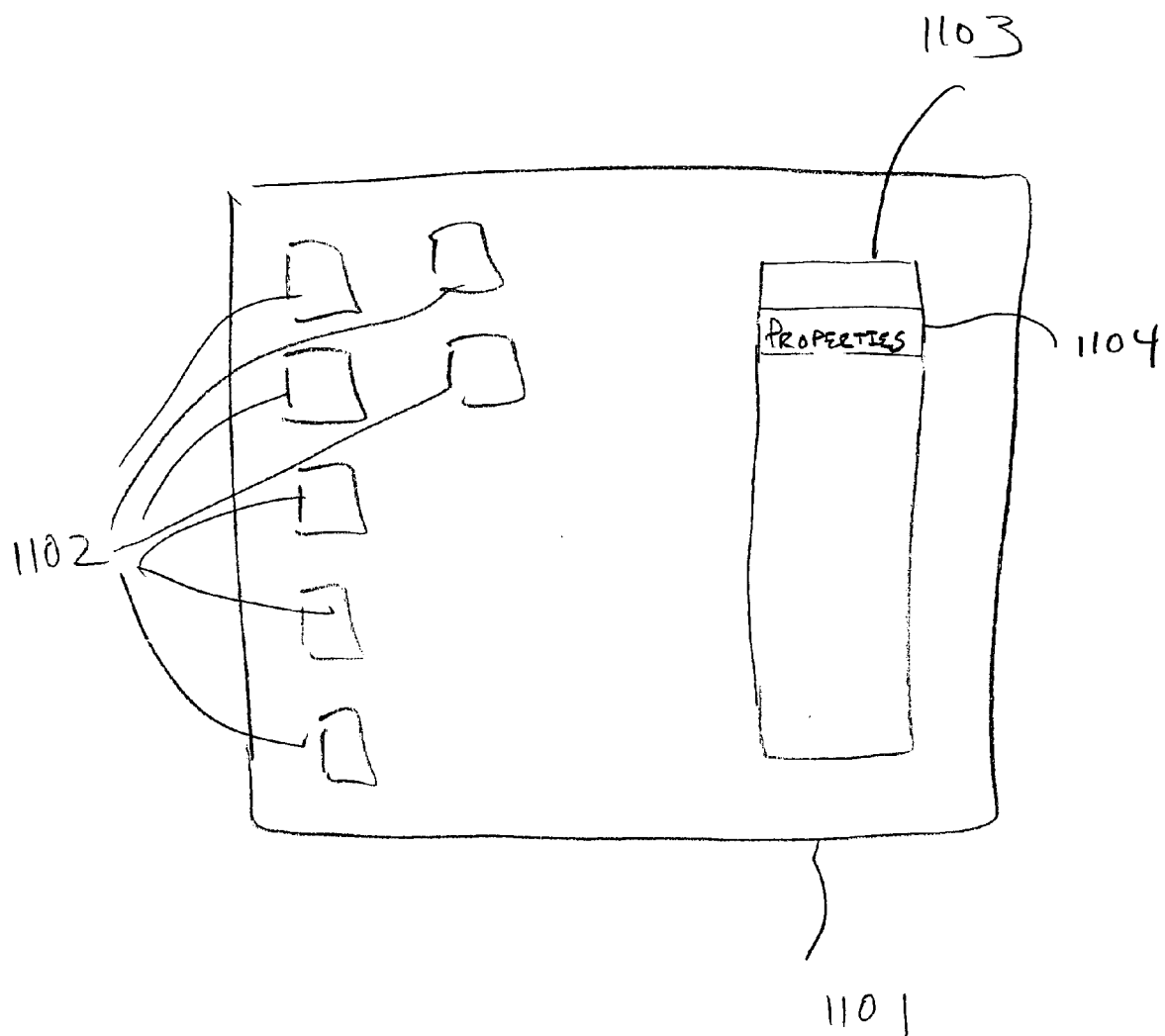


FIG. 11

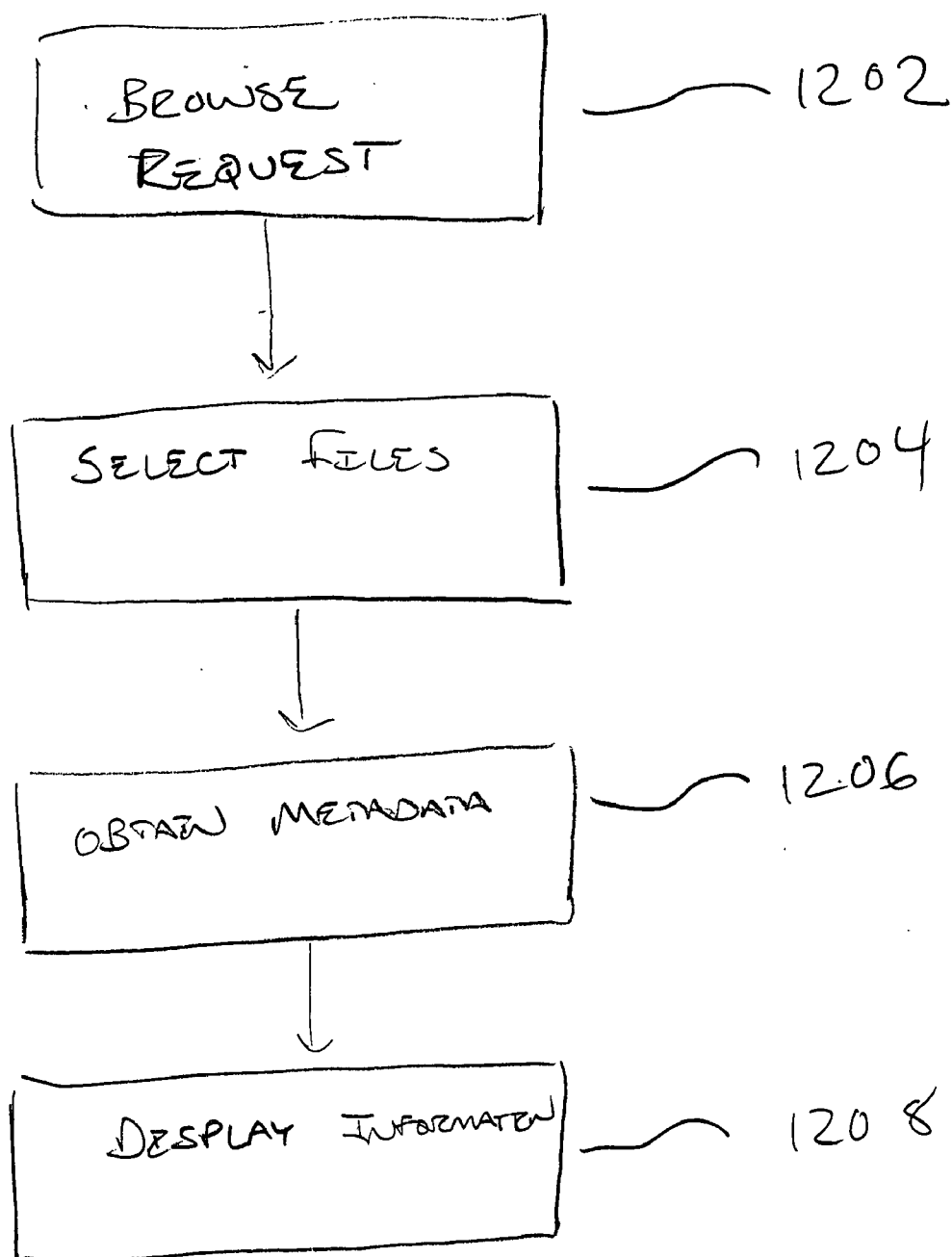


FIG. 12

SYSTEMS AND METHODS FOR FACILITATING STORAGE OPERATIONS USING NETWORK ATTACHED STORAGE DEVICES

RELATED APPLICATIONS

[0001] This application is a continuation of U.S. patent application Ser. No. 11/640,730, filed Dec. 18, 2006, and is related to the following patents and pending applications, each of which is hereby incorporated herein by reference in its entirety:

[0002] U.S. Pat. No. 6,418,478, titled PIPELINED HIGH SPEED DATA TRANSFER MECHANISM, issued Jul. 9, 2002, attorney docket number 4982/6;

[0003] Application Ser. No. 09/610,738, titled MODULAR BACKUP AND RETRIEVAL SYSTEM USED IN CONJUNCTION WITH A STORAGE AREA NETWORK, filed Jul. 6, 2000, attorney docket number 4982/8;

[0004] U.S. Pat. No. 6,542,972, titled Logical View and Access to Physical Storage in Modular Data and Storage Management System, issued Apr. 1, 2003, attorney docket number 4982/10;

[0005] Application Ser. No. 10/658,095, titled DYNAMIC STORAGE DEVICE POOLING IN A COMPUTER SYSTEM, filed Sep. 9, 2003, attorney docket number 4982/18;

[0006] Application Ser. No. 10/818,749, titled SYSTEM AND METHOD FOR PERFORMING STORAGE OPERATIONS IN A COMPUTER NETWORK, filed Apr. 3, 2004, attorney docket number 4982/35;

[0007] Application titled SYSTEMS AND METHODS FOR GRANULAR RESOURCE MANAGEMENT IN A STORAGE NETWORK, filed Dec. 15, 2006 attorney docket number 25963/008.

BACKGROUND OF THE INVENTION

[0008] The invention relates generally to performing storage operations on electronic data in a computer network, and more particularly, to facilitating storage operations including data stored on a network attached storage device.

[0009] The storage of electronic data has evolved over time. During the early development of the computer, storage of electronic data was limited to individual computers. Electronic data was stored in Random Access Memory (RAM) or some other storage medium such as a magnetic tape or a hard drive that was a part of the computer itself.

[0010] With the advent of network computing, the storage of electronic data gradually moved from the individual computer to dedicated storage devices accessible via a network. Some of these network storage devices evolved over time into networked tape drives, optical libraries, Redundant Arrays of Inexpensive Disks (RAID), CD-ROM jukeboxes, and other devices. Common architectures also include network attached storage devices (NAS devices) that are coupled to a particular network (or networks) and are used to provide storage capability for various storage operations that may be required by a particular network (e.g., backup operations, archiving, and other storage operations including the management and retrieval of such information).

[0011] A NAS device typically utilizes a specialized file server or network attached storage system that connects to the network. A NAS device often contains a reduced capacity or minimized operating and file management system (e.g., a microkernel) and normally processes input/output (I/O)

requests by supporting common file sharing protocols such as the Unix network file system (NFS), DOS/Windows, and server message block/common Internet file system (SMB/CIFS). Using traditional local area network protocols such as Ethernet and transmission control protocol/internet protocol (TCP/IP), a NAS device typically enables additional storage to be quickly added by connecting to a network hub or switch.

[0012] Certain storage management procedures, such as hierarchical storage management (HSM) procedures provides for movement of files from hard disk to slower, less-expensive storage media, or secondary storage over time. As shown in FIG. 1, one migration scheme may include data transfer from a magnetic disk **10** on a computing device to an optical disk **20** and later to a tape **30**. Conventional data management software usually monitors hard disk capacity and moves data from one storage level to the next (e.g., from production level to primary storage and/or from primary storage to secondary storage, etc.) based on storage criteria associated with that data such as a storage policy, age, category or other criteria as specified by the network or system administrator. For example, an email system such as MICROSOFT OUTLOOK™ may have attachments “aged off” (i.e., migrated when age requirement is met) from production level storage to a network attached storage device.

[0013] Referring to FIG. 2, there is shown a network architecture of a system **200** for performing storage operations on electronic data in a computer network in accordance with the prior art. As shown, system **200** includes a storage manager **201** and one or more of the following: a data store computer **285**, a data store **290**, a data agent **295**, a jobs agent **240**, a plurality of media management components **205**, which may be referred to as media agents, a plurality of storage devices **215**, a plurality of media management component index caches **210** and a storage manager index cache **230**.

[0014] Data agent **295** is generally a software module that may be responsible for archiving, migrating, and recovering data of data store computer **285** stored in a data store **290** or other memory location. Each data store computer **285** may have a data agent **295** and system **200** can support many data store computers **285**.

[0015] Each media management component **205** may maintain an index cache **210** which stores index data that system **200** generates during storage operations. The system may maintain two copies of the index data regarding particular stored data. A first copy may be stored with the data copied to a storage device **215**. Thus, a tape may contain the stored data as well as index information related to the stored data. In the event of a system restore, the index data stored with the stored data can be used to rebuild a media management component index **205** or other index useful in performing storage operations.

[0016] In addition, the media management component **205** that controls the storage operation also may write an additional copy of the index data to its index cache **210**. The data in the media management component index cache **210** may be stored on faster media, such as magnetic media, and is thus readily available to the system for use in connection with storage operations and other activities without having to be first retrieved from a slower storage device **215**.

[0017] Storage manager **201** may also maintain an index cache **230**. Storage manager index cache **230** may be used to indicate, track, and associate logical relationships and associations between components of system **200**, user preferences, management tasks, and other useful data. For example,

storage manager **201** may use its index cache **230** to track logical associations between media management components **205** and storage devices **215**. Index caches **230** and **210** may reside on their corresponding storage component's hard disk or other fixed storage device. For example, the media management component **205** may retrieve data from storage manager index cache **230** regarding a storage policy and storage operation to be performed or scheduled for a particular client **285**. The media management component **205**, either directly or via an interface module, may communicate with the data agent **295** at data store computer **285** regarding the details of an upcoming storage operation.

[0018] Jobs agent **240** may also retrieve from index cache **230** information relating to a storage policy **260** associated with data store computer **285**. This information may be used in coordinating or establishing actions performed by one or more data agents **295** and one or more media management components **205** associated with performing storage operations for that particular data store computer **285**. Such information may also include other information regarding the storage operation to be performed such as retention criteria, encryption criteria, streaming criteria, path information, etc.

[0019] Data agent **295** may package or otherwise manipulate client data stored in client data store **290** in accordance with storage policy **260** and/or according to a user preference, and communicate client data to the appropriate media management component(s) **205** for processing. The media management component(s) **205** may store the data according to storage preferences associated with storage policy **260** including storing the generated index data with the stored data, as well

[0020] as storing a copy of the generated index data in the media management component index cache **210**.

[0021] As shown in FIG. 2, a network attached storage device **250** and corresponding file server **254** are also connected to storage manager **201**. NAS **250** and file server **254** are dedicated applications without a general purpose operating system and generally do not by themselves support software applications, such as a back-up.

[0022] NAS devices typically interface with other components, such as those of storage management system **200**, or a relatively limited basis. One reason for this is because NAS devices tend to be proprietary. Accordingly, other storage system designers have a limited knowledge of implementation particulars needed to design fully compatible and integrated interfaces for their products.

[0023] Moreover, there are many different types of NAS devices, such as WAFL by NETWORK APPLIANCE of Sunnyvale, Calif., the EMC CELERA file system by the EMC Corporation of Hopkinton, Mass., the NETWARE file system by NOVELL of Provo, Utah, and other vendors. Most of these systems export their file systems to host computers such as the common Internet file system (CIFS) or the network file system (NFS), but provide no mechanism to run software on their operating systems or reside on the file system stack to intercept read/write or other data requests.

[0024] One solution to this problem is through the use of a proxy media management component **252** connected to file server **254**. Proxy media agent **252** runs the applicable software used to move data to NAS **250**. Proxy media management component **252** may, for example, issue commands using the Network Data Management Protocol ("NDMP").

[0025] Referring now to FIG. 3, a representation of a data structure **310** is shown that may be used by system **200** in

moving data to NAS **250**. As shown, data structure **310** includes the actual data being moved in a payload **314** as well as a NDMP header **312** preceding payload **314** and NDMP trailer **316** following the payload.

[0026] As discussed above, index cache **230** in storage manager **200** may keep track of certain information including the status of storage operations. If a storage operation copying data from data store **290** to NAS **250** is interrupted, for example, index cache **230** may be used to restart the operation and may keep track of the data path, data transferred, data remaining, etc. If data from NAS **250** needs to be restored, data in index cache **230** may also be used to facilitate such a restore operation.

[0027] One shortcoming of the NAS architecture described above is the vulnerability associated with the dedicated data transfer path which includes proxy **252**. For example, if proxy media management component **252** becomes inoperative or otherwise unavailable, there is generally no way to send data to NAS **250**. Similarly, if other media management components in the system are handling less of a load than proxy media management component **252**, they are unable to assist media management component **252** as it is the sole media management component designated for NAS **250**.

[0028] Moreover, should storage manager **201** become inoperative or otherwise unavailable, or its data or associated indexes be corrupted, incomplete, or otherwise unavailable, there is generally no way to rebuild index **230** to with data from NAS **250**.

[0029] Furthermore, with conventional systems, it is difficult to verify the contents of NAS **250** after data is stored thereon. As discussed above, in general, NAS systems are proprietary and a simple request to verify the data stored on a NAS cannot be performed nor can information regarding the data, such as helpful metadata, be made available.

[0030] Therefore, it would be desirable to provide a more robust storage operation system that can more effectively interoperate with NAS devices.

SUMMARY OF THE INVENTION

[0031] A system and method for communicating, browsing, verifying and routing data in storage operation systems using network attached storage devices is provided. In some embodiments, the system may include a management module and a media management component connected to the management server, which interoperate with network attached storage devices to provide the communicating, browsing, verifying and routing functions.

BRIEF DESCRIPTION OF THE DRAWINGS

[0032] The invention is illustrated in the figures of the accompanying drawings which are meant to be exemplary and not limiting, in which like references are intended to refer to like or corresponding parts throughout, and in which:

[0033] FIG. 1 illustrates a storage operation in accordance with the prior art;

[0034] FIG. 2 is a system diagram illustrating a prior art storage system;

[0035] FIG. 3 is a prior art data arrangement;

[0036] FIG. 4 is a system diagram illustrating a storage operation system constructed in accordance with an embodiment of the invention;

[0037] FIG. 5 is a flow chart illustrating some of the steps associated with a method in accordance with an embodiment of the present invention;

[0038] FIG. 6 is a representation of a data structure constructed in accordance with an embodiment of the present invention;

[0039] FIG. 7 is a flow chart illustrating some of the steps associated with a method in accordance with an embodiment of the present invention;

[0040] FIG. 8 is a flow chart illustrating some of the steps associated with a method in accordance with an embodiment of the present invention;

[0041] FIG. 9 is a representation of a data structure constructed in accordance with an embodiment of the present invention;

[0042] FIG. 10 is a flow chart illustrating some of the steps associated with a method in accordance with an embodiment of the present invention;

[0043] FIG. 11 is an illustration of a graphical user interface constructed in accordance with an embodiment of the present invention;

[0044] FIG. 12 is a flow chart illustrating some of the steps associated with a method in accordance with an embodiment of the present invention;

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0045] Referring to FIG. 4, a system 400 in accordance with one embodiment of the present invention is shown. System 400 is similar in certain respects to the system described in FIG. 2, and generally includes components and functional blocks which have been numbered similarly to denote some similar functionality and general correspondence. For example, system 400 includes a storage manager 401 (storage manager 201 in FIG. 2), one or more computing devices 485 (computer 285 in FIG. 2, which include clients), data store(s) 490 (data store(s) 290 in FIG. 2), data agent 495, (data agent(s) 295 in FIG. 2) jobs agent(s) 440 (jobs agent 240 in FIG. 2) media management components 405 (components 205 in FIG. 2), storage device 415 (storage devices 215 in FIG. 2), media management component index caches 410 and a storage manager index cache 430 (caches 210 and 230, respectively in FIG. 2) The system and elements thereof are further described in application Ser. No. 09/610,738, now U.S. Pat. No. 7,035,880 which is incorporated by reference in its entirety.

[0046] One way in which system 400 has been improved as compared to the systems shown in FIGS. 1-3, is by the addition of NAS module 406, which may include verification component 407, routing component 408, browsing component 409, and metadata component 411. In some embodiments, NAS module 406 may be installed or distributed across some or all components 405 to facilitate data routing, data verification and browsing, and communication with NAS device 450, among other things.

[0047] Data agent 495 is generally a software module that is responsible for archiving, migrating, and otherwise coordinating the transfer and recovery of data from computer 485 or data store 490 to another memory location such as storage device 415 or NAS device 450. Each computer 485 may have one or more associated data agent(s) 495 and system 400 can support many computers 485. System 400 may provide a plurality of data agents 495 each of which may backup, migrate, and recover data associated with a different applica-

tion. For example, different individual data agents 495 may be designed to handle MICROSOFT EXCHANGE data, LOTUS NOTES data, MICROSOFT WINDOWS file system data, MICROSOFT ACTIVE DIRECTORY OBJECTS data, and other types of data known in the art.

[0048] In the case where data store computer 485 has two or more types of data, one data agent 495 may be used for each data type to archive, migrate, and restore computer 485 data (however, in other embodiments, one data agent may handle multiple types of data). For example, to backup, migrate, and restore some or all of the data on a MICROSOFT EXCHANGE 2000 server, computer 485 may use a MICROSOFT EXCHANGE 2000 Mailbox data agent 495 to backup the EXCHANGE 2000 mailboxes, a MICROSOFT EXCHANGE 2000 Database data agent 495 to backup the EXCHANGE 2000 databases, a MICROSOFT EXCHANGE 2000 Public Folder data agent 495 to backup the Exchange 2000 Public Folders, and a MICROSOFT WINDOWS File System data agent 495 to backup the file system. In some embodiments, data agents 495 would may be treated as four separate data agents 495 by system 400 even though they may reside on the same data store computer 485.

[0049] Each media management component 405 may maintain an index cache 410 which may store index data system 400 generates during storage operations. This may include, for example, storage operations for MICROSOFT EXCHANGE that generate index data. Index data may include, for example, metadata or other information regarding the location of the stored data on a particular media, information regarding the content of the data stored such as file names, sizes, creation dates, formats, application types, and other file-related criteria, information regarding one or more clients associated with the data stored, information regarding one or more storage policies (discussed below), storage criteria, or storage preferences associated with the data stored, compression information, retention-related information, encryption-related information, stream-related information, and other types of information.

[0050] Data in index cache 410 thus provides the system with an efficient mechanism for performing storage operations including information useful in locating files for recovery operations and for managing and tracking stored data. The system generally maintains two or more copies of the index data regarding particular stored data. A first copy may be stored with the data copied to a storage device 415, the other in the index itself. Thus, a tape may contain the stored data as well as index information related to the stored data.

[0051] In the event of a system restore or other data restore or refresh operation, the index data included with the stored data may located on storage device 415 and be used to rebuild a media management component index 405 or other index useful in performing storage operations which may include repopulating its index cache 410.

[0052] Storage manager 401 may maintain an index cache 430. Storage manager index cache 430 may used to indicate, track, and associate logical relationships and associations between components of system 400, user preferences, management tasks, and other useful data. For example, storage manager 401 may use its index cache 430 to track logical associations between media management components 405 and storage devices 415. Storage manager 401 may also use its index cache 430 to track the status of storage operations to be performed, storage patterns associated with the system components such as media use, storage growth, network

bandwidth, service level agreement (“SLA”) compliance levels, data protection levels, storage policy information, storage criteria associated with user preferences, retention criteria, storage operation preferences, and other storage-related information.

[0053] Index caches may **430** and **410** may reside on their corresponding storage component’s hard disk or other fixed or dynamic storage device or on other associated memory. For example, media management component **405** may retrieve information from storage manager index cache **410** regarding a storage policy and storage operation to be performed or scheduled for a particular computer **485**. Media management component **405**, either directly or via an interface module, may communicate with the data agent **295** at the data store computer **485** regarding the storage operation.

[0054] Jobs agent **440** may also retrieve from index cache **430** a storage policy **460** associated with the data store computer **485** and use information from one or more storage policies **460** to communicate to data agent **495** one or more media management components **405** associated with performing storage operations for that particular data store computer **485** as well as other information regarding the storage operation to be performed such as retention criteria, encryption criteria, streaming criteria, etc.

[0055] A storage policy is generally a data structure or other information, which may include a set of preferences and other storage criteria for performing storage operations. The preferences and storage criteria may include, but are not limited to: a storage location, relationships between system components, network pathway to utilize, retention policies, data characteristics, compression or encryption requirements, preferred system components to utilize in a storage operation, and other criteria relating to a storage operation. A storage policy may be stored to a storage manager index, to archive media as metadata for use in restore operations or other storage operations, or to other locations or components of the system.

[0056] Data agent **495** may package or otherwise manipulate client data stored in data store **490** in accordance with storage policy **460** and/or according to a user preference, and may communicate this data to the appropriate media management component(s) **405** for processing. Media management component(s) **405** may store the data according to storage preferences associated with the storage policy including storing the generated index data with the stored data, as well as storing a copy of the generated index data in the media management component index cache **410**.

[0057] Media management component **405** may further include a NAS module **406** including a metadata component **411**. NAS module **406** may be implemented as a software module that may be installed on one or more media management components **405**. NAS module **406** may interoperate with components **405**, data agents **495**, and/or storage manager(s) **401** to coordinate and verify the transfer of data from computer(s) **485** to NAS device **450**.

[0058] As shown in FIG. 4, media management component **405** may further include a NAS component **235** associated with NAS module **406**. NAS modules **406** may be associated or resident with each component **405** or may be distributed across various component **405** in system **400** (e.g., every other component **405** may have a NAS module **406**, one in three component **405** may have a NAS module **406**, or any other desired distribution). Moreover, in some embodiments, one or more components **405** may share a NAS module **406**

which may be moved as necessary (e.g., float) to accommodate system storage operations. This provides system **400** with a robust set of routing options to reach NAS devices **450** through multiple components **405**.

[0059] In some embodiments, each media management component **405** is may be capable of handling storage operations to either a standard storage device **415**, or to work with a file server **454** to handle copy operations with a NAS storage device **450**. NAS components **435** may, for example, operate using the NDMP protocol. Resource allocation for moving data from data store **490** to NAS **450** may now be based on other relevant characteristics instead of being limited to a particular defined proxy media agent. For example, a data paths **402** and **404** from data from data store **490** to NAS **450** may be defined by properties of NAS **450**—for example if NAS **450** includes only particular types of data such as EXCHANGE or OUTLOOK or if NAS **450** is further defined as being the OUTLOOK data for a particular individual—such characteristics may be taken into account when defining which media management component **405** to use for NAS device **450**.

[0060] Data path **402** and **404** may also be based on storage policy **460** defined for data store **490**. For example, a storage policy **460** may be defined for each particular type of data or application—such as EXCHANGE or OUTLOOK. The storage policy could define, for example, where data is to be stored, the duration of the storage, and how many copies should be made. Each storage policy may define one or multiple data paths for moving data from data store **490** to NAS **450**. Each data path may include a single media management component or multiple management components and a set of storage devices **415** and/or NAS devices **450**.

[0061] Storage policies **460** may define a preferred data path **402** and **404** for handling data moving from data store **490** to NAS **450**, or define a load balancing algorithm so that data paths may be utilized that have more availability than other data paths. The actual data path selected may, for example, be first based on the properties of NAS device **450**, such as data types or application, and then chosen based on storage policy **460** for a preferred data path and/or load balancing. For example, additional NAS devices **450** (not shown) may be operating a minimum capacity with others operating at a higher or near maximum capacity (also not shown). In this case, it may be desirable to distribute data load to underutilized resources. This may be done based on preferences in storage policy or certain load distribution algorithms the govern system operation. For example, in one embodiment, it may be desirable to distribute load across multiple NAS modules **406**, NAS devices **450** and associated transmission links such that a substantially even distribution is obtained. In other circumstances, for example, when a certain NAS module **406** is unavailable, it may be desired to continue to send data to a certain NAS device **450**, using, however, a different NAS module **406**. This may be accomplished with embodiments of the present invention by routing data storage operations to a different NAS module **406** which may communicate and supervise storage operations to the original NAS device **450**, even though the now unavailable module **406** was previously handling such operations.

[0062] In this way, if a certain media management component **405** becomes inoperative, storage manager **400** may select another media management component **405** in defining a data path to NAS **450**. Moreover, storage policy **460** may be defined to perform load balancing. When load balancing is

chosen, one option is for storage manager **400** to assign the least loaded media management component to handle a particular storage operation. For example, if a storage policy has three (3) data paths and ten (10) storage operations to perform, the ten jobs may be spread across available media management components. This may mean that multiple operations for a single NAS **450** may be balanced across multiple media management components **405**. File server **454** may be used to keep track of applicable copy information so as to improve routing inter-connectivity.

[0063] Storage manager **401** may be also used to monitor the capacities of respective media management components **405** and dynamically alter data path **402** and **404**. In this way, a copy operation may begin using a first media management component and a first data path and then, because of, for example, a change in the load in system **400**, a second data path including a different media management component may be selected to complete the copy operation. For example, if the first media management component that started the copy operation becomes overburdened or inoperative, storage manager **400** may decide to move the copy operation to a different media management component. Moreover, storage manager **401** may choose a data path so as to most efficiently utilize storage media. For example, if using a first data path may result in using a last portion of a first storage medium and a first portion of a second storage medium, storage manager **401** may choose a different data path which results in only a single storage medium being used.

[0064] Referring now to FIG. 5, some steps associated with a method for facilitating storage operations in accordance with an embodiment of the invention is shown. The method shown in the figure could operate using, for example, system **400** shown in FIG. 4. As shown, at step **502**, a request may be received to perform a storage operation including sending data to a network attached storage device. At step **504**, a data path may be determined for the source data to the network attached storage device. This may include a determination based on a storage policy or other routing preference. It may also further involve an analysis of the network operating conditions including network congestion or utilization rate of the resources defined in the transmission path, and certain specified load balancing requirements or preferences.

[0065] If one or more resources as initially specified are beyond a threshold, the path and/or the destination NAS device may be altered to ensure certain system management operating conditions are achieved/maintained. For example, if a certain media management component, transmission path or destination device is operating beyond a specified level, some or all of those resources may be altered, such that other resources are used instead to meet or maintain system operating specifications. This may include specifying another media management component to handle storage operations to a NAS device that was previously associated with another different media management component. Moreover, it will be understood that such resource reallocation may occur dynamically (e.g., upon consideration of network operating conditions), or as specified by a user, or in recognition of a system change, such as removal of certain hardware such as a NAS device, or computing devices, etc. This may be accomplished by a storage manager or other system management module.

[0066] At step **506**, once the route and destination resource has been established, the storage operation is performed. This may involve moving one or more chunks of data associated

with an archive file, etc. This may also involve confirming that the data has successfully been stored, and writing file location, size, and other index information to indexes in media management components or storage managers.

[0067] Moreover, in some embodiments, the system may perform additional monitoring of the resources operation while performing the storage operation to determine the effect of the operation on system resources (step **508**). Such load measurement may determine, for example, whether particular devices in the system are more loaded with data than other devices and also determines whether particular devices have no load—such as, for example, if such devices have become inoperative. This additional information may be used at step **510**, where the system may determine if there is a load imbalance or whether that a device on the chosen data path is no longer operative. This may be confirmed by performing certain tests on the data path to determine, if, it is in fact inoperative. In some embodiments, such further monitoring may be performed at step **504**. Moreover, based on this information, the system may further redefine or reallocate system resources to distribute data load substantially evenly across network resources, or conserve media usage, or promote other load balancing goals (e.g., distribute data operations across network resources in view of a the need to remain within a copy or backup window, or other operating condition such as a data path preference etc.). Moreover, it will be understood that system monitoring and analysis with respect to load balancing may be performed iteratively at steps **504** and **510** with resource reallocation occurring on a continuous basis based on the results of the monitoring and analysis.

[0068] Referring back now to FIG. 4, media management component **405** may further include a metadata component **411**. As data is copied from data store **490** is to NAS **450**, the path may include media management component **405** and metadata component **411**. Metadata component **411** may generate certain storage metadata, which may include, for example a storage header and trailer, including management information associated with system **400**, which may be appended to any data copied (such as a data “chunk”). This metadata may include the size of the data, path information, offsets, client ID information, information relating to the source data such as, which archive file a chunk may be associated with, what files are in the chunk, chunk number, and any other data useful for data management, etc. This metadata may be separated into a header and trailer and appended to the data as described below.

[0069] For example, in FIG. 6, data chunk **690** and metadata header **678** and metadata trailer **682** may be combined to produce the data structure shown. Metadata may be separated between header **678** and trailer **682** as desired for system management purposes. For example, header **678** may include the size of the data, and offsets, while trailer **682** may include path information, client ID information, job ID, information relating to the source data such as, which archive file a chunk may be associated with, what files are in the chunk, chunk number, etc. However, any useful arrangement may be used, if desired.

[0070] In operation, media management component **405** and metadata component **411** may generate the data arrangement shown in FIG. 6 and using write commands, for example, the NDMP protocol, may send data along with metadata as shown in FIG. 6 to file server **454** for subsequent storage in NAS **450**. In some embodiments, file server **454** may remove any NDMP header or trailer, and add metadata

created by component 411 and store the combination on NAS 450. Data may be retrieved by a media management component when a computing device 485 or storage manager 401 issues a data or system restore request. In this case, the media management component may query file server 454 on behalf of the requesting device for the requested information (which may be retrieved via paths 413 and 414 and subsequently routed to the appropriate destination).

[0071] Referring to FIG. 7, there is shown a method for facilitating storage operations including NAS in accordance with an embodiment of the invention. The process could be used with, for example, system 400 shown in FIG. 4. As step 702, a request is received by a storage manager or media management component to move data to a network attached storage device. At step 704, a routine, which may be recursive, is initiated for each group of data, such as a chunk, to be moved. This may involve evaluating path and destination information or preferences, and obtaining metadata relating to the information as further described herein.

[0072] At step 706, a chunk of data from the data source is moved or copied to or processed by an applicable media management component. This may involve processing on a source computing device prior to movement a NAS device. At step 708, the media management component may issue an applicable write command and generates metadata for the received data (in some embodiments, a data structure including data and metadata, is created). At step 710, a file server receives the write command, the data and the metadata. At step 712, the file server combines the data and metadata, writes the resulting data structure to the NAS.

[0073] Referring to FIG. 8 there is shown a process for retrieving data from a NAS in accordance with an embodiment of the invention. The process may be used with, for example, system 400 shown in FIG. 4. At step 820, a request is received to restore data stored on a NAS. At step 822, a routine, which may be recursive, is initiated for each set or chunk of data which may request a specific chunk, monitor data received and response to the request, and terminate the retrieval operation once the desired data is retrieved.

[0074] At step 824, a file server retrieves data and associated metadata from a NAS in response to the request. In some embodiments, the file server may separate, chunk data and metadata and provide each to the media management device. At step 826, the data and metadata may be forwarded to a media management component or a computing device. At step 828, the media management component may use the data and metadata to identify and route the requested information to the source requesting such information and “unpack” or otherwise obtain the data (e.g., a computing device requesting a data restore operation). In certain circumstances, the metadata associated with the stored data may be desired, for example, in the event of a system restore, to reconstruct one or more media management or storage manager indexes.

[0075] Referring again back to FIG. 4, NAS module 406 may further include a NAS verification component 407 for verifying that the information requested information is properly stored (or retrieved) from NAS device 450. For example, in operation, NAS verification module 411 may issue a request to file server 454 to restore certain data stored in NAS device 450 to a data destination 412. The request may include, for example, the path stored on NAS device 450, whether backup information stored on NAS 450 was a full or incremental backup, a list of the paths to restore, a destination path, an option not to write the data stored in NAS 450, etc. This

request may be issued in order to obtain metadata associated with stored data which may be compared with certain index information to confirm the requested data is available, was properly stored and may be retrieved.

[0076] File server 454 may send a retrieve request to NAS 450, that reads data stored on NAS 450, and forwards the responsive data to media management component 405. NAS verification module 407 may read the metadata associated with the retrieved data and forward the data to destination 412. In some embodiments, data destination 412 may be a null or other empty port such as air and does not include a storage medium. This may be performed in cases where only metadata is desired for verification purposes.

[0077] Referring to FIG. 9, a data structure 906 constructed in accordance with the principles of the present invention is shown. As shown, data structure 906 may include a dump header 920, a data payload 922 and a trailer 922. Dump header 920 may include, for example, a description and size of data payload 922, and a list of the files in payload 922. File server 454 may further modify data structure 906 and add a label field 926, a file marker field 928, a chunk header 930, a file marker 932, a file marker 936, a chunk trailer 938, a file marker 940, and a trailer 942 to produce data structure 908. Label 926 may label data structure 908. File markers 928, 932, 936 and 940 may separate different portions of data structure 908. Chunk header 930 may include header information for each chunk of data 922. Similarly, chunk trailer includes trailer information for each chunk of data 922.

[0078] NAS verification component 407 receives data from NAS 450 in response to a request and may compare certain information in data structure 908 with information from an index in media management component 405 or storage manager 401. For example, verification component 407 may compare information in dump header 920 to a corresponding entry stored in index cache 430 or in index cache 410. Such information was initially generated when data 906 was first stored on NAS 450. Further, when NAS verification module 407 restores data 908, the restoration process itself generates metadata which may also be compared with meta data stored in index cache 430 or index cache 410 (e.g., unique or somewhat unique rebuild information, etc.).

[0079] The results of such comparisons may be used to verify the contents of payload 922 and may be performed using techniques known in the art—such as checksums, hashing, etc. if the comparisons are favorable, media management component 405 then forwards data 908 to data destination 412. In some embodiments, data destination 412 may be a “dummy” device, air or other null port and excludes a storage medium, if the metadata is desired. This increases the speed of the process because less time is used as compared to media management component 405 actually writing data structure 408 to a storage medium. Further, no extra storage space needs to be utilized.

[0080] Referring to FIG. 10, a flow chart illustrating some of the steps involved in a method for validating data stored on a network attached storage device in accordance with an embodiment of the invention is shown. The method in FIG. 10 may be implemented using, for example, system 400 discussed above. At step 1002, a request is made to restore data stored on a NAS. This may involve a computing device or storage manager contacting a media management component with such as request. Next, at step 1004, a media management component may communicate with a NAS device through a file server and locate the data requested. After the requested

data has been located, data from the NAS may be retrieved and sent from the file server and restored. At step **1006**, certain metadata such as a dump header or other portions from the data stored in the NAS may be extracted and compared with metadata stored in an index cache. If the comparison is favorable, the metadata or other data may be used for system restore or management purposes. For example, this information may be used to verify that the requested information is stored on the NAS device and may be retrieved, if necessary (e.g., for repopulating a computing device with application data or repopulating a management database with system management information, etc.).

[0081] If the comparison is unfavorable, the query may be processed several times, until a favorable result is obtained, or until a certain number of unfavorable results are obtained, in which the query may terminate as a time out, and system indexes or processes updated to reflect the requested information could not be found, and therefore not retrieved. Assuming that a favorable result is obtained, the metadata may be retrieved and the data from the NAS is sent to a null port which does not include a storage medium (step **1008**). Moreover, data that has been obtained may be merged into any existing media management or storage manager database to repopulate an index or other databases in the event a restore is desired.

[0082] Thus, by requesting to restore data from a NAS and reading header and/or other metadata from the NAS data, a system and method for verifying data stored on a NAS is realized.

[0083] As shown in FIG. 4, NAS module **406** may further include a NAS browsing component **409** for allowing a user to browse information or data stored on NAS device **450**. For example, in operation, NAS browsing component **409** may be invoked, for example, on a computing device **485** and issue a request to media management component **405** for information regarding certain data stored in NAS device **450**. The request may include, for example, a request for a file system overview of the files stored on NAS device **450**, including certain information or properties about those files.

[0084] Once the information is obtained, a graphical user interface on a computing device may display the information in a familiar graphical file format with an icon representing each file. This is generally shown in FIG. 11. As shown, screen **1101** may be similar to a WINDOWS file system display screen, which may include file icons **1102** representing NAS files under management. In operation, a user may select one of icons **1102** and invoke a pulldown or other menu **1103** which lists a set of options or operations that may be performed on the various files shown. For example, this may include a properties selection which may display various properties of the files under management including the size of the data, path information, offsets, client ID information, application information, date copied, storage policies associated with the data, information relating to the source data such as, which archive file a chunk may be associated with, what files are in the chunk, chunk number, and any other data useful for data management, etc.

[0085] The user may obtain this information by choosing a particular file **1102** and selecting the properties option **1104** from pull down menu **1103**. It will be understood each property may be presented in a layered or tiered format that additional details may be obtained by clicking or selecting a particular property and that other options may also be available(not shown).

[0086] Referring to FIG. 12, a flow chart illustrating some of the steps involved in a method for browsing data stored on a network attached storage device in accordance with an embodiment of the invention is shown. The method in FIG. 12 may be implemented using, for example, system **400** discussed above. At step **1202**, a request may be made to view files stored on one or more NAS devices. This may involve a user invoking a browser on a computing device. At this point the user may be prompted to select a certain network of interest. Once selected, the system may scan the selected network(s) for associated NAS devices. A list of available devices may be displayed, from which the user may choose one or more devices of interest. In some embodiments, a user may request a list of files or application programs, and the system may generate a list of available NAS devices that include the specified information.

[0087] Once a user selects one or more NAS devices from the list, a query may sent to the system of those devices for a list of files under management. The system may obtain these lists, for example, from a system index or through direct queries to the NAS devices. Next, at step **1204**, the user may select certain files to receive more information about them. This may involve the restoration process described above where files are restored to obtain metadata for browsing purposes (step **1206**).

[0088] Next, at step **1208**, the information may be displayed to the user for further inspection. At this point, the user may further direct operations with respect to the displayed files. For example, the user may direct a full or partial data restore from the NAS device and may direct certain clients be repopulated with the restored data. Similarly, the user may update or refresh certain system management components such as a storage manager database or index or media management component index to be updated. Other operations are also contemplated, such as directing the copying of data or management information to a disk, other portable media or to another network location.

[0089] Systems and modules described herein may comprise software, firmware, hardware, or any combination(s) of software, firmware, or hardware suitable for the purposes described herein. Software and other modules may reside on servers, workstations, personal computers, computerized tablets, PDAs, and other devices suitable for the purposes described herein. Software and other modules may be accessible via local memory, via a network, via a browser or other application in an ASP context, or via other means suitable for the purposes described herein. Data structures described herein may comprise computer files, variables, programming arrays, programming structures, or any electronic information storage schemes or methods, or any combinations thereof, suitable for the purposes described herein. User interface elements described herein may comprise elements from graphical user interfaces, command line interfaces, and other interfaces suitable for the purposes described herein. Screenshots presented and described herein can be displayed differently as known in the art to input, access, change, manipulate, modify, alter, and work with information.

[0090] Moreover, it will be appreciated that the systems and methods provided herein are intended to exemplary and not limiting and that additional elements or steps may be added or performed in different order, if desired.

[0091] While the invention has been described and illustrated in connection with preferred embodiments, many variations and modifications as will be evident to those skilled

in this art may be made without departing from the spirit and scope of the invention, and the invention is thus not to be limited to the precise details of methodology or construction set forth above as such variations and modification are intended to be included within the scope of the invention.

What is claimed is:

1. A method for verifying data stored in a storage system, the storage system comprising a management server, a media management component connected to the management server, at least one network attached storage device connected to the media management component, and a data source connected to the management server, the method comprising:

sending a request to restore stored data, the stored data being stored on a network attached storage device and including first metadata;
receiving the stored data;
comparing the first metadata with second metadata stored in the storage system;
verifying the stored data based on the comparing; and
writing the stored data to a data destination, the data destination excluding a first storage medium.

2. The method of claim 1, wherein the first metadata comprises a dump header.

3. The method of claim 2, wherein the dump header includes a list of files in the stored data.

4. The method of claim 1, wherein the second metadata is stored in an index cache.

5. The method of claim 1, wherein the second metadata is stored in the management server.

6. The method of claim 1, wherein the second metadata is stored in the media management component.

7. The method of claim 1, wherein the data destination is an empty port.

8. The method of claim 1, further comprising:

restoring the stored data; and
producing the second metadata based on the restoring.

9. The method of claim 1, wherein the second metadata is generated when the stored data is stored on the network attached storage device.

10. A system for verifying stored data, the system comprising:

a management server;
a media management component connected to the management server;
at least one network attached storage device connected to the media management component, the network attached storage device including stored data, the stored data including first metadata; and

a data source connected to the management server; wherein:

the media management component sends a request to restore the stored data, receive the stored data, and compare the first metadata with second metadata stored in the storage system, the media management component further being effective to verify the stored data based on the comparing and write the stored data to a data destination, the data destination excluding a first storage medium.

11. The system of claim 10, wherein the first metadata comprises a dump header.

12. The system of claim 12, wherein the dump header comprises a list of files in the stored data.

13. The system of claim 10, wherein the second metadata is stored in an index cache.

14. The system of claim 10, wherein the second metadata is stored in the management server.

15. The system of claim 10, wherein the second metadata is stored in the media management component.

16. The system of claim 10, wherein the data destination is an empty port.

17. The system of claim 10, wherein the media management component is configured to restore the stored data and produce the second metadata based on the restoration.

18. The system of claim 10, wherein the media management component is configured to generate the second metadata when the stored data is stored on the network attached storage device.

19. A computer readable storage medium including computer executable code effective to verify data stored in a storage system, the storage system comprising a management server, a media management component connected to the management server, at least one network attached storage device connected to the media management component, and a data source connected to the management server, the method comprising:

sending a request to restore stored data, the stored data being stored on a network attached storage device and including first meta data;
receiving the stored data;
comparing the first meta data with second meta data stored in the storage system;
verifying the stored data based on the comparing; and
writing the stored data to a data destination, the data destination excluding a first storage medium.

20. The computer-readable medium of claim 19 having further instructions stored thereon including sequences of instructions which, when executed by one or more processors cause an electronic device to restore the stored data; and
produce the second metadata based on the restoring.

* * * * *