

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
18 May 2006 (18.05.2006)

PCT

(10) International Publication Number
WO 2006/052872 A2

(51) International Patent Classification:
G06F 12/00 (2006.01)

(21) International Application Number:
PCT/US2005/040258

(22) International Filing Date:
7 November 2005 (07.11.2005)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/625,746 5 November 2004 (05.11.2004) US
60/626,076 8 November 2004 (08.11.2004) US

(71) Applicant (for all designated States except US): **COM-MVAULT SYSTEMS, INC.** [US/US]; 2 Crescent Place, Oceanport, NJ 07757-0900 (US).

(72) Inventors; and

(75) Inventors/Applicants (for US only): **AMARENDRAN, Arun, Prasad** [IN/IN]; 2406 E Block, Jothy Illam, Sahakara, Nagar, Bellary Road, Bangalore 560092 (IN). **VOKAUAM RETNAMMA, Manoj, Kumar** [IN/US]; 2140 Aldrin Road, Ocean, NJ 07712 (US).

(74) Agent: **SHANAHAN, Michael, E.**; Brown Raysman Millstein Felder & Steiner LLP, 900 Third Avenue, New York, NY 10022 (US).

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

(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— without international search report and to be republished upon receipt of that report

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: SYSTEM AND METHOD TO SUPPORT SINGLE INSTANCE STORAGE OPERATIONS

(57) Abstract: Systems and methods for single instance storage operations are provided. Systems constructed in accordance with the principals of the present invention may process data containing a payload and associated metadata. Often, chunks of data are copied to traditional archive storage wherein some or all of the chunk, including the payload and associated metadata are copied to the physical archive storage medium. In some embodiments, chunks of data are designated for storage in single instance storage devices. The system may remove the encapsulation from the chunk and may copy the chunk payload to a single instance storage device. The single instance storage device may return a signature or other identifier for items copied from the chunk payload. The metadata associated with the chunk may be maintained in separate storage and may track the association between the logical identifiers and the signatures for the individual items of the chunk payload which may be generated by the single instance storage device.



WO 2006/052872 A2

**SYSTEM AND METHOD TO SUPPORT
SINGLE INSTANCE STORAGE OPERATIONS**

PRIORITY CLAIM

[0001] This application claims the benefit of U.S. provisional application no. 60/626,076 titled SYSTEM AND METHOD FOR PERFORMING STORAGE OPERATIONS IN A COMPUTER NETWORK, filed November 8, 2004, and U.S. provisional application no. 60/625,746 titled STORAGE MANAGEMENT SYSTEM filed November 5, 2004, each of which is incorporated herein by reference in its entirety.

COPYRIGHT NOTICE

[0002] A portion of the disclosure of this patent document contains material which is subject to copyright protection. The copyright owner has no objection to the facsimile reproduction by anyone of the patent document or the patent disclosure, as it appears in the Patent and Trademark Office patent files or records, but otherwise reserves all copyright rights whatsoever.

RELATED APPLICATIONS

[0003] This application is related to the following patents and pending applications, each of which is hereby incorporated herein by reference in its entirety:

[0004] Application Serial No. 09/354,058, titled HIERARCHICAL BACKUP AND RETRIEVAL SYSTEM, filed July 15, 1999, attorney docket number 4982/5;

[0005] U.S. Patent No. 6,418,478, titled PIPELINED HIGH SPEED DATA TRANSFER MECHANISM, issued July 9, 2002, attorney docket number 4982/6;

[0006] Application Serial No. 60/460,234, SYSTEM AND METHOD FOR PERFORMING STORAGE OPERATIONS IN A COMPUTER NETWORK, filed April 3, 2003, attorney docket number 4982/35;

[0007] Application Serial No. 60/482,305, HIERARCHICAL SYSTEM AND METHOD FOR PERFORMING STORAGE OPERATIONS IN A COMPUTER NETWORK, filed June

[0008] 25, 2003, attorney docket number 4982/39; Application Serial No. 60/519,526, SYSTEM AND METHOD FOR PERFORMING PIPELINED STORAGE OPERATIONS IN A COMPUTER NETWORK, filed November 13, 2003, attorney docket number 4982/46P;

[0009] Application Serial No. 10/803,542, METHOD AND SYSTEM FOR TRANSFERRING DATA IN A STORAGE OPERATION, filed March 18, 2004, attorney docket number 4982/49;

[0010] Application Serial Number to be assigned, titled SYSTEM AND METHOD FOR PERFORMING MULTISTREAM STORAGE OPERATIONS, filed November 7, 2005, attorney docket number 4982-59;

[0011] Application Serial Number to be assigned, titled METHOD AND SYSTEM OF POOLING STORAGE DEVICES, filed November 7, 2005, attorney docket number 4982-61;

[0012] Application Serial Number to be assigned, titled METHOD AND SYSTEM FOR SELECTIVELY DELETING STORED DATA, filed November 7, 2005, attorney docket number 4982-67;

[0013] Application Serial Number to be assigned, titled METHOD AND SYSTEM FOR GROUPING STORAGE SYSTEM COMPONENTS, filed November 7, 2005, attorney docket number 4982-69;

[0014] Application Serial Number to be assigned, titled SYSTEMS AND METHODS FOR RECOVERING ELECTRONIC INFORMATION FROM A STORAGE MEDIUM, filed November 7, 2005, attorney docket number 4982-68; and

[0015] Application Serial Number to be assigned, titled METHOD AND SYSTEM FOR MONITORING A STORAGE NETWORK, filed November 7, 2005, attorney docket number 4982-66.

BACKGROUND OF THE INVENTION

[0016] The invention disclosed herein relates generally to performing storage operations in a computer network. More particularly, the present invention relates to systems and methods for supporting single instance storage devices in a computer network.

[0017] Storage of electronic data has evolved through many forms. During the early development of the computer, storage of data was limited to individual computers. Electronic data was stored in the Random Access Memory (RAM) or some other storage medium such as a hard drive or tape drive that was an actual part of the individual computer.

[0018] Later, with the advent of networked computing, storage of electronic data gradually migrated from the individual computer to stand-alone storage devices and other storage devices accessible via a network, for example a tape library accessible via a network server or other computing device. These network storage devices soon evolved in the form of networked tape drives, libraries, optical libraries, Redundant Arrays of Inexpensive Disks (RAID), CD-ROM jukeboxes, and other devices. System administrators often use network storage devices to perform storage operations and make backup copies and other copies of data stored on individual client computers in order to preserve data against accidental loss, corruption, physical damage, and other risks.

[0019] Storage systems evolved to handle increasingly complex storage operations and increasingly large volumes of data. For example, some storage management systems began organizing system components and system resources into logical groupings and hierarchies such as storage operation cells of the CommVault QiNetix storage management system, available from CommVault Systems, Inc. of Oceanport, NJ, and as further described as further described in Application Serial No. 60/482,305 and Application Serial No. 09/354,058 which are hereby incorporated by reference in their entirety.

[0020] Another factor contributing to increasingly large volumes of data is storage of multiple copies of the same file or data item. For example, a large enterprise might have several hundred users each keeping a copy of the same e-mail attachment. Alternatively, individual users may also lose track of or otherwise retain several copies of a file on their own personal hard drive or network share. Thus, storage space on systems is being wasted by multiple instances of the same data.

[0021] To address this problem, companies have developed storage devices that support single instance storage. Data items copied to a single instance storage device are processed to determine a unique signature for each file. Thus, copies or instances of the same file will generate the same unique signature. One well known technique for generating such a signature is generating a cryptographic hash of the file or other similar checksum based on the file contents. Storage devices can then compare the signature for a file to be stored with a list of previously stored signatures to determine whether a copy of the file already exists in storage and thus the file need not be copied again. Some storage systems also use content addressable storage ("CAS") in single instance storage devices in which the signature or hash of the file is also used as the address of the file in the storage device.

[0022] One problem associated with single instance storage solutions is that they are not designed to process backup data stored as chunks. When a copy of a production data store or other large volume of data is made, the data is often divided into a number of smaller parts for easier transmission to archive media via the network. These smaller parts typically become encapsulated as the payload for chunks of data which include metadata, such as tag headers and footers as previously described in U.S. Application No. 10/803,542 and U.S. Patent No. 6,418,478 each of which is hereby incorporated by reference in its entirety, and the chunks of data are sent over the network to the archive storage. For example, each chunk may contain a payload of several thousand files from a larger data store containing hundreds of thousands of

files with each file or item having a logical identifier such as a filename. Metadata for each chunk describing the contents of the payload (logical identifiers, etc.) and other information may be stored along with the payload data as further described herein. In addition, the metadata from each chunk may be used by system components to track the content of the payload of each chunk and also contains storage preferences and other information useful for performing storage operations.

[0023] Each chunk of data, however, usually contains different metadata. Thus, two instances of the same file will likely be encapsulated by different metadata in two different chunks even if the payload of the two chunks is the same. Similarly, current single instance storage systems would generate different signatures for each chunk of data and therefore store a different copy of each chunk of data even though the payload of each chunk is the same.

BRIEF SUMMARY OF THE INVENTION

[0024] Systems and methods for single instance storage operations are provided. Systems constructed in accordance with the principals of the present invention may process data containing a payload and associated metadata. Often, chunks of data are copied to traditional archive storage wherein some or all of the chunk, including the payload and associated metadata are copied to the physical archive storage medium. In some embodiments, chunks of data are designated for storage in single instance storage devices. The system may remove the encapsulation from the chunk and may copy the chunk payload to a single instance storage device. The single instance storage device may return a signature or other identifier for items copied from the chunk payload. The metadata associated with the chunk may be maintained in separate storage and may track the association between the logical identifiers and the signatures for the individual items of the chunk payload which may be generated by the single instance storage device.

BRIEF DESCRIPTION OF THE DRAWINGS

[0025] 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, and in which:

[0026] Fig. 1 is a block diagram of a storage operation cell in a system to perform storage operations on electronic data in a computer network according to an embodiment of the invention;

[0027] Fig. 2 is a block diagram of a hierarchically organized group of storage operation cells in a system to perform storage operations on electronic data in a computer network according to an embodiment of the invention;

[0028] Fig. 3 is a block diagram of a hierarchically organized group of storage operation cells in a system to perform storage operations on electronic data in a computer network according to an embodiment of the invention;

[0029] Fig. 4 is a flow diagram of a method to store chunks of data in a single instance storage device.

[0030] Fig. 5 is a flow diagram of a method for retrieving chunk payload data from a single instance storage device.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0031] With reference to Figs. 1 through 5, embodiments of the invention are presented. Systems and methods are presented for performing multi-stream storage operations including multi-stream storage operations associated with a single sub-client.

[0032] Fig. 1 presents a block diagram of a storage operation cell in a system to perform storage operations on electronic data in a computer network according to an embodiment of the invention. As shown, the storage operation cell may include a storage management component, such as storage manager 100 and one or more of the following: a client 85, a data store 90, a data

agent 95, a media management component, such as a media agent 125, a media management component index cache 130, a storage device 135, a storage management component index cache 105, a jobs agent 110, an interface module 115, and a management agent 120. The system and elements thereof are exemplary of a modular storage management system such as the CommVault QiNetix storage management system, available from CommVault Systems, Inc. of Oceanport, NJ, and further described in Application Serial Number 09/610,738 which is incorporated herein by reference in its entirety.

[0033] A storage operation cell generally includes combinations of hardware and software components directed to performing storage operations on electronic data. Exemplary storage operation cells according to embodiments of the invention include CommCells as embodied in the QNet storage management system and the QiNetix storage management system by CommVault Systems of Oceanport, New Jersey, and as further described in Application Serial No. 60/482,305 and Application Serial No. 09/354,058 which are hereby incorporated by reference in their entirety.

[0034] According to some embodiments of the invention, storage operations cells are related to backup cells and provide all of the functionality of backup cells as further described in Application Serial No. 09/354,058. Storage operation cells also perform additional types of storage operations and provide other types of storage management functionality. According to embodiments of the invention, storage operation cells perform storage operations which also include, but are not limited to, creation, storage, retrieval, migration, deletion, and tracking of primary or production volume data, secondary volume data, primary copies, secondary copies, auxiliary copies, snapshot copies, backup copies, incremental copies, differential copies, HSM copies, archive copies, Information Lifecycle Management (“ILM”) copies, and other types of copies and versions of electronic data. In some embodiments, storage operation cells also

provide an integrated management console for users or system processes to interface with to perform storage operations on electronic data as further described herein.

[0035] A storage operation cell can be organized and associated with other storage operation cells forming a logical hierarchy among various components of a storage management system as further described herein. Storage operation cells generally include a storage manager 100, and, according to some embodiments, one or more other components including, but not limited to, a client computer 85, a data agent 95, a media management component 125, a storage device 135, and other components as further described herein.

[0036] For example, a storage operation cell may contain a data agent 95 which is generally a software module that is generally responsible for performing storage operations related to client computer 85 data stored in an data store 90 or other memory location, for example archiving, migrating, and recovering client computer data. In some embodiments, a data agent performs storage operations in accordance with one or more storage policies or other preferences. A storage policy is generally a data structure or other information that may include a set of preferences and other storage criteria for performing a storage operation. 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. As further described herein, storage policies 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.

[0037] Each client computer 85 generally has at least one data agent 95 and the system can support many client computers 85. The system also generally provides a plurality of data agents 95 each of which is intended to perform storage operations related to data associated with a different application, for example to backup, migrate, and recover application specific data. For

example, different individual data agents 95 may be designed to handle Microsoft Exchange data, Lotus Notes data, Microsoft Windows 2000 file system data, Microsoft Active Directory Objects data, and other types of data known in the art.

[0038] If a client computer 85 has two or more types of data, one data agent 95 is generally required for each data type to perform storage operations related to client computer 85 data. For example, to backup, migrate, and restore all of the data on a Microsoft Exchange 2000 server, the client computer 85 would use one Microsoft Exchange 2000 Mailbox data agent 95 to backup the Exchange 2000 mailboxes, one Microsoft Exchange 2000 Database data agent 95 to backup the Exchange 2000 databases, one Microsoft Exchange 2000 Public Folder data agent 95 to backup the Exchange 2000 Public Folders, and one Microsoft Windows 2000 File System data agent 95 to backup the client computer's 85 file system. These data agents 95 would be treated as four separate data agents 95 by the system even though they reside on the same client computer 85. In some embodiments, separate data agents may be combined to form a virtual data agent (not shown) for performing storage operations related to a specific application. Thus, the four separate data agents of the previous example could be combined as a virtual data agent suitable for performing storage operations related to all types of Microsoft Exchange 2000 and/or Windows 2000 data.

[0039] The storage manager 100 is generally a software module or application that coordinates and controls storage operations performed by the storage operation cell. The storage manager 100 communicates with all elements of the storage operation cell including client computers 85, data agents 95, media management components 125, and storage devices 135 regarding storage operations, for example to initiate and manage system backups, migrations, and recoveries. The storage manager 100 also communicates with other storage operation cells as further described herein.

[0040] The storage manager 100 includes a jobs agent 110 software module which monitors the status of all storage operations that have been performed, that are being performed, or that are scheduled to be performed by the storage operation cell. The jobs agent 110 is communicatively coupled with an interface agent 115 software module. The interface agent 115 provides presentation logic, such as a graphical user interface (“GUI”), an application program interface (“API), or other interface by which users and system processes can retrieve information about the status of storage operations and issue instructions to the storage operations cell regarding performance of storage operations as further described herein. For example, a user might modify the schedule of a number of pending snapshot copies or other types of copies. As another example, a user might use the GUI to view the status of all storage operations currently pending in all storage operation cells or the status of particular components in a storage operation cell.

[0041] The storage manager 100 also includes a management agent 120 software module. The management agent 120 generally provides an interface with other management components 100 in other storage operations cells through which information and instructions regarding storage operations may be conveyed. For example, in some embodiments as further described herein, a management agent 120 in first storage operation cell can communicate with a management agent 120 in a second storage operation cell regarding the status of storage operations in the second storage operation cell. In some embodiments, a management agent 120 in first storage operation cell can communicate with a management agent 120 in a second storage operation cell to control the storage manager 100 (and other components) of the second storage operation cell via the management agent 120 contained in the storage manager 100 for the second storage operation cell. In other embodiments, the management agent 120 in the first storage operation cell communicates directly with and controls the components in the second storage management cell and bypasses the storage manager 100 in the second storage management cell.

Storage operation cells can thus be organized hierarchically among cells and as further described herein.

[0042] A media management component 125 is generally a software module that conducts data, as directed by a storage manager 100, between client computers 85 and one or more storage devices 135. The media management component 125 is communicatively coupled with and generally configured to control one or more storage devices 135. For example, the media management component 125 might instruct a storage device 135 to use a robotic arm or other means to load or eject a media cartridge, and to archive, migrate, or restore application specific data. The media management component 125 generally communicates with storage devices 135 via a local bus such as a SCSI adaptor. In some embodiments, the storage device 135 is communicatively coupled to the media management component 125 via a Storage Area Network ("SAN").

[0043] Each media management component 125 maintains an index cache 130 which stores index data the system generates during storage operations as further described herein. For example, storage operations for Microsoft Exchange data generate index data. Index data may include, for example, 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, 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. Index data thus provides the system with an efficient mechanism for performing storage operations including locating user files for recovery operations and for managing and tracking stored data. The system generally maintains two copies of the index data regarding particular stored data. A first copy is generally stored with the data copied to a storage

dévice 135. 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 130 or other index useful in performing and/or managing storage operations. In addition, the media management component 125 that controls the storage operation also may generally write an additional copy of the index data to its index cache 130. The data in the media management component index cache 130 is generally stored on faster media, such as magnetic media, and is thus readily available to the system for use in storage operations and other activities without having to be first retrieved from the storage device 135.

[0044] The storage manager 100 may also maintain an index cache 105. Storage manager index data may be, among other things, used to indicate, track, and associate logical relationships and associations between components of the system, user preferences, management tasks, and other useful data. For example, the storage manager 100 might use its index cache 105 to track logical associations between media management components 125 and storage devices 135. The storage manager 100 may also use index cache 105 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. Index caches 105 and 130 typically reside on their corresponding storage component’s hard disk or other fixed storage device.

[0045] For example, jobs agent 110 of a storage manager component 100 may retrieve storage manager index 105 data regarding a storage policy and storage operation to be performed or scheduled for a particular client 85. The jobs agent 110, either directly or via the interface module 115, communicates with the data agent 95 at the client 85 regarding the storage

operation. In some embodiments, the jobs agent 110 also retrieves from index cache 105 a storage policy associated with client 85 and uses information from the storage policy to communicate to data agent 95 one or more media management components 125 associated with performing storage operations for that particular client 85 as well as other information regarding the storage operation to be performed such as retention criteria, encryption criteria, streaming criteria, etc. Data agent may 95 then package or otherwise manipulate client data stored in client data store 90 in accordance with the storage policy information and/or according to a user preference, and may communicate this client data to the appropriate media management component(s) 125 for processing. Media management component(s) 125 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 130.

[0046] In some embodiments, components of the system may reside and execute on the same computer. In some embodiments, a client computer 85 component such as a data agent 95, a media management component 125, or a storage manager 100 coordinates and directs storage operations as further described in Application Number 09/610,738. This client computer 85 component can function independently or together with other similar client computer 85 components.

[0047] Fig. 2 presents a block diagram of a hierarchically organized group of storage operation cells in a system to perform storage operations on electronic data in a computer network according to an embodiment of the invention. As shown, the system may include a master storage manager component 140, a first storage operation cell 145, a second storage operation cell 150, a third storage operation cell 155, a fourth storage operation cell 160, a fifth storage operation cell 165, and an nth storage operation cell 170.

[0048] As previously described, storage operation cells are often communicatively coupled and hierarchically organized. For example, as shown in Fig. 2, a master storage manager 140 is associated with, communicates with, and directs storage operations for a first storage operation cell 145, a second storage operation cell 150, a third storage operation cell 155, a fourth storage operation cell 160, a fifth storage operation cell 165, and an nth storage operation cell 170. In some embodiments, the master storage manager 140 is not part of any particular storage operation cell. In other embodiments (not shown), the master storage manager 140 may itself be part of a storage operation cell.

[0049] Thus, the master storage manager 140 communicates with the manager agent of the storage manager of the first storage operation cell 145 (or directly with the other components of the first cell 145) regarding storage operations performed in the first storage operation cell 145. For example, in some embodiments, the master storage manager 140 instructs the first storage operation cell 145 how and when to perform storage operations including the type of operation to perform and the data on which to perform the operation.

[0050] In other embodiments, the master storage manager 140 tracks the status of its associated storage operation cells, such as the status of jobs, system components, system resources, and other items, by communicating with manager agents (or other components) in the respective storage operation cells. In other embodiments, the master storage manager 140 tracks the status of its associated storage operation cells by receiving periodic status updates from the manager agents (or other components) in the respective cells regarding jobs, system components, system resources, and other items. For example, in some embodiments, the master storage manager 140 uses methods to monitor network resources such as mapping network pathways and topologies to, among other things, physically monitor storage operations and suggest alternate routes for storing data as further described herein. The master storage manager 140 also uses

methods to monitor primary and secondary storage trends, storage status, media usage, data protection levels, and other storage-related information as further described herein.

[0051] In some embodiments, the master storage manager 140 stores status information and other information regarding its associated storage operation cells and the system in an index cache or other data structure accessible to the master storage manager 140. In some embodiments, as further described herein, the presentation interface of the master storage manager 140 accesses this information to present users and system processes with information regarding the status of storage operations, storage operation cells, system components, and other information of the system.

[0052] Storage operation cells may thus be organized hierarchically. Consequently, storage operation cells may inherit properties from “parent” or hierarchically superior cells or be controlled by other storage operation cells in the hierarchy. Thus, in some embodiments as shown in Fig. 2, the second storage operation cell 150 controls or is otherwise superior to the third storage operation cell 155, the fourth storage operation cell 160, the fifth storage operation cell 165, and the nth storage operation cell 170. Similarly, the fourth storage operation cell 160 controls the fifth storage operation cell 165, and the nth storage operation cell 170.

[0053] Storage operation cells may also be organized hierarchically according to criteria such as function (e.g., superior or subordinate), geography, architectural considerations, or other factors useful in performing storage operations. For example, in one embodiment storage operation cells are organized according to types of storage operations: the first storage operation cell 145 may be directed to performing snapshot copies of primary copy data, and the second storage operation cell 150 may be directed to performing backup copies of primary copy data or other data. In another embodiment, the first storage operation cell 145 may represent a geographic segment of an enterprise, such as a Chicago office, and a second storage operation cell 150 represents a different geographic segment, such as a New York office. In this example,

the second storage operation cell 150, the third storage operation cell 155, the fourth storage operation cell 160, the fifth storage operation cell 165, and the nth storage operation cell 170 could represent departments within the New York office. Alternatively, these storage operation cells could be further divided by function performing various types of copies for the New York office or load balancing storage operations for the New York office.

[0054] In some embodiments, hierarchical organization of storage operation cells may facilitate, among other things, system security and other considerations. For example, in some embodiments, only authorized users are allowed to access or control certain storage operation cells. For example, a network administrator for an enterprise might have access to all storage operation cells including the master storage manager 140. But a network administrator for only the New York office, according to a previous example, might only satisfy access criteria to have access to the second storage operation cell 150, the third storage operation cell 155, the fourth storage operation cell 160, the fifth storage operation cell 165, and the nth storage operation cell 170 which comprise the New York office storage management system.

[0055] In some embodiments, hierarchical organization of storage operation cells facilitates storage management planning and decision-making. For example, in some embodiments, a user of the master storage manager 140 can view the status of all jobs in the associated storage operation cells of the system as well as the status of each component in every storage operation cell of the system. The user can then plan and make decisions based on this global data. For example, the user can view high-level report of summary information regarding storage operations for the entire system, such as job completion status, component availability status, resource usage status (such as network pathways, etc.), and other information. The user can also drill down through menus or use other means to obtain more detailed information regarding a particular storage operation cell or group of storage operation cells.

[0056] In other embodiments, the master storage manager 140 may alert a user or system administrator when a particular resource is unavailable (e.g., temporary or permanent) or congested. A storage device may be full or require additional media. Alternatively, a storage manager in a particular storage operation cell may be unavailable due to hardware failure, software problems, or other reasons. In some embodiments, the master storage manager 140 (or another storage manager within the hierarchy of storage operation cells) may utilize the global data regarding its associated storage operation cells at its disposal to suggest solutions to such problems when they occur or even before they occur. For example, the master storage manager 140 might alert the user that a storage device in a particular storage operation cell was full or otherwise congested, and then suggest, based on job and data storage information contained in its index cache, an alternate storage device.

[0057] As another example, in some embodiments the master storage manager 140 (or other network storage manager) contains programming directed to analyzing the storage patterns and resources of its associated storage operation cells and which suggests optimal or alternate methods of performing storage operations. Thus, for example, the master storage manager 140 may analyze traffic patterns to determine that snapshot data should be sent via a different network segment or to a different storage operation cell or storage device. In some embodiments, users can direct specific queries to the master storage manager 140 regarding predicting storage operations or regarding storage operation information.

[0058] Fig. 3 presents a block diagram of a hierarchically organized group of storage operation cells in a system to perform storage operations on electronic data in a computer network according to an embodiment of the invention. As shown, Fig. 3 includes a first storage operation cell 175, a second storage operation cell 180, a third storage operation cell 185, a client 190 in communication with a primary volume 195 storing production or other "live" data, a storage manager component 200 in communication with a storage manager index data store 205,

a media management component 210 in communication with a media management component index 215 a secondary storage device or volume 220, and a master storage manager component 225 in communication with a master storage manager index data store 230.

[0059] According to an embodiment of the invention, the first storage operation cell 175 may be directed to a particular type storage operation, such as SRM storage operations. For example, the first storage operation cell 175 monitors and performs SRM-related calculations and operations associated with primary volume 195 data. Thus, the first storage operation cell 175 includes a client component 190 in communication with a primary volume 195 storing data. For example, the client 190 may be directed to using Exchange data, SQL data, Oracle data, or other types of production data used in business applications or other applications and stored in primary volume 195. Storage manager component 200 in cell 175 contains SRM modules or other logic directed to monitoring or otherwise interacting with attributes, characteristics, metrics, and other information associated with the data stored in primary volume 195. Storage manager 200 tracks and stores this information and other information in storage manager index 205. For example, in some embodiments, storage manager component 200 tracks the amount of available space and other similar characteristics of data associated with primary volume 195. In some embodiments, as further described herein, storage manager component 200 may also issue alerts or take other actions when the information associated with primary volume 195 satisfies certain criteria, such as alert criteria.

[0060] The second storage operation cell 180 may be directed to another type storage operation, such as HSM storage operations. For example, the second storage operation cell 180 may perform backups, migrations, snapshots, or other types of HSM-related operations known in the art. For example, in some embodiments, data is migrated from faster and more expensive storage such as magnetic storage to less expensive storage such as tape storage.

[0061] In some embodiments, storage operation cells may also contain logical groupings of the same physical devices. Thus, the second storage operation cell 180 includes the client component 190 in communication with the primary volume 195 storing data, and client component 190 and primary volume 195 in the second storage operation cell 180 are the same physical devices as the client component 190 and primary volume 195 in the first storage operation cell 175. Similarly, in some embodiments, the storage manager component 200 and index 205 in the second storage operation cell 180 are the same physical devices as the storage manager component and index in the first storage operation cell 175. The storage manager component 200, however, also contains HSM modules or other logic associated with the second storage operation cell 180 directed to performing HSM storage operations on primary volume 195 data.

[0062] The second storage operation cell 180 therefore also contains a media management component 210, a media management component index 215, and a secondary storage volume 220 directed to performing HSM-related operations on primary copy data. For example, storage manager 200 migrates primary copy data from primary volume 195 to secondary volume 220 using media management component 210. Storage manager 200 also tracks and stores information associated with primary copy migration and other similar HSM-related operations in storage manager index 205. For example, in some embodiments, storage manager component 200 directs HSM storage operations on primary copy data according to a storage policy associated with the primary copy 195 and stored in the index 205. In some embodiments, storage manager 200 also tracks where primary copy information is stored, for example in secondary storage 220.

[0063] The third storage operation cell 185 contains a master storage manager 225 and a master storage manager index 230. In some embodiments (not shown), additional storage operation cells might be hierarchically located between the third storage operation cell 185 and

the first storage operation cell 175 or the second storage operation cell 180. In some embodiments, additional storage operation cells hierarchically superior to the third storage operation cell 185 may also be present in the hierarchy of storage operation cells.

[0064] In some embodiments, the third storage operation cell 185 is also generally directed to performing a type of storage operation, such as integration of SRM and HSM data from other storage operation cells, such as the first storage operation cell 175 and the second storage operation cell 180. In other embodiments, the third storage operation cell 185 also performs other types of storage operations and might also be directed to HSM, SRM, or other types of storage operations. In some embodiments, the master storage manager 225 of the third storage operation cell 185 aggregates and processes network and storage-related data provided by other manager components 200 in other storage operation cells 175 and 180 in order to provide, among other information, reporting information regarding particular cells, groups of cell, or the system as a whole.

[0065] Fig. 4 presents a flow diagram of a method to store chunks of data in a single instance storage device. The system may receive or generate an instruction to copy one or more chunks of data to a single instance storage device, step 235. For example, the system may receive a message to copy twenty chunks of archive data comprising a client e-mail data store containing thousands of files in each chunk. As discussed, each chunk of data may contain payload information representing the data items from the client data store as well as metadata describing the contents of each chunk, storage preferences associated with each chunk, associations between chunks, etc.

[0066] The chunk metadata may be separated from the chunk payload at step 240. For example, data pipe modules as further described herein may unencapsulate the chunk to extract payload information or otherwise process the chunk to separate the chunk metadata from the chunk payload. The chunk metadata may be copied to a data store, for example a storage

management component index of a media management component index, and associated with the chunk payload to be stored in the single instance storage device as further described herein (step 245). In some embodiments, the chunk metadata may also be copied to a single instance storage device, but be logically or physically separated (e.g., maintained as a separate container or data item from the chunk payload). Thus, the chunk payload may be saved as a single instance while the metadata can still be preserved.

[0067] Separation of metadata from payload data may be performed in a number of ways. For example, a data agent residing on a client may separate the chunk metadata from the payload data and transmit each portion to a single instance storage device either separately or together to one or more destinations. This may also involve, as described herein, transmitting metadata and payload data to different logical or physical locations, with the appropriate update of management components or indicies to maintain correlation between the payload data and the metadata. Other arrangements may include one or more a media agents examining or analyzing chunks to be transmitted and separating metadata from payload data at the client or sub-client based on direction from a media agent, or such separation may occur while the chunk is substantially in transit, with the media agents routing the payload to one location and the metadata to another. Moreover, such separation may occur at the storage device(s) with certain metadata and/or payload data tagged or otherwise deemed suitable for single instance storage and separated, for example, while queued at the storage device, whereas other metadata and payload data, not suitable for single instance storage, may be stored elsewhere.

[0068] Items from the chunk payload may be copied on an item-by-item basis to the single instance storage device at step 250. Thus, the chunk payload may contain several thousand e-mail messages and attachments, each of which is copied to the single instance storage device for storage. The single instance storage device may generate a suitable identifier, such as a signature, a cryptographic hash, or other identifier, for each item, and store each item as

appropriate according to its identifier (e.g., items previously stored are generally not stored again and may simply be overwritten, new items for which identifiers do not already exist on an index, map, or other tracking means are stored, etc.). The identifier may be communicated for each item to a system component, such as a storage management component or a media management component, responsible for maintaining the index data containing the metadata associated with the chunk payload (step 255).

[0069] In some embodiments, however, certain sections of payload data may not be suitable for single instance storage. In this case, such data may be separated from the other chunk data and stored using conventional means. Thus, single instance data and other data from the chunk may be stored in logically or physically separate locations. This allows at least certain portions of chunk data to be stored using single instance storage techniques. A data index, database or other management component may keep track of the various locations to facilitate subsequent restore or copy operations.

[0070] The identifier returned for each item from the single instance storage device may be associated with the identifier for the item maintained in the metadata associated with the chunk (step 260). For example, chunk metadata generally tracks the contents of the payload of each chunk on an item by item basis. A chunk containing a payload of 1000 e-mail messages and attachments may contain metadata identifying the 1000 items of the payload, for example by file name, message index ID, or other identifier. Thus, the chunk metadata may maintain logical identifiers such as file system identifiers or other identifiers corresponding to each item stored in the chunk payload.

[0071] When single instance storage identifiers are returned for each item processed and stored by the single instance storage device, the single instance identifiers are associated with the logical identifiers previously maintained in the chunk metadata. Thus, a map or other index structure may be created and maintained in the copy of metadata associated with the chunk

payload items stored in single instance storage that correlates single instance storage identifiers or physical storage identifiers with the logical identifiers maintained in the original chunk metadata prior to single instance storage. For example, the original chunk metadata may contain separate entries for a File A and a File B which are actually instances of the same copy of data, for example of the same e-mail attachment. When File A and File B are processed by the single instance storage device, they may each generate the same single instance storage identifier, such as a hash or signature, and the single instance storage device would know to only store one copy or instance of the data. Nevertheless, the single instance storage device would still return an identifier for each file. Thus, when File A was sent to the single instance storage device, its signature would be returned and associated with File A in the chunk metadata. Similarly, when File B was sent to the single instance storage device, the same signature would be returned, but this time associated with File B in the chunk metadata. This arrangement allows the single instance of the attachment to be referenced by both files rather than storing two instances of the same attachment.

[0072] Thus, the chunk metadata can still be used to recreate the original chunk of data or to retrieve various files from the chunk according to the original chunk metadata. For example, a user or other system component could be presented with a logical view of the chunk by recreating a representation of the chunk contents using the chunk metadata. Thus, although only 600 of 1000 files might be stored in single instance storage due to multiple instances of data, etc., the system could still present a logical view of the chunk containing 1000 files (including, for example the same instances of data with different file names, etc.). If a user wanted to retrieve a file, as further described herein, the system may use the map correlating logical identifiers of the original chunk metadata/payload with single instance storage identifiers to retrieve the requested item from single instance storage. Similarly, users can also browse or otherwise navigate or interact with items in the single instance storage device to view items actually stored, etc. For

example, a user might wish to interact with contents of a chunk payload containing 1000 files which would normally use 1000MB of storage space on non-single instanced storage and only occupies 500MB on single instanced storage.

[0073] In this case, the user may perform storage operations regarding each of the 1000 files according to the 1000 logical identifiers maintained by the chunk metadata, determine that the 1000 files are only costing the system 500MB in terms of storage (due to their storage on single instance storage), understand that the 1000 files are stored as 500 files, for example, on the single instance storage device, and also understand that the 1000 files would require 1000MB of storage space on non-single instanced storage.

[0074] In some embodiments, the system may also support cross-referencing between copies of metadata regarding different chunks (step 265). For example, the system may cross-reference single instance storage identifiers to identify duplications among items contained in a plurality of chunks of data in order to more accurately provide storage-related information and perform storage operations. For example, the system may track on a system-wide level across all chunks how much data is stored as multiple instances, etc.

[0075] Fig. 5 presents a flow diagram of a method for retrieving chunk payload data from a single instance storage device. At step 335, the system may receive or generate a request to retrieve data from single instance storage. For example, the system may receive a request to migrate the payload of a chunk to less expensive storage media such as tape media.

[0076] At step 340, the logical identifier of the first item of the chunk payload as identified by the metadata may be correlated to its corresponding single instance storage identifier. This item may be requested from the single instance storage device using its single instance storage identifier. For example, an item as originally contained in the chunk payload may have been described in the chunk metadata as File A whereas File A may be associated with a single instance storage signature of 207778627604938. To restore File A, the system may request the

item having this storage signature from the single instance storage device. Other files in the chunk payload may also have the same single instance identifier, but will likely have different logical identifiers in the chunk metadata. Thus, each item may be retrieved from the single instance storage device using its single instance storage identifier and then reassociated in a new chunk with its previous logical identifier as further described herein (step 345).

[0077] At step 350, the system may consult the chunk metadata to determine whether additional items remain in the original payload of the chunk. If additional items remain, the system may return to step 340 and the next item logically identified by the chunk metadata is retrieved from the single instance storage device using its single instance storage identifier and then reassociated with its previous logical identifier. When no further items remain to be retrieved from the single instance storage device, the system finishes recreating the chunk by encapsulating all of the items retrieved with the appropriate chunk metadata (step 355), and copies the new copy of the original chunk to the desired storage location (step 360).

[0078] In some embodiments, the chunk may be recreated on an item by item basis as items are returned from the single instance storage device. In other embodiments, items are first returned to a buffer or other temporary storage location until all items are returned and then the chunk is recreated. Thus, the new copy of the chunk is generally an exact copy of the chunk before it was stored in single instance storage, yet the metadata regarding the chunk is preserved for use in future storage operations.

[0079] Systems and modules described herein may comprise software, firmware, hardware, or any combination(s) of software, firmware, hardware, or other means 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, methods, or means, 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, physical 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 generally input, access, change, manipulate, modify, alter, and work with information.

[0080] 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 performing a storage operation on a computer network comprising:
 - receiving a request to perform the storage operation on a first set of data;
 - analyzing the first set of data;
 - characterizing the first set of data into a first portion and a second portion based on characteristics observed in the analyzing step;
 - copying the first portion of the data to a first single instance storage location; and
 - associating an identifier with the first portion of data stored at the first single instance storage location.

2. The method of claim 1 further comprising updating a database associated with a management component of a storage operation cell with the identifier information.

3. The method of claim 1 wherein the copying further comprises routing the first portion of the data to the first single instance storage location via a media agent.

4. The method of claim 1 wherein the second portion of the data is metadata relating to the second portion of data.

5. The method of claim 3 wherein the first portion of the data is payload data.

6. The method of claim 1 wherein the first portion of data is copied to the first single instance location, based at least in part, on the characterization step.

7. The method of claim 6 wherein the first portion of the data is copied item by item to the first single instance storage location.

8. The method of claim 7 further comprising associating an identifier with the first portion of data stored at the first single instance storage location.

9. The method of claim 8 wherein an identifier is associated with the first portion of data stored at the first single instance storage location, the method further comprising correlating the identifier associated with the first portion of data with an identifier associated with the second portion of data.

10. The method of claim 1 wherein the characterization further comprises:
characterizing the first portion of data into first and second sub-portions of data;
wherein the first sub-portion of the first portion of data is suitable for single instance storage and the second sub-portion of the first portion is suitable for conventional storage.

11. The method of claim 10 wherein the copying further comprises copying the second sub-portion of the first portion of data to a first storage device.

12. The method of claim 10 wherein the copying further comprises copying the second sub-portion of the first portion of data to a first storage device.

13. The method of claim 10 further comprising:

copying the second sub-portion of the first portion of data to a first storage device;

copying the second sub-portion of the first portion of data to the first storage device; and

updating a database associated with a management component of a storage operation cell to reflect copy operations associated with the data.

14. A method for recreating data stored in a storage network comprising:

receiving a request to retrieve a portion data stored in a storage network;

identifying a location of the of the portion of data, wherein at least some of the portion of data is located in a single instance storage device;

retrieving from the single instance storage device data identified in the identifying step;

consulting the retrieved data to determine whether additional data relating to the retrieved data is available; and

recreating the data portion based, at least in part, on data retrieved from the single instance storage device.

15. The method of claim 16 further comprising retrieving additional data from the single instance storage device if it is determined in the consulting step that additional data relating to the retrieved data is available.

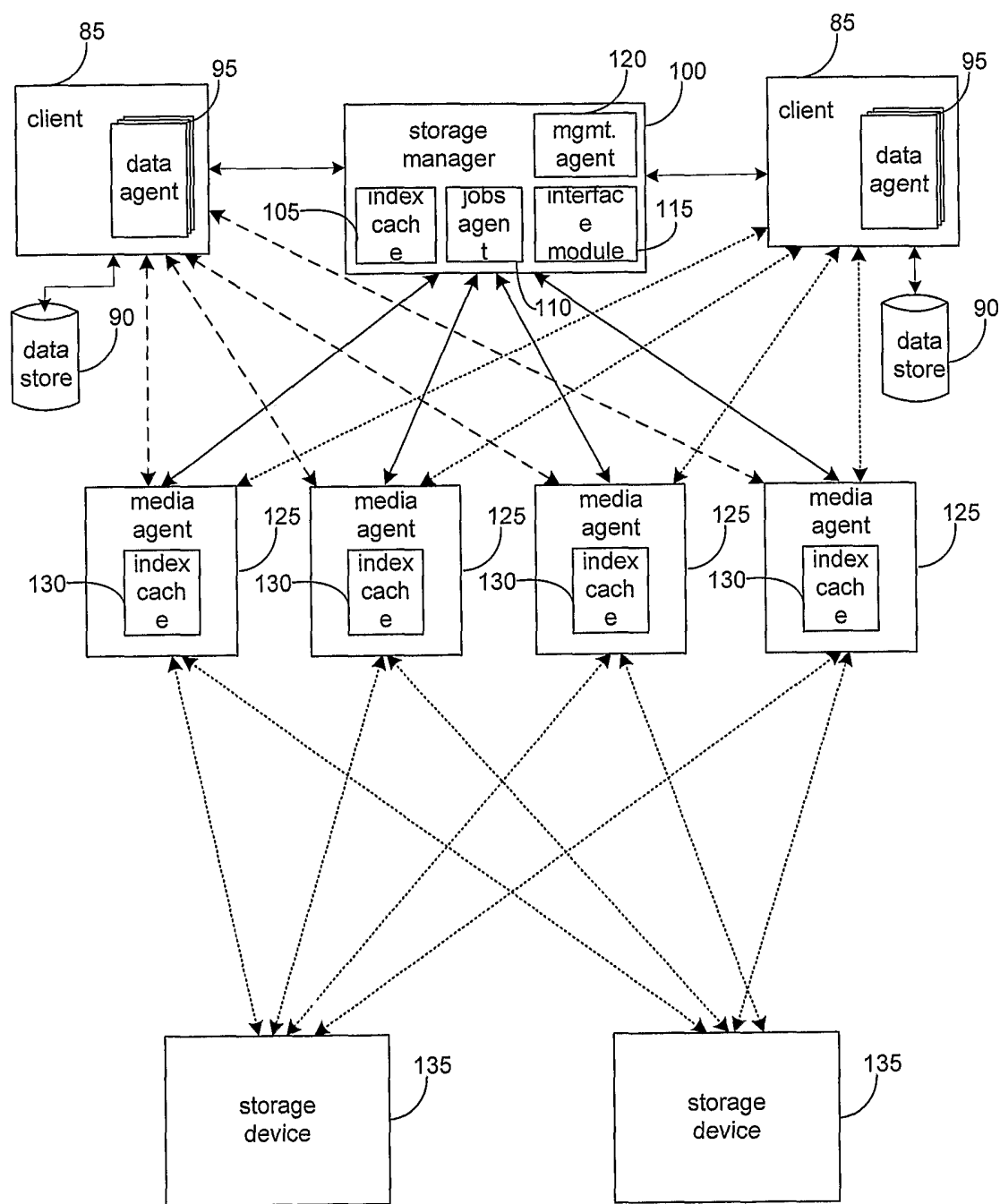


FIG. 1

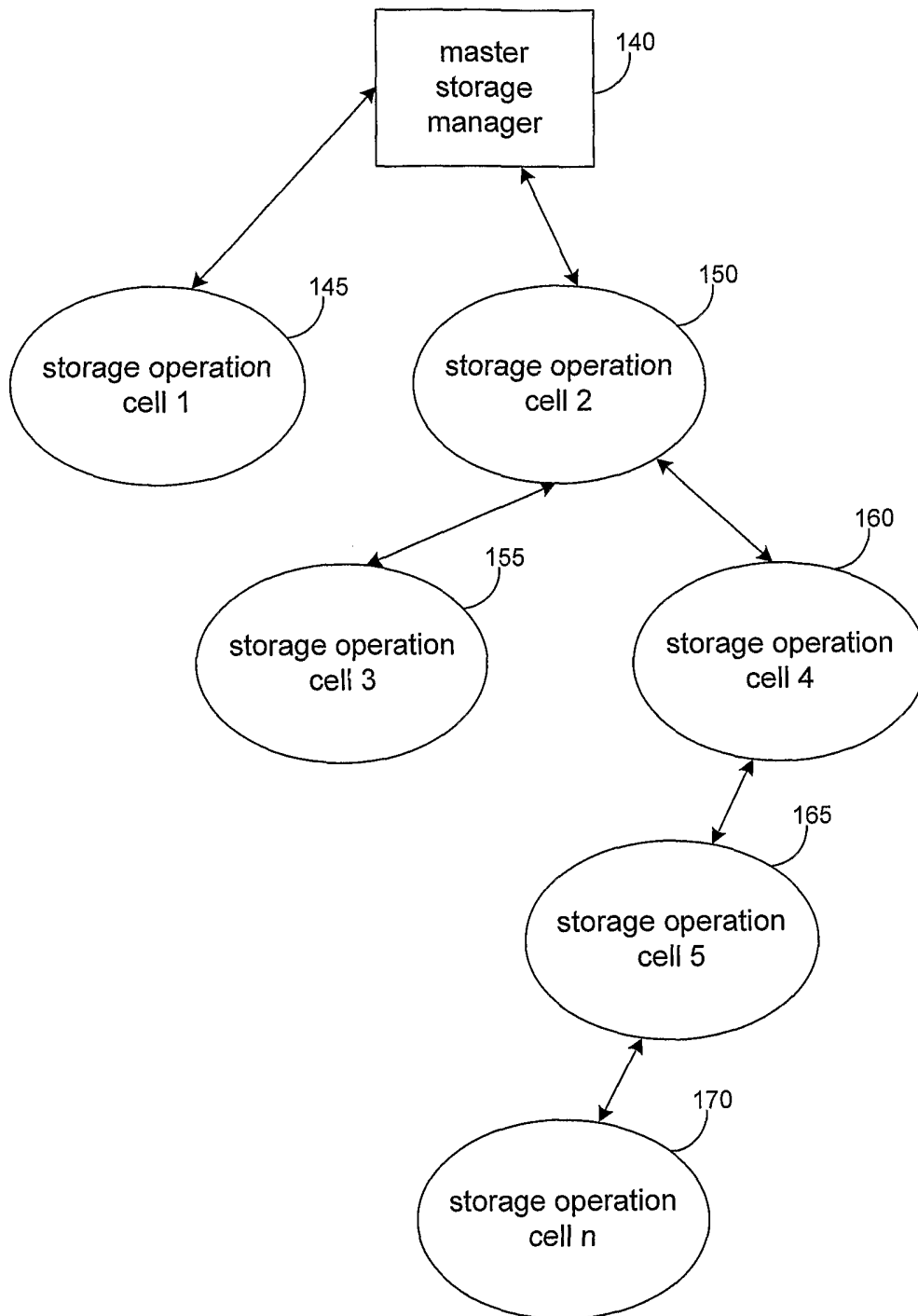


FIG. 2

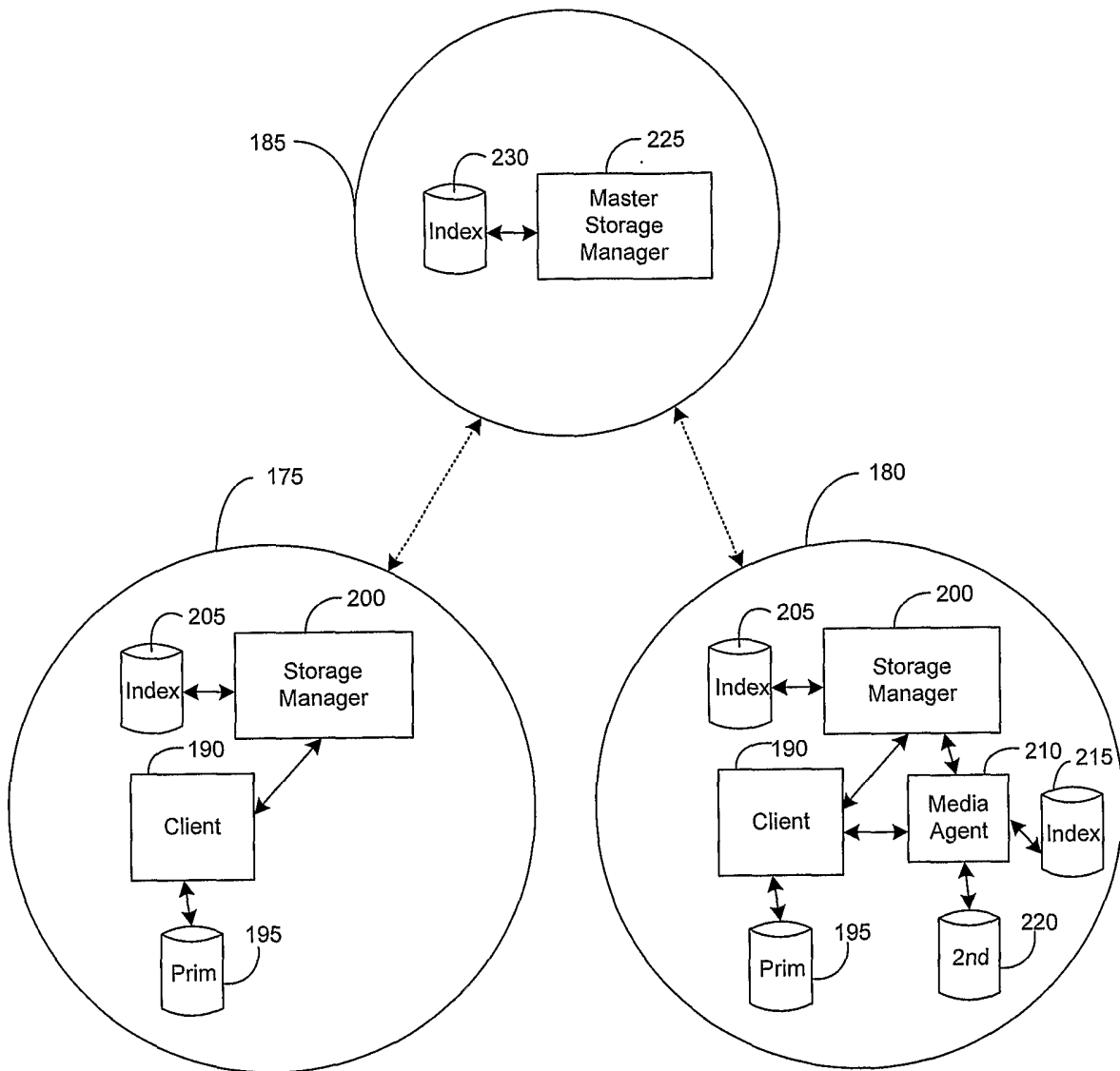


FIG. 3

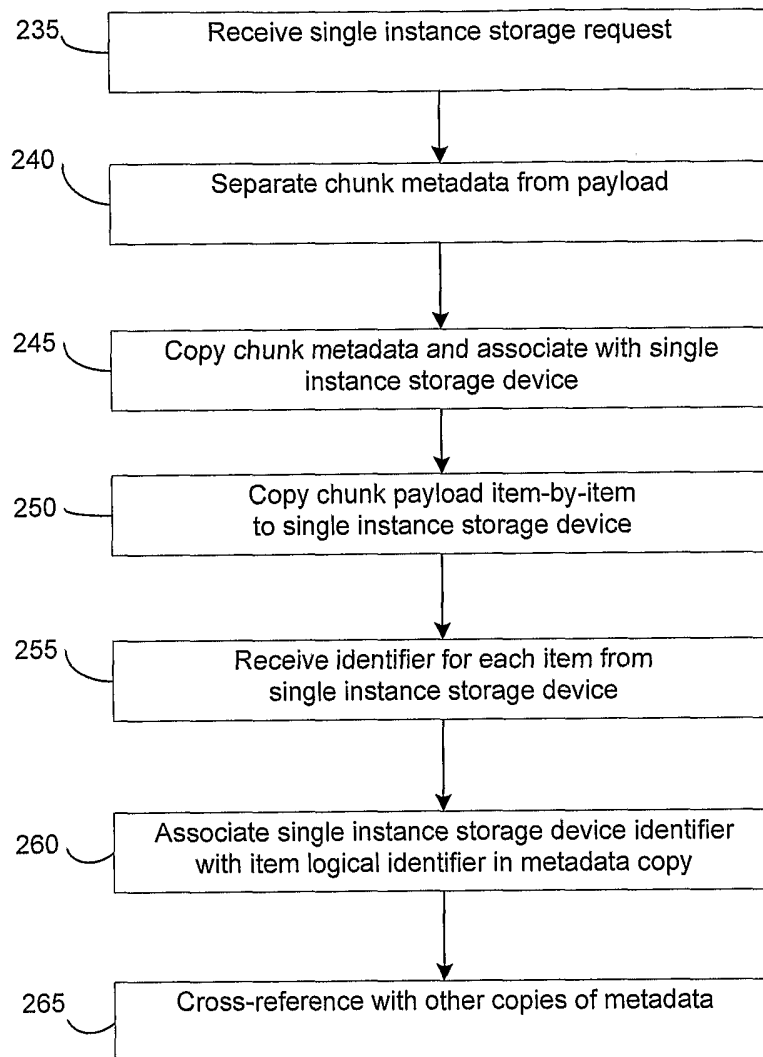


FIG. 4

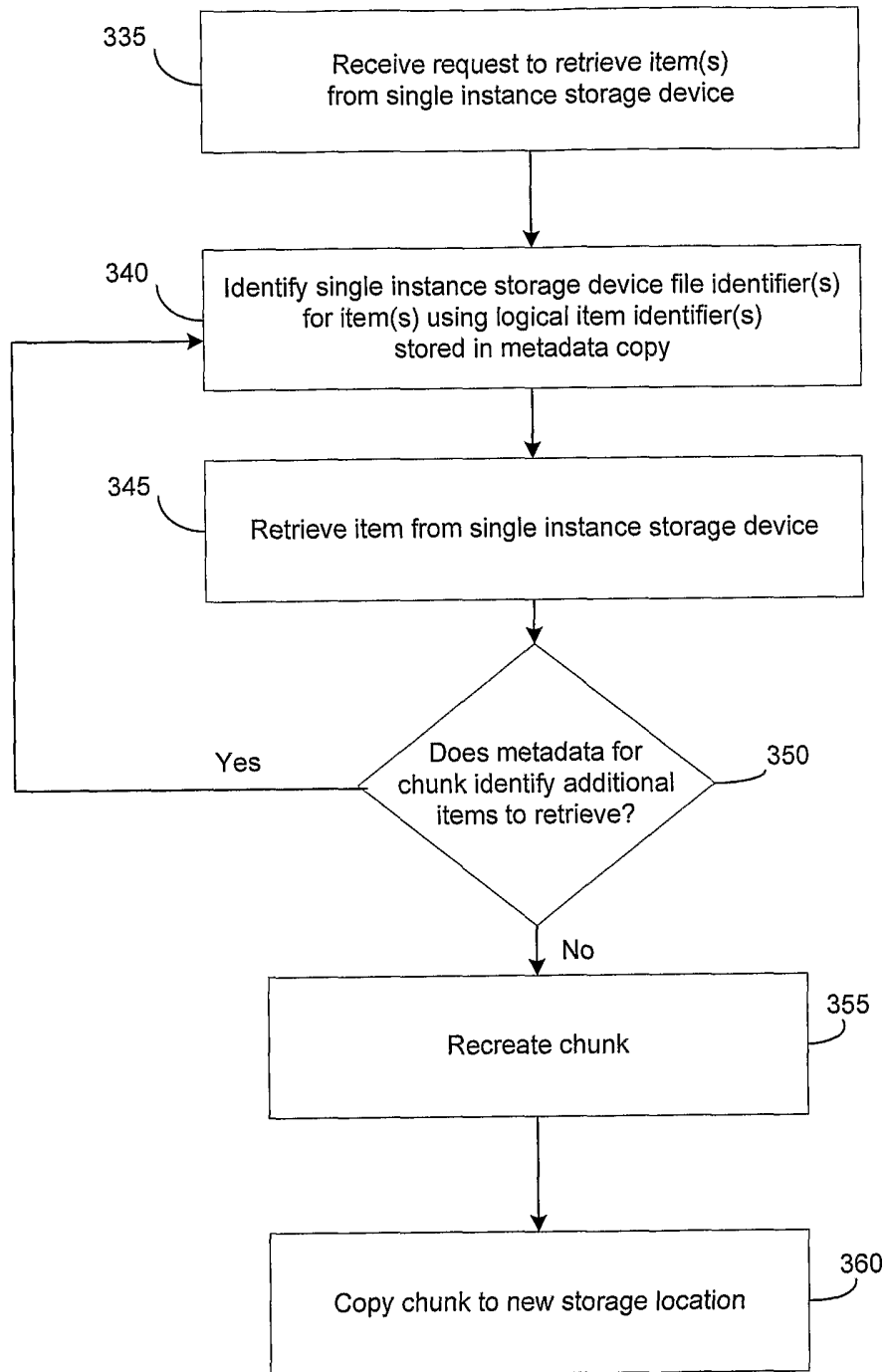


FIG. 5