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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
31 May 2007 (31.05.2007)

PCT

(10) International Publication Number
WO 2007/062429 A2

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

(21) International Application Number:
PCT/US2006/061304

(22) International Filing Date:
28 November 2006 (28.11.2006)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/740,686 28 November 2005 (28.11.2005) US
60/752,203 19 December 2005 (19.12.2005) US

(71) Applicant *(for all designated States except US)*: **COMMVAULT SYSTEMS, INC.** [US/US]; 2 CRESCENT PLACE, Oceanport, NJ 07757-0090 (US).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **PRAHLAD, Anand** [US/US]; 3 BUCKNELL DRIVE, East Brunswick, NJ 08816 (US). **SCHWARTZ, Jeremy, A.** [US/US]; 30 DRUMMOND PL, Red Bank, NJ 07701 (US). **NGO, David** [US/US]; 118 BORDEN ST, Shrewsbury, NJ 07702 (US). **BROCKWAY, Brian** [US/US]; 3 BRADY ROAD, Shrewsbury, NJ 07702 (US). **MULLER, Marcus, S.** [US/US]; 4 FENNEC COURT, Tinton Falls, NJ 07757 (US).

(74) Agents: **DALEY-WATSON, Christopher, J.** et al.; PERKINS COIE LLP, P.O. BOX 1247, Seattle, WA 98111-1247 (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, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LV, LY, MA, MD, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, 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.

WO 2007/062429 A2

(54) **Title:** SYSTEMS AND METHODS FOR CLASSIFYING AND TRANSFERRING INFORMATION IN A STORAGE NETWORK

(57) **Abstract:** Systems and methods for data classification to facilitate and improve data management within an enterprise are described. The disclosed systems and methods evaluate and define data management operations based on data characteristics rather than data location, among other things. Also provided are methods for generating a data structure of metadata that describes system data and storage operations. This data structure may be consulted to determine changes in system data rather than scanning the data files themselves.

SYSTEMS AND METHODS FOR CLASSIFYING AND TRANSFERRING INFORMATION IN A STORAGE NETWORK

CROSS-REFERENCE TO RELATED APPLICATION(S)

[0001] This application claims the benefit of United States Provisional Application No. 60/740,686, entitled SYSTEMS AND METHODS FOR CLASSIFYING INFORMATION IN A STORAGE NETWORK, filed November 28, 2005, attorney docket no. 60692-8029.US00 and United States Provisional Application No. 60/752,203, entitled SYSTEMS AND METHODS FOR CLASSIFYING AND TRANSFERRING INFORMATION IN A STORAGE NETWORK, filed December 19, 2005, attorney docket no. 60692-8029.US01 both of which are hereby incorporated herein by reference in their entirety.

[0002] This application is related to the following patents and pending U.S. applications, each of which is hereby incorporated herein by reference in its entirety:

- Application Serial No. 09/354,058, titled HIERARCHICAL BACKUP AND RETRIEVAL SYSTEM, filed July 15, 1999, docket number 4982/5;
- U.S. Patent No. 6,418,478, titled PIPELINED HIGH SPEED DATA TRANSFER MECHANISM, issued July 9, 2002, docket number 4982/6;
- Application Serial No. 09/610,738, titled MODULAR BACKUP AND RETRIEVAL SYSTEM USED IN CONJUNCTION WITH A STORAGE AREA NETWORK, filed July 6, 2000, docket number 4982/8;
- U.S. Patent No. 6,542972 titled LOGICAL VIEW AND ACCESS TO PHYSICAL STORAGE IN MODULAR DATA AND STORAGE MANAGEMENT SYSTEM, issued April 1, 2003, docket number 4982/10;
- U.S. Patent No. 6,658,436, titled LOGICAL VIEW AND ACCESS TO DATA MANAGE BY A MODULAR DATA AND STORAGE

MANAGEMENT SYSTEM, issued December 2, 2003, docket number 4982/13;

- Application Serial No. 10/658,095, titled DYNAMIC STORAGE DEVICE POOLING IN A COMPUTER SYSTEM, filed September 9, 2003, docket number 4982/18;
- Application Serial No. 10/262,556, titled METHOD FOR MANAGING SNAPSHOTS GENERATED BY AN OPERATING SYSTEM OR OTHER APPLICATION, filed September 30, 2002, docket number 4982/20;
- Application Serial No. 10/818,749, SYSTEM AND METHOD FOR DYNAMICALLY PERFORMING STORAGE OPERATIONS IN A COMPUTER NETWORK, filed April 5, 2004, docket number 4982/35;
- Application Serial No. 10/877,831, HIERARCHICAL SYSTEM AND METHOD FOR PERFORMING STORAGE OPERATIONS IN A COMPUTER NETWORK, filed June 25, 2004, docket number 4982/39;
- Application entitled SYSTEM AND METHOD FOR CONTAINERIZED DATA STORAGE AND TRACKING, filed December 19, 2005, docket number 4982/93
- Application Serial No. 60/752,198, entitled "Systems and Methods for Granular Resource Management in a Storage Network" filed December 19, 2005, docket number 4982/84
- Application Serial No. 11/313,224, entitled "Systems and Methods for Performing Multi-Path Storage Operations" filed December 19, 2005, docket number 4982/88
- Application Serial No. 60/752,196, entitled "Systems and Methods for Migrating Components in a Hierarchical Storage Network" filed December 19, 2005, docket number 4982/95

- Application Serial No. 60/752,202, entitled "Systems and Methods for Unified Reconstruction of Data in a Storage Network" filed December 19, 2005, docket number 4982/97
- Application Serial No. 60/752,201, entitled "Systems and Methods for Resynchronizing Storage Operations" filed December 19, 2005, docket number 4982/98
- Application Serial No. 60/752,197, entitled "Systems and Methods for Hierarchical Client Group Management" filed December 19, 2005, docket number 4982/102

BACKGROUND

[0003] Aspects of the invention disclosed herein relate generally to performing operations on electronic data in a computer network. More particularly, aspects of the present invention relate to detecting data interactions within a computer network and/or performing storage-related operations for a computer network according to a specified classification paradigm.

[0004] Current storage management systems employ a number of different methods to perform storage operations on electronic data. For example, data can be stored in primary storage as a primary copy or in secondary storage as various types of secondary copies including, as a backup copy, a snapshot copy, a hierarchical storage management copy ("HSM"), as an archive copy, and as other types of copies.

[0005] A primary copy of data is generally a production copy or other "live" version of the data which is used by a software application and is generally in the native format of that application. Primary copy data may be maintained in a local memory or other high-speed storage device that allows for relatively fast data access if necessary. Such primary copy data is typically intended for short term retention (e.g., several hours or days) before some or all of the data is stored as one or more secondary copies, for example to prevent loss of data in the event a problem occurred with the data stored in primary storage.

[0006] Secondary copies include point-in-time data and are typically for intended for long-term retention (e.g., weeks, months or years depending on

retention criteria, for example as specified in a storage policy as further described herein) before some or all of the data is moved to other storage or discarded. Secondary copies may be indexed so users can browse and restore the data at another point in time. After certain primary copy data is backed up, a pointer or other location indicia such as a stub may be placed in the primary copy to indicate the current location of that data.

[0007] One type of secondary copy is a backup copy. A backup copy is generally a point-in-time copy of the primary copy data stored in a backup format as opposed to in native application format. For example, a backup copy may be stored in a backup format that is optimized for compression and efficient long-term storage. Backup copies generally have relatively long retention periods and may be stored on media with slower retrieval times than other types of secondary copies and media. In some cases, backup copies may be stored at on offsite location.

[0008] Another form of secondary copy is a snapshot copy. From an end-user viewpoint, a snapshot may be thought as an instant image of the primary copy data at a given point in time. A snapshot generally captures the directory structure of a primary copy volume at a particular moment in time, and also preserves file attributes and contents. In some embodiments, a snapshot may exist as a virtual file system, parallel to the actual file system. Users typically gain a read-only access to the record of files and directories of the snapshot. By electing to restore primary copy data from a snapshot taken at a given point in time, users may also return the current file system to the prior state of the file system that existed when the snapshot was taken.

[0009] A snapshot may be created instantly, using a minimum of file space, but may still function as a conventional file system backup. A snapshot may not actually create another physical copy of all the data, but may simply create pointers that are able to map files and directories to specific disk blocks.

[0010] In some embodiments, once a snapshot has been taken, subsequent changes to the file system typically do not overwrite the blocks in use at the time of snapshot. Therefore, the initial snapshot may use only a small amount of disk space needed to record a mapping or other data structure representing or otherwise tracking the blocks that correspond to the current state of the file system. Additional

disk space is usually only required when files and directories are actually modified later. Furthermore, when files are modified, typically only the pointers which map to blocks are copied, not the blocks themselves. In some embodiments, for example in the case of copy-on-write snapshots, when a block changes in primary storage, the block is copied to secondary storage before the block is overwritten in primary storage and the snapshot mapping of file system data is updated to reflect the changed block(s) at that particular point in time. An HSM copy is generally a copy of the primary copy data, but typically includes only a subset of the primary copy data that meets a certain criteria and is usually stored in a format other than the native application format. For example, an HSM copy might include only that data from the primary copy that is larger than a given size threshold or older than a given age threshold and that is stored in a backup format. Often, HSM data is removed from the primary copy, and a stub is stored in the primary copy to indicate its new location. When a user requests access to the HSM data that has been removed or migrated, systems use the stub to locate the data and often make recovery of the data appear transparent even though the HSM data may be stored at a location different from the remaining primary copy data.

[0011] An archive copy is generally similar to an HSM copy, however, the data satisfying criteria for removal from the primary copy is generally completely removed with no stub left in the primary copy to indicate the new location (i.e., where it has been moved to). Archive copies of data are generally stored in a backup format or other non-native application format. In addition, archive copies are generally retained for very long periods of time (e.g., years) and in some cases are never deleted. Such archive copies may be made and kept for extended periods in order to meet compliance regulations or for other permanent storage applications.

[0012] In some embodiments, application data over its lifetime moves from more expensive quick access storage to less expensive slower access storage. This process of moving data through these various tiers of storage is sometimes referred to as information lifecycle management ("ILM"). This is the process by which data is "aged" from more forms of secondary storage with faster access/restore times down through less expensive secondary storage with slower access/restore times, for example, as the data becomes less important or mission critical over time.

[0013] Examples of various types of data and copies of data are further described in the above-referenced related applications that are hereby incorporated by reference in their entirety. One example of a system that performs storage operations on electronic data that produce such copies is the QiNetix storage management system by CommVault Systems of Oceanport, New Jersey.

[0014] The QiNetix system leverages a modular storage management architecture that may include, among other things, storage manager components, client or data agent components, and media agent components as further described in U.S. Patent Application Serial No. 10/818,749 which is hereby incorporated herein by reference in its entirety. The QiNetix system also may be hierarchically configured into backup cells to store and retrieve backup copies of electronic data as further described in U.S. Patent Application Serial No. 09/354,058 which is hereby incorporated by reference in its entirety.

[0015] Regardless of where data is stored, conventional storage management systems perform storage operations associated with electronic data based on location-specific criteria. For example, data generated by applications running on a particular client is typically copied according to location-specific criteria, such as from a certain location such as a specific folder or subfolder, according to a specified data path, etc. A module installed on the client or elsewhere in the system may supervise the transfer of data from the client to another location in a primary or secondary storage. Similar data transfers associated with location-specific criteria are performed when restoring data from secondary storage to primary storage. For example, to restore data, a user or system process must specify a particular secondary storage device, piece of media, archive file, etc. Thus, the precision with which conventional storage management systems perform storage operations on electronic data is generally limited by the ability to define or specify storage operations based on data location rather than information relating to or describing the data itself.

[0016] Moreover, when identifying data objects, such as files associated with performing storage operations, conventional storage systems often scan the file system of a client or other computing device to determine which data objects on the client should be associated with the storage operation. This may involve collecting file and/or folder attributes by traversing the file system of the client prior to

performing storage operations. This process is typically time-consuming and uses significant client resources that might be more desirably spent performing other tasks associated with production applications. There is thus a need for systems and methods for performing more precise and efficient storage operations.

BRIEF DESCRIPTION OF THE DRAWINGS

[0017] 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:

[0018] Figure 1 is a flow chart in accordance with an embodiment of the present invention;

[0019] Figure 2 is a system constructed in accordance with an embodiment of the present invention;

[0020] Figure 3a is a system constructed in accordance with an embodiment of the present invention;

[0021] Figure 3b is a flow chart in accordance with an embodiment of the present invention;

[0022] Figure 4 is a flow chart in accordance with an embodiment of the present invention;

[0023] Figure 5 is a system constructed in accordance with an embodiment of the present invention;

[0024] Figure 6 is a flow chart in accordance with an embodiment of the present invention;

[0025] Figure 7 is a system constructed in accordance with an embodiment of the present invention;

[0026] Figure 8 is a flow chart in accordance with an embodiment of the present invention;

[0027] Figure 9 is a system constructed in accordance with an embodiment of the present invention;

[0028] Figure 10 is a flow chart in accordance with an embodiment of the present invention;

[0029] Figure 11 is a flow chart in accordance with an embodiment of the present invention;

[0030] Figure 11a is a system constructed in accordance with an embodiment of the present invention;

[0031] Figure 12 is a flow chart in accordance with an embodiment of the present invention;

[0032] Figure 13 is a system constructed in accordance with an embodiment of the present invention;

[0033] Figure 14 is a flow chart in accordance with an embodiment of the present invention;

[0034] Figure 15 is a system constructed in accordance with an embodiment of the present invention;

[0035] Figure 16 is a data arrangement in accordance with an embodiment of the present invention;

[0036] Figure 17 is a flow chart in accordance with an embodiment of the present invention;

[0037] Figure 18 is a system constructed in accordance with an embodiment of the present invention;

[0038] Figure 19 is a flow chart in accordance with an embodiment of the present invention; and

[0039] Figure 20 is a flow chart in accordance with an embodiment of the present invention.

COPYRIGHT NOTICE

[0040] A portion of the disclosure of this patent document contains material that is subject to copyright protection. The copyright owner has no objection to the facsimile reproduction by anyone of the patent document or the patent disclosures,

as it appears in the Patent and Trademark Office patent files or records, but otherwise reserves all copyright rights whatsoever.

DETAILED DESCRIPTION

[0041] Aspects of the present invention are generally concerned with systems and methods that analyze and classify various forms of data that, among other things, facilitates identification, searching, storage and retrieval of data that satisfies certain criteria. Although described in connection with certain specific embodiments, it will be understood that the inventions disclosed herein have broad-based applicability to any wireless or hard-wired network or data transfer system that stores and conveys data from one point to another, including communication networks, enterprise networks, storage networks, and the like.

[0042] Aspects of the present invention provide systems and methods for data classification to facilitate and improve data management within an enterprise. The disclosed systems and methods evaluate and define data management operations based on data characteristics rather than data location, among other things. Also provided are methods for generating a data structure of metadata that describes system data and storage operations. This data structure may be consulted to determine changes in system data rather than scanning the data files themselves.

[0043] Generally, the systems and methods described in detail below are for analyzing data and other information in a computer network (sometimes referred to herein as a "data object") and creating a database or index of information which may describe certain pertinent aspects of the data objects that allow a user or system process to consult the database to obtain information regarding the network data. For example, a data collection agent may traverse a network file system and obtain certain characteristics and other attributes of data in the system. In some embodiments, such a database may be a collection of metadata and/or other information regarding the network data and may be referred to herein as a "metabase." Generally, metadata refers to data or information about data, and may include, for example, data relating to storage operations or storage management, such as data locations, storage management components associated with data, storage devices used in performing storage operations, index data, data application type, or other data.

[0044] With this arrangement, if it is desired to obtain information regarding network data, a system administrator or system process may simply consult the metabase for such information rather than iteratively access and analyze each data item in the network. Thus, this significantly reduces the amount of time required to obtain data object information by substantially eliminating the need to obtain information from the source data, and furthermore minimizes the involvement of network resources in this process, substantially reducing the processing burden on the host system.

[0045] Various embodiments of the invention will now be described. The following description provides specific details for a thorough understanding and enabling description of these embodiments. One skilled in the art will understand, however, that the invention may be practiced without many of these details. Additionally, some well-known structures or functions may not be shown or described in detail, so as to avoid unnecessarily obscuring the relevant description of the various embodiments.

[0046] The terminology used in the description presented below is intended to be interpreted in its broadest reasonable manner, even though it is being used in conjunction with a detailed description of certain specific embodiments of the invention. Certain terms may even be emphasized below; however, any terminology intended to be interpreted in any restricted manner will be overtly and specifically defined as such in this Detailed Description section.

[0047] A flow chart 100 illustrating some of the steps associated with one embodiment of the present invention is shown in Figure 1. In order to perform some of the functions described herein, it may be necessary at the outset to install certain data classification software or data classification agents on computing devices within at least parts of the network (step 102). This may be done, for example, by installing classification software on client computers and/or servers within a given network. In some embodiments, classification agents may be installed globally on a computing device or with respect to certain subsystems on a computing device. The classification software may monitor certain information regarding data objects generated by the computers and classify this information for use as further described herein.

[0048] Next, at step 104, a monitor agent may be initialized. Such a monitoring agent may be resident or installed on each computing device similar to the deployment of classification agents described above and may be configured to monitor and record certain data interactions within each machine or process. For example, the monitor agent may include a filter driver program and may be deployed on an input/output port or data stack and operate in conjunction with a file management program to record interactions with computing device data. This may involve creating a data structure such as a record or journal of each interaction. The records may be stored in a journal data structure and may chronicle data interactions on an interaction by interaction basis. The journal may include information regarding the type of interaction that has occurred along with certain relevant properties of the data involved in the interaction. One example of such a monitor program may include Microsoft's Change Journal or similar program.

[0049] Prior to populating a metabase with metadata, the portions of the network or subject system may be quiesced such that no data interactions are permitted prior to completing an optional scan of system files as described in conjunction with step 106 below. This may be done in order to obtain an accurate point in time picture of the data being scanned and to maintain referential integrity within the system. For example, if the system were not quiesced, data interactions would continue and be allowed to pass through to mass storage and thus data would change. However, in some embodiments, the subject system may be allowed to continue to operate, with operations or instructions queued in a cache. These operations are typically performed after the scan is complete so that any such data interactions occur based on the cached operations are captured by the monitor agent.

[0050] Generally, the file scanning of step 106 may be performed by a data classification agent and may include traversing the file system of a client to identify data objects or other files, email or other information currently stored or present in the system and obtain certain information regarding the information such as any available metadata. Such metadata may include information about data objects or characteristics associated with data objects such as the data owner (e.g., the client or user that generates the data or other data manager), the last modified time (e.g., the time of the most recent modification), the data size (e.g., number of bytes of

data), information about the data content (e.g., the application that generated the data, the user that generated the data, etc.), to/from information for email (e.g., an email sender, recipient or individual or group on an email distribution list), creation date (e.g., the date on which the data object was created), file type (e.g., format or application type), last accessed time (e.g., the time the data object was most recently accessed or viewed), application type (e.g., the application which generated the data object), location/network (e.g., a current, past or future location of the data object and network pathways to/from the data object), frequency of change (e.g., a period in which the data object is modified), business unit (e.g., a group or department that generates, manages or is otherwise associated with the data object), and aging information (e.g., a schedule, which may include a time period, in which the data object is migrated to secondary or long term storage), etc. The information obtained in the scanning process may be used to initially populate the metabase of information regarding network data at step 108.

[0051] After the metabase has been populated, the network or subject system may be released from the quiesced state and normal operation may resume. Next, at step 110, the monitor agent may monitor system operations to record changes to system data in the change journal database as described above. The change journal database may include a database of metadata or data changes and may comprise log files of the data or metadata changes. In some embodiments, the data classification agent may periodically consult the change journal database for new entries. If new entries exist, these entries may be examined, and if deemed relevant, then analyzed, parsed, and written to the metabase as an update (step 112). In other embodiments, change journal entries may be supplied substantially in parallel to the journal database and data classification agent. This allows the metabase to maintain substantially current information regarding the state of system data at any given point in time.

[0052] As mentioned above, one benefit of such a metabase is it significantly reduces the amount of time required to obtain information by substantially eliminating the need to obtain information directly from the source data. For example, assume a system administrator desires to identify data objects that a certain user has interacted with that contain certain content or other characteristics. Rather than search each file in each directory, a very time consuming process, the

administrator may simply search the metabase to identify such data objects and any properties associated with those objects, (e.g., metadata, which may include location, size, etc.), resulting in significant time savings.

[0053] Moreover, use of the metabase for satisfying data queries also minimizes the involvement of network resources in this process, substantially reducing any processing burden on the host system. For example, as described above, if an administrator desires to identify certain data objects, querying the metabase rather than the file system virtually removes the host system from the query process (i.e., no brute force scanning of directories and files required), allowing host computing devices to continue performing host tasks rather than be occupied with search tasks.

[0054] Figure 2 shows one embodiment of a client 200 constructed in accordance with principles of the present invention. As shown, client 200 may include a classification agent 202 and a monitor agent 206, which, in some embodiments, may be combined as an update agent 204, and which may be a single module encompassing the functionality of both agents. Client 200 may also include an internal or external data store 209, metabase 210, and change record 212.

[0055] Generally, client 200 may be a computing device, or any portion of a computing device that generates electronic data. Data store 209 generally represents application data such as production volume data used by client 200. Metabase 210, which may be internal or external to client 200 may contain information generated by classification agent 202 as described above. Similarly, change journal 212, which also may be internal or external to client 200, may contain information generated by monitor agent 206 as described above.

[0056] In operation, data interactions occurring within client 200 may be monitored with update agent 204 or monitor agent 206. Any relevant interaction may be recorded and written to change record 206. Data classification agent 202 may scan or receive entries from monitor agent 206 and update metabase 210 accordingly. Moreover, in the case where update agent 204 is present, monitored data interactions may be processed in parallel with updates to change record 212

and written to data store 208 and metabase 210 occurring accordingly. A file system 207 may be used to conduct or process data from the client to a data store 209.

[0057] Figure 3a provides a more detailed view of the journaling and classification mechanisms of client 200 generally shown in Figure 2. As shown, system 300 may include a memory 302, an update agent 304 which may include a separate or integrated monitor agent 306, classification agents 312a and 312b, a content agent 315, a monitor program index 310, metabase 314 and mass storage device 318.

[0058] In operation, data interactions that occur between memory 302 and mass storage device 318 may be monitored by monitor agent 306. In some embodiments, memory 302 may include random access memory (RAM) or other memory device employed by a computer in client 200 in performing data processing tasks. Certain information from memory 302 may be periodically read or written to mass storage device 318 which may include a magnetic or optical disk drive such as a hard drive or other storage device known in the art. Such data interactions are monitored by monitoring agent 306 which, in some embodiments, may include any suitable monitoring or journaling agent as further described herein.

[0059] As shown, system 300 may also include an administrative file system program 316, such as a file system program known in the art, which may include operating system programs, a FAT, an NTFS, etc. that may be used to manage data movement to and from mass storage device 318. Thus, in operation, data may be written from memory 302 to mass storage device 318 via file system program 316. Such an operation may occur, for example, to access data used to service an application running on a computing device. During this operation, monitor agent 306 may capture this interaction and generate a record indicating that an interaction has occurred and store the record in index 310. The data may be stored in mass storage 318 under the supervision of file system manager 316.

[0060] As shown in Figure 3a, monitor agent 306 may analyze data interactions such as interactions between memory 302 and mass storage 318 via the file system manager 316, and record any such interactions in monitor index 310. Thus, as described above, monitor index 310 may represent a list of data interactions wherein each entry represents a change that has occurred to client data along with certain

information regarding the interaction. In embodiments where Microsoft Change Journal or other similar software is used, such entries may include a unique identifier such as an update sequence number (USN), certain change journal reason codes identifying information associated with a reason(s) for the change made, along with data or metadata describing the data and certain data properties, data copy types, etc.

[0061] Thus, in operation, as data moves from memory 302 to mass storage 318 (or vice versa), monitor agent 304 may create and write an entry to index 310 which may in turn, be analyzed and classified by classification agent 312b for entry in metabase 314. In some embodiments, classification agent 312a may be coupled with mass storage device (either directly or through file system manager 316) and write metadata entries to both metabase 314 and mass storage device 318. In some embodiments, the metabase information may be stored on mass storage device 318. Moreover, in an alternate embodiment, classification agent 312b may periodically copy or backup metabase 314 to the storage device under the direction of a storage manager and/or pursuant to a storage policy (not shown) such that the information in metabase 314 may be quickly restored if lost, deleted or is otherwise unavailable.

[0062] In some embodiments, optional classification agent 312a may operate in conjunction with monitor agent 306 such that data moving to mass storage device 318 is classified as further described herein and written to device 318. With this arrangement, the data, along with the processed metadata describing that data, is written to mass storage device 318. This may occur, for example in embodiments in which monitor agent 306 and classification agent 312a are combined into update agent 304. Writing metadata in this way allows it to be recovered or accessed from mass storage device 318 if necessary, for example, when metabase 314 is missing certain information, busy, or otherwise inaccessible.

[0063] Content agent 315 may be generally used to obtain or filter data relating to content of the data moving from memory 302 to mass storage 318. For example, content agent 315 may read data payload information and generate metadata based on the operation for storage in metabase 314 and may include a pointer to the data item in mass storage 318. The pointer information may optionally be stored in an index. This metadata may also be stored with the data item in mass storage 318 or

as an entry functioning in place of or in addition to metabase 314. Storing metadata relating to data content in metabase 314 provides the capability to perform content searches for data in the metabase 314, instead of searching entries in mass storage 318. This allows the system to quickly locate information satisfying a content query that may be retrieved from mass storage 318, if necessary.

[0064] Moreover, such content metadata may be generated and used in locating data based on content features throughout a hierarchy within a storage operation system (e.g., content metadata may be generated and stored at each or certain levels of storage within the system (primary. secondary, tertiary, etc.) to facilitate location and retrieval of data based on content). As will be understood by one of skill in the art, the functionality provided by the content agent 315, classification agent 312a & b and monitor agent 306 may be provided by one or more modules or components such that the modules may be integrated into a single module providing the described functions, or may be implemented in one more separate modules each providing some of the functions.

[0065] Figure 3b is a flow chart 350 illustrating some of the steps that may be involved in the journaling process described above. At step 355, the monitor program may be initialized, which may include instantiating a data structure or index for recording interaction entries, and the assignment of a unique journal ID number which allows the system to differentiate between various journaling data structures that may be operating within the system. As mentioned above, the monitor program may include a filter driver or other application that monitors data operations (step 360). During the monitoring process, the monitor agent may observe data interactions between memory and mass storage to determine that certain data interactions have occurred. Information relating to the interactions may be captured and used to populate the metabase. In some instances, interaction types or certain aspects of interactions are captured. Such types or aspects may be defined in an interaction definition, which may be a change journal reason codes as used by Microsoft's Change Journal program, or be defined by a user or network administrator to capture some or all data interactions in order to meet user needs. For example, certain change definitions may record every data interaction that occurs regardless of whether any data actually changes or not. Such information

may be useful, for example, to determine users or processes that have "touched scanned or otherwise accessed data without actually changing it.

[0066] Thus, it is possible to employ interaction definitions that may capture a relatively broad or narrow set of operations, allowing a user to customize the monitor program to meet certain goals. Such interaction definitions may define or describe data movement, changes, manipulations or other operations or interactions that may be of interest to a system user or administrator (e.g., any operation that "touches" data may be recording along with the action or operation that caused the interaction (e.g. read, write, copy, parse, etc.) Moreover, change definitions may evolve over time or may be dynamic based on the entries sent to the index. For example, if expected results are not obtained, change definitions may be modified or additional definitions used until appropriate or desired results are obtained. This may be accomplished, for example by globally linking certain libraries of interaction definitions and selectively enabling libraries on a rolling basis until acceptable results are achieved. This process may be performed after the initial activation of the monitor agent and periodically thereafter, depending on changing needs or objectives.

[0067] Moreover, in some embodiments, the system may support the use of "user tags" that allow certain types of information to be tagged so they may be identified and tracked throughout the system. For example, a user may designate a particular type of data or information such as project information, or information shared between or accessed by particular group of users to be tracked across the system or through various levels of storage. This may be accomplished through a user interface (not shown) that allows a user to define certain information to be tagged, for example, by using any available attribute within the system such as those specified above with respect to the classification agent or filter used in the system. In some embodiments, the user may define one or more tags using these or other attributes which may be further refined by combining them through the use of logical or Boolean operators to a define a certain tag expression.

[0068] For example, a user may define a certain tag by specifying one or more criteria to be satisfied such as certain system users, a certain data permission level, a certain project, etc. These criteria may be defined using a logical operators such as an AND or OR operators to conditionally combine various attributes to create a

condition that defines a tag. All information satisfying those criteria may be tagged and tracked within the system. For example, as data passes through the monitor agent 306 (or other module within update agent 304), the data satisfying these criteria may be identified and tagged with a header or a flag or other identifying information as is known in the art. This information may be copied or otherwise noted by metabase 314 and mass storage 318 so that the information may be quickly identified. For example, the metabase may contain entries keeping track of all entries satisfying the tag criteria along with information relating to the types of operations performed on the information as well as certain metadata relating to the data content and its location in mass storage 318. This allows the system to search the metabase at a particular level of storage for the information, and quickly locate it within mass storage device for potential retrieval.

[0069] Next, a step 365, the monitor agent may continue to monitor data interactions based on change definitions until an interaction satisfying a definition occurs. Thus, a system according to one embodiment of the present invention may continue to monitor data interactions at steps 360 and 365 until a defined interaction, such as an interaction satisfying or corresponding to a selection criterion, such as an interaction definition, etc., occurs. If a defined interaction does occur, the monitor agent may create a record, which may be stored in a monitor program index, and in some embodiments, assign an interaction code that describes the interaction observed on the data object. Next, a step 370, the monitor program may identify a data object identifier associated with the data and that is associated with the data interaction, such as a file reference number (FRN) related to the data object. The FRN may include certain information such as the location or path of the associated data object. Any additional information (e.g., data properties, copy properties, storage policy information, etc.) associated with the FRN may also be obtained in order to enrich or enhance the record. In some embodiments, this may further involve obtaining information from other system files including master file tables (MFTs) to further enhance the metabase entries. Additional processing or formatting of the metabase entries may also occur in accordance with certain defined classification paradigms in order to populate the metabase with optimal or preferred information.

[0070] Next, at step 375 the record may be assigned record identifier such as a unique update sequence number (USN) that may be used to uniquely identify the entry within the index, and, in some embodiments, act as an index memory location. Thus a particular record may be quickly located with a data structure based on the USN. Next at step 380, the information described above may be concatenated or otherwise combined with other data or metadata data obtained by the monitor agent and arranged in an appropriate or expected format to create the record that may be written to the monitor index.

[0071] In alternate embodiments, the information described above may be written to the index and arranged at the index into an expected format or may be written to the record "as received" and include metadata or other information, such as a header describing that information such that adherence to a strict data format is not required. For example, some records may contain more or less information than other records, as appropriate. After the record has been constructed and deemed complete, the record may be "closed" by the system at step 385 and the system may then assign another USN to the next detected change. However, if the record is determined to be incomplete, the monitor agent or update agent may request any missing information to complete the entry. If such information is not received, the monitor agent may set a flag within the record to indicate it contains incomplete information and the record may then be closed.

[0072] Figure 4 is a flow chart 400 illustrating some of the steps that may be involved in a data classification process. At step 410, the classification agent may be initialized, which may include activating, reserving and/or clearing certain buffers and/or linking to libraries associated with deployment of the classification agent. Prior to scanning the interaction records generated by the monitor agent as described above, the classification agent may classify existing stored data by, for example, traversing the file and directory structure of an object system to initially populate the metabase as described herein.

[0073] Next, at step 420, during normal operation, the classification agent may scan the entries in the interaction journal to determine whether new entries exist since any previous classification processing was performed, for example, by determining whether the most recent entry currently in the journal is more or less recent than the last journal entry analyzed. This may be accomplished in several

ways. One method may include scanning a time or date information associated with the last journal entry examined and comparing it to the most recent time or date information than the entry currently present in the journal. If it is determined that the most recent journal entry occurred after a previous classification process, this process may be performed iteratively by "walking backwards" through the journal entries until the last entry previously analyzed by the classification agent is found. All entries with time information after that point may be considered new or unprocessed by the classification agent (step 440). If the last entry analyzed has the same time stamp as the most recent entry in the journal, no new entries exist and the system may return to step 420 to continue monitoring, etc.

[0074] Another method of identifying new journal entries may include comparing record identifiers such as USN numbers assigned to each journal entry (step 430). Journal entries with a larger USN number than the last entry previously analyzed may be considered new or unprocessed. If the last entry analyzed has the same USN number as the current entry, no new entries exist and the system may return to step 420 to continue monitoring, etc. This comparison may be performed until new entries are located (step 440) or until it is determined that no new entries exist.

[0075] In other embodiments, rather than scanning the journal data structure for new entries, any entries created by the journal agent may be automatically sent to the classification agent and the identification process may be largely unnecessary (except for the case where such scanning is necessary or desirable, for example, to repopulate the journal or verify certain information, etc.).

[0076] Next, at step 450, assuming new journal entries are identified, the system may determine if a metabase record already exists for the data object associated with those entries. This may be accomplished by comparing data object identifiers, such as FRNs of metabase entries with data object identifiers such as FRNs of journal entries. Matching these and other unique data characteristics may be used to match or correlate metabase and journal entries.

[0077] If no corresponding metabase record is identified, a new record may be created at step 460. This may involve creating a new metabase entry ID, analyzing the journal entry and parsing the entry into a predetermined format, and copying certain portions of the parsed data to the new metabase entry (steps 460 and 470),

as further described herein. Any additional metadata or file system information may also be associated with the new entry to enhance its content, including information from an FRN or information derived from an interaction code present in the journal entry, file system, e.g., MFT, etc. (step 480).

[0078] On the other hand, if a corresponding metadata entry is identified, the new journal entry may be processed as described above and may overwrite some or all of the corresponding entry. Such an updated pre-existing entry may receive an updated time stamp to indicate a current revision. However, in some embodiments, even if a corresponding entry is located, a new entry may be created and written to the metabase and optionally associated with the existing record. In such a system, the older related record may be maintained, for example, archival, recreation, historical or diagnostic purposes, and in some embodiments, may be marked or indicated as outdated or otherwise superseded. Such corresponding entries may be linked to one another via a pointer or other mechanism such that entries relating to the history of a particular data object may be quickly obtained.

[0079] Next, at step 490 the system may process any additional new journal entries detected by returning to step 450, where those additional new entries may be processed as described above. If no new entries are detected, the system may return to step 420 to perform additional scans on the journal data structure and continue monitoring.

[0080] Figure 5 illustrates an embodiment of the present invention in which a secondary processor performs some or all of functions associated with the data classification process described herein, including certain search functions. As shown, system 500 may include a manager module 505 which may include an index 510, a first computing device 515, (which may include a first processor 520, a journal agent 530, and a data classification agent 535), and a second computing device 540 which may include a second processor 545 and a data classification agent 535. System 500 may also include data store 550, a metabase 555 and change journal 560.

[0081] Computing devices 515 and 544 may be any suitable computing device as described herein and may include clients, servers or other network computers running software, such as applications or programs that create, transfer, and store

electronic data. In some embodiments, metabase 555 and journal 560 may be physically located within computing device 515, e.g., stored on local mass storage. In other embodiments the metabase 555 and journal 560 may be external to computing device 515 (or distributed between the two). In yet other embodiments, metabase 555 is accessible via a network and journal 560 is a local device.

[0082] In operation, computing device 515 may operate in a substantially similar manner as system 300 shown in Figure 3 with second processor 545 in second computing device 540 performing certain functions. For example, as shown, data classification agent 535 and journaling agent 530 may operate substantially as described in connection with Figure 3, i.e., journaling agent monitors data interactions on computing device 515 and records the interactions in journal 535 and classification agent processes journal entries and populates metabase 555, etc.

[0083] However, certain of the functions may be initiated or performed in whole or in part by second processor 545. Computing operations associated with journal agent 530 and/or classification agent 535 may run on or be directed by second processor 545 and may also utilize support resources located on or associated with computing device 540 such that the resources on computing device 515 are substantially unimpacted by these operations. This may serve to offload certain non-critical tasks from the host system (515) and have them performed by a secondary computing device 545.

[0084] For example, in some embodiments, the processing burden associated with some or all of the following tasks normally performed by first computing device 515 may be performed by processor 545 and associated resources in second computing device 540: (1) the initial scan of client files by the classification agent 535 and population of metabase 555, (2) the ongoing monitoring of data interactions of computing device (e.g., 515) and generation of interaction records for storage in journal 560, (3) processing and classification of journal information for updating metabase 555; and (4) searching or otherwise analyzing or accessing metabase 555 and/or journal 560 for certain information. However, in some embodiments it may be preferred to assign the secondary computing device the certain tasks such as those associated with searching metabase 555, while other tasks such as updating the journal and metabase may be performed by the primary computing device.

[0085] Performing such operations using a secondary or other processor may be desirable, for example, when the primary processor (e.g., processor 520) is unavailable, over utilized, unavailable or otherwise heavily used, or when it is otherwise desired to remove the primary processor and other primary system resources from performing certain tasks such as the ones described above. For example, in the case where it is desired to search or access metabase 555 for certain information, it may be preferable to avoid using processor 520 for this task so it remains free to perform other tasks associated with programs operating on computing device 515 (e.g., when computing device 515 is busy performing other network or application -related functions).

[0086] In some embodiments, the secondary processor may be located on computing device 515 (e.g., processor 525) and may perform the operations described herein in connection with processor 545. Moreover, some embodiments may include a manager module 505 which may coordinate overall operations between the various computing devices. For example, manager module 505 may monitor or otherwise be cognizant of the processing load on each computing device and may assign processing tasks based on availability (e.g., load balance). For example, if processor 520 is idle or operating at a low capacity, a request to search metabase 555 may be handled by processor 520. However, if processor 520 is busy or scheduled to perform or is performing priority work, manager 505 may assign the task to processor 545. Manager 505 may act as a general arbiter for such processor assignments to ensure system 500 is making efficient use of system resources.

[0087] Figure 6 is flow chart 600 illustrating some of the steps involved in performing a query on a metabase in a multiple processor system similar to the system shown in Figure 5. At step 610, a query may be received by the system for certain information. This request may be processed and analyzed by a manager module or other system process that determines or otherwise identifies which metabase or metabases within the system likely include at least some of the requested information, step 630. For example, the query itself may suggest which metabases to search and/or the management module may consult an index that contains information regarding metabase content within the system as further described herein. It will be understood that the identification process may require

searching and identifying multiple computing devices within an enterprise or network that may contain information satisfying search criteria.

[0088] In other embodiments, search requests may be automatically referred to a secondary processor to minimize processing demands on the computing device that may have created or is otherwise associated with the identified metabase(s). In some embodiments, it is preferable that the computing device that created or is otherwise associated with the identified metabase(s) not be involved in processing search operations as further described herein. Thus, the secondary computing device may consult with a manager or index associated with other computing devices to identify metabases with responsive information.

[0089] Next at step 640, the secondary processor may search metabases to identify appropriate data set that may potentially have information related to the query. This may involve performing iterative searches that examine results generated by previous searches and subsequently searching additional, previously unidentified metabases to find responsive information that may not have been found during the initial search. Thus the initial metabase search may serve as a starting point for searching tasks that may be expanded based on returned or collected results. Next, at step 650, the returned results may be optionally analyzed for relevance, arranged, and placed in a format suitable for subsequent use (e.g., with another application), or suitable for viewing by a user and reported (step 650).

[0090] Figure 7 presents a system 700 constructed in accordance with the principles of the present invention employing a centralized metabase 760 that may serve multiple computing devices 715-725. For example, as shown, system 700 may include computing devices 715-725, each of which may include a journaling agent (730-740 respectively), a classification agent (745-755 respectively), and centralized metabase 760, and in some embodiments, a manager module 705 with an index 710.

[0091] In operation, system 700 may operate substantially similarly to system 300 shown in Figure 3 with each computing device 715-725 storing classification entries in centralized metabase 760 rather than each computing device having its own dedicated metabase. For example, as shown, data classification agents 745-755 may operate substantially as described herein and communicate results to

centralized metabase 760. That is, analyze and process entries within the respective journals associated with journaling agents 730-740, and report results to metabase 760. With this arrangement, the classification agent may provide each metabase entry with an ID tag or other indicia that identifies which computing device 715-725 the entry originated from to facilitate future searches and efficiently designate entry ownership, or other associations between entries and computing devices.

[0092] Moreover, each entry to metabase 760 may be assigned a unique identifier for management purposes. As mentioned above, this number may represent the index location or offset of the entry within centralized metabase 760. In some embodiments, entries may be communicated to metabase 760 from the computing devices 715-725 on a rolling basis and may be arranged and formatted for storage by the metabase 760. For example, metabase 760 may receive multiple entries at substantially the same point in time from multiple computing devices 715-725 and may be responsible for queuing and arranging such entries for storage within the metabase 760.

[0093] In some embodiments, system 700 may include manager module 705 that may be responsible for assigning or removing associations between certain computing devices 715-725 and a particular centralized metabase 760. For example, in accordance with certain system preferences defined in index 710, manager 705 may direct certain computing devices 715-725 to write classification entries to a particular centralized metabase 760. Information indicating an association of the metabase 760 and the computing devices 715-725 may be stored in the index 710. This allows system 700 to reassign resources (globally or locally) to optimize system performance without the need to change device pointers or code associated with each computing device 715-725 that may be affected by a particular reallocation. For example, manager 705 may reassign certain computing devices 715-725 to another metabase by changing a destination address in an appropriate index.

[0094] Figure 8 is flow chart 800 illustrating some of the steps involved in using a centralized metabase with multiple computing devices similar to the one shown in Figure 7. At step 810, a centralized metabase may be instantiated by a manager module or in accordance with certain system management or provisioning policies.

This may involve securing certain processing, storage, and management resources for performing the task, loading certain routines into various memory buffers and informing the management module that the metabase is ready for operation.

[0095] Next, at step 820, the management module may review system resources, management policies, operating trends, and other information, for example, to identify computing devices to associate with the instantiated centralized metabase. This may further involve identifying pathways to the metabase from the various computing devices, locating operational policies governing the computing devices and, creating certain logical associations between the centralized metabases and the identified computing devices. These associations, once created, may be stored in an index or database for system management purposes.

[0096] After the metabase has been instantiated and associated with computing devices, classification agents within each associated computing device may scan existing files or data on the computing devices or clients (step 825) and populate the centralized metabase as further described herein (step 830). During the scanning process, a computing device identifier or other indicia may be appended or otherwise associated with the entry prior to transmission to the metabase such that each entry in the metabase can be tracked to its associated source computing device (step 840). Next, the centralized metabase may be populated with entries (step 850) and may communicate with the management module to establish and monitor a list of computing devices serviced by the centralized metabase and return to step 830. At this point, the system continues to monitor the associated computing devices for data interactions, which may be reported to the centralized metabase on an ongoing, periodic, or rolling basis.

[0097] In certain circumstances, the centralized metabase may need to assimilate or otherwise integrate existing entries with new entries reported by the computing devices. For example, the centralized metabase may become disconnected or unavailable for a period of time and subsequently be required to integrate a large number of queued entries. In this case, the metabase or management module may examine existing metabase entries as described herein and communicate with computing devices to identify: (1) the amount of time the object computer and the metabases have been disconnected, (2) the number of queued entries at the computing devices that need to be processed (for example,

entries cached once the centralized metabase was inaccessible for write operations), (3) whether there are any duplicative entries, and (4) which entries need to be integrated and in what order of preference (assuming multiple computing devices contain queued entries).

[0098] Based on these criteria, the management module or centralized metabase may assimilate the relevant entries into the metabase in the appropriate order until the backlog is eliminated and the system returns to normal operation. If it is determined during this process that certain information is lost to cache overflow, accidental deletion, corruption, or other reasons, the metabase and/or manager module may indicate such a discontinuity with the metadata or index associated with the centralized storage device or management module. In this case, clients, computing devices or other data sources may be rescanned to replace or repair the faulty entries. In other embodiments, the points of discontinuity may be noted and interpolation or other data healing techniques may be employed to provide derived information for the unknown points based on known information.

[0099] Figure 9 presents a system 900 constructed in accordance with the principles of the present invention including a computing device that interacts with a network attached storage device (NAS). As shown, system 900 may include a management module 905 and index 910, computing devices 915-925, each of which may include a journaling agent (945-955 respectively), a classification agent (930-940 respectively), data stores 960 and 965, and metabases 970-980. System 900 may also include NAS device 995 which may include NAS storage device 990 and NAS file system manager 985. Moreover, computing device 925 may be configured to operate as a NAS proxy device supervising the transfer of data to and from NAS device 995.

[00100] In operation, system 900 may operate substantially similar to system 300 shown in Figure 3a with exception of the NAS portion shown on the right-hand side. For example, as shown, data classification agents 930-940 may operate substantially as described herein and communicate results to their respective metabases 970-980. That is, analyze and process entries within the respective journals associated with journaling agents 945-955, and report results to metabases 970-980 which may be supervised in whole or in part by management module 905.

[00101] Data from computing device 925 may be journaled and classified using methods similar to those described herein. For example, journaling agent 955 may reside on computing device 925 and track each or certain data interactions between NAS device 995 and external applications. The location of the journaling agent 955 may be external to the NAS device 995 due, at least in part, to its proprietary nature (i.e., a closed system) and the difficulty associated with attempting to run other programs on the NAS device 995 itself.

[00102] The NAS portion 995 of system 900 may operate somewhat differently. For example computing device 925 may operate as a NAS proxy for moving data files to and from NAS device 995 using a specialized protocol such as the Network Data Management Protocol (NDMP) that is an open network protocol designed to perform data backups over heterogeneous networks. NDMP may be used to enhance performance by transferring data over a network by separating data and control paths, while maintaining centralized backup administration.

[00103] Journaling agent 955 may record any interactions between NAS data and external applications and record those interactions in computing device 925 as described herein. In some embodiments, such a journaling agent may include specialized routines for interpreting and processing data in NAS format. Data classification agent 940 may analyze journal entries and populate metabase 980 initially and periodically as further described herein.

[00104] Once initially populated, it may be desired to search the metabases of system 900 for certain information. This is discussed in more detail below in connection with the flow chart 1100 of Figure 11. In some embodiments, this may be handled by manager 905 or other system process which may initially evaluate any search request and consult index 910 or other information stores to determine which metabases within the system are likely to include responsive information. The results of this evaluation may be provided to the computing device handling the search request and may be in the form of pointers or other indicia or identifiers identifying a metabase such as a metabase ID. This may allow the computing device posing the search request to contact and search the identified metadata directly. In other embodiments manager 905 may process the request and provide substantially complete results to the computing device that submitted the query.

[00105] Figure 10 is flow chart 1000 illustrating some of the steps that may be involved in using the NAS system similar to or the same as the one shown of Figure 9. At step 1010 a copy operation may be initiated that directs data from computing device to a NAS device. This may involve identifying certain data to moved, for example, based on a data management or storage policy. Other factors that may also be considered may include data size, the last time the data was moved to the NAS device, the file owner, application type, etc.

[00106] It will be understood that in some embodiments it may be preferred to use computing device 925 as a NAS proxy that routes data from other network computing devices (not shown) to NAS device 995 with the computing device 925 supervising the data movement using certain specialized transfer programs to assist in the effort (step 1020). As the data is routed though computing device 925, journaling agent 955 may monitor interactions with NAS device 995 and create interaction entries for an interaction journal (step 1030). This may be accomplished by consulting with NAS file manager 985 and identifying which files in NAS 995 that have been involved in a data interaction as further described herein (step 1040). Next, journal entries may be created or updated to reflect data interactions currently detected as previously described herein (step 1050). The interaction journal may then be scanned to analyze the journal records (step 1060) and perform the classification process as further described herein to create metabase entries (step 1070). At this point metabase entries may be assigned an identifier and used to populate metabase 980 (step 1080).

[00107] As mentioned above, under certain circumstances, it may be desired to search a system that includes multiple metabases for certain information such as system 900 shown in Figure 9 whether or not NAS included. Figure 11 includes a flow chart 1100 illustrating some of the steps that may be performed in searching a multiple metabase system in accordance with certain aspects of the present invention.

[00108] Assume, for example, a user wants to locate and copy all data relating to a certain specified criteria such as data relating to a specific marketing project created and edited by a specific group of users over a certain period of time. First, the requestor may formulate such a request through a user interface (not shown) using techniques known in the art and submit the request to the system for

processing. This may also be accomplished by an automated computerized process, for example, when the system is performing certain management functions. Next the system may receive and analyze this query (step 1110). In some embodiments, this may be performed by a computing device configured to support the user interface. In other embodiments, the computing device may simply pass the request to the system where a management module or other system process computing device may perform the analysis. The analysis may include determining characteristics of data in the metabase that may satisfy the selected criteria.

[00109] Once the search request or query has been analyzed or parsed, the system may identify all metabases likely to contain records related to relevant data objects based on a query. This may be accomplished by using information obtained from analyzing or parsing the request as well as consulting with a management module that may have a substantially global view of metabases within the system that includes index information or a general overview of the information the metabases contain. After a set of metabases have been identified, the management module or other computing device may perform the search to identify a data set satisfying a query as further described herein and return a set of results (step 1130). At step 1140 the results may optionally be normalized. If normalization is not required, the results may be reported at step 1150. If normalization is desired, the system may analyze the results for both content and completeness. If, based on the returned results, other unsearched metabases are implicated as potentially having information that satisfies the search criteria, those metabases may be searched as well. This process may continue in an iterative fashion until a substantially complete set of results is obtained. Even if no additional metabases are implicated, these results may then be optionally normalized by performing certain functions such as locating and removing duplicative results, identifying network pathways to data objects identified in the search, and formatting or arranging the results for further processing (whether another computing process or for a user). For example, the returned results may be used to locate and retrieve the responsive data objects that may include information located on primary or secondary storage devices within the system or for other purposes as further described herein.

[00110] In some embodiments, the systems and methods of the present invention may be employed to identify and track some or all data interactions on a

user or group basis. For example, a system administrator or user may wish to record and keep track of all data interactions involving some or all system groups or users. This may include, for example, read and write operations performed on the user's or group's behalf, information and applications used or accessed, viewed web pages, electronic gaming interactions, chat, instant messages, and other communication interactions, multimedia usage, and other Internet or network based electronic interactions as known in the art. Thus, the system identifies, captures, classifies, and may otherwise track user and group interactions with electronic data creating a data store or other repository of these interactions and metadata associated with these interactions. In some embodiments, this repository may serve as a "digital or electronic life record" that effectively chronicles and catalogues some or all user or group interactions with electronic information and data during a given time period as further described herein.

[00111] For example, FIG 11a illustrates a system constructed in accordance with the principles of the present invention that identifies, captures, classifies, and otherwise tracks user and group interactions with electronic data. As shown, the system may generally include computing device 1162, one or more classification agents 1164, one or more journaling agents 1165, metabase 1166, change record 1167, and database 1168.

[00112] In operation computing device 1162 may be coupled to or interact with various other applications, networks, and electronic information such as, for example multimedia applications 1170, instant messaging/chat applications 1172, network applications 1174 such as an enterprise WAN or LAN, Internet 1176, and gaming applications 1178. It will be understood, however, that these are only examples and that any other network, application, or type of electronic information suitable for the purposes described herein may be added if desired.

[00113] Journaling agents 1165 and classification agents 1164 may operate in conjunction with one another to detect and record data interactions as further described herein. For example, each type of electronic data interaction (e.g. – instant messaging, web surfing, Internet search activities, electronic gaming, multimedia usage, etc.) may be identified, captured, classified, and otherwise tracked by a different journaling agent 1165 and classification agent 1164, for example an interaction-specific journaling agent 1165 or classification agent 1164

dedicated to processing a single type of interaction with electronic data. Thus, the system may have a first journaling agent 1165 and a first classification agent 1164 monitoring network traffic on a given network interface (not shown) directed to interactions associated with Internet usage, and a second journaling agent 1165 and a second classification agent 1164 monitoring a different system resource directed to interactions associated with electronic gaming (e.g. – recording and classifying gaming interactions such as recording games played, opponents played, win/loss records, etc.) or directed to interactions associated with use of an Internet browser to "surf" web (e.g. – tracking pages visited, content, use patterns, etc.) In some embodiments, journaling agent 1165 and classification agent 1164 may function as a single module capable of performing some or all functions associated with journaling agent 1165 and a classification agent 1164.

[00114] Thus, as a user or group interacts with various types of electronic information, some or all of those interactions may be captured and recorded in database 1168. Change record 1167 and metabase 1166 may record certain aspects of the interactions as further described herein and may represent an interaction by interaction log of the user's computing activities.

[00115] For example, in operation, a user of computing device 1162 may interact with certain applications such as multimedia application 1170 and instant messaging application 1172. This may include sending, receiving, viewing and responding to various audio/video files in any suitable format and may include instant, text or email messages. Journaling agent 1165 may detect the interactions between these applications and computing device 1162 and classification agent 1164 may classify and record information (e.g., metadata) associated with these interactions in metabase 1166 as further described herein.

[00116] Moreover, in some embodiments, some or all the content being exchanged or otherwise associated with these interactions may be captured and stored in database 1168 or other storage locations in the system. This may include capturing screen shots or summaries of information exchanges during data interactions. For example, the system may download all content associated with web pages viewed thus being able to recreate the original page content and interaction without access to the original or source version of the page on the Internet or other network. This may be advantageous, for example, if a user wishes

to interact with content associated with a previous interaction when that content is no longer available, as is common with web pages and other network resources over time. As another example, the system may also capture or otherwise store data associated with other interactions, for example chat transcripts, video game replays, search queries, search results, and associated search content, songs accessed, movies accessed, stored songs and movies, in addition to metadata, etc.

[00117] Moreover, in some embodiments, specialized classifications agents may be employed for some or all of the applications that a user or administrator desires to track and record. For example, the multimedia and instant messaging applications described above may each have a dedicated classification agent that analyzes journal records to create entries for metabase 1166. Further still, each classification agent may have its own associated metabase and or repository for source data (not shown), so application histories and content may be quickly indexed and searched. In other embodiments, however, a "universal" classification agent may be used that recognizes the application type (e.g., based on the journaling agent entries) and process interactions accordingly (which may include routing metadata to one or more specialized metabases).

[00118] As shown in Figure 11a, computing device 1162 may also interact with various network applications 1174 such as LAN or WAN applications. These may include interaction with certain distributed programs such as Microsoft Word or Outlook. Users may also interact with Internet 1176 and download various web pages and other information. In accordance with an aspect of the present invention, interactions with these networks/applications may also be journaled as described above with certain information regarding these interactions stored in metabase 1166. Portions of exchanged content may also be stored in database 1166. For example, Word documents, emails, web pages, web addresses and HTML content may be captured and stored on database 1168 such that it substantially represents a record of all user interactions with computing device 1162, or other system devices. For example, user interactions may be recorded with respect to any identified user based on identifiers and tracked at any network computing device.

[00119] Thus, if desired a user may retrieve captured data and review or replay certain data exchanges or save such records for future reference. For example, a

user may store all instant messaging interactions for replay or transmission to another. In some instances, it may be desirable to not record certain interactions, such as personal or private information. In some embodiments, this may be accomplished by "disabling" the appropriate classification agent for a certain period of time, etc.

[00120] Likewise, interactions with gaming applications (network or stand alone) may also be recorded stored with appropriate information stored in database 1168 and metabase 1166. Thus, a user may have the ability to retrieve, replay and transmit certain saved gaming sequences to third parties.

[00121] In some embodiments, database 1168 may become large and thus some information stored thereon may be moved to single instance storage from database 1168 with a pointer placed in the logical address of the instanced information (not shown). This may be performed as a memory saving measure as at least some of the entries in database 1168 are likely to be duplicative.

[00122] Some of the steps associated with the method generally described above are illustrated in chart 1200 of Figure 12 and may include the following. At the outset, a group or user of interest may be identified based on certain user related information or other network characteristics (step 1210). Such characteristics may include Active Directory privileges, network login, machine ID, or certain biometrics associated with a user or group member. These characteristics may be combined together or associated with one another to create a user or group profile. Such profiles may be stored in a database or index within a management module of the system and may be used as classification definitions within the system. When it is desired to identify or classify data items associated with a particular interaction, the system may compare certain attributes of the data involved in a detected interaction and associate that interaction with a particular group or user based on profile information (step 1220).

[00123] Such associations may be stored in a metabase created to keep track of user or group interactions. Thus, in one embodiment, the metabase essentially represents a list of all data interaction for a particular group or user. If desired, a list or copy of all the data items touched (e.g., interacted with) by a group or user may be quickly obtained.

[00124] In operation, the system may, through the use of a journaling agent or the like, monitor data interactions for a particular computing device as described herein. The interactions may be analyzed by a classification agent as described herein and associated with one or more profiles (step 1230). The association may be recorded in an identified metabase(s) that keeps track of a user's or group's interactions (step 1240) which may include references to the data object(s) identified, the attributes compared, and the basis for the association, etc. As discussed herein, the journaling agent may continue to monitor data interactions throughout operation, so that each metabase is updated and continues to accurately represent the data touched by a particular group or user. The identified metabases are associated with an identified group or user (step 1250), such as by storing an indication of the association in an index.

[00125] Figure 13 presents a system 1300 constructed in accordance with the principles of the present invention for communicating metadata and/or data objects between two or more computing devices. As shown, system 1300 may generally include first and second computing devices 1310 and 1320, respectively, associated data stores 1330 and 1340, and metabases 1350 and 1360. Computing devices in system 1300 may store data objects and metadata in their respective metabases and data stores as further described herein. In certain situations, however, it may be desired to transfer certain metadata between metabases 1350 and 1360 and certain data objects between data stores 1330 and 1340. This may be desirable for example, to move certain data from one computing device to another, to recreate a certain application at another location, or to copy or backup certain data objects and associated metadata.

[00126] Figure 14 presents a flow chart 1400 illustrating some of the steps associated with moving data between the computing devices described above. First, at step 1410, data objects and/or associated metadata may be identified for movement from one computing device to another. This may be accomplished by forming a query for certain data, such as a search for data that may be to be moved or copied pursuant to a data management or storage policy, or in response to a request to move data relating to certain processes or applications from one computing device to another, for any other suitable purpose such as disaster recovery, resource reallocation or reorganization, load balancing, etc.

[00127] At step 1420, the query may be analyzed and a first data store associated with a first computer may be searched for data objects satisfying the search criteria. Data objects identified during this process may then be transferred to a second data store associated with a second computing device (step 1430). Metadata associated with the transferred data objects may also be identified in a first metabase associated with the first computing device and transferred to an appropriate second metabase associated with the second computing device (step 1440). Such a transfer may involve copying data objects and metadata from one data store and metabase to another, or in some embodiments, may involve migrating the data from its original location to a second location and leaving a pointer or other reference to the second location so the moved information may be quickly located from information present at the original location.

[00128] Figure 15 illustrates a one arrangement of resources in a computing network in accordance with the principles of the present invention. As shown, storage operation cell 1550 may generally include a storage manager 1501, a data agent 1595, a media agent 1505, a storage device 1515, and, in some embodiments, may include certain other components such as a client 1585, a data or information store 1590, database 1511, jobs agent 1520, an interface module 1525, and a management agent 1530. Such system and elements thereof are exemplary of a modular storage system such as the CommVault QiNetix system, and also the CommVault GALAXY backup system, available from CommVault Systems, Inc. of Oceanport, NJ, and further described in U.S. Patent Application Serial Number 09/610,738 which is incorporated herein by reference in its entirety.

[00129] A storage operation cell, such as cell 1550, may generally include combinations of hardware and software components associated with performing storage operations on electronic data. According to some embodiments of the invention, storage operations cell 50 may be related to backup cells and provide some or all of the functionality of backup cells as described in Application Serial No. 09/354,058, which is hereby incorporated by reference in its entirety. However, in certain embodiments, storage operation cells may also perform additional types of storage operations and other types of storage management functions that are not generally offered by backup cells.

[00130] In accordance with certain embodiments of the present invention, additional storage operations performed by storage operation cells may include creating, storing, retrieving, and migrating primary storage data (e.g., 1590) and secondary storage data (which may include, for example, snapshot copies, backup copies, HSM copies, archive copies, and other types of copies of electronic data) stored on storage devices 1515. In some embodiments, storage operation cells may also provide one or more integrated management consoles for users or system processes to interface with in order to perform certain storage operations on electronic data as further described herein. Such integrated management consoles may be displayed at a central control facility or several similar consoles distributed throughout multiple network locations to provide global or geographically specific network data storage information.

[00131] In some embodiments, storage operations may be performed according to various storage preferences, for example as expressed by a user preference or storage policy. A storage policy is generally a data structure or other information source that includes a set of preferences and other storage criteria associated with 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. Thus, a storage policy may indicate that certain data is to be stored in a specific storage device, retained for a specified period of time before being aged to another tier of secondary storage, copied to secondary storage using a specified number of streams, etc. A storage policy may be stored in a storage manager database 1511, to archive media as metadata for use in restore operations or other storage operations, or to other locations or components of the system.

[00132] A schedule policy may specify when to perform storage operations and how often and may also specify performing certain storage operations on sub-clients of data and how to treat those sub-clients. A sub-client may represent static or dynamic associations of portions of data of a volume and are typically mutually exclusive. Thus, a portion of data may be given a label and the association is stored as a static entity in an index, database or other storage location used by the system.

Sub-clients may also be used as an effective administrative scheme of organizing data according to data type, department within the enterprise, storage preferences, etc.

[00133] For example, an administrator may find it preferable to separate e-mail data from financial data using two different sub-clients having different storage preferences, retention criteria, etc. Storage operation cells may contain not only physical devices, but also may represent logical concepts, organizations, and hierarchies. For example, a first storage operation cell 1550 may be configured to perform a first type of storage operations such as HSM operations, which may include backup or other types of data migration, and may include a variety of physical components including a storage manager 1501 (or management agent 1530), a media agent 1505, a client component 1585, and other components as described herein. A second storage operation cell may contain the same or similar physical components, however, it may be configured to perform a second type of storage operations such as SRM operations, and may include as monitoring a primary data copy or performing other known SRM operations.

[00134] Thus, as can be seen from the above, although the first and second storage operation cells are logically distinct entities configured to perform different management functions (i.e., HSM and SRM respectively), each cell may contain the same or similar physical devices in both storage operation cells. Alternatively, in other embodiments, different storage operation cells may contain some of the same physical devices and not others. For example, a storage operation cell 1550 configured to perform SRM tasks may contain a media agent 1505, client 1585, or other network device connected to a primary storage volume, while a storage operation cell 1550 configured to perform HSM tasks may instead include a media agent 1505, client 1585, or other network device connected to a secondary storage volume and not contain the elements or components associated with and including the primary storage volume. These two cells, however, may each include a different storage manager 1501 that coordinates storage operations via the same media agents 1505 and storage devices 1515. This "overlapping" configuration allows storage resources to be accessed by more than one storage manager 1501 such that multiple paths exist to each storage device 1515 facilitating failover, load balancing and promoting robust data access via alternative routes.

[00135] Alternatively, in some embodiments, the same storage manager 1501 may control two or more cells 1550 (whether or not each storage cell 1550 has its own dedicated storage manager 100). Moreover, in certain embodiments, the extent or type of overlap may be user-defined (through a control console (not shown)) or may be automatically configured to optimize data storage and/or retrieval.

[00136] Data agent 1595 may be a software module or part of a software module that is generally responsible for copying, archiving, migrating, and recovering data from client computer 1585 stored in an information store 1590 or other memory location. Each client computer 1585 may have at least one data agent 1595 and the system can support multiple client computers 1585. In some embodiments, data agents 1595 may be distributed between client 1585 and storage manager 1501 (and any other intermediate components (not shown)) or may be deployed from a remote location or its functions approximated by a remote process that performs some or all of the functions of data agent 1595.

[00137] Embodiments of the present invention may employ multiple data agents 1595 each of which may backup, migrate, and recover data associated with a different application. For example, different individual data agents 1595 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. Other embodiments may employ one or more generic data agents 1595 that can handle and process multiple data types rather than using the specialized data agents described above.

[00138] If a client computer 1585 has two or more types of data, one data agent 1595 may be required for each data type to copy, archive, migrate, and restore the client computer 1585 data. For example, to backup, migrate, and restore all of the data on a Microsoft Exchange 2000 server, the client computer 1585 may use one Microsoft Exchange 2000 Mailbox data agent 1595 to backup the Exchange 2000 mailboxes, one Microsoft Exchange 2000 Database data agent 1595 to backup the Exchange 2000 databases, one Microsoft Exchange 2000 Public Folder data agent 1595 to backup the Exchange 2000 Public Folders, and one Microsoft Windows 2000 File System data agent 1595 to backup the client computer's 1585 file system. These data agents 1595 would be treated as four separate data agents 1595 by the system even though they reside on the same client computer 1585.

[00139] Alternatively, other embodiments may use one or more generic data agents 1595, each of which may be capable of handling two or more data types. For example, one generic data agent 1595 may be used to back up, migrate and restore Microsoft Exchange 2000 Mailbox data and Microsoft Exchange 2000 Database data while another generic data agent may handle Microsoft Exchange 2000 Public Folder data and Microsoft Windows 2000 File System data, etc.

[00140] Data agents 1595 may be responsible for arranging or packing data to be copied or migrated into a certain format such as an archive file which is discussed in more detail in connection with Figure 16 herein. Nonetheless, it will be understood this represents only one example and any suitable packing or containerization technique or transfer methodology may be used if desired. Such an archive file may include a list of files or data objects copied in metadata, the file and data objects themselves. Moreover, any data moved by the data agents may be tracked within the system by updating indexes associated appropriate storage managers or media agents.

[00141] Generally speaking, storage manager 1501 may be a software module or other application that coordinates and controls storage operations performed by storage operation cell 1550. Storage manager 1501 may communicate with some or all elements of storage operation cell 1550 including client computers 1585, data agents 1595, media agents 1505, and storage devices 1515, to initiate and manage system backups, migrations, and data recovery.

[00142] Storage manager 1501 may include a jobs agent 1520 that monitors the status of some or all storage operations previously performed, currently being performed, or scheduled to be performed by storage operation cell 1550. Jobs agent 1520 may be communicatively coupled with an interface agent 1525 (typically a software module or application). Interface agent 1525 may include information processing and display software, such as a graphical user interface ("GUI"), an application program interface ("API"), or other interactive interface through which users and system processes can retrieve information about the status of storage operations. Through interface 1525, users may optionally issue instructions to various storage operation cells 1550 regarding performance of the storage operations as described and contemplated by the present invention. For example, a user may modify a schedule concerning the number of pending snapshot copies or

other types of copies scheduled as needed to suit particular needs or requirements. As another example, a user may employ the GUI to view the status of pending storage operations in some or all of the storage operation cells in a given network or to monitor the status of certain components in a particular storage operation cell (e.g., the amount of storage capacity left in a particular storage device).

[00143] Storage manager 1501 may also include a management agent 1530 that is typically implemented as a software module or application program. In general, management agent 1530 provides an interface that allows various management components 1501 in other storage operation cells 1550 to communicate with one another. For example, assume a certain network configuration includes multiple cells 1550 adjacent to one another or otherwise logically related in a WAN or LAN configuration (not shown). With this arrangement, each cell 1550 may be connected to the other through each respective interface agent 1525. This allows each cell 1550 to send and receive certain pertinent information from other cells 1550 including status information, routing information, information regarding capacity and utilization, etc. These communication paths may also be used to convey information and instructions regarding storage operations.

[00144] For example, a management agent 1530 in first storage operation cell 1550 may communicate with a management agent 1530 in a second storage operation cell 1550 regarding the status of storage operations in the second storage operation cell. Another illustrative example includes the case where a management agent 1530 in first storage operation cell 1550 communicates with a management agent 1530 in a second storage operation cell to control the storage manager 1501 (and other components) of the second storage operation cell via the management agent 1530 contained in the storage manager 100.

[00145] Another illustrative example is the case where management agent 130 in the first storage operation cell 1550 communicates directly with and controls the components in the second storage management cell 1550 and bypasses the storage manager 1501 in the second storage management cell. If desired, storage operation cells 1550 can also be organized hierarchically such that hierarchically superior cells control or pass information to hierarchically subordinate cells or vice versa.

[00146] Storage manager 1501 may also maintain an index, a database, or other data structure 1511. The data stored in database 1511 may be used to indicate logical associations between components of the system, user preferences, management tasks, media containerization and data storage information or other useful data. For example, the storage manager 1501 may use data from database 1511 to track logical associations between media agent 1505 and storage devices 1515 (or movement of data as containerized from primary to secondary storage).

[00147] Generally speaking, a media agent, which may also be referred to as a secondary storage computing device, 1505 may be implemented as software module that conveys data, as directed by storage manager 1501, between a client computer 1585 and one or more storage devices 1515 such as a tape library, a magnetic media storage device, an optical media storage device, or any other suitable storage device. In one embodiment, secondary computing device 1505 may be communicatively coupled with and control a storage device 1515. A secondary computing device 1505 may be considered to be associated with a particular storage device 1515 if that secondary computing device 1505 is capable of routing and storing data to particular storage device 1515.

[00148] In operation, a secondary computing device 1505 associated with a particular storage device 1515 may instruct the storage device to use a robotic arm or other retrieval means to load or eject a certain storage media, and to subsequently archive, migrate, or restore data to or from that media. Secondary computing device 1505 may communicate with a storage device 1515 via a suitable communications path such as a SCSI or fiber channel communications link. In some embodiments, the storage device 1515 may be communicatively coupled to a data agent 105 via a Storage Area Network ("SAN").

[00149] Each secondary storage computing device 1505 may maintain a index, a database, or other data structure 1506 which may store index data generated during backup, migration, and restore and other storage operations as described herein. For example, performing storage operations on Microsoft Exchange data may generate index data. Such index data provides a secondary computing device 1505 or other external device with a fast and efficient mechanism for locating data stored or backed up. Thus, in some embodiments, a secondary storage computing device index 1506, or a storage manager database 1511, may store data associating a

client 1585 with a particular secondary computing device 1505 or storage device 1515, for example, as specified in a storage policy, while a database or other data structure in secondary computing device 1505 may indicate where specifically the client 1585 data is stored in storage device 1515, what specific files were stored, and other information associated with storage of client 1585 data. In some embodiments, such index data may be stored along with the data backed up in a storage device 1515, with an additional copy of the index data written to index cache in a secondary storage device. Thus the data is readily available for use in storage operations and other activities without having to be first retrieved from the storage device 1515.

[00150] Generally speaking, information stored in cache is typically recent information that reflects certain particulars about operations that have recently occurred. After a certain period of time, this information is sent to secondary storage and tracked. This information may need to be retrieved and uploaded back into a cache or other memory in a secondary computing device before data can be retrieved from storage device 1515. In some embodiments, the cached information may include information regarding format or containerization of archive or other files stored on storage device 1515.

[00151] In some embodiments, certain components may reside and execute on the same computer. For example, in some embodiments, a client computer 1585 such as a data agent 1595, or a storage manager 1501 coordinates and directs local archiving, migration, and retrieval application functions as further described in U.S. Patent Application Number 09/610,738. This client computer 1585 can function independently or together with other similar client computers 1585.

[00152] Moreover, as shown in Figure 15, clients 1585 and secondary computing devices 1505 may each have associated metabases (1525 and 1560, respectively). However in some embodiments each "tier" of storage, such as primary storage, secondary storage, tertiary storage, etc., may have multiple metabases or a centralized metabase, as described herein. For example, in Figure 15, rather than a separate metabase 1525 associated with each client 1585, the metabases on this storage tier may be centralized as discussed further herein. Similarly, second and other tiers of storage may have either centralized or distributed metabases. Moreover, mixed architectures systems may be used if desired, that may include a

first tier centralized metabase system coupled to with a second tier storage system having distributed metabases and vice versa, etc.

[00153] Moreover, in operation, a storage manager 1501 or other management module may keep track of certain information that allows the storage manager to select, designated or otherwise identify metabases to be searched in response to certain queries as further described herein. Movement of data between primary and secondary storage may also involvement movement of associated metadata and other tracking information as further described herein.

[00154] Figure 16 is a diagram illustrating one arrangement of data that may be used in constructing an archive file according to one aspect of the invention. As shown, archive file 1600 may include header section 1610, index section 1620 and payload section 1630. Such an archive file may be constructed by a data agent at a client computing device when migrating data, for example, from primary to secondary storage, primary storage to other primary storage, etc. The payload section 1610 may include the data objects that are to be moved from a first location to a second location within the system (e.g., primary to secondary storage). These data objects may be identified by a data agent and designated to be moved pursuant to a storage preference such as a storage policy, a user preference, etc. Header 1610 may include routing and path information that identifies the origin and destination of the payload data and may include other information such as a list of files copied, checksums, etc. Index section 1620 may include certain other information regarding the payload data objects such as size, file type, and any offset or other logical indexing information that may been tracked by a storage management component or other component previously managing the data objects in the payload.

[00155] In some embodiments, storage managers may index information regarding archive files and related payload by time and storage on certain media so the archive files can be quickly located and/or retrieved. For example, it may be desired to identify certain data based on a query. The query may be analyzed and a certain time frame of interest may be identified. The system may use this information as a basis for a query search of certain index information (e.g., only search for records concerning operations that occurred during a specific time). This

streamlines the search and retrieval process by narrowing the universe of data needs to be searched to locate responsive information.

[00156] Figure 17 presents a flow chart 1700 that illustrates some of the steps that may be performed in moving data from primary storage to other storage devices within the system. First, at step 1702, a query seeking certain data may be identified. The query may include aspects of data such as a schedule policy, storage policy, storage preference or other preference. The query may be analyzed and a primary metabase searched to identify data objects that satisfy the query (step 1704). This may include parsing the query into constituent parts and analyzing each part alone or in combination with other portions as part of the evaluation process. At step 1706, it may be determined, whether data objects satisfying the query are to be copied to other primary storage devices, to secondary storage devices or both (pursuant to a storage policy, etc.).

[00157] If at least some data objects satisfying the search criteria are to be copied to other primary storage devices, those data objects may be identified as further described herein and the target primary storage device(s) identified. This may involve consulting a storage policy or storage manager to determine the destination point. In some embodiments, destination maybe determined dynamically, such that it is selected based on certain system preferences or optimization routines that select a storage device based on storage capacity, availability, data paths to the destination, etc.

[00158] At step 1708 the identified data objects may be copied from primary storage of a first computing device (the source) to primary storage of a second computing device (the target or destination). Any metadata associated with the first computing device describing the copied data may also be copied to a metabase associated with the second computing device such that this description information is not abandoned or lost, but rather travels with the copied data for subsequent use (step 1710).

[00159] Next, at step 1712, it may be determined whether the copied data objects and associated metadata are to be deleted from the source computing device. For example, this may be done in order to free storage space on the source computer or in accordance with certain data aging or migration criteria. If it is

decided to delete the data objects (and associated metadata) the memory locations which include the data may be erased or designated for overwrite (step 1714 and 1716).

[00160] In some embodiments the data objects may be deleted but certain metadata may be retained. If it is decided not delete the data objects, the data is retained and an index in an associated storage manager may be updated (step 1718), for example by updating an index to reflect a new location, data object status, any changes, etc., and return to step 1702. In other embodiments, if data is deleted from the system, for example, a user permanently deletes certain data from an application, that associated data may also be deleted from both primary and secondary storage devices and associated metabases to free storage space within the system.

[00161] Returning to step 1706, it is also determined whether certain data objects currently stored in primary storage are to be migrated to one or more secondary storage devices. If so, an archive file similar to the one described in Figure 16 or other data structure suitable for transport may be constructed or created by the source computing device with identified data objects placed in the payload section and header and index information added (step 1722). Data may be moved from primary to secondary storage in predefined chunks which are constructed from such archive files, for example, using a data pipe, such as the data pipe described in Patent No. 6,418,478 titled PIPELINED HIGH SPEED DATA TRANSFER MECHANISM.

[00162] Next, at step 1724 one or more target secondary storage devices may be identified. This may involve consulting a storage policy or storage manager to determine the destination point. In some embodiments, destination maybe determined dynamically, such that it is selected based on certain system preferences or optimization routines that select a storage device based on storage capacity, availability, data paths to the destination, etc. Once the secondary storage device(s) are identified, the archive files may be routed to a media agent, storage manager, or other system component that supervises the transfer to the target secondary storage device (steps 1724 and 1728). This may involve selecting and appropriate data transfer route and ensuring the proper resources and are available (e.g., bandwidth) such that the data may be copied with a certain period of time.

Supervision may further include parsing a copy operation into several portions with each portion being transferred by certain media agent or other resources, etc, to meet system or transfer requirements (e.g., a time window).

[00163] Next, the appropriate media within the target storage device may be identified (step 1730) and the archive files may be transferred from the media management device to the secondary storage device (step 1732). Such media may be selected from available media already associated with a similar data transfer or may be selected and reserved from an available media pool or scratch pool within the storage device. During or after the transfer, a media agent index or storage manager index associated with the secondary storage device may be updated to reflect the transfer (step 1733). This may include copying the appropriate management files to the media management index such as offset, media ID file name or other management information.

[00164] At step 1734, any metadata stored in a first metabase associated with the transferred data objects may also be transferred and used to update a second metabase associated with the target secondary storage device. Such metadata may be copied from the first metabase to the second metabase using network transmission resources. In some embodiments, the metadata in the first metabase may be deleted after it is confirmed the metadata has been copied to the second metabase. In other embodiments, the metadata may remain in both first and second metabases.

[00165] At step 1736, it may be determined whether the data objects transferred from the primary storage device are to be deleted. If so, the data objects and associated metadata in a first metabase may be erased or otherwise designated for overwrite (steps 1738 and 1740). In some cases, a pointer or other reference such as a file stub may be left in the original data location

[00166] Figure 18 presents a generalized 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 in accordance with an embodiment of the present invention. It will be understood that although the storage operation cells generally depicted in Figure 18 have different reference numbers than the storage

operation cell 1550 shown in Figure 15, these cells may be configured the same as or similar to the storage cell 1550 as depicted in Figure 15.

[00167] As shown, the system illustrated in Figure 18 may include a master storage manager component 1835 and various other storage operations cells. As shown, the illustrative embodiment in Figure 18 includes a first storage operation cell 1840, a second storage operation cell 1845, a third storage operation cell 1850, a fourth storage operation cell 1855, and may be extended to include nth storage operation cell, if desired (not shown). However, it will be understood this illustration is only exemplary and that fewer or more storage operation cells may be present or interconnected differently if desired.

[00168] Storage operation cells, such as the ones shown in Figure 18 may be communicatively coupled and hierarchically organized. For example, a master storage manager 1835 may be associated with, communicate with, and direct storage operations for a first storage operation cell 1840, a second storage operation cell 1845, a third storage operation cell 1850, and fourth storage operation cell 1855. In some embodiments, the master storage manager 1835 may not be part of any particular storage operation cell. In other embodiments (not shown), master storage manager 1835 may itself be part of a certain storage operation cell. This logical organization provides a framework in which data objects, metadata and other management data may be hierarchically organized and associated with appropriate devices components (e.g., storage devices).

[00169] The storage operation cells may be configured in any suitable fashion, including those which involve distributed or centralized metabases. For example, storage operation cell 1840 may include a centralized primary storage metabase and a centralized secondary storage metabase, storage operation cell 1845 may include a centralized primary storage metabase and multiple secondary storage metabases, storage operation cell 1850 may include multiple primary storage metabases and a centralized secondary storage metabase, and storage operation cell 1855 may include multiple primary storage metabases and multiple secondary storage metabases (not shown). However, it will be understood that this is merely illustrative, and any other suitable configuration may be used if desired.

[00170] Thus, in operation, master storage manager 1835 may communicate with a management agent of the storage manager of the first storage operation cell 1840 (or directly with the other components of first cell 1840) with respect to storage operations performed in the first storage operation cell 1840. For example, in some embodiments, master storage manager 1835 may instruct the first storage operation cell 1840 with certain commands regarding a desired storage operation such as how and when to perform particular storage operations including the type of operation and the data on which to perform the operation.

[00171] Moreover, metabases associated with each storage operation cell may contain information relating to data and storage operations as described herein. In some embodiments, master storage manager 1835 may include a master metabase index or database (not shown) that reflects some or all of the metadata information from the hierarchically subordinate storage operation cells within the system. This allows the system to consult the master storage index or database for information relating to data within those storage operation cells rather than requiring each cell be contacted of polled directly for such information.

[00172] In other embodiments, master storage manager 1835 may track 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. Moreover, master storage manager 1835 may track 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, master storage manager 1835 may use methods to monitor network resources such as mapping network pathways and topologies to, among other things, physically monitor storage operations and suggest, for example, alternate routes for storing data as further described herein.

[00173] In some embodiments, master storage manager 1835 may store status information and other information regarding its associated storage operation cells and other system information in an index cache, database or other data structure accessible to manager 1835. A presentation interface included in certain embodiments of master storage manager 1835 may access this information and present it to users and system processes with information regarding the status of

storage operations, storage operation cells, system components, and other information of the system.

[00174] In some embodiments, master storage manager 1835 may store and/or track metadata and other information regarding its associated storage operation cells and other system information in an index cache, database or other data structure accessible to manager 1835. Thus, during a search procedure as further described herein, queries can be directed to a specific storage operation cell or cells based on the cell's function, past involvement, routing or other information maintained within the storage manager or other management component.

[00175] As mentioned above, storage operation cells may be organized hierarchically. With this configuration, storage operation cells may inherit properties from hierarchically superior storage operation cells or be controlled by other storage operation cells in the hierarchy (automatically or otherwise). Thus, in the embodiment shown in Fig. 18, storage operation cell 1845 may control or is otherwise hierarchically superior to storage operation cells 1850 and 1855. Similarly, storage operation cell 1850 may control storage operation cells 1855. Alternatively, in some embodiments, storage operation cells may inherit or otherwise be associated with storage policies, storage preferences, storage metrics, or other properties or characteristics according to their relative position in a hierarchy of storage operation cells.

[00176] Storage operation cells may also be organized hierarchically according to function, geography, architectural considerations, or other factors useful or desirable in performing storage operations. For example, in one embodiment, storage operation cell 1840 may be directed to create snapshot copies of primary copy data, storage operation cell 1845 may be directed to create backup copies of primary copy data or other data. Storage operation cell 1840 may represent a geographic segment of an enterprise, such as a Chicago office, and storage operation cell 1845 may represents a different geographic segment, such as a New York office. In this example, the second storage operation cells 1845, 1850 and 1855 may 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.

[00177] As another example, and as previously described herein, different storage operation cells directed to different functions may also contain the same or a subset of the same set of physical devices. Thus, one storage operation cell in accordance with the principles of the present invention may be configured to perform SRM operations and may contain the same, similar or a subset of the same physical devices as a cell configured to perform HSM or other types of storage operations. Each storage operation cell may, however, share the same parent or, alternatively, may be located on different branches of a storage operation cell hierarchy tree. For example, storage operation cell 1845 may be directed to SRM operations whereas storage operation cell 1855 may be directed to HSM operations. Those skilled in the art will recognize that a wide variety of such combinations and arrangements of storage operation cells are possible to address a broad range of different aspects of performing storage operations in a hierarchy of storage operation cells.

[00178] In some embodiments, hierarchical organization of storage operation cells facilitates, among other things, system security and other considerations. For example, in some embodiments, only authorized users may be allowed to access or control certain storage operation cells. For example, a network administrator for an enterprise may have access to many or all storage operation cells including master storage manager 1835. But a network administrator for only the New York office, according to a previous example, may only have access to storage operation cells 1845-1855, which form the New York office storage management system.

[00179] Moreover, queries performed by the system may be subject to similar restrictions. For example, depending on access privileges, users may be limited or otherwise excluded from searching a certain cell or cells. For example, a user may be limited to searching information in cells or metabases within the system that are unrestricted or to those which specific access rights have been granted. For example, certain users may not have privileges to all information within the system. Accordingly, in some embodiments, as a default setting, users may have access privileges to information in cells that they interact with. Thus, confidential and sensitive information may be selectively restricted except only to certain users with express privileges (e.g., financial or legal information, etc.). For example, certain classification information within the metabases in the system may be restricted and therefore accessed only by those with the proper privileges.

[00180] Other restrictions on search criteria may include the scope of the search. For example, in a large network with many storage cells may require dedicating significant amounts of resources to perform go global or comprehensive searches. Thus, if a certain resource threshold is exceeded by a proposed search, the system may prompt that search to be modified or otherwise cancelled.

[00181] In other embodiments master storage manager 1835 may alert a user such as a system administrator when a particular resource is unavailable or congested. For example, a particular storage device might be full or require additional media. For example, a master storage manager may use information from an HSM storage operation cell and an SRM storage operation cell to present indicia or otherwise alert a user or otherwise identify aspects of storage associated with the storage management system and hierarchy of storage operation cells.

[00182] 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, master storage manager 1835 (or another storage manager within the hierarchy of storage operation cells) may utilize the global data regarding its associated storage operation cells to suggest solutions to such problems when they occur (or act as a warning prior to occurrence). For example, master storage manager 1835 may alert the user that a storage device in a particular storage operation cell is full or otherwise congested, and then suggest, based on job and data storage information contained in its database, or associated metabase, or an alternate storage device. Other types of corrective actions based an such information may include suggesting an alternate data path to a particular storage device, or dividing data to be stored among various available storage devices as a load balancing measure or to otherwise optimize storage or retrieval time. In some embodiments, such suggestions or corrective actions may be performed automatically, if desired. This may include automatically monitoring the relative health or status of various storage operation cells and searching for information within the cells of the system relating to systems or resource performance within that cell (e.g., index, metabase, database, etc.) for use in diagnostics or for suggesting corrective action.

[00183] In alternate embodiments, HSM and SRM components may be aware of each other due to a common database or metabase of information that may include

normalized data from a plurality of cells. Therefore, in those embodiments there is no need for such information to pass through a master storage manager as these components may be able to communicate directly with one another. For example, storage operation cell 1845 may communicate directly with storage operation cell 1855 and vice versa. This may be accomplished through a direct communications link between the two or by passing data through intermediate cells.

[00184] Moreover, in some embodiments searches may be performed across a numerous storage cells within the hierarchy. For example, a query may be posed to master storage manager 1835 that may pass the query down through the hierarchy from cells 1840 to 1845 to 1850 and 1855, etc. This may be accomplished by passing the query form one manager component of each cell to another, or from one data classification agent to another, one metabase to another etc. The results may be passed upward through the hierarchy and compiled with other results such that master storage manager 1835 has a complete set of results to report. In other embodiments, each storage manager cell may report results directly to the requestor or to a designated location.

[00185] Figure 19 presents a flow chart 1900 that illustrates some of the steps that may be involved in performing searches for data objects across systems that include multiple primary and secondary storage devices. First, at step 1905, a query seeking certain data may be identified (e.g., from a storage policy, user preference, other process, etc.). The query may be analyzed to identify system components, such as clients potentially having information such as certain data objects or metadata that may satisfy the query (e.g., by excluding certain clients that are unlikely to have data being sought based on certain query parameters such as location, time frame, client or other component, department, application type, or any other criteria used to classify data as described herein, etc. (step 1910)). Results may be presented based on a confidence factor indicating the likelihood that the results meet the specified parameters. For example, results substantially satisfying most or all criteria may be listed first with the confidence factors provided based on a percentage of the criteria satisfied (e.g., a query that returned results having three out of four criteria satisfied may be represented with a 75% confidence factor etc.). Less relevant results may be listed subsequently with the confidence factor provided

based on any suitable relevant factor such as number of parameters satisfied, how close the match is, etc.

[00186] The search process may further involve consulting one or more indexes associated with the clients to identify where responsive data objects or other copies of client data, etc., may be located within the system. At step 1915, it may be determined whether client data objects satisfying the query are located in primary storage, secondary storage, or both (e.g., based on index information in a storage manager). This may be based on polling various storage managers or a master storage manager that includes information the covers or represents whole system or the portion of system specified for search.

[00187] If it is determined that responsive data objects are only located on client(s) in primary storage, that client may be added to the list of clients to be searched (step 1955). If it is determined that responsive data objects are located in secondary storage devices (or other primary storage locations that may be identified), the system may consult a storage manager index to identify archive files (or other files) based on certain query parameters such as a specified point in time, origination point, etc., or on index data stored in a storage manager index identifying archive files or other file associated with the data objects.

[00188] Next at step 1920, storage managers may be consulted to identify responsive archive files. At step 1925, media management components that may have handled responsive data objects are be identified. This may be based on information retrieved from the storage manager index regarding archive files, e.g., an association of archive files with media agents and media items. It may then be determined whether the identified media management components have metadata relating to the identified archive files available readily available in an index cache (step 1930).

[00189] This may be accomplished by searching for reference information relating to the identified archive files. If such information is already present in the cache, responsive data objects may be identified and retrieved using the index cache information, which may include, offsets and any file identifiers, etc., by the media management component, and the system may proceed to step 1940 (determine whether another media management component needs to be analyzed).

[00190] If not, the index information may need to be loaded from the secondary storage device so archive files may be retrieved and accessed. This may involve identifying the particular media on which the index data is stored and upload it to the media management component cache (step 1935). In some embodiments, a master storage manager or other component with information relating to files may be consulted to identify media containing the responsive information. These media may be mounted in drive or other input/ output device and examined to locate the proper files or data structures. Index information may then be located and uploaded to an index or database associated with the appropriate media management component (e.g., media agent). This allows the media management component to locate and retrieve specific data objects on the media that satisfy the search criteria.

[00191] Next, if no further media management components have been identified, a list of media management components to be searched may be compiled (step 1945). At step 1950, a list of clients identified as potentially having responsive data objects may also be compiled. After a complete list of secondary storages devices and clients potentially having responsive data objects is identified, the associated metabases are queried for these components, step 1960, and results are returned indicating data objects that may satisfy the search criteria, step 1965. In some embodiments, these results may be reviewed and analyzed to ensure relevance, with only reasonably relevant or responsive data objects actually being retrieved.

[00192] Figure 20 presents a flow chart 2000 that illustrates some of the steps that may be involved in retrieving data objects from secondary storage (or other tiers or other storage locations) in accordance with principles of the present invention. This may be accomplished generally as follows. Certain data (e.g. data objects or associated metadata) from the system may need to be retrieved. That data may be requested and communicated to the system in the form of a query. The query may be used to search the system and identify media on which responsive data may be located. Once located, data satisfying the selection criteria may be uploaded and retrieved and analyzed for relevance, or other action may be taken. Or, alternatively, the identified data may be moved to other tiers of storage. More specific steps involved in this process may be as follows.

[00193] First, at step 2002, a query seeking certain data may be identified. The query may be analyzed to ascertain certain additional information that may assist in

identifying responsive information such as identifying a certain point in time to search (step 2004). This may involve consulting storage manager and/or media agent index or database for responsive information relating to a certain point in time. This may also involve consulting certain metabases for similar information that may be associated with these or other media management components providing copy and management functions. Point in time information may be specified by the user or may be assigned by the system absent a specific time frame established by the user. For example, a user may specify a certain time range within the query (e.g., a time range, a certain date, all information related to a project since its inception etc.). The system however, may assign a certain time limit based on the query (e.g., such as based on the specifics of the query (e.g., only have data relating to a certain time frame)), and may limit the search to the time frame of information present in certain metabases, master storage manager, or index within the system, and/or poll or otherwise communicate with storage devices within the system to determine the range or time frame of available data within the system and present the user with options for retrieving it (e.g., some, all within a time frame, etc.)

[00194] Next, at step 2006 certain archive files may be identified and associated media agents (step 2008) that may have been involved in transferring responsive data objects. This may be determined by consulting a master storage manager or other media management component index or metabase to determine whether the archive files have been handled by such components. Once the appropriate media agents have been identified, it may be determined whether information regarding the identified archive files is present in a cache or index associated with the media agents (step 2010). If not, the index information may need to be uploaded so the appropriate archive files may be retrieved and accessed. This process may be performed until all identified media agents have the appropriate index information loaded and/or until it is determined that no responsive information has been handled by the media agents and therefore no index information need be uploaded.

[00195] Next, at step 2016 data objects satisfying the query criteria may be identified by searching metabases and/or indexes. In some embodiments, such data objects may be compiled into a list of data objects for present or subsequent retrieval. For example, such a list of responsive data objects may be provided to the

user (which may itself satisfy the query) and then provide the user with the option to actually retrieve all or certain selected identified data objects.

[00196] At step 2018, the new destination for the data objects may be determined. For example, if certain data objects are being migrated off as part of an ILM operation, the query or other information may indicate the intent or reason for the search and the data object's destination. This may be useful in determining whether certain data objects are responsive to search criteria or query. At step 220 it may be determined whether the new destination is primary storage (a restore operation) or secondary or other tier of storage (ILM). Such information may be further useful in determining whether the data objects are likely to fall within a time frame or category of interest and thus may be useful in further identifying data objects of interest.

[00197] If the identified data objects are moving to other secondary storage tiers, the data objects may be repackaged into form suitable for secondary storage, which may include repackaging into an archive file, converting to a new format, compressing of the data objects and associated files, encryption, or any other containerization technique known in the art (step 2022).

[00198] Once the data objects are in a suitable format, they may be copied to the appropriate storage destination by the system. This may be accomplished by a media agent or media component in conjunction with a storage manager or other media management component that coordinate routing and the specifics involved with file transfer (step 2024), as further described herein. Metadata relating to the copied data objects may then be copied to a metabase associated with a computing device at the destination (step 2026).

[00199] For example, metadata relating to the data being copied may be copied along with the data to the secondary storage device and may be copied to an index in the media agent or other media management component involved in the data transfer. This allows the media management component to locate and retrieve and otherwise manage the stored data. Such metadata may also useful when performing searches of secondary storage devices (or other tiers) as further described herein. Metadata stored along with the data on the secondary storage device may be useful to restore or refresh the media agent index in the case of lost

or corrupt data and also may be transferred along with the data on storage media in the case whether it is necessary to copy all such data (or actually physically relocate) to another storage device. A master storage manager index or metabase associated with destination computing device may be updated reflecting the arrival and new location of the transferred data objects and/or archive file for system management purposes (step 2034).

[00200] In some embodiments, the copied data objects and metadata may be deleted from the source location (steps 2028-2032). For example, at step 2028, it may be determined whether the copied data objects should be deleted based user preferences, storage policy requirements or other system constraints such has diminished storage capacity, etc. At steps 2030 and 2032 the data objects and records may be deleted. However, a stub, pointer or other referential element may be placed at the same logical location to act as a marker for the moved data. This allows subsequent operations to quickly track down and locate the moved data at its new location.

[00201] If, however, at step 2020, it is determined that the identified data objects are moving to primary storage, accordingly, the data objects may be reformatted (e.g., unpacked from archive file format) for copying to a computing device (step 2038). Next the unpacked data may be copied to a target computing device along with any associated metadata (steps 2040 and 2042). For example, this may involve reading metadata and/or index information from the archive file and repopulating the metabase and/or management component indexes with this information as further described herein. For example, metadata from the archive file may be retrieved and integrated into a metabase associated with the target computing device including information relating to data management and as well as certain content and storage information as further described herein with respect to the classification process and metabase population. Thus, such archive information may be fully restored to primary storage and any associated information, such as metabase information may be searched and retrieved accordingly.

[00202] Moreover, information relating to system management may be uploaded and used to repopulate storage management components within the system such as a storage manager or master storage manager reflecting the return of the retrieved data to primary storage (step 2050). For example, a storage manager index may be

updated to reflect the presence of the retrieved data along with certain management information such as logical offsets and location of the retrieved information such that the retrieved information may be located and accessed. Other management components, such as a master storage manager may also be updated with the appropriate identification and location information to reflect the return of the retrieved data within the system.

[00203] In certain embodiments, the copied data and metadata may be deleted from the source location (steps 2044-2048). For example, at step 2044, it may be determined whether the copied data objects in secondary storage should be deleted based user preferences, storage policy requirements or other system constraints such has diminished storage capacity, etc. At steps 2046 and 2048 the data objects and records may be deleted within the system including any metabase or other system management information associated with the retrieved data. Storage management components such as storage managers, media agents may also be updated to reflect the removal or deletion of such information (step 2050).

[00204] 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, personal digital assistants (PDAs), and other devices suitable for the purposes described herein. In other words, the software and other modules described herein may be executed by a general-purpose computer, e.g., a server computer, wireless device or personal computer. Those skilled in the relevant art will appreciate that aspects of the invention can be practiced with other communications, data processing, or computer system configurations, including: Internet appliances, hand-held devices (including PDAs), wearable computers, all manner of cellular or mobile phones, multi-processor systems, microprocessor-based or programmable consumer electronics, set-top boxes, network PCs, mini-computers, mainframe computers, and the like. Indeed, the terms "computer," "server," "host," "host system," and the like are generally used interchangeably herein, and refer to any of the above devices and systems, as well as any data processor. Furthermore, aspects of the invention can be embodied in a special purpose computer or data processor that is specifically

programmed, configured, or constructed to perform one or more of the computer-executable instructions explained in detail herein.

[00205] 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. Aspects of the invention can also be practiced in distributed computing environments where tasks or modules are performed by remote processing devices, which are linked through a communications network, such as a Local Area Network (LAN), Wide Area Network (WAN), or the Internet. In a distributed computing environment, program modules may be located in both local and remote memory storage devices. 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.

[00206] Aspects of the invention may be stored or distributed on computer-readable media, including magnetically or optically readable computer discs, hard-wired or preprogrammed chips (e.g., EEPROM semiconductor chips), nanotechnology memory, biological memory, or other data storage media. Indeed, computer implemented instructions, data structures, screen displays, and other data under aspects of the invention may be distributed over the Internet or over other networks (including wireless networks), on a propagated signal on a propagation medium (e.g., an electromagnetic wave(s), a sound wave, etc.) over a period of time, or they may be provided on any analog or digital network (packet switched, circuit switched, or other scheme).

[00207] Unless the context clearly requires otherwise, throughout the description and the claims, the words "comprise," "comprising," and the like are to be construed in an inclusive sense, as opposed to an exclusive or exhaustive sense; that is to say, in the sense of "including, but not limited to." As used herein, the terms "connected," "coupled," or any variant thereof, means any connection or coupling,

either direct or indirect, between two or more elements; the coupling of connection between the elements can be physical, logical, or a combination thereof. Additionally, the words "herein," "above," "below," and words of similar import, when used in this application, shall refer to this application as a whole and not to any particular portions of this application. Where the context permits, words in the above Detailed Description using the singular or plural number may also include the plural or singular number respectively. The word "or," in reference to a list of two or more items, covers all of the following interpretations of the word: any of the items in the list, all of the items in the list, and any combination of the items in the list.

[00208] The above detailed description of embodiments of the invention is not intended to be exhaustive or to limit the invention to the precise form disclosed above. While specific embodiments of, and examples for, the invention are described above for illustrative purposes, various equivalent modifications are possible within the scope of the invention, as those skilled in the relevant art will recognize. For example, while processes or blocks are presented in a given order, alternative embodiments may perform routines having steps, or employ systems having blocks, in a different order, and some processes or blocks may be deleted, moved, added, subdivided, combined, and/or modified to provide alternative or subcombinations. Each of these processes or blocks may be implemented in a variety of different ways. Also, while processes or blocks are at times shown as being performed in series, these processes or blocks may instead be performed in parallel, or may be performed at different times.

[00209] The teachings of the invention provided herein can be applied to other systems, not necessarily the system described above. The elements and acts of the various embodiments described above can be combined to provide further embodiments. Any patents and applications and other references noted above, including any that may be listed in accompanying filing papers, are incorporated herein by reference. Aspects of the invention can be modified, if necessary, to employ the systems, functions, and concepts of the various references described above to provide yet further embodiments of the invention.

[00210] These and other changes can be made to the invention in light of the above Detailed Description. While the above description describes certain embodiments of the invention, and describes the best mode contemplated, no

matter how detailed the above appears in text, the invention can be practiced in many ways. Details of the system and method for classifying and transferring information may vary considerably in its implementation details, while still being encompassed by the invention disclosed herein. As noted above, particular terminology used when describing certain features or aspects of the invention should not be taken to imply that the terminology is being redefined herein to be restricted to any specific characteristics, features, or aspects of the invention with which that terminology is associated. In general, the terms used in the following claims should not be construed to limit the invention to the specific embodiments disclosed in the specification, unless the above Detailed Description section explicitly defines such terms. Accordingly, the actual scope of the invention encompasses not only the disclosed embodiments, but also all equivalent ways of practicing or implementing the invention under the claims.

[00211] While certain aspects of the invention are presented below in certain claim forms, the inventors contemplate the various aspects of the invention in any number of claim forms. For example, while only one aspect of the invention is recited as embodied in a computer-readable medium, other aspects may likewise be embodied in a computer-readable medium. Accordingly, the inventors reserve the right to add additional claims after filing the application to pursue such additional claim forms for other aspects of the invention.

CLAIMS

I/We claim:

1. A method of classifying data generated within a computer system, wherein the computer system includes a data storage device, the method comprising:

intercepting the generated data, wherein the intercepted data is to be stored at the data storage device;

generating a log entry within a log, wherein the log contains entries associated with the intercepted data;

analyzing entries within the log to identify entries that satisfy at least one data management criterion;

storing the intercepted data generated within the computer system on the data storage device; and

adding metadata to a metabase, wherein the metadata includes information that describes the intercepted data associated with the identified log entries and refers to the data stored on the data storage device.

2. The method of claim 1 further comprising accessing the data storage device to retrieve additional information related to the intercepted data, and adding the retrieved additional information to the metabase.

3. The method of claim 1 wherein intercepting comprises trapping the generated data and wherein the data is generated by an application program.

4. The method of claim 1 wherein analyzing entries within the log comprises parsing the log entries into a predetermined format that includes a file reference number.

5. The method of claim 1 wherein the storage criterion identifies a particular user and wherein analyzing entries identifies entries related to intercepted

data that has been accessed by the identified user and classifies the intercepted data by characteristics associated with the user.

6. The method of claim 1 wherein the storage criterion identifies a department within an organization and wherein analyzing entries identifies entries related to data that has been accessed by a member of the identified department and classifies the data based on the identified department.

7. The method of claim 1 wherein the storage criterion identifies a pattern of access and wherein analyzing entries identifies entries related to data that has been accessed in accordance with the identified pattern and classifies the data based on the identified pattern.

8. The method of claim 1 wherein adding metadata comprises associating one or more predefined classifications with the data.

9. The method of claim 1 wherein adding metadata comprises associating one or more user-defined classifications with the data.

10. The method of claim 1 wherein adding metadata comprises associating one or more classifications with the data based on the similarity of the data to previously classified data and storing the metadata based on the classifications.

11. A system for classifying data stored within a computer system, wherein the computer system includes an application program configured to generate program data to be employed by the computer system, the system comprising:

a monitor agent configured to process at least part of the program data to create at least one log entry related to the program data;

a data collection agent configured to analyze the at least one log entry and select at least part of the program data associated with the at least one log entry;

a data classification agent configured to identify characteristics of and assign classifications to the selected program data, wherein a classification associates the selected program data with other program data having similar characteristics; and,

a metabase configured to store metadata information about the selected program data, wherein the metadata contains at least the assigned classifications of the selected program data, and wherein the metabase is configured to receive requests from the computer system to perform data management operations based on the assigned classifications.

12. The system of claim 11 wherein the monitor agent processes program data by capturing a request to store the program data.

13. The system of claim 11 wherein the monitor agent processes program data by traversing a file system containing the program data.

14. The system of claim 11 wherein the monitor agent is further configured to identify changes in the program data.

15. The system of claim 11 wherein the data classification agent identifies characteristics of the selected program data that are not stored with the selected program data.

16. The system of claim 11 wherein the data classification agent identifies characteristics including an aging criteria.

17. The system of claim 11 wherein the data classification agent identifies characteristics including a state of protection that identifies how recently an archive copy of the data has been made.

18. The system of claim 11 wherein the data classification agent identifies characteristics including a pattern of use.

19. The system of claim 11 wherein the data collection agent captures additional information about the selected program data as metadata describing the selected program data and wherein the metadata includes at least one of the following types of information: an owner, a last modified time, a size, an application that generated the selected program data, a user that generated the selected program data, header information, a creation date, a file type, a last accessed time, an application type, a location, a frequency of change, a business unit, usage trends associated with the selected program data, or aging information.

20. The system of claim 11 wherein the data collection agent captures additional information about the selected program data if changes are detected in the program data.

21. The system of claim 11 wherein the monitor agent is further configured to create a log entry when at least part of the program data changes, and the monitor agent is a filter driver.

22. The system of claim 11 wherein the monitor agent is further configured to create a log entry when at least part of the program data changes, and the monitor agent is Microsoft Change Journal.

23. The system of claim 11 wherein the monitor agent is further configured to create a log entry when at least part of the program data changes, and the monitor agent is a separate update agent.

24. The system of claim 11 wherein the monitor agent is further configured to create a log entry when at least part of the program data changes.

25. The system of claim 11 further comprising a change journal database configured to store log entries created by the monitor agent.

26. The system of claim 11 wherein the metabase is further configured to perform the data management operations without traversing a data store where the program data is stored.

27. A data storage system, comprising:

a processor for executing instructions;

an input/output component coupled to the processor, wherein the input/output component receives a set of data to be stored; and,

a data storage device coupled to the processor,

wherein the processor is configured to:

analyze a data object associated with the set of data,

determine one or more categorizations for the data object, wherein the categorizations identify one or more characteristics of the data object that are not stored with the data object itself,

facilitate storage of the set of data in the data storage device, and

perform a storage operation associated with the set of data with respect to the data storage device, based on a rule determined at least in part from the one or more categorizations for the data object, and not based solely on the location where the block of data is to be stored.

28. The system of claim 27 wherein the rule is predetermined.

29. The system of claim 27 wherein the rule is calculated based on the one or more categorizations.

30. A system for classifying data stored within a computing device, the system comprising:

a means for storing data;

a means for collecting stored data comprising a data collection agent configured to identify data objects within the data store and obtain information about identified data objects;

a means for classifying data comprising a data classification agent for assigning classifications to identified data objects, wherein the classifications are related to one or more applications that may use the data object; and,

a means for storing metadata comprising a metabase configured to store information about identified data objects and their classifications.

31. The system of claim 30 wherein the means for collecting stored data further comprises a means for journaling changes.

32. The system of claim 30 further comprising a means for administering the configuration of data movement to and from the means for storing data.

33. A computer-readable medium containing instructions for classifying data stored in a data storage system, by a method comprising:

receiving a file to be stored in a data storage device;

analyzing the file to determine characteristics associated with the file;

determining one or more of multiple categorizations for the file based on the determined characteristics, wherein the categorizations identify trends associated with the file that are related to trends identified about other files; and,

facilitating a storage operation on the file with respect to the data storage device, based at least in part on the one or more categorizations for the file.

34. The computer-readable medium of claim 33 wherein the system is part of a network and the processor is coupled to the data storage device through the network.

35. The computer-readable medium of claim 33 wherein the input/output device is a port for receiving the file from a network.

36. The computer-readable medium of claim 33 wherein the processor executes an agent specialized to manipulate data related to an application that is associated with the file.

37. A method of accessing data stored on a first computing device using a second computing device, comprising:

receiving data associated with a first computing device and classifying the received data to create metadata including one or more classifications, wherein the classifications contain information describing characteristics of the received data wherein at least one characteristic is not stored with the identified data;

storing the metadata at the second computing device based on the classifications;

receiving a data management request, wherein the data management request comprises a request to identify data and then perform a data management operation on the identified data;

performing a query of the metadata stored at the second computing device without accessing the first computing device, wherein the query identifies at least part of the received data based on the data management request; and

performing the data management operation on the identified data wherein the data management operation is performed to reduce an impact of the data management request on the first computing device.

38. The method of claim 37 wherein receiving data comprises determining if a copy of the data associated with the first computing device is accessible at a computing device other than the first computing device, and when the data is accessible at a computing device other than the first computing device, accessing the data at the computing device other than the first computing device.

39. The method of claim 37 wherein performing the data management operation comprises determining if a copy of the data associated with

the first computing device is accessible at a computing device other than the first computing device, and when the data is accessible at a computing device other than the first computing device, performing the data management operation on the data at the computing device other than the first computing device.

40. The method of claim 37 wherein performing the data management operation comprises identifying characteristics of the received data including a purpose for the received data indicated by a user that created the received data.

41. The method of claim 37 wherein classifying comprises identifying characteristics of the received data including characteristics for finding the received data in response to a legal discovery request.

42. The method of claim 37 wherein classifying comprises identifying characteristics of the received data including characteristics related to a purpose for retrieving the received data.

43. The method of claim 37 wherein the data is application-specific data, and wherein the metadata includes information identifying an application associated with the data.

44. The method of claim 37, further comprising:

accessing other data stored at other computing devices;

classifying the accessed other data to create other metadata;

storing the other metadata at the second computing device wherein metadata that classifies data stored at multiple computing devices is centrally accessible at the second computing device; and

performing a storage operation based on the other metadata stored at the second computing device.

45. The method of claim 37 wherein receiving data comprises intercepting an application request to store data at the first computing device.

46. The method of claim 37 wherein receiving data associated with a first computing device comprises determining if data stored at the first computing device is available at an additional computing device, and when it is determined that the data is available at the additional computing device, accessing the data at the additional computing device without accessing the first computing device.

47. The method of claim 37 wherein the metadata stored at the second computing device contains a reference to the data stored at the first computing device.

48. A system to perform a data management operation on data stored at a first computing device using a second computing device, comprising:

a data discovery component configured to discover data stored on the first computing device;

a data classification component configured to classify the discovered data to create metadata including one or more classifications that identify characteristics of the discovered data;

a metadata storing component configured to store the metadata on the second computing device; and

a data management operation component configured to perform a data management operation that first identifies data and then performs an action on the identified data wherein the data management operation component reduces a number of accesses of the first computing device by using the classifications stored within the metadata of the second computing device to identify data related to the data management operation without accessing the first computing device.

49. The system of claim 48 wherein the data management operation component performs the data management operation at a computing device other than the first computing device.

50. The system of claim 48 wherein the data discovery component discovers data by traversing a file system of the first computing device.

51. The system of claim 48 wherein the data discovery component discovers data by intercepting requests to store data at the first computing device.

52. A computer-readable media encoded with instructions for accessing data stored at a first data processing system using a second data processing system, by a method comprising:

identifying data stored on the first data processing system;

categorizing the identified data to create descriptive data including one or more categorizations that specify information about the identified data for finding the data in response to a data management request;

storing the descriptive data on the second data processing system; and

analyzing the descriptive data at the second data processing system to find data in response to a data management request, wherein an analysis regarding the data at the first data processing system is performed without accessing the data at the first data processing system.

53. The computer-readable media of claim 52 wherein the data is application-specific data, and wherein the metadata includes information identifying an application associated with the data.

54. A first data storage system coupled to a second data storage system, comprising:

a processor for executing instructions;

an input/output device coupled to the processor, wherein the input/output device receives a block of data; and,

a data storage device coupled to the processor,

wherein the processor is configured to:

analyze the block of data,

determine one or more of multiple categorizations for the block of data wherein the categorizations identify properties associated with the block of data that are not stored with the block of data, and

facilitate performance of a storage operation with respect to the block of data based on the one or more categorizations for the block of data, and not necessarily based on a storage location of the block of data.

55. The first data storage system of claim 54 wherein the processor facilitates performance of a storage operation by offloading a portion of the storage operation that identifies data related to the storage operation to the second data storage system, wherein the categorizations are accessed by the second data storage system to identify data stored at the first data storage system related to the storage operation without accessing the first data storage system.

56. The first data storage system of claim 54 wherein the first and second data storage systems are part of a network, and wherein the processor is coupled to the data storage device through the network.

57. The first data storage system of claim 54 wherein the input/output device is a port for receiving the block of data from a network.

58. The first data storage system of claim 54 wherein the processor executes an agent specialized for an application that is associated with the block of data and the block of data is a data file.

59. A system for distributing metadata related to data stored in a network, comprising:

a first storage device containing data accessible via the network;

a second storage device containing data accessible via the network;

a first metabase containing metadata including classifications of data stored on at least one of the first and second storage devices; and

a second metabase containing metadata including classifications of data stored on at least one of the first and second storage devices,

wherein the first and second metabases are accessed to make determinations about the data stored on the first and second storage devices, and the classifications included within the metadata of the first and second metabases includes information for facilitating the performance of a storage management operation on at least one of the first and second storage devices.

60. The system of claim 59 wherein the first metabase contains metadata classifying the data stored on the first storage device and the second metabase contains metadata classifying the data stored on the second storage device.

61. The system of claim 59 wherein both the first and second metabases contain copies of metadata classifying the data stored on the first and second storage devices.

62. The system of claim 59 further comprising a metabase manager component for directing storage requests to a metabase, and wherein the metabase manager component directs requests based at least in part on a past level of activity associated with each metabase tracked by the metabase manager component.

63. The system of claim 59 further comprising a metabase manager component for directing storage requests to a metabase, and wherein the metabase manager component directs requests based at least in part on a current level of activity associated with each metabase monitored by the metabase manager component.

64. The system of claim 59 further comprising a metabase manager component for directing storage requests to a metabase, and wherein the metabase manager component directs requests based at least in part on a projected level of activity associated with each metabase determined by the metabase manager component.

65. The system of claim 59 further comprising a metabase manager component for directing storage requests to a metabase, and wherein the metabase manager component directs requests based on the data to be accessed by the storage request and identifying a metabase associated with the data to be accessed.

66. The system of claim 59 further comprising a metabase manager component for selecting a metabase to which to direct a storage query, and wherein the metabase manager component selects a metabase and if the metabase does not contain data that satisfies the query, the metabase manager component selects another metabase.

67. The system of claim 59 further comprising a metabase manager component for selecting a metabase to which to direct a storage request, and wherein the metabase manager component selects the metabase based on access control information and determining whether the storage request identifies an entity with access to the metabase.

68. The system of claim 59 wherein the first and second metabases are arranged hierarchically based on geography.

69. The system of claim 59 wherein the first and second metabases are arranged hierarchically based on a hierarchy of departments within a business organization that stores data at the first and second storage devices.

70. The system of claim 59 wherein the first and second metabases are arranged based on a relationship between subsidiaries of a business organization.

71. A method of accessing data through a distributed metabase, comprising:

receiving a request to access data;

selecting at least one of multiple distributed metabases containing metadata related to the data, wherein the metadata includes information not stored with the data to facilitate data management operations on the data;

sending the request to the at least one selected metabase; and

accessing the data based on the metadata contained by the at least one selected metabase, wherein accessing comprises offloading at least some accesses of a primary storage device storing the data to the metabase based on the metadata.

72. The method of claim 71 wherein selecting comprises selecting one of the multiple metabases based on a level of activity of each metabase, wherein the level is past, current, or projected.

73. The method of claim 71 wherein selecting comprises determining a content type of the data and selecting a metabase based on the content type.

74. The method of claim 71 wherein selecting further comprises:

selecting a first metabase;

determining whether the first selected metabase is available; and

if the first selected metabase is not available, selecting a second metabase.

75. The method of claim 71 wherein selecting comprises determining which of the multiple metabases contains the most recent metadata related to the data.

76. The method of claim 71 wherein selecting comprises determining which of the multiple metabases contains metadata associated with the most recent copy of the data.

77. The method of claim 71 wherein selecting at least one of multiple metabases comprises consulting a metabase index to identify a set of data without accessing the data itself.

78. The method of claim 71 wherein the request comprises a query to find identified data, and the selected metabases find the data without accessing the data itself.

79. A system for accessing information through a distributed database system, each database containing descriptive data describing information, the system comprising:

a means for receiving a request to find identified information;

a means for determining one or more of multiple databases that contain data describing the identified information;

a means for directing the request to the one or more determined databases; and

a means for accessing the identified information based on the data describing the identified information contained by the one or more determined databases.

80. The system of claim 79 wherein the means for directing the request routes requests based on access control information.

81. The system of claim 79 wherein the means for directing the request routes requests based on an index of data available from each metabase.

82. The system of claim 79 further comprising a means for performing a data management operation based on the identified information.

83. A method of creating metadata identifying data accesses by a client, comprising:

at each of multiple computing devices:

receiving a request to access data from a requesting computing device;

recording a record of the request, wherein the record at least identifies the requesting computing device;

creating metadata classifying the record, wherein the metadata at least identifies the requesting computing device and stores information describing the request to access data; and

storing the metadata in a metabase that contains metadata describing data accessed by each of the multiple computing devices.

84. The method of claim 83 wherein the receiving comprises intercepting the request using a filter driver.

85. The method of claim 83 further comprising searching the metabase to identify data based upon a pattern of access of the data.

86. The method of claim 83 further comprising producing a report of the interactions of a particular computing device with data.

87. The method of claim 83 wherein the recording comprises writing an entry in a log that journals requests to access data.

88. The method of claim 83 wherein storing the data comprises storing the data in network attached storage.

89. The method of claim 83 wherein the request is received by a storage proxy that supervises the transfer of data to and from network attached storage device.

90. The method of claim 83 wherein storing the metadata in a metabase comprises updating an index of computing devices that have accessed data.

91. A computer-readable media encoded with computer-executable instructions for querying a metabase for certain data interacted with by one of multiple computers, by a method comprising:

receiving a query identifying one or more computers selected from the set of computers that have accessed the certain data;

searching the metabase to locate at least one matching record, wherein the metabase contains records of interactions with the certain data at multiple computers, and wherein each record at least identifies one or more of the set of computers that requested the interaction; and

providing the at least one matching record in response to the query.

92. The computer-readable media of claim 91 further comprising normalizing a set of results containing the at least one matching record, and wherein normalizing includes removing duplicate records associated with multiple computers within the set of results.

93. The computer-readable media of claim 91 further comprising normalizing a set of results containing the at least one matching record, and wherein normalizing includes identifying a storage location of one or more data objects identified by the set of results and excluding at least one result based on the storage location of at least one of the one or more data objects.

94. The computer-readable media of claim 91 further comprising normalizing a set of results containing the at least one matching record, and wherein normalizing includes formatting the set of results based on an operating system that will receive the results.

95. The computer-readable media of claim 91 further comprising normalizing a set of results containing the at least one matching record, and wherein normalizing includes formatting the set of results based on storage parameters to be used by a data management program that will perform a data management operation related to the results.

96. The computer-readable media of claim 91 further comprising normalizing a set of results containing the at least one matching record.

97. A system for creating descriptive data identifying data accesses by a user, comprising:

a means for receiving a request from the user to access information, wherein the user is one of multiple users with access to the information;

a means for recording a record of the request, wherein the record at least identifies the one user;

a means for creating descriptive data categorizing the record, wherein the descriptive data at least identifies the one user; and

a means for storing the descriptive data in a data storage device.

98. The system of claim 97 further comprising a means for producing a report of a user's accesses of the information.

99. The system of claim 97 wherein the record identifies the information accessed by the received request.

100. A first data processing system connected via a network to at least a second data processing system, the first data processing system comprising:

a processor for executing instructions;

an input/output device coupled to the processor, wherein the input/output device receives a request from the second data processing system to access data stored in a data store coupled to the network; and,

a data storage device coupled to the processor,

wherein the processor is configured to:

create a data element identifying the request, wherein the data element at least identifies the second data processing system;

define metadata categorizing the data element, wherein the metadata at least identifies the second data processing system and describes storage characteristics of the data element; and

store the metadata in the data storage device.

101. The first data processing system of claim 100 wherein first data processing system is a central provider of a metabase to multiple other data processing systems.

102. The first data processing system of claim 100 wherein creating a data element identifying the request further comprises determining if the request is a duplicate of a previously received request, and when it is determined that the request is a duplicate, only storing the data element once within the data storage device.

103. A method of locating data objects on which to perform a data management operation within a data storage management system, comprising:

receiving a request to perform a data management operation that includes at least one criterion for identifying data objects managed by the storage management system;

accessing a metabase containing metadata describing data objects managed by the storage management system and finding specific metadata that identifies data objects managed by the storage management system satisfying the criteria;

resolving references in the specific metadata to data objects managed by the storage management system that satisfy the criteria; and

providing results that contain the resolved references to the data objects managed by the storage management system that satisfy the criteria on which to perform the data management operation.

104. The method of claim 103 wherein at least one criterion identifies a pattern of access of one or more data objects.

105. The method of claim 103 wherein accessing the metabase comprises locating metadata identifying a particular user.

106. The method of claim 103 wherein resolving references comprises locating data objects within an auxiliary copy and specifying a path for accessing the data objects using the auxiliary copy.

107. The method of claim 103 wherein resolving references comprises removing metadata describing data objects to which a user making the request does not have access.

108. A system for managing storage operations between multiple servers containing instances of a data object, comprising:

a storage operation receiving component that receives a storage request related to the data object;

a metabase component containing metadata that at least identifies one or more locations where instances of the data object can be found, wherein the metadata contains classifications of the data object that identify characteristics of the data object related to performing a storage operation;

a storage manager component that accesses the metadata and determines which of multiple devices should handle the request; and

a storage operation execution component that forwards the storage request to the determined device to handle the request.

109. The system of claim 108 wherein the storage operation receiving component receives requests to read data from a file server.

110. The system of claim 108 wherein the storage operation receiving component receives requests to access data from a web server.

111. The system of claim 108 wherein the metabase component comprises a data classification component that adds one or more data classifications to the metadata, wherein the classifications describe at least some characteristics of the data object not stored with the data object itself.

112. The system of claim 108 wherein the metabase component identifies the one or more locations based on the results of a search for data objects identified in the storage request.

113. The system of claim 108 wherein the storage manager component determines which of multiple servers should handle the request based on one or more criteria received with the storage request.

114. The system of claim 108 wherein the storage manager component determines which of multiple servers should handle the request based on a level of activity of the multiple servers, wherein the level of activity is past, current, or projected.

115. The system of claim 108 wherein the storage manager component determines which of multiple servers should handle the request based on network proximity of each of the multiple servers to the source of the request.

116. The system of claim 108 wherein the storage manager component determines which of multiple servers should handle the request based on a proximity of each of the multiple servers to the data object identified.

117. A computer-readable media encoded with instructions for controlling a data storage system to perform data management operations, by a method comprising:

receiving a request to perform a data management operation specifying a time-based criteria;

searching a metabase containing metadata that describes data objects managed by the data storage system to identify metadata

satisfying the time-based criteria, wherein the metadata contains at least categories associated with the data objects that relate to the manner in which storage operations are performed on the data objects;

determining data objects managed by the data storage system that are associated with the identified metadata; and

accessing at least one of the determined data objects managed by the data storage system to perform the requested data management operation.

118. The computer-readable media of claim 117 wherein the received request comprises locating data objects accessed during a specified time range.

119. The computer-readable media of claim 117 wherein the received request comprises locating data objects changed at a particular frequency.

120. The computer-readable media of claim 117 wherein the received request comprises locating data objects accessed by a particular user after a specified time.

121. The computer-readable media of claim 117 wherein the received request comprises a request to access data containing one or more keywords.

122. The computer-readable media of claim 117 wherein the received request comprises a request to create a copy of data according to a specified schedule.

123. The computer-readable media of claim 117 wherein the received request comprises a request to locate and restore based on a restoration criterion.

124. The computer-readable media of claim 117 wherein the received request comprises a request to locate data having a user-specified data classification.

125. The computer-readable media of claim 117 further comprising:

storing a data object in primary storage;

copying the data object to secondary storage; and

restoring the data object from secondary storage to primary storage based on classifications associated with the data object.

126. In a data storage network, a system for storing information related to data accessible within a data storage network, comprising:

a means for identifying data within the data storage network managed by a data management system;

a means for obtaining time-based information about the data within the data storage network;

a means for creating metadata that describes the identified data and that contains the obtained time-based information;

a means for storing the created metadata; and

a means for performing storage operations within the data storage network based on the metadata.

127. The system of claim 126 wherein the means for identifying data comprises a means for intercepting requests to store data.

128. The system of claim 126 wherein the means for obtaining time-based information comprises a means for obtaining a manner in which the identified data was created.

129. The system of claim 126 wherein the means for obtaining time-based information obtains a pattern of modification of the data.

130. The system of claim 126 wherein the means for obtaining time-based information comprises a means for obtaining a time that a particular user last accessed the data.

131. The system of claim 126 wherein the means for obtaining time-based information comprises a means for obtaining a time that a particular storage operation was last performed on the data.

132. The system of claim 126 wherein the means for obtaining time-based information comprises a means for obtaining a time the data was deleted.

133. The system of claim 126 wherein the means for obtaining time-based information comprises a means for obtaining a schedule for accessing the data.

134. A method of performing a data storage operation with respect to at least one secondary storage location, the method comprising:

receiving a request to perform a storage operation with respect to the at least one secondary storage location;

determining which of a primary and secondary metabase to access for information related to the storage request, wherein the primary and secondary metabases are not identical;

accessing the determined metabase to obtain information related to the storage request; and

performing the storage request based on the request and the information related to the request.

135. The method of claim 134 wherein the receiving comprises receiving selection criteria for locating one or more data objects.

136. The method of claim 134 wherein the determining is based on a distribution of metadata describing data stored within the secondary storage location.

137. The method of claim 134 wherein the determining is based on a distribution architecture of the primary and secondary metabase.

138. The method of claim 134 wherein the determining is based at least in part on a characteristic of data identified in the storage operation request.

139. The method of claim 134 wherein the determining is based on the availability of the secondary storage location.

140. The method of claim 134 wherein the accessing comprises obtaining parameters for the storage operation.

141. The method of claim 134 wherein the accessing comprises reading a storage policy related to the storage request.

142. The method of claim 134 wherein the received storage request comprises a request to backup or restore data.

143. A method of identifying changed data, comprising:

selecting an entry of a log that journals changes to data and determining if the selected entry refers to data tracked by a data classification system that stores classifications associated with data that describe the manner in which data management operations are performed on data;

if the selected entry refers to data tracked by the data classification system, identifying at least a first change to the data represented by the entry; and

updating the data classification system based on the identified change.

144. A system for identifying data based on entities associated with the data, comprising:

an entity definition component that defines one or more entities;

a data identification component that locates data and determines one or more entities associated with the data;

a data classification component that determines at least one characteristic of the data related to patterns describing

interactions with the one or more entities associated with the data; and

a metabase component that stores the metadata and responds to requests to perform data storage operations based on the metadata.

145. The system of claim 144 wherein the entity definition component comprises a Microsoft Active Directory.

146. The system of claim 144 wherein the entity definition component defines one or more users of the system.

147. The system of claim 144 wherein the entity definition component defines one or more groups of users of the system and the groups are based on job functions within an organization.

148. The system of claim 144 wherein the entity definition component defines one or more groups of users of the system.

149. The system of claim 144 wherein the data identification component determines an entity that created the data.

150. The system of claim 144 wherein the data identification component determines one or more entities that accessed the data.

151. A method of finding data objects based on users associated with the data objects, comprising:

receiving a request to identify one or more data objects under the control of a data management system, wherein the request specifies one or more criteria based on a pattern of user interaction with the data objects;

searching a metabase for metadata identifying data objects under the control of the data management system that satisfy the criteria; and

providing the identified data objects in response to the request.

152. The method of claim 151 wherein the criteria identifies like data objects based on similarities among the data objects.

153. The method of claim 151 wherein the criteria identifies data objects based on the results of past searches for data objects.

154. The method of claim 151 wherein the criteria identifies data objects that were previously stored within the data management system and have been deleted.

155. The method of claim 151 wherein the criteria comprises a request to identify data objects accessed by a specified user.

156. The method of claim 151 wherein the criteria comprises a request to identify data objects accessed according to a particular pattern of behavior by one or more users.

157. The method of claim 151 wherein the criteria comprises a request to identify data objects associated with a group of users.

158. The method of claim 151 wherein the criteria comprises a request to identify data objects related to users assigned to a particular project.

159. The method of claim 151 wherein the criteria comprises a request to identify data objects based on user access control information.

160. The method of claim 151 wherein the criteria comprises a request to identify data objects made unavailable by a particular user.

161. A method of distributing metadata throughout a network, comprising:

receiving an indication to perform a copy operation on one or more data objects;

analyzing a first metabase to identify data objects in a first data store associated with the first metabase that satisfy a selection criteria;

copying the identified data objects from the first data store to a second data store; and

copying metadata associated with the identified data objects from the first metabase to the second metabase, wherein the second metabase is associated with the second data store.

162. The method of claim 161 further comprising, after copying the identified data objects, updating the first metabase with information describing the copied data objects

163. The method of claim 161 wherein the selection criteria is based on a type of the identified data objects.

164. The method of claim 161 wherein the selection criteria is based on a source of the identified data objects.

165. The method of claim 161 wherein the selection criteria is based on a topology of the network.

166. The method of claim 161 wherein the selection criteria is based on information within the metadata.

167. The method of claim 161 wherein sending the selection criteria is based on a migration policy.

168. The method of claim 161 wherein the first data store and first metabase are associated with a first computing device and the second data store and second metabase are associated with a second computing device.

169. The method of claim 161 wherein a computing device accesses data at the second data store when the first data store is unavailable.

170. The method of claim 161 wherein copying metadata includes adding one or more data classifications describing an availability of the copied data at the first data store and the second data store.

171. The method of claim 161 wherein copying metadata includes adding one or more data classifications describing the copy operation.

172. A system of distributing metadata throughout a network, comprising:

a primary data store that contains a production copy of stored data;

a primary metabase component that stores metadata associated with the primary data store, wherein the metadata contains information about data stored at the primary data store and describes characteristics of the data that are not stored with the data;

a secondary data store that contains one or more secondary copies of the data;

a secondary metabase that stores metadata associated with the secondary data store; and

a data management operation component that performs data management operations on the stored data and uses the primary and secondary metabases to facilitate performing data management operations.

173. The system of claim 172 further comprising a metadata migration component that migrates metadata from the primary metabase to the secondary

metabase when data is copied from the primary data store to the secondary data store.

174. The system of claim 172 further comprising a data restoration component that copies a subset of the stored data from the secondary data store to the primary data store upon the occurrence of an event, wherein the subset of the stored data is determined using the metadata stored at the primary and secondary metabases.

175. A computer-readable media encoded with instructions for controlling a computer system to move data objects from a primary data store to a secondary data store, by a method comprising:

analyzing a primary metabase to determine data objects stored within a primary data store satisfying a data object query;

identifying a secondary data store based on metadata stored within a secondary metabase to receive the determined data objects;

copying the data objects from the primary data store to the secondary data store; and

updating the secondary metabase with information describing the copied data objects.

176. The computer-readable medium of claim 175 further comprising, after copying the data objects from the primary data store, deleting the data objects from the primary data store.

177. The computer-readable medium of claim 175 further comprising, after copying the data objects from the primary data store, deleting metadata describing the data objects from the primary metabase.

178. The computer-readable medium of claim 175 further comprising, after copying the data objects from the primary data store, updating the primary metabase with information describing the copied data objects.

179. A system for providing redundant access to metadata over a network, comprising:

a means for storing at a centralized primary storage metabase, metadata that contains information about data stored at a computing device and describes characteristics of the data that are not stored with the data;

a means for selecting one of multiple distributed secondary storage metabases to store a copy of at least some of the metadata stored at the centralized primary storage metabase; and

a means for storing metadata at the selected one of multiple distributed secondary storage metabases via the network.

180. The system of claim 179 further comprising a means for determining which of the centralized primary storage metabases and the one of multiple distributed secondary storage metabases to use to access the metadata.

181. One or more computer memories collectively storing a data structure for describing interactions with stored data managed by a data management system, comprising:

an interaction type field, wherein the interaction type field describes a type of interaction that modified the stored data managed by the data management system;

a data identifying field, wherein the data identifying field describes the stored data that was modified;

an updated value field that specifies a new value for the stored data; and

a data classification field, wherein the data classification field specifies one or more data classifications associated with the stored data, and wherein the one or more data classifications relate the stored data to other stored data having similar characteristics that are not stored with the data.

182. The data structure of claim 181 further comprising an identifier field that distinguishes one instance of the data structure from other instances of the data structure.

183. The data structure of claim 181 further comprising a user field that identifies a user associated with the interaction.

184. The data structure of claim 181 further comprising a device field that identifies a device associated with the interaction.

185. The data structure of claim 181 wherein the data structure is part of a journal entry data structure.

186. The data structure of claim 181 wherein the data classification field identifies an application associated with the data.

187. The data structure of claim 181 further comprising an update sequence number that is used to determine recent activity associated with the stored data.

188. One or more computer memories collectively storing a data structure for describing a data storage operation, comprising:

a storage operation field that describes a storage operation to be performed;

a storage criteria field that specifies a manner in which the storage operation is to be performed based on one or more data classifications that describe characteristics of data that are not stored in a primary location of the data;

a metabase determining field that provides information to determine which of multiple metabases to query for information related to the storage operation; and

an operation status field that describes a status of the storage operation and coordinates at least a stage of the storage operation that

accesses the metabase and a stage of the storage operation that accesses a data storage device.

189. The data structure of claim 188 wherein the storage operation field specifies a copy of data from the primary location of the data to a secondary location.

190. The data structure of claim 188 wherein the storage criteria specifies a network location.

191. The data structure of claim 188 wherein the storage criteria contains a schedule describing when the storage operation should be performed.

192. The data structure of claim 188 wherein the storage criteria describes relationships between components of a storage system, and wherein the relationships are associated with tiers in a multi-tiered storage system.

193. The data structure of claim 188 wherein the storage criteria describes relationships between components of a storage system.

194. The data structure of claim 188 wherein the storage criteria describes retention policies.

195. The data structure of claim 188 wherein the storage criteria describes encryption requirements.

196. The data structure of claim 188 wherein the storage criteria describes components of a system to utilize to perform the storage operation.

197. The data structure of claim 188 wherein the storage criteria describes media to be used by the storage operation.

198. The data structure of claim 188 further comprising a user preferences field that describes user settings related to the storage operation.

199. One or more computer memories collectively storing a data structure for describing data stored within a storage system, comprising:

a data identification field that distinguishes a data object from other data objects stored within the storage system;

a device field that identifies a device associated with the data object; and

a data classification field that specifies one or more classifications that describe the data object and a manner in which storage operation are performed on the data object,

wherein the storage system is configured to perform data storage operations using the data structure.

200. The data structure of claim 199 further comprising a data object field that contains at least some of the contents of the data object.

201. The data structure of claim 199 further comprising a reference count field that counts a number of instances of the data object stored within the storage system, and wherein data storage operations access the data at a location based on the reference count field.

202. The data structure of claim 199 further comprising a storage policy field that specifies a storage policy for the data object.

203. The data structure of claim 199 further comprising a data availability field that specifies an estimate of an amount of time required to access the data.

204. The data structure of claim 199 further comprising a hierarchy information field that describes a relationship of the data object to other data objects in a hierarchy, and wherein the hierarchy is based on a network topography.

205. The data structure of claim 199 further comprising a hierarchy information field that describes a relationship of the data object to other data objects in a hierarchy, and wherein the hierarchy is based on geographic information.

206. The data structure of claim 199 further comprising a hierarchy information field that describes a relationship of the data object to other data objects in a hierarchy.

207. The data structure of claim 199 further comprising an operation status field that describes a progress of an ongoing storage operation.

208. A method of classifying data, comprising:

installing a data classification agent on a computer system, wherein the data classification agent classifies data stored on the computer system according to characteristics of the data;

classifying data using the data classification agent installed on the computer system;

populating a metabase with data classified by the data classification agent; and

performing a data management operation on the data based on the classifications of the data.

209. The method of claim 208 further comprising:

monitoring changes to the data using the data classification agent; and

updating the metabase based on the monitored changes.

210. The method of claim 208 wherein the installing comprises installing a data classification agent for classifying data produced by a single application.

211. The method of claim 208 wherein the classifying comprises receiving data classifications from a user of the computer system.

212. The method of claim 208 wherein the classifying comprises determining an application that created the data.

213. The method of claim 208 wherein the classifying comprises adding access control information associated with the data to the metabase.

214. The method of claim 208 wherein the classifying comprises storing information about user interactions with the data.

215. The method of claim 208 wherein the classifying comprises storing information about the relationship of the data to other data.

216. The method of claim 208 wherein the classifying comprises identifying a storage policy for the data.

217. A computer-readable media encoded with instructions for controlling a computer system to operate as a data classification agent, by a method comprising:

identifying data stored within a data store;

determining one or more classifications that describe the stored data, wherein the classifications describe characteristics of the data, and wherein at least some of the classifications are not stored with the data and are based on information about an application that generated the stored data;

creating metadata containing information about the stored data including the determined one or more classifications; and

storing the metadata in a non-volatile storage device.

218. The computer-readable media of claim 217 wherein identifying data comprises trapping requests to store data within the data store.

219. The computer-readable media of claim 217 wherein the determining comprises identifying an application associated with the stored data.

220. The computer-readable media of claim 217 wherein the determining comprises identifying keywords within the stored data.

221. The computer-readable media of claim 217 wherein the determining comprises identifying structure within the stored data.

222. A system for classifying data, comprising:

a data store containing multiple data objects generated by one or more applications;

multiple data classification agents configured to classify the multiple data objects into at least one of multiple data classifications, wherein the data classifications categorize the data based on characteristics related to the use of the data; and

a metabase update component configured to store metadata describing the multiple data objects and the at least one of multiple data classifications for each data object in a metabase.

223. The system of claim 222 wherein each data classification agent is specialized to classify data produced by a single application.

224. The system of claim 222 wherein at least one data classification agent classifies data produced by Microsoft Exchange Server.

225. The system of claim 222 wherein at least one data classification agent classifies data produced by Microsoft SQL Server.

226. The system of claim 222 wherein at least one data classification agent classifies data produced by a database application.

227. The system of claim 222 wherein at least one data classification agent classifies data produced by a web application.

228. The system of claim 222 wherein at least one data classification agent classifies data produced by a word processing application.

229. The system of claim 222 wherein at least one data classification agent classifies data produced by a spreadsheet.

230. A system for classifying application data, comprising:

a means for determining an application that created a particular data object;

a means for parsing certain data within the data object to identify a structure of the certain data;

a means for assigning one or more data classifications to the parsed data; and

a means for storing the structure of the data and the one or more assigned data classifications as metadata describing the data object.

231. The system of claim 230 wherein the means for determining determines the application based on an extension of a file containing the data object.

232. The system of claim 230 wherein the means for determining determines the application based on header information within the data object.

233. The system of claim 230 wherein the means for parsing divides the data object into logical units of application-specific data.

234. The system of claim 230 wherein the means for assigning assigns classifications based on the user that created the data object.

235. The system of claim 230 wherein the means for assigning assigns classifications based on the contents of the data object.

236. The system of claim 230 wherein the means for assigning assigns a classification that describes an original location of the data object.

237. The system of claim 230 wherein the means for storing stores the metadata in a metabase.

238. A method of classifying data stored within a storage management system based on user input, comprising:

receiving an indication of a data object within the storage management system;

receiving user input providing one or more classifications for the selected data object that categorize the data object within the storage management system, wherein the classifications describe characteristics of the data object and facilitate storage operations on the data object; and

storing the user-provided one or more classifications as metadata associated with the data object, such that the one or more classifications are accessible for performing storage operations based on the classifications.

239. The method of claim 238 wherein receiving an indication comprises displaying a graphical user interface for receiving user input selecting a data object.

240. The method of claim 238 wherein receiving user input comprises displaying a list of available classifications.

241. The method of claim 238 wherein receiving user input comprises providing an interface for the user to create new classifications based on a user's purpose for interacting with the selected data object.

242. The method of claim 238 wherein the one or more classifications comprise classifications related to the source of the data object.

243. The method of claim 238 wherein the one or more classifications comprise classifications related to the purpose of the data object.

244. The method of claim 238 wherein the one or more classifications comprise classifications related to intended consumers of the data object.

245. The method of claim 238 wherein the one or more classifications comprise classifications related to content of the data object.

246. The method of claim 238 wherein the one or more classifications comprise classifications related to a pattern of interaction with the data object.

247. A system for extending metadata related to stored data objects within a storage management system based on user-specified classifications, comprising:

a metadata identifying component configured to identify metadata related to one or more data objects, wherein the data objects are selected by a user and stored within the storage management system;

a data classification component configured to create and receive data classifications from a user for associating with the identified metadata, wherein the classifications associate characteristics with the data objects that are not stored at least in part with the data objects themselves; and

a metadata update component configured to associate classifications received from a user with the data object, wherein the received classifications categorize the metadata within the storage management system.

248. The system of claim 247 wherein the metadata update component stores the classifications as tags associated with the metadata.

249. A computer-readable media encoded with instructions for controlling a computer system to locate data objects within a data management system based on user-defined attributes, by a method comprising:

receiving a search query specifying one or more user-defined attributes;

locating metadata within the data management system matching the specified one or more user-defined attributes; and

providing search results based on the located metadata within the data management system,

wherein a user-defined attribute specifies a data classification associated with a data object that classifies the data object within the data management system and facilitates the manner in which a storage operation is to be performed on one or more data objects associated with the metadata.

250. The computer-readable media of claim 249 wherein receiving a search query comprises receiving user-defined attributes identifying a project associated with at least one data object.

251. The computer-readable media of claim 249 wherein receiving a search query comprises receiving user-defined attributes identifying a user responsible for at least one data object.

252. The computer-readable media of claim 249 wherein receiving a search query comprises receiving user-defined attributes identifying a group with access to at least one data object.

253. The computer-readable media of claim 249 wherein receiving a search query comprises receiving user-defined attributes identifying a purpose associated with at least one data object.

254. The computer-readable media of claim 249 wherein receiving a search query comprises receiving user-defined attributes identifying a pattern of interaction with at least one data object.

255. The computer-readable media of claim 249 wherein receiving a search query comprises receiving user-defined attributes identifying access control information associated with at least one data object.

256. The computer-readable media of claim 249 wherein receiving a search query comprises receiving user-defined attributes identifying an event associated with at least one data object.

257. The computer-readable media of claim 249 wherein receiving a search query comprises receiving user-defined attributes identifying keywords associated with at least one data object.

258. A method of updating metadata associated with data objects, comprising:

receiving an update identifying a data object and one or more data classifications related to the data object, wherein the data classifications describe one or more storage characteristics of the data object;

searching for metadata previously associated with the object;

if previous metadata is not found, creating metadata associated with the object based on the received update and identified data classifications; and

if previous metadata is found, updating the metadata based on the received update and identified data classifications.

259. The method of claim 258 wherein the received update comprises a change journal entry.

260. The method of claim 258 wherein the received update identifies a data object that has been deleted.

261. The method of claim 258 wherein the received update identifies a data object that has been modified by a particular user.

262. The method of claim 258 wherein the received update identifies a data object that has been moved to a new location.

263. The method of claim 258 wherein the received update identifies a data object that has been copied from a first computing device to a second computing device.

264. The method of claim 258 wherein the identified data classifications specify a storage retention policy for the data object.

265. The method of claim 258 wherein the identified data classifications specify a storage aging policy for the data object.

266. The method of claim 258 wherein the identified data classifications specify an access history of the data object.

267. A system for integrating metadata with a metabase, comprising:

a metabase component that stores metadata identifying one or more data objects and one or more data classifications associated with each data object, wherein the data classifications describe one or more characteristics of the data object;

a metadata receiving component that receives metadata for integration with the metabase;

a metadata locating component for locating metadata in the metabase related to the received metadata; and,

a metadata integration component for adding the received metadata into the metabase.

268. The system of claim 267 wherein the metadata locating component locates metadata that refers to a data object referred to by the received metadata.

269. The system of claim 267 wherein the metadata locating component locates metadata associated with an identified user.

270. The system of claim 267 wherein the metadata receiving component receives metadata that identifies the location of auxiliary copies of data.

271. The system of claim 267 wherein the metadata receiving component receives notifications of storage operations performed on data objects and updates the metadata based on the received notifications.

272. The system of claim 267 wherein the metadata integration component comprises a metabase migration component for moving metadata when a copy of data associated with the metadata is made.

273. The system of claim 267 further comprising a data classification component for adding data classifications to the received metadata.

274. A computer-readable media encoded with instructions for controlling a computer system to create a copy of a metabase, by a method comprising:

receiving a storage operation containing a request to create a secondary copy of a metabase, wherein the metabase contains metadata that includes one or more data categories that classify data tracked by the metabase;

identifying a layout for the secondary copy of the metabase;

partitioning the metadata according to the identified layout; and

copying at least a portion of the metadata from a source metabase to a target location to create a secondary copy of the metabase.

275. The computer-readable media of claim 274 wherein identifying the layout comprises identifying a distribution architecture for a distributed secondary metabase.

276. The computer-readable media of claim 274 wherein partitioning the metadata comprises determining the metadata for which the secondary copy of the metabase will handle requests.

277. The computer-readable media of claim 274 wherein partitioning the metadata comprises identifying a distribution of data associated with the metadata among data storage devices associated with the secondary copy of the metabase.

278. A system for managing data associated with a data storage component, wherein the data storage component is coupled to at least one computer, and stores data from one or more applications running on the computer, the system comprising:

a data storage management component for managing data stored within the data storage component,;

a data classification component for associating classifications with the stored data managed by the data storage management component, wherein the classifications describe characteristics of the stored data that describe a data management state of the stored data; and

a metabase associated with the data storage management component that stores metadata referring to the stored data managed by the data storage management component, wherein the metadata includes the associated classifications for the stored data,

wherein at least the data storage management component or the data classification component is separate from the data storage component, and is configured to track interactions between the data storage component and the one or more applications running on the at least one computer.

279. The system of claim 278 wherein the data storage component comprises network attached storage, and wherein at least one of the data storage management component, the data classification component, and the metabase are

specifically configured for use with the network attached storage and for communication with the network attached storage via an open network protocol designed to perform backups over heterogeneous networks.

280. The system of claim 278 wherein the data storage component comprises direct attached storage, and wherein at least one of the data storage management component, the data classification component, and the metabase are specifically configured for use with the direct attached storage.

281. The system of claim 278 wherein the data management state of the stored data comprises a data protection state of the stored data.

282. The system of claim 278 wherein the data storage component comprises network attached storage, and wherein at least one of the data storage management component, the data classification component, and the metabase are specifically configured for use with the network attached storage.

283. The system of claim 278 wherein the data storage management component comprises a journaling agent to record interactions between the data storage component and the at least one application.

284. The system of claim 278 wherein the data storage component employs a closed, proprietary system, and wherein at least one of the data storage management component, the data classification component, and the metabase are specifically configured for use with the data storage component.

285. The system of claim 278 wherein the data classifications identify a type of media used by the data storage component.

286. The system of claim 278 wherein the data classifications identify an aging criteria of the stored data.

287. The system of claim 278 wherein the data storage management component facilitates storage operations by interpreting and processing the stored data to accommodate a format specific to the data storage device.

288. A method of accessing documents available through a network, wherein the documents are stored on one or more data storage devices coupled to the network, the method comprising:

receiving a request to access one or more documents;

searching a metabase to locate metadata describing one or more documents satisfying the request based on data classifications stored within the metadata that describe characteristics of the documents;

resolving references in the located metadata to identify one of the one or more data storage devices containing the data, wherein the resolving at least assists in determining which of the more or more data storage devices to access the data from, and wherein the access is at least in part based on a data access criteria; and

providing a signal to access, via the network and in response to the request, the one or more documents at the one or more identified data storage devices.

289. The method of claim 288 wherein the data access criteria is based on reducing the impact of the request on a primary data storage device.

290. The method of claim 288 wherein the resolving of references determines an availability of the one or more documents at each data storage device.

291. The method of claim 288 wherein the metabase is distributed across multiple server systems and the searching of the metabase comprises identifying one or more metabases containing metadata describing documents satisfying the request.

292. The method of claim 288 wherein the resolving of references comprises determining which of multiple copies of the document is currently available for access.

293. The method of claim 288 wherein the resolving of references comprises determining at least an access time associated with multiple types of data storage devices.

294. The method of claim 288 wherein at least one identified data storage device is network attached storage.

295. A system for backing up data, comprising:

a means for managing storage of data;

a means for creating a backup of the stored data;

a means for creating one or more copies of the backup using multiple storage devices; and

a means for storing metadata describing an availability of the backup data from the multiple storage devices, wherein the means for storing metadata includes means for tracking interactions with the backup data by components external to the means for storing metadata.

296. The system of claim 295 wherein the means for creating one or more copies of the backup creates a secondary copy from a primary copy of the backup.

297. The system of claim 295 wherein the means for creating one or more copies modifies the backup by performing storage operations on the backup before creating the one or more copies.

298. A method of classifying application-specific data, the method comprising:

receiving data from an application, wherein the data is specific to the application;

classifying the application-specific data to create metadata containing one of multiple defined classifications of the application-specific data, wherein the classifications describe characteristics of the application-specific data that facilitate the performance of storage operations on the application-specific data; and,

storing the created metadata in a metabase, wherein the metabase stores metadata created by multiple different applications, wherein the metadata is specific to each of the multiple different applications.

299. The method of claim 298 wherein the receiving comprises receiving email data generated by an email application.

300. The method of claim 298 wherein the receiving comprises receiving database data generated by a database application.

301. The method of claim 298 wherein the receiving comprises receiving data generated by a network application.

302. The method of claim 298 wherein the receiving comprises receiving data generated by an Internet application.

303. The method of claim 298 wherein the receiving comprises receiving storage data generated by a storage application.

304. The method of claim 298 wherein the receiving comprises receiving multimedia data generated by a multimedia application.

305. The method of claim 298 wherein the metadata identifies the application that generated the application-specific data.

306. A computer-readable media encoded with instructions for controlling a computing system to identify documents complying with a request for documents, by a method comprising:

receiving a request for documents, wherein the request relates to satisfying a compliance criteria;

based on the received request, searching a metabase to locate metadata describing documents satisfying the compliance criteria, wherein the metadata contains data classifications that describe characteristics of the documents that are not stored with the documents; and,

providing report data, wherein the report data relates to documents satisfying the compliance criteria based on the located metadata.

307. The computer-readable media of claim 306 wherein the compliance criteria is a legal discovery request.

308. The computer-readable media of claim 306 wherein the metadata identifies an availability of the documents.

309. The computer-readable media of claim 306 wherein the metadata identifies a retention policy of the documents.

310. The computer-readable media of claim 306 wherein the metadata identifies one or more keywords contained in the documents.

311. The computer-readable media of claim 306 wherein the metadata identifies one or more users with access to the documents.

312. A system for recovering data in an enterprise, comprising:

a data object identification component for receiving criteria from a user identifying one or more data objects to recover;

a metabase component containing metadata describing data objects and classifications associated with the data objects that identify information relevant to searching for the data objects;

a search component for identifying metadata within the metabase that satisfies the received criteria; and,

a data recovery component for recovering data described by the metadata identified by the search component.

313. The system of claim 312 wherein the data object identification component comprises an application selection component that receives an application selection from the user that identifies a type of data the user wants to recover.

314. The system of claim 312 wherein the data object identification component receives a selection from the user of a time range of the data objects that the criteria identifies.

315. The system of claim 312 wherein the data recovery component recovers email deleted by the user based on the received criteria.

316. The system of claim 312 wherein the metabase component comprises metadata describing objects that have been archived.

317. The system of claim 312 wherein the received criteria contains user-defined data classifications.

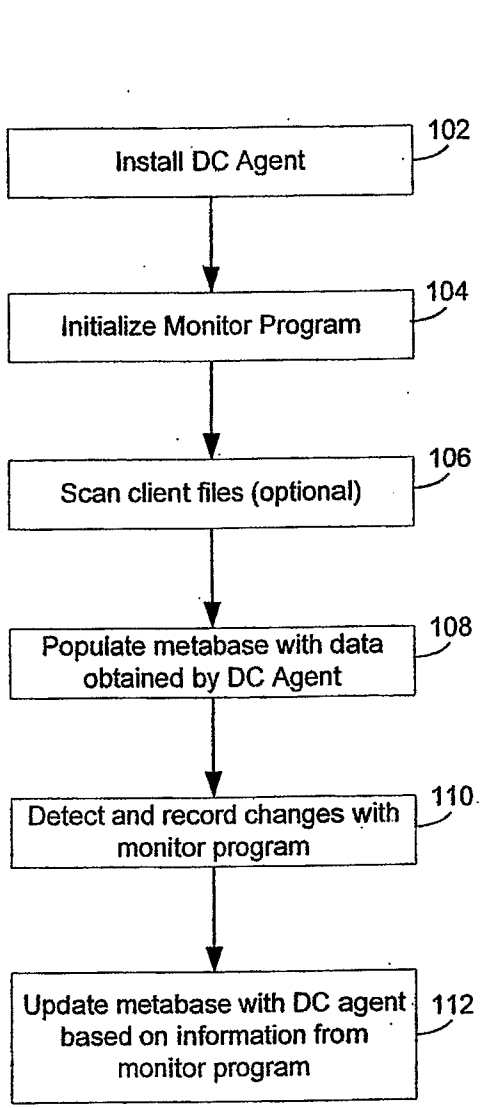
100
Install DC Agent
102
Initialize Monitor Program
104
Scan client files (optional)
106
Populate metabase with data obtained by DC Agent
108
Detect and record changes with monitor program
110
Update metabase with DC agent based on information from monitor program
112

FIG. 1

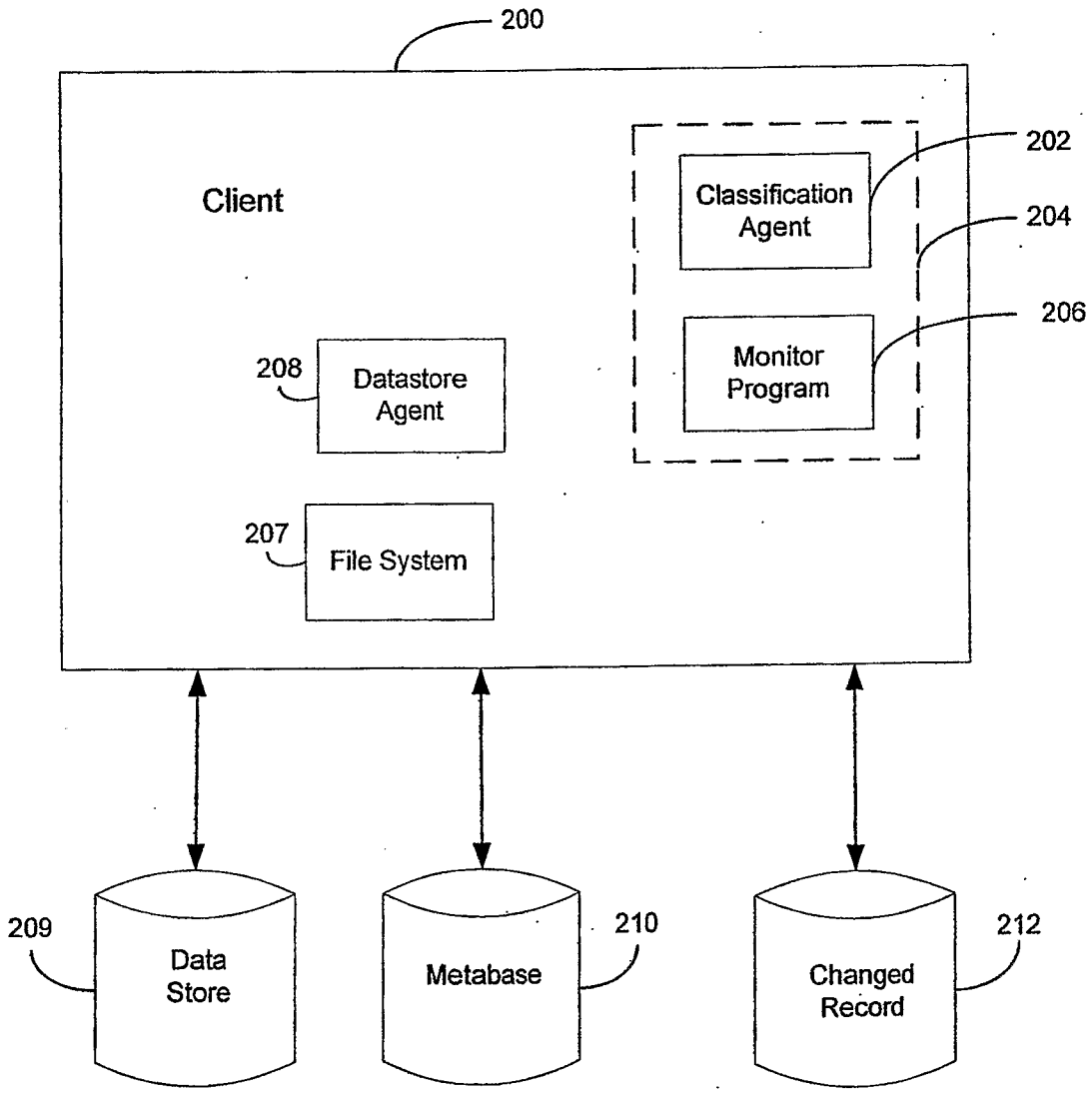
200
Client
202
Classification Agent
204
206
Monitor Program
208
Datastore Agent
207
File System
209
Data Store
210
Metabase
212
Changed Record

FIG. 2

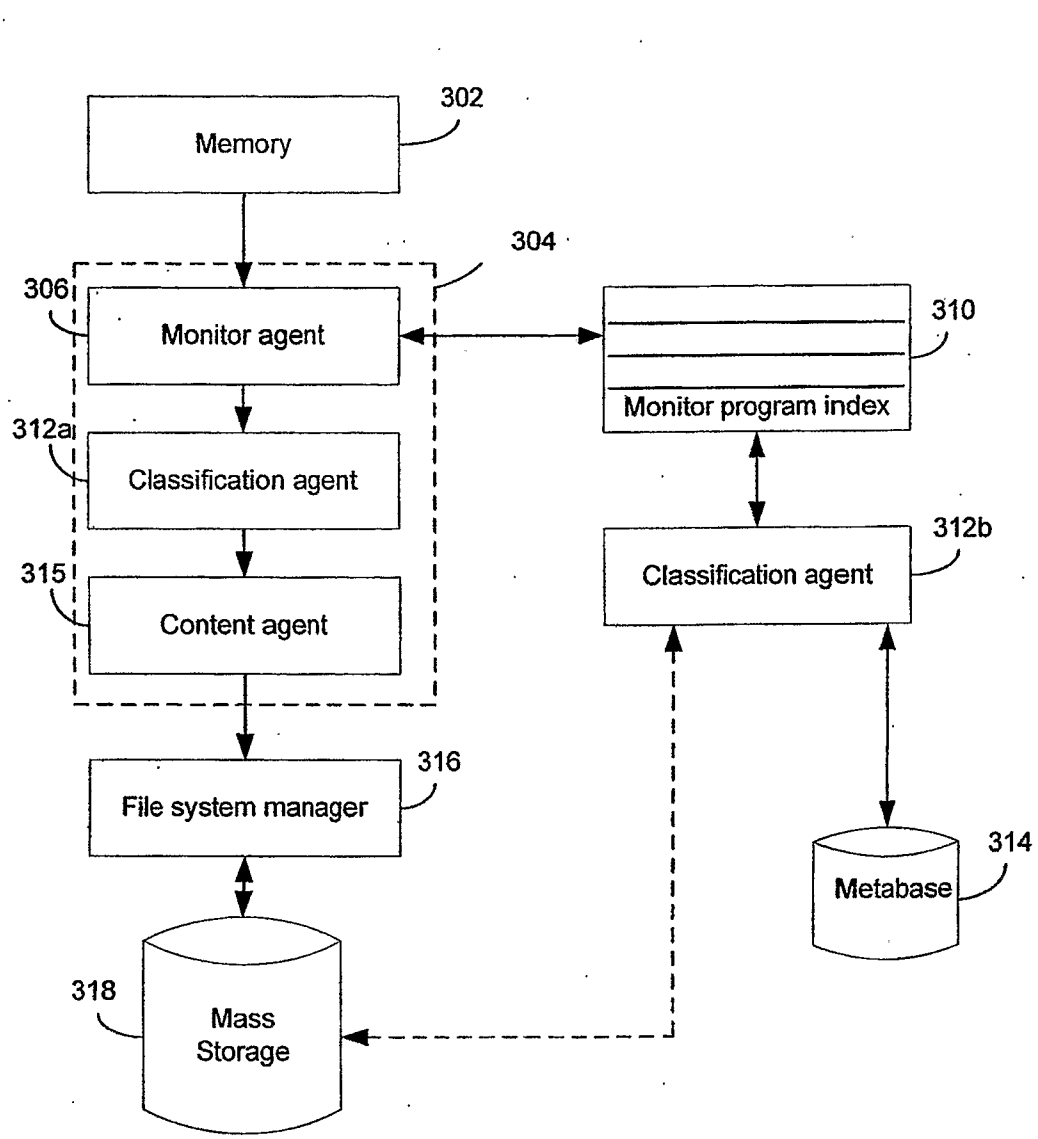
300
Memory
302
304
306
Monitor agent
310
Monitor program index
312a
Classification agent
312b
Classification agent
315
Content agent
316
File system manager
314
Metabase
318
Mass Storage

FIG. 3a

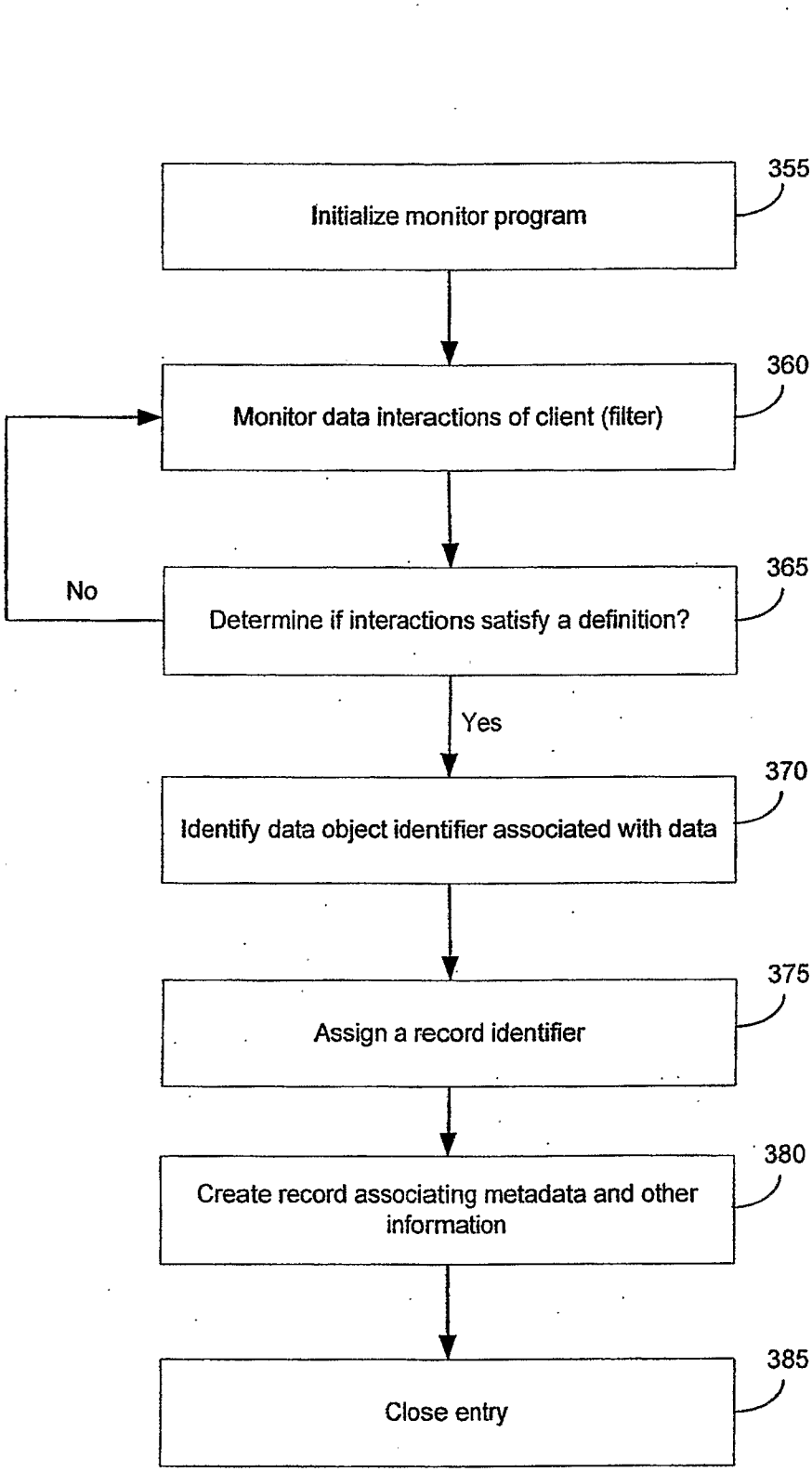
350
Initialize monitor program
355
Monitor data interactions of client (filter)
360
No
Determine if interactions satisfy a definition?
365
Yes
Identify data object identifier associated with data
370
Assign a record identifier
375
Create record associating metadata and other information
380
Close entry
385

FIG. 3b

400

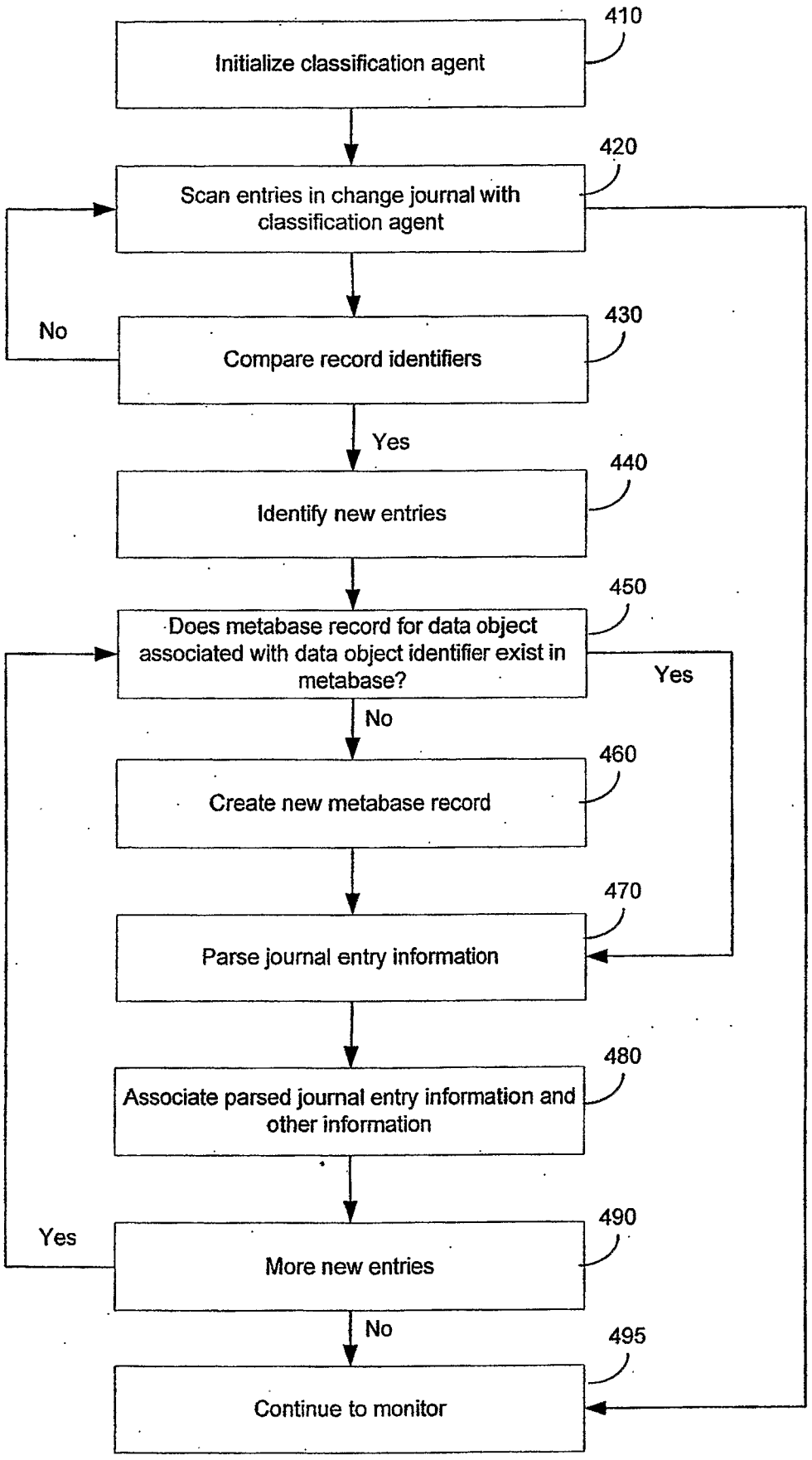
410
Initialize classification agent
420
Scan entries in change journal with classification agent
No
430
Compare record identifiers
Yes
440
Identify new entries
450
Does metabase record for data object associated with data object identifier exist in metabase?
Yes
No
460
Create new metabase record
470
Parse journal entry information
480
Associate parsed journal entry information and other information
Yes
490
More new entries
No
495
Continue to monitor

FIG. 4

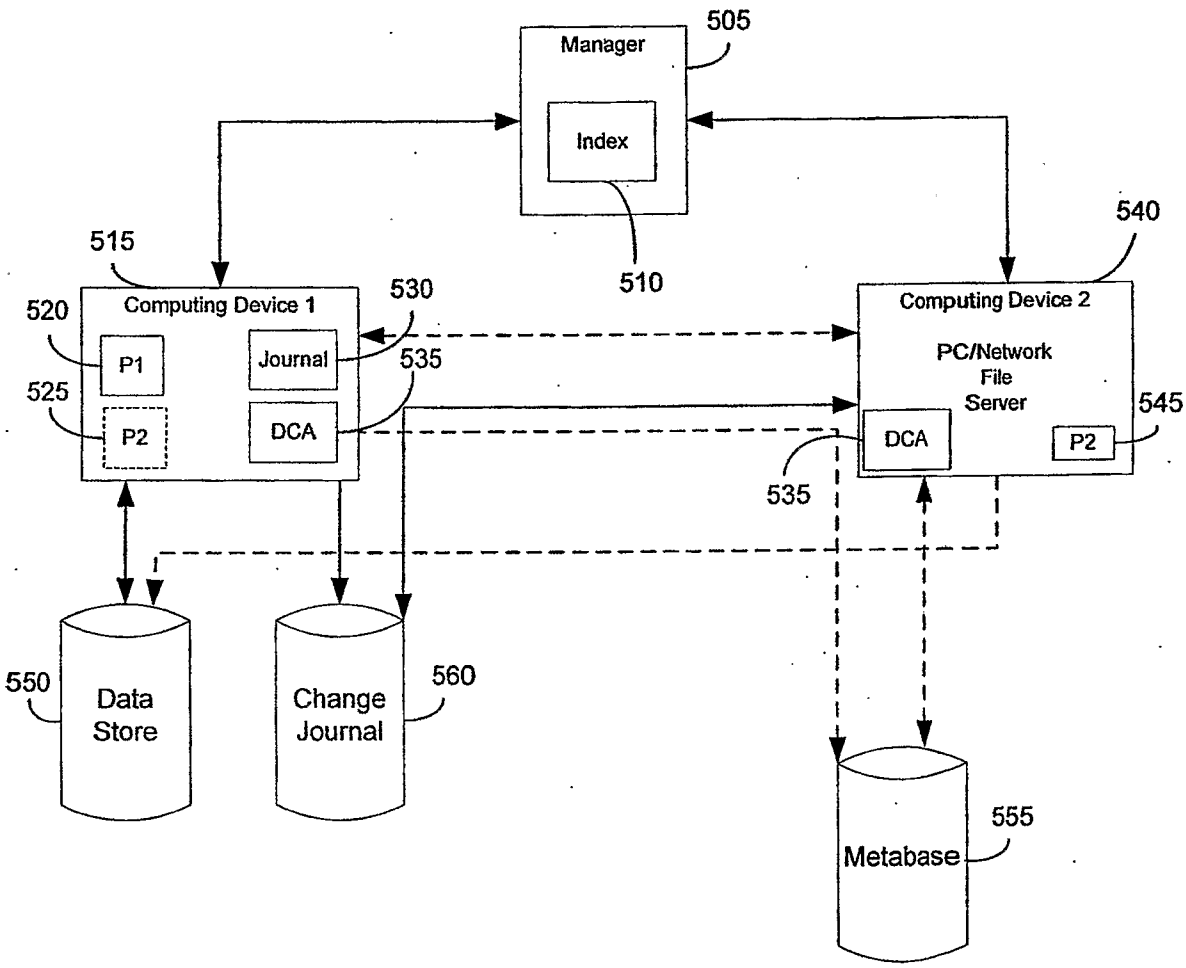
500
Manager
505
Index
510
515
Computing Device 1
520
P1
525
P2
Journal
530
535
DCA
540
Computing Device 2
PC/Network
File
Server
545
DCA
P2
535
550
Data
Store
Change
Journal
560
Metabase
555

FIG. 5

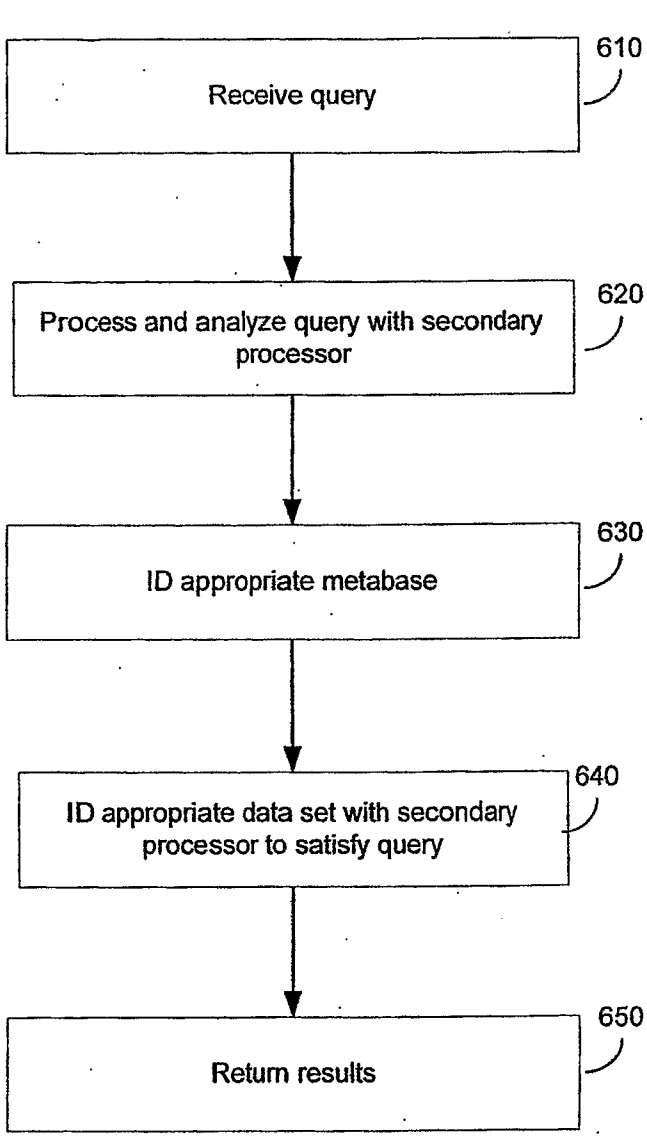
600
610
Receive query
620
Process and analyze query with secondary processor
630
ID appropriate metabase
640
ID appropriate data set with secondary processor to satisfy query
650
Return results

FIG. 6

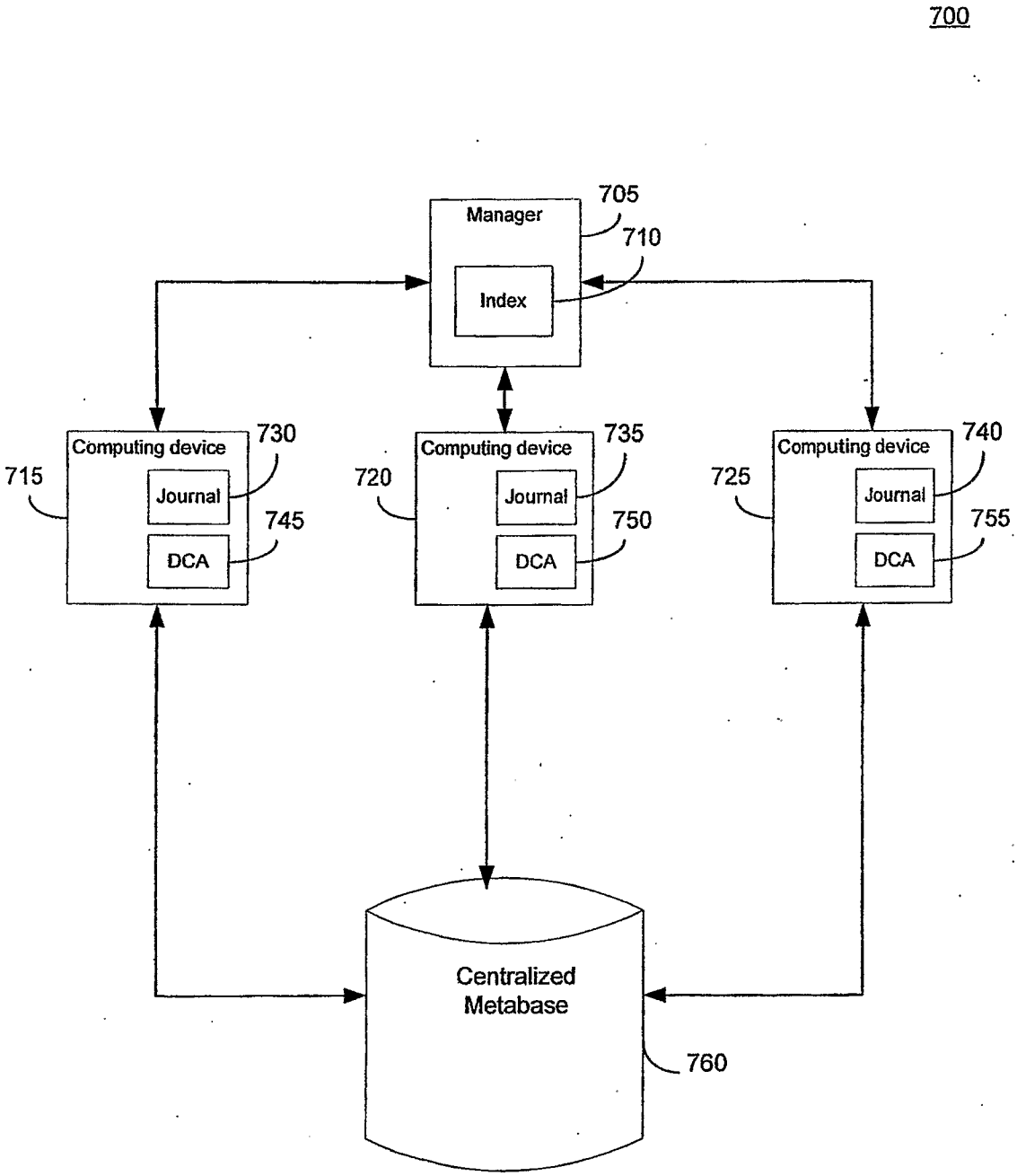
700
Manager
705
710
Index
Computing device
730
715
Journal
745
DCA
Computing device
735
720
Journal
750
DCA
Computing device
740
725
Journal
755
DCA
Centralized
Metabase
760

FIG. 7

800

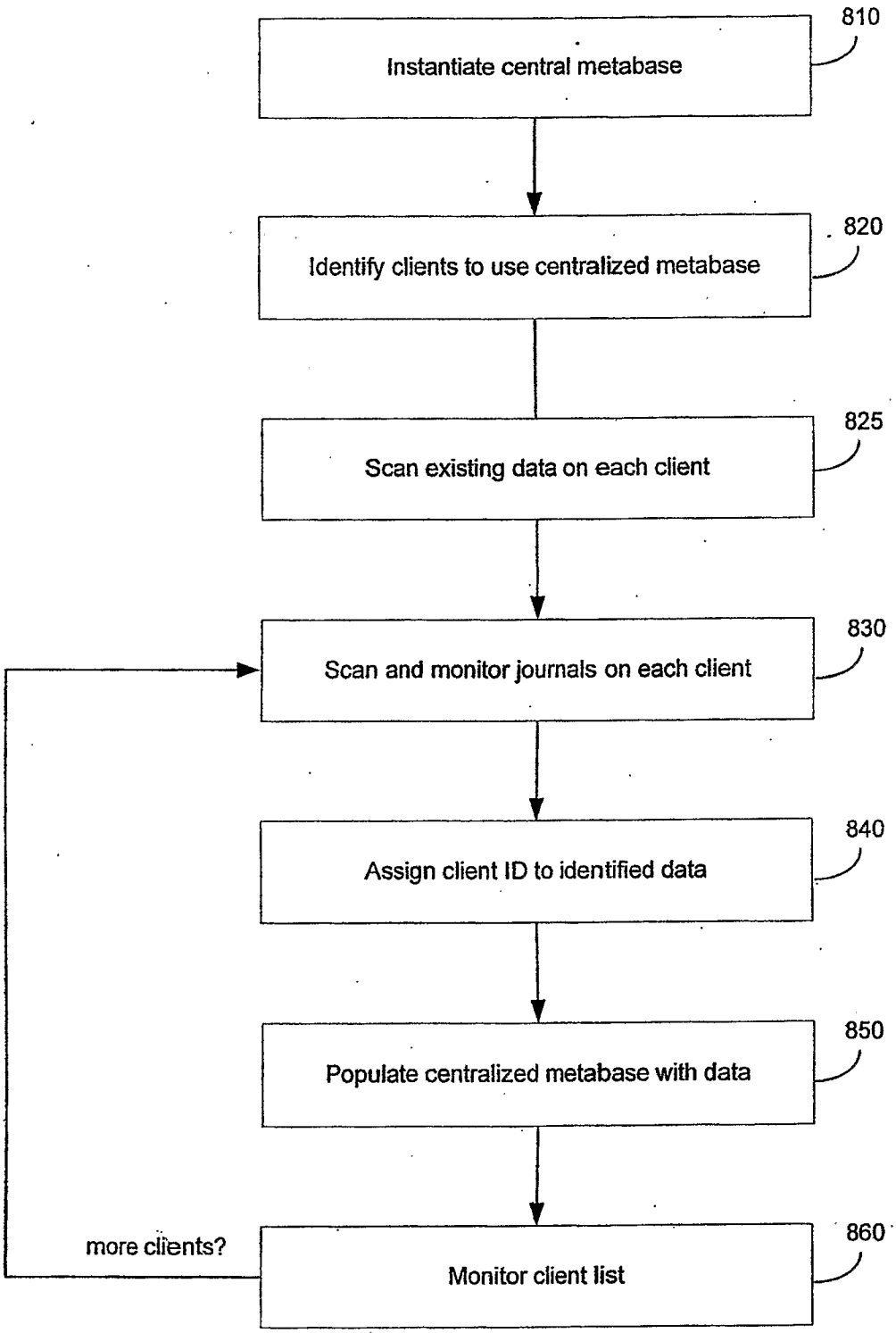
Instantiate central metabase
810
Identify clients to use centralized metabase
820
Scan existing data on each client
825
Scan and monitor journals on each client
830
Assign client ID to identified data
840
Populate centralized metabase with data
850
Monitor client list
860
more clients?

FIG. 8

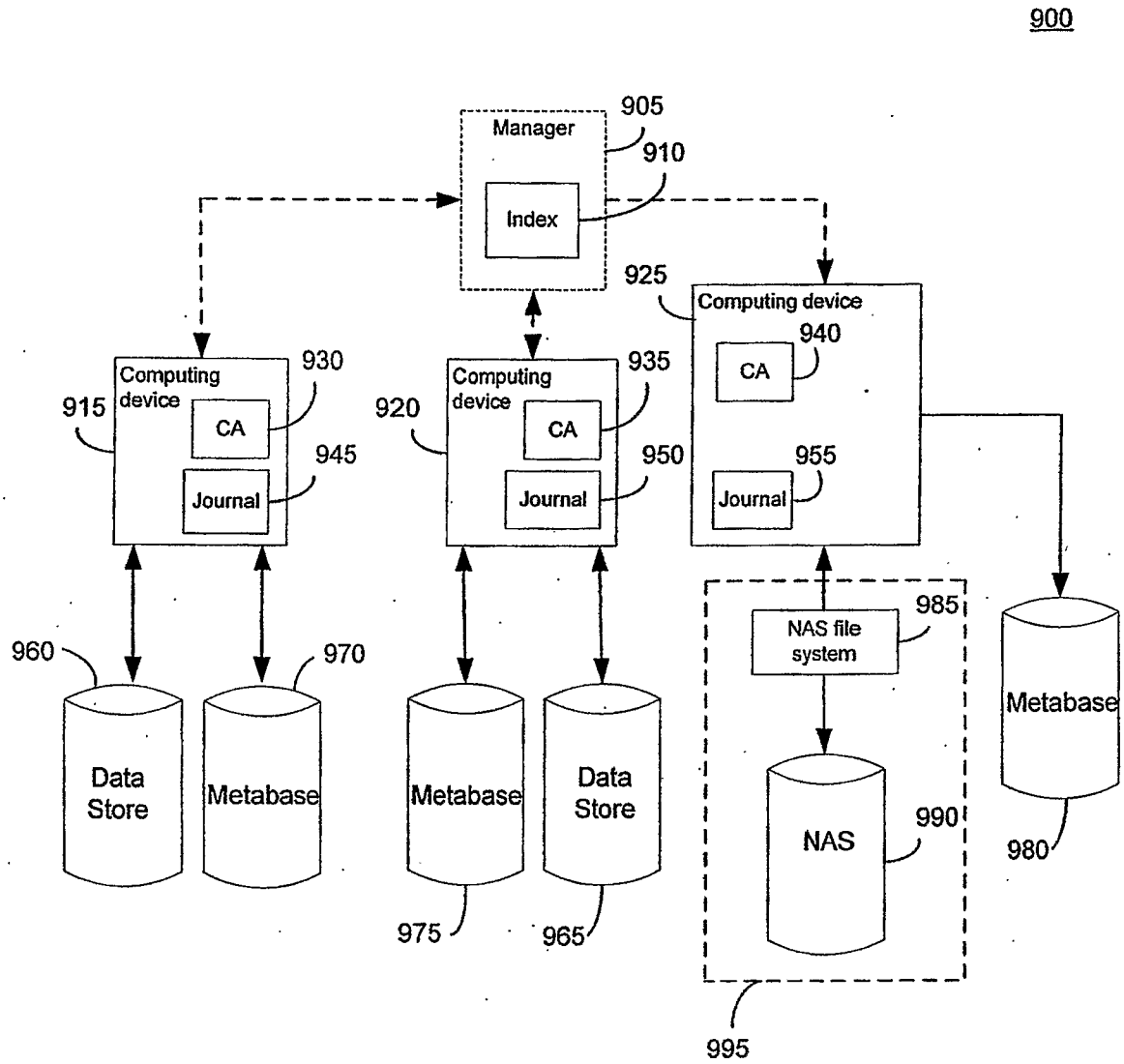
900
Manager
905
910
Index
925
Computing device
940
CA
955
Journal
915
Computing device
930
CA
945
Journal
920
Computing device
935
CA
950
Journal
960
Data Store
970
Metabase
975
Metabase
965
Data Store
985
NAS file system
990
NAS
995
Metabase
980

FIG. 9

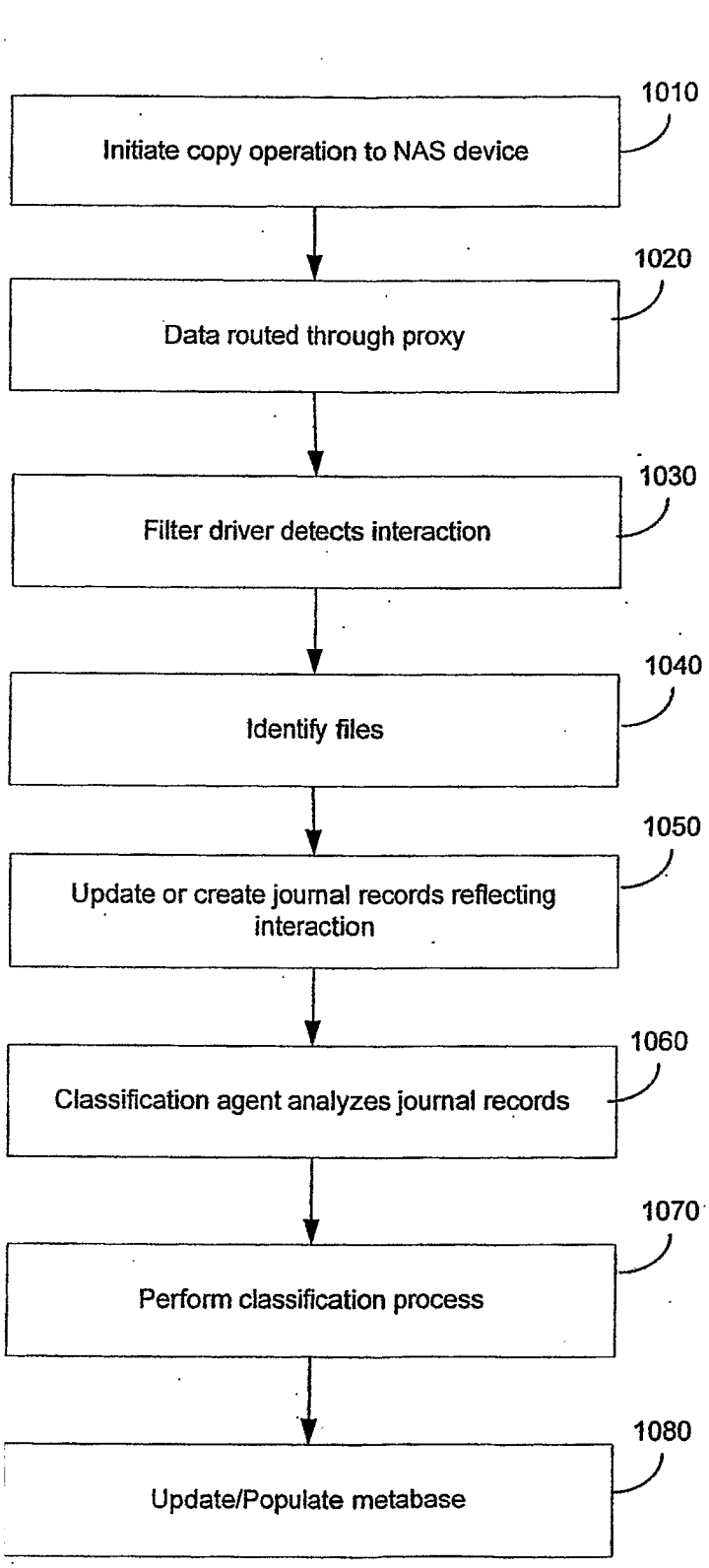
1000
1010
Initiate copy operation to NAS device
1020
Data routed through proxy
1030
Filter driver detects interaction
1040
Identify files
1050
Update or create journal records reflecting interaction
1060
Classification agent analyzes journal records
1070
Perform classification process
1080
Update/Populate metabase

FIG. 10

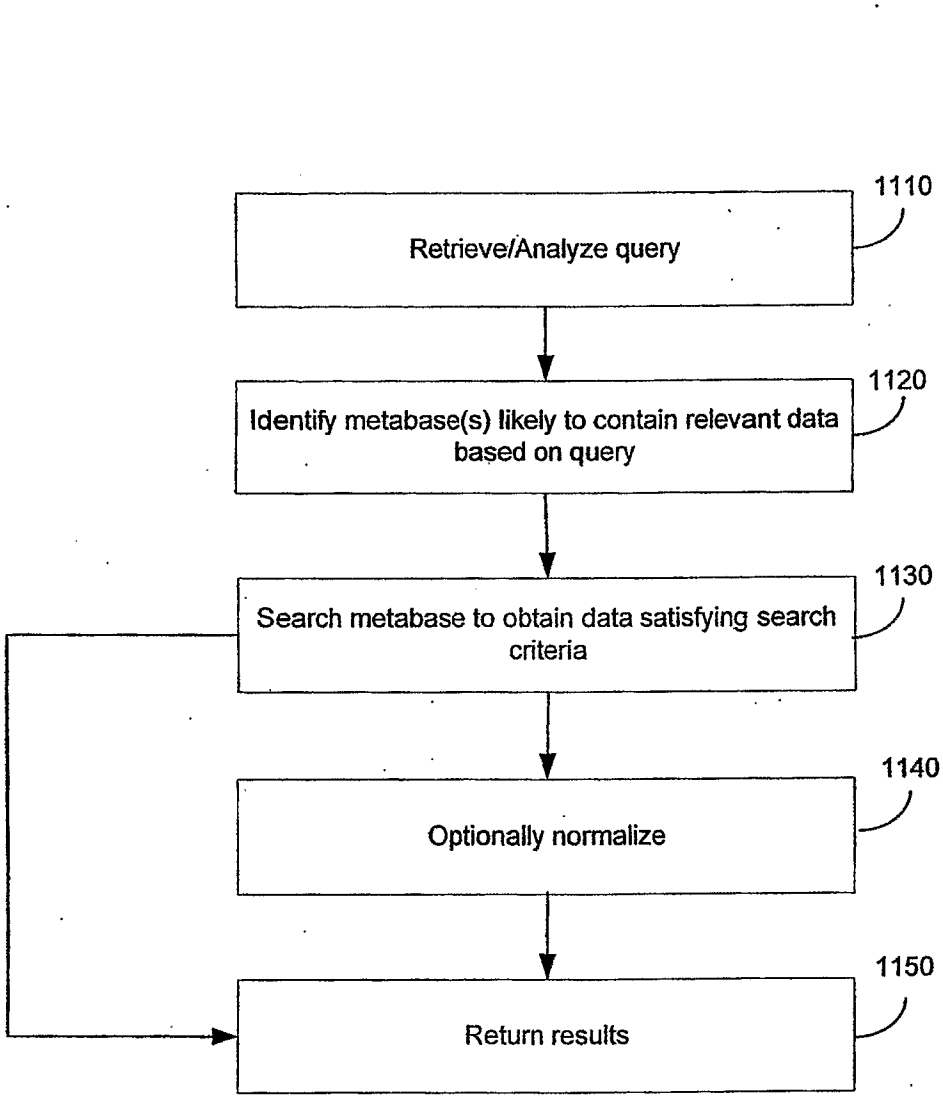
1100
1110
Retrieve/Analyze query
1120
Identify metabase(s) likely to contain relevant data based on query
1130
Search metabase to obtain data satisfying search criteria
1140
Optionally normalize
1150
Return results

FIG. 11

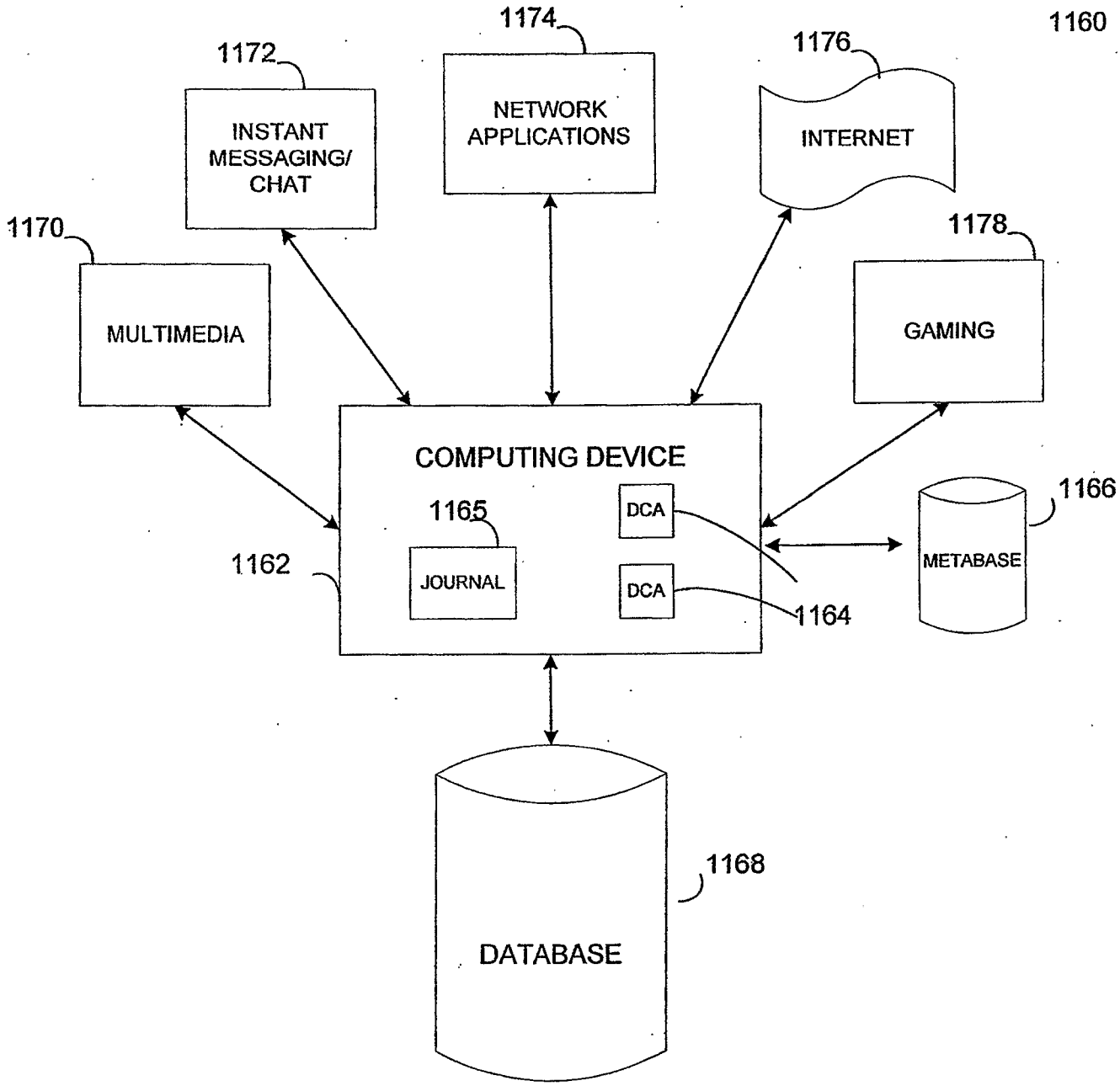
1160
1172
INSTANT MESSAGING/ CHAT
1174
NETWORK APPLICATIONS
1176
INTERNET
1170
MULTIMEDIA
1178
GAMING
COMPUTING DEVICE
1165
JOURNAL
DCA
DCA
1162
1164
1166
METABASE
1168
DATABASE

Figure 11a

1200

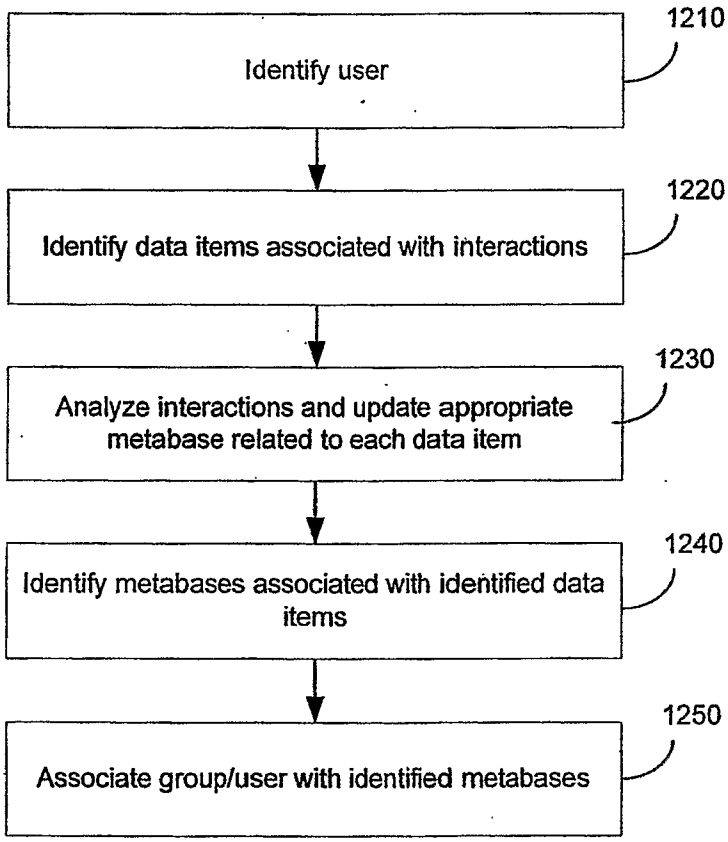
Identify user
1210
Identify data items associated with interactions
1220
Analyze interactions and update appropriate metabase related to each data item
1230
Identify metabases associated with identified data items
1240
Associate group/user with identified metabases
1250

FIG. 12

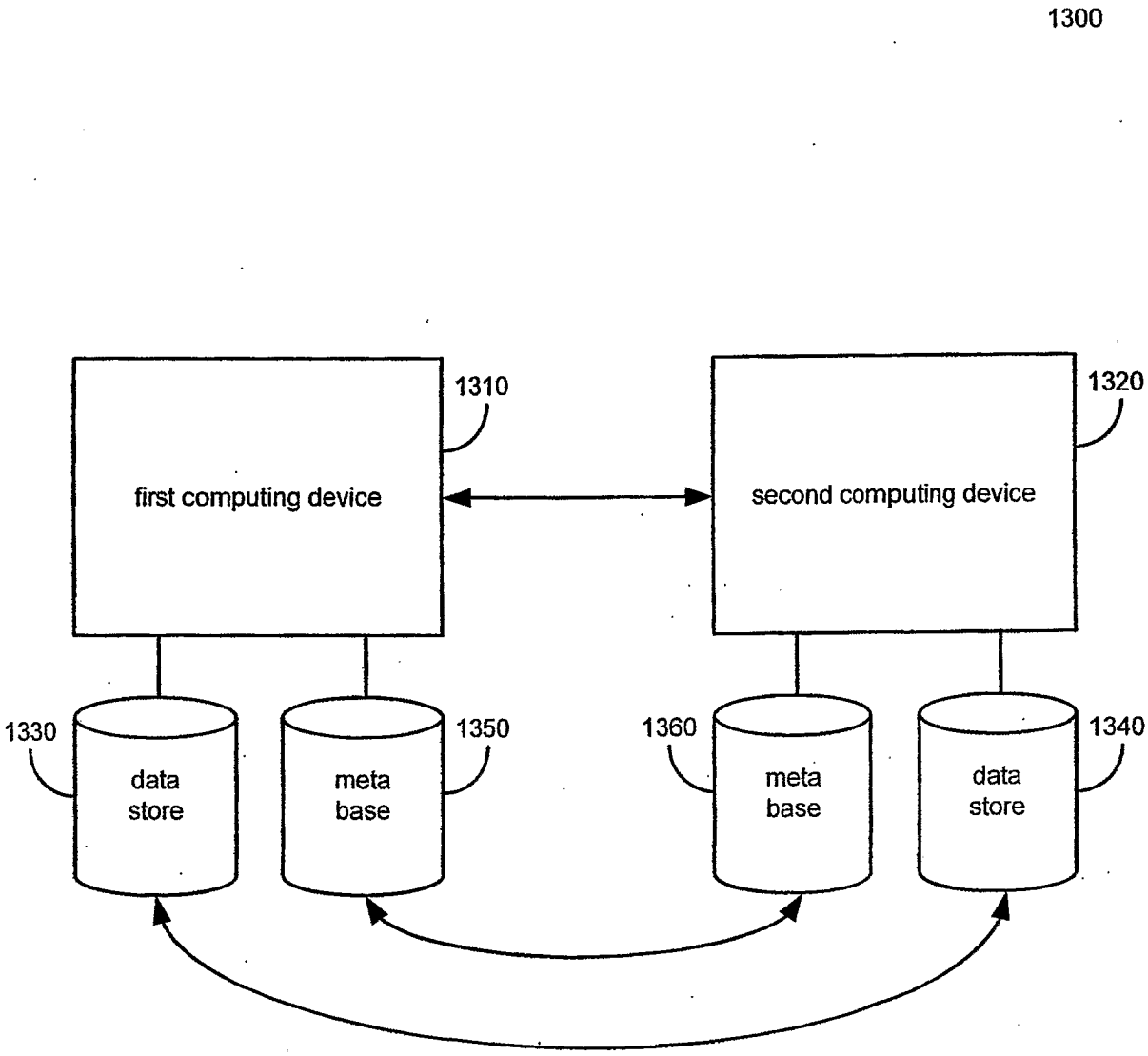
1300
1310
first computing device
1320
second computing device
1330
data store
1350
meta base
1360
meta base
1340
data store

FIG. 13

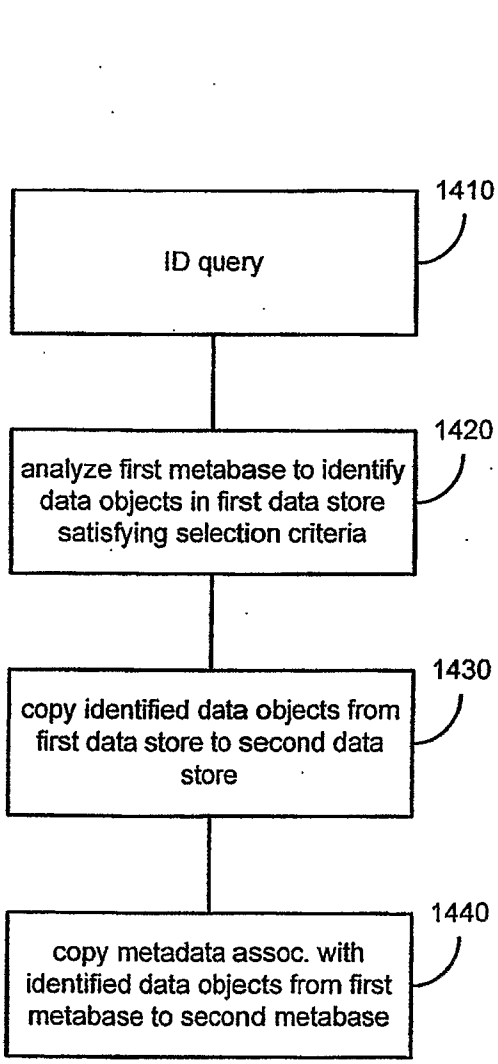
1400
1410
ID query
1420
analyze first metabase to identify data objects in first data store satisfying selection criteria
1430
copy identified data objects from first data store to second data store
1440
copy metadata assoc. with identified data objects from first metabase to second metabase

FIG. 14

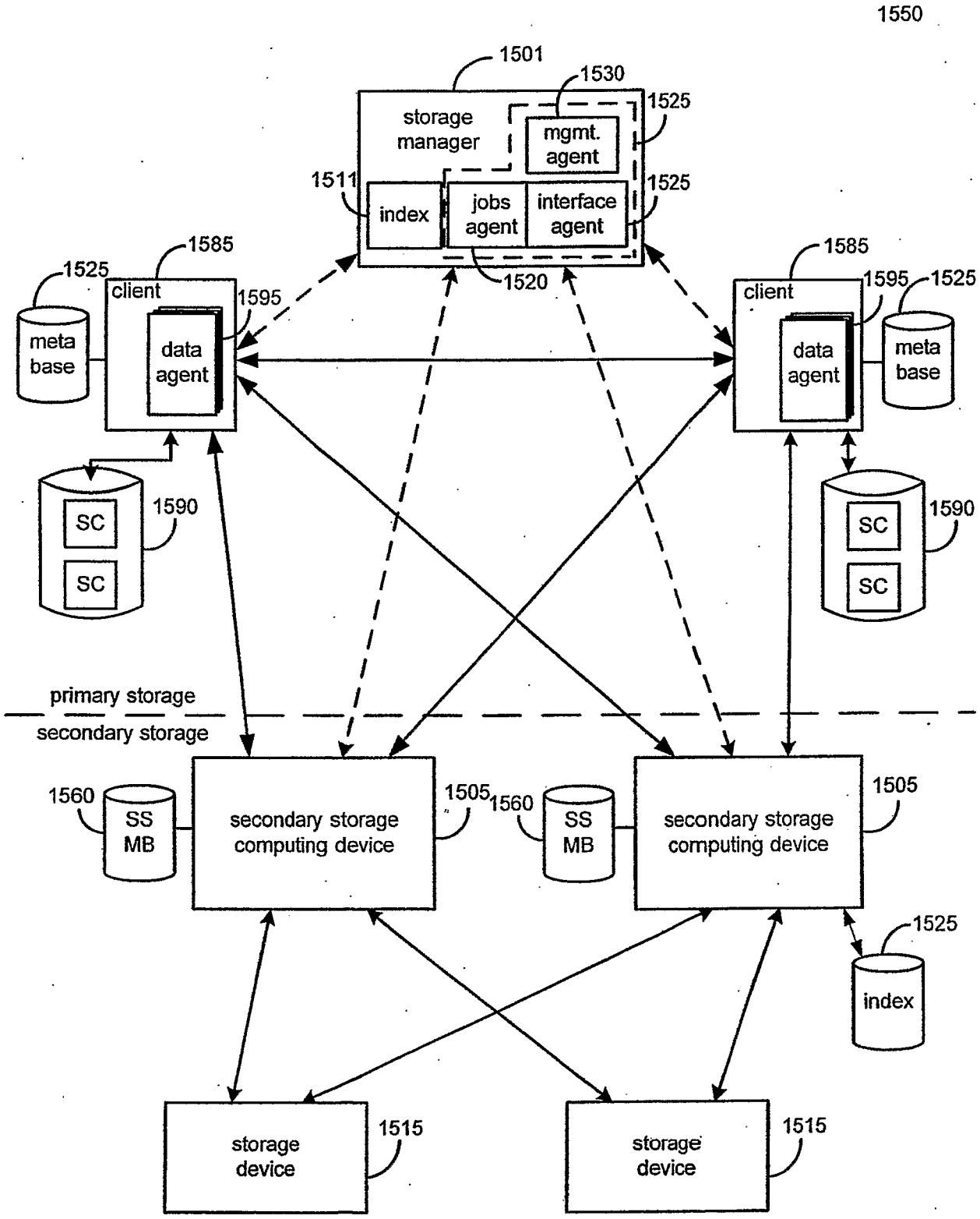
1550
1501
1530
storage manager
mgmt. agent
1525
1511
index
jobs agent
interface agent
1525
1520
1585
client
1595
1525
meta base
data agent
1585
client
1595
1525
data agent
meta base
SC
SC
1590
SC
SC
1590
primary storage
secondary storage
1560
SS MB
secondary storage computing device
1505
1560
SS MB
secondary storage computing device
1505
1525
index
storage device
1515
storage device
1515

FIG. 15

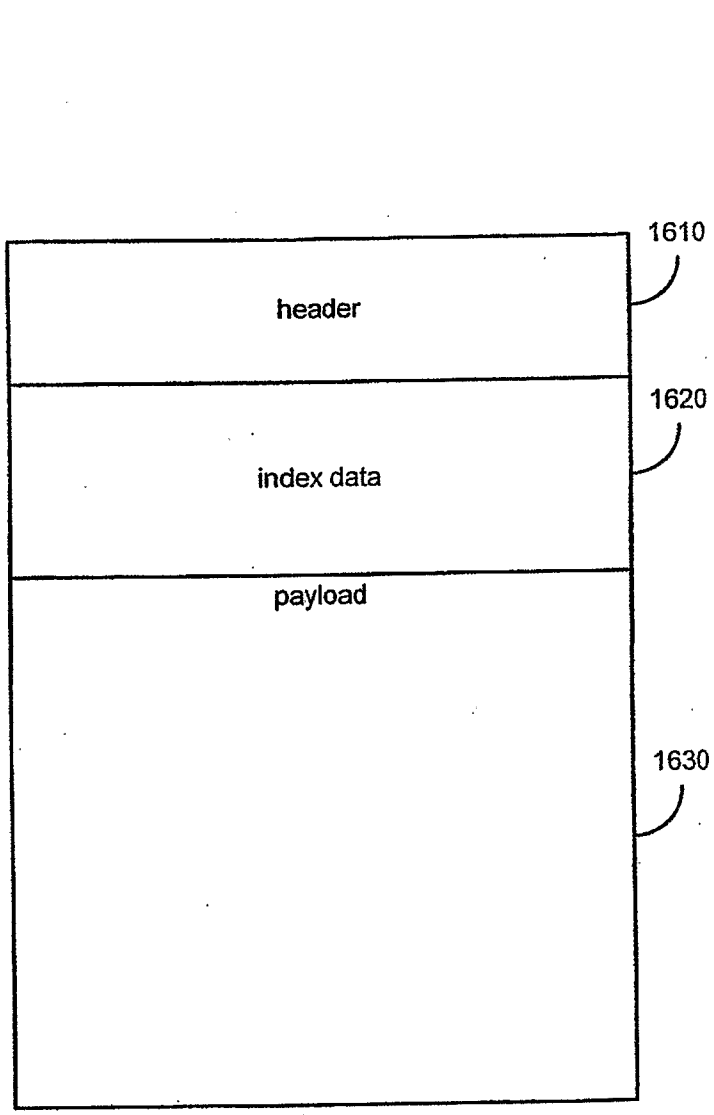

FIG. 16

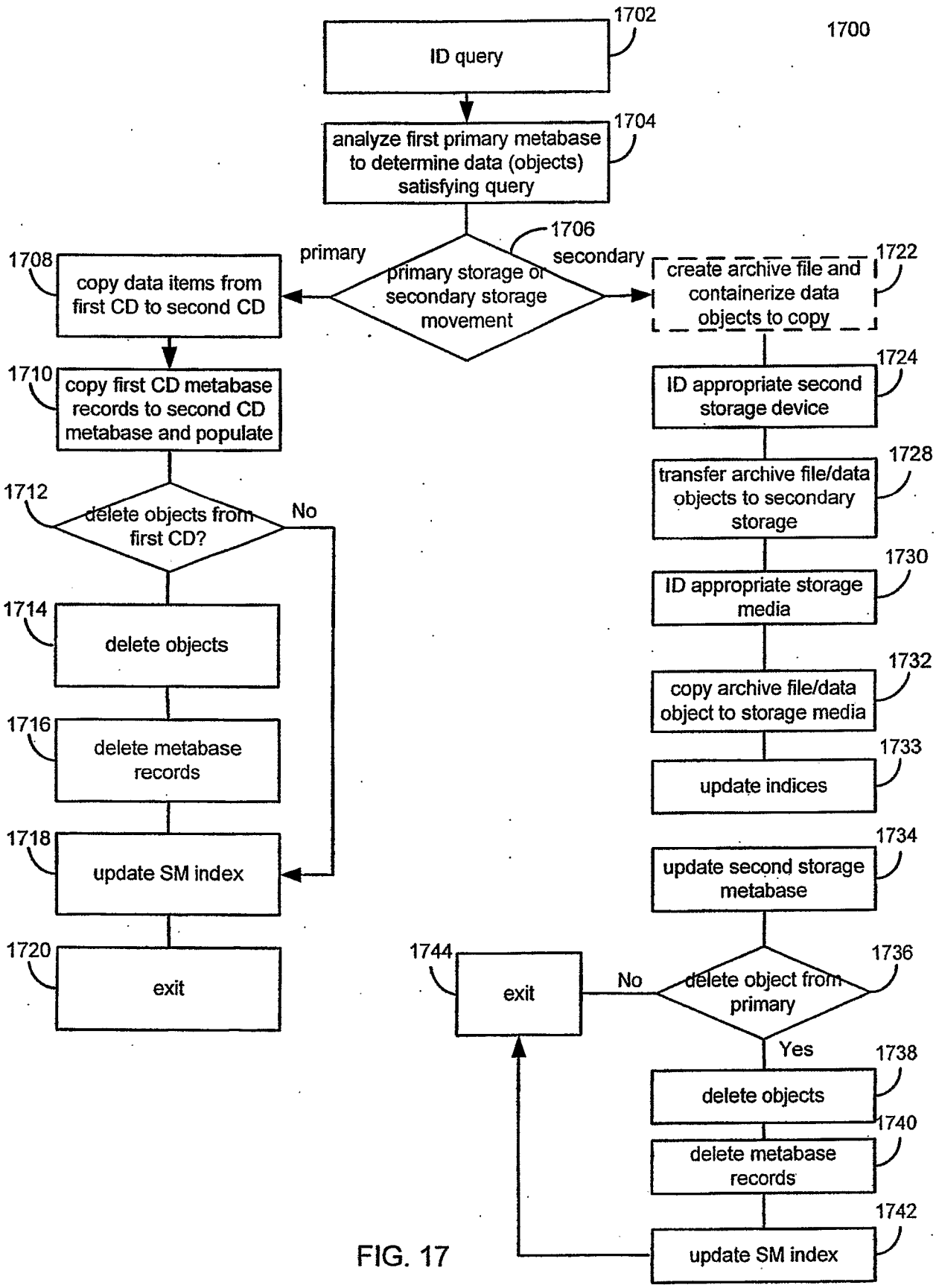
1700
1702
ID query
1704
analyze first primary metabase to determine data (objects) satisfying query
1706
primary storage or secondary storage movement
primary
secondary
1708
copy data items from first CD to second CD
1710
copy first CD metabase records to second CD metabase and populate
1712
delete objects from first CD?
No
1714
delete objects
1716
delete metabase records
1718
update SM index
1720
exit
1722
create archive file and containerize data objects to copy
1724
ID appropriate second storage device
1728
transfer archive file/data objects to secondary storage
1730
ID appropriate storage media
1732
copy archive file/data object to storage media
1733
update indices
1734
update second storage metabase
1736
delete object from primary
No
Yes
1738
delete objects
1740
delete metabase records
1742
update SM index
1744
exit

FIG. 17

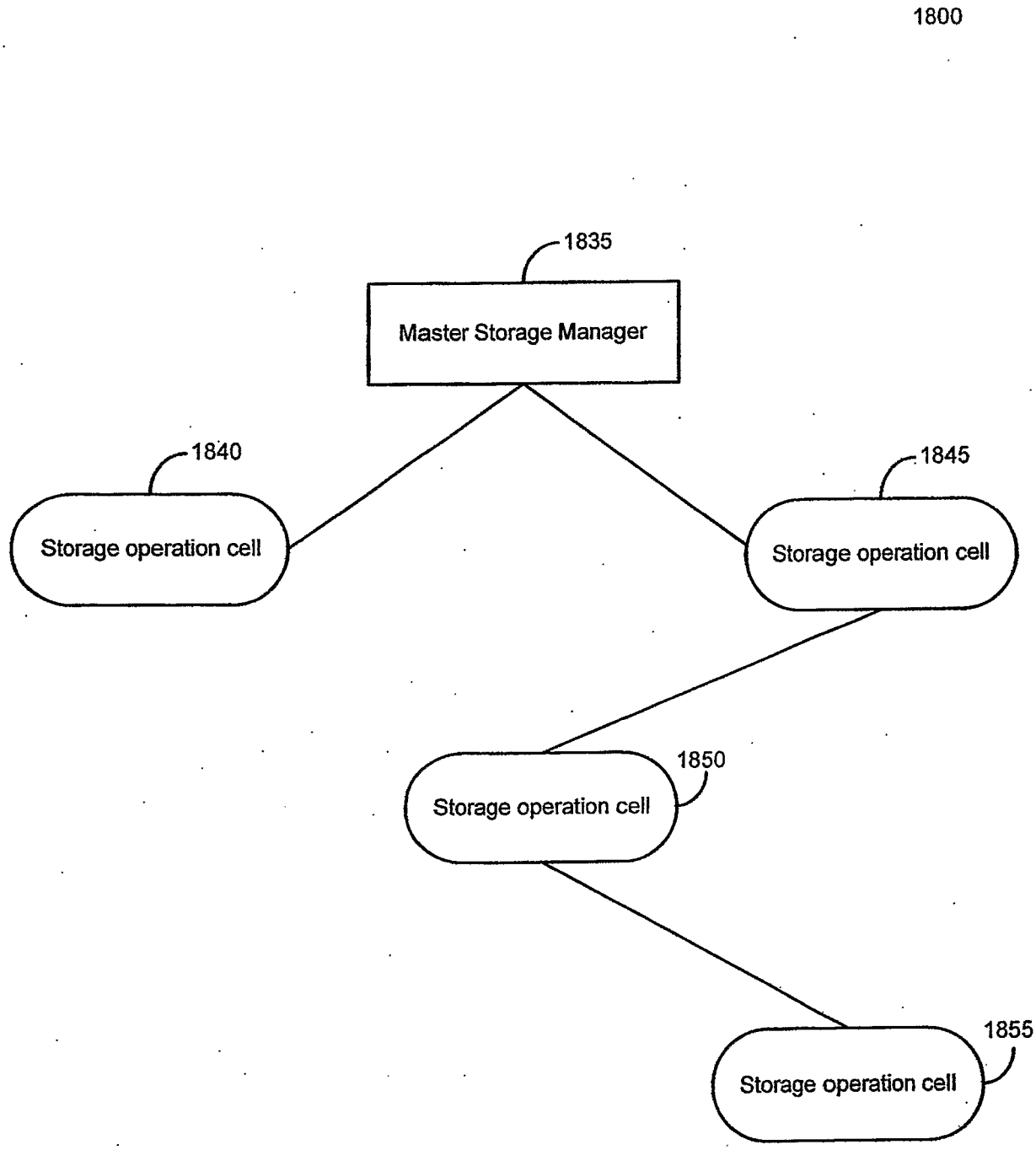
1800
1835
Master Storage Manager
1840
Storage operation cell
1845
Storage operation cell
1850
Storage operation cell
1855
Storage operation cell

FIG. 18

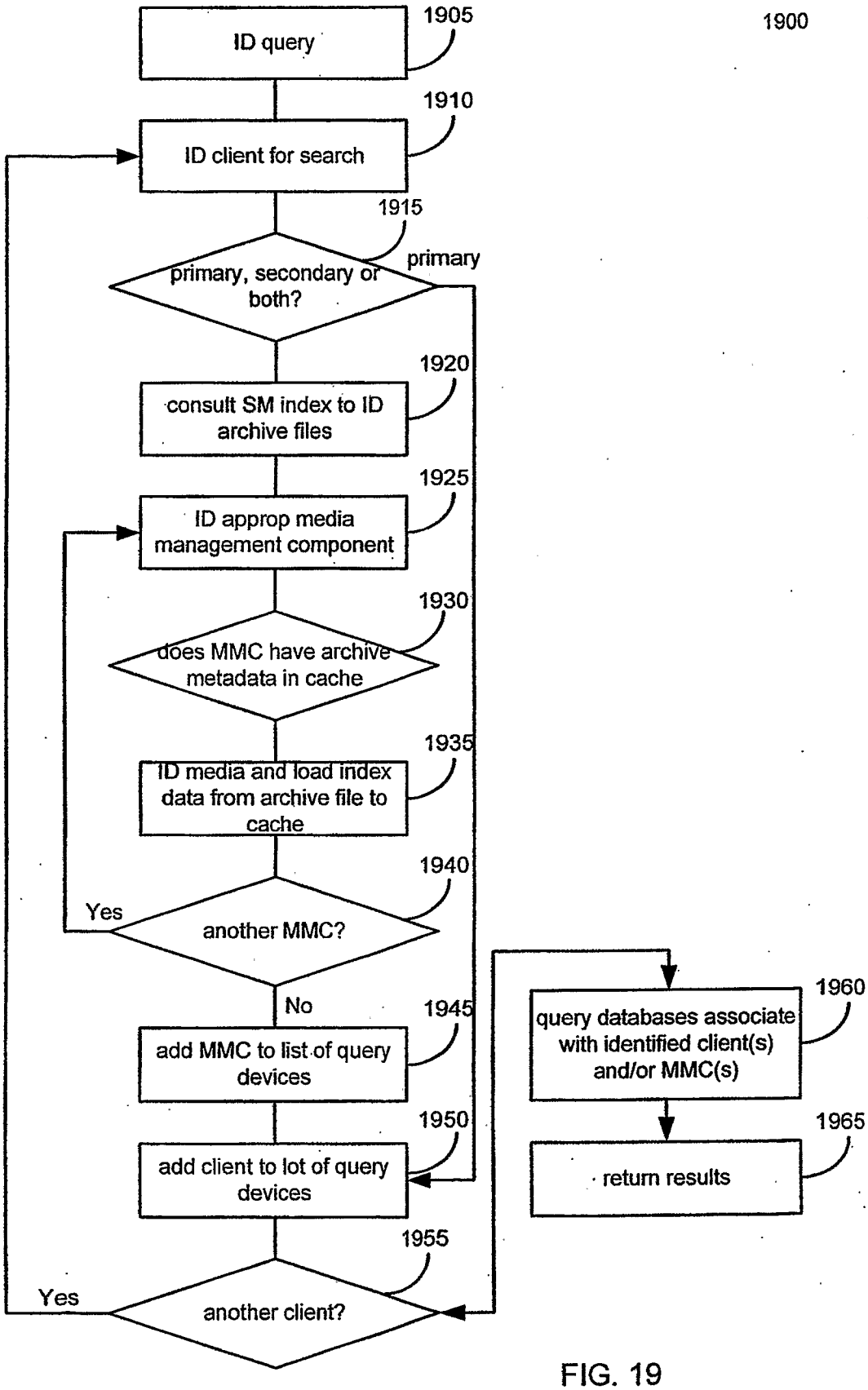
1900
ID query
1905
1910
ID client for search
1915
primary, secondary or both?
primary
1920
consult SM index to ID archive files
1925
ID approp media management component
1930
does MMC have archive metadata in cache
1935
ID media and load index data from archive file to cache
1940
Yes
another MMC?
No
1945
add MMC to list of query devices
1950
add client to lot of query devices
1955
Yes
another client?
1960
query databases associate with identified client(s) and/or MMC(s)
1965
return results

FIG. 19

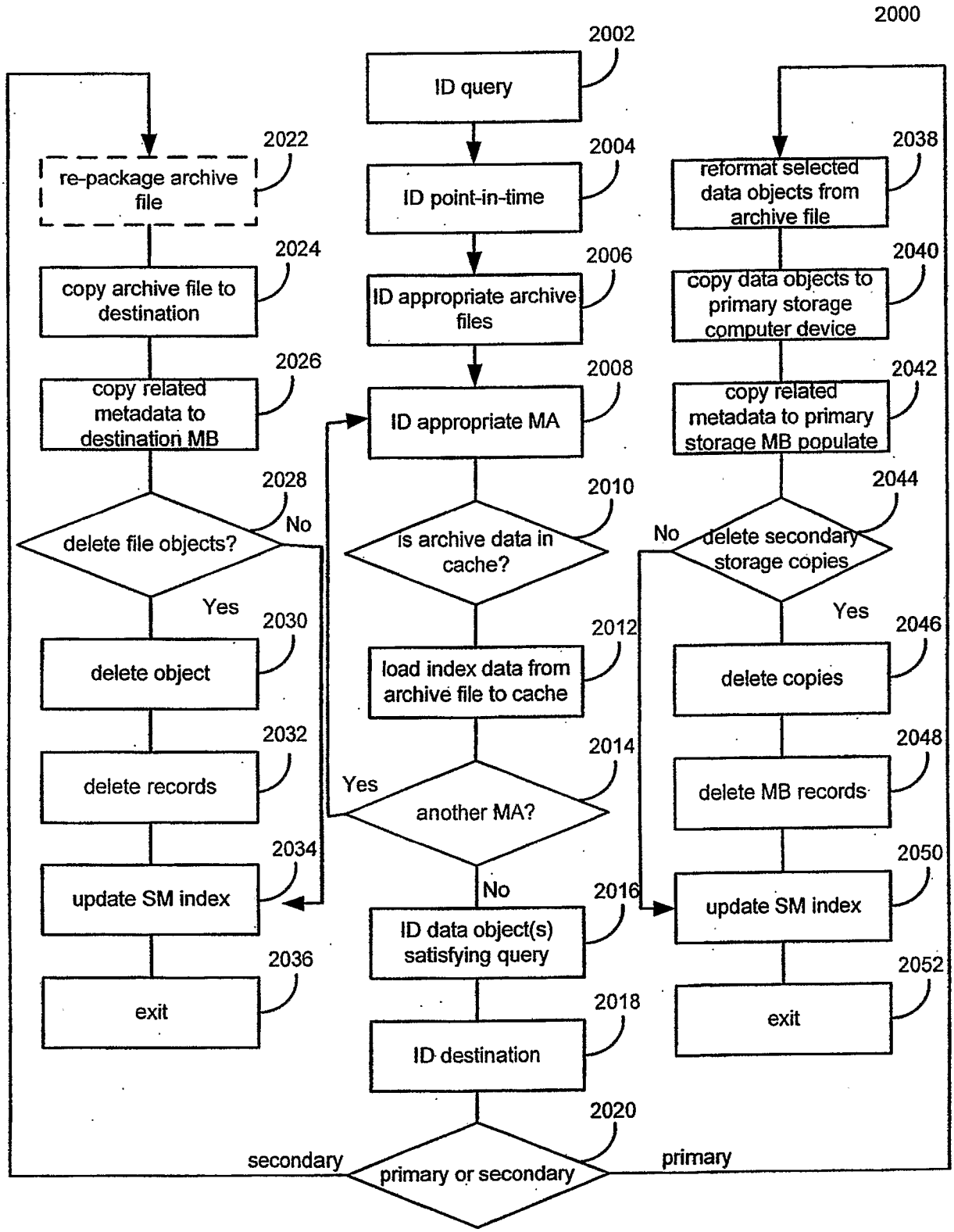
2000
2002
ID query
2004
ID point-in-time
2006
ID appropriate archive files
2008
ID appropriate MA
2010
is archive data in cache?
No
2012
load index data from archive file to cache
2014
another MA?
Yes
No
2016
ID data object(s) satisfying query
2018
ID destination
2020
primary or secondary
secondary
primary
2022
re-package archive file
2024
copy archive file to destination
2026
copy related metadata to destination MB
2028
delete file objects?
No
Yes
2030
delete object
2032
delete records
2034
update SM index
2036
exit
2038
reformat selected data objects from archive file
2040
copy data objects to primary storage computer device
2042
copy related metadata to primary storage MB populate
2044
delete secondary storage copies
No
Yes
2046
delete copies
2048
delete MB records
2050
update SM index
2052
exit

FIG. 20