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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
18 July 2013 (18.07.2013)

WIPO | PCT

(10) International Publication Number
WO 2013/106590 A2

(51) International Patent Classification:
G06F 19/00 (2011.01) *A61B 5/00* (2006.01)

(21) International Application Number:
PCT/US2013/021058

(22) International Filing Date:
10 January 2013 (10.01.2013)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
61/585,137 10 January 2012 (10.01.2012) US

(71) Applicant: **MEMEO, INC.** [US/US]; 469 El Camino Real, Suite 201, Santa Clara, CA 95050 (US).

(72) Inventors; and
(71) Applicants : **SAVAGE, Bret** [US/US]; 469 El Camino Real, Suite 201, Santa Clara, CA 95050 (US). **MARSHALL, Casey** [US/US]; 469 El Camino Real, Suite 201, Santa Clara, CA 95050 (US). **STUTCHMAN, Geoffrey** [US/US]; 469 El Camino Real, Suite 201, Santa Clara, CA 95050 (US). **ELTHERINGTON, Ross** [US/US]; 469 El Camino Real, Suite 201, Santa Clara, CA 95050 (US). **OWENS, Steve** [US/US]; 469 El Camino Real, Suite 201, Santa Clara, CA 95050 (US). **NORTHUP, George** [US/US]; 469 El Camino Real, Suite 201, Santa Clara, CA 95050 (US).

(74) Agent: **GREGORY, Richard, L.**; Gregory & Sawrie LLP, 2018 Bissonnet Street, Houston, TX 77005 (US).

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

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV,

[Continued on next page]

(54) Title: CLOUD-BASED DISTRIBUTED DATA SYSTEM

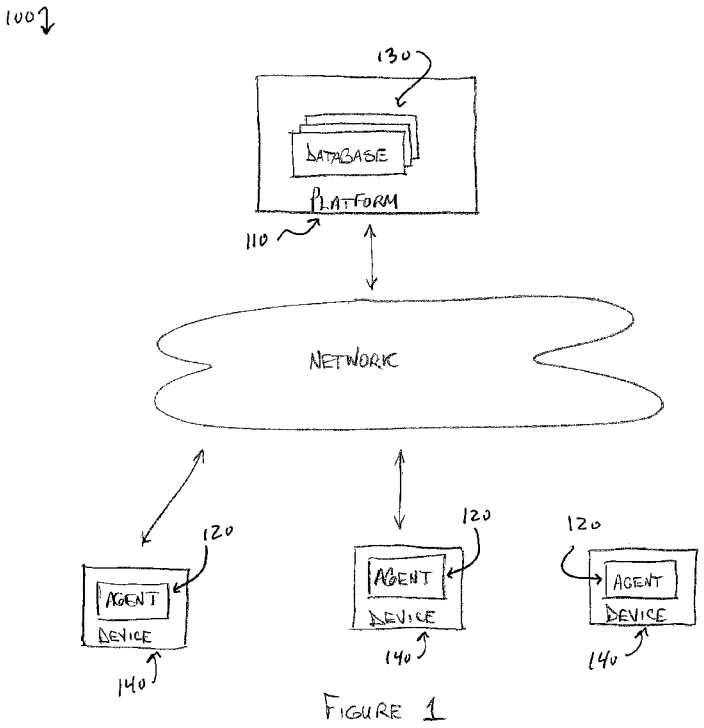


FIGURE 1

(57) Abstract: Embodiments of a distributed data management system include a platform comprising a processor coupled to databases. A grid including agents is coupled to the platform, and each agent is an agent of the platform independently running on a client device. The agents generate and provide to the platform metadata that corresponds to the content of memory accessible by the client devices corresponding to the agents. The agents generate the metadata by hashing fragments of the content. The platform uses the metadata instead of the content to generate and assign to the agents tasks including tasks controlling at least one of storing, transferring and processing of the content. A task is a processing operation performed on content accessible by the agent responsible for the task.

WO 2013/106590 A2

MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— *without international search report and to be republished upon receipt of that report (Rule 48.2(g))*

CLOUD-BASED DISTRIBUTED DATA SYSTEM

RELATED APPLICATION

This application claims the benefit of United States (US) Patent Application Number 61/585,137, filed January 10, 2012.

TECHNICAL FIELD

Embodiments described herein relate to data processing and, more particularly, to distributed storage and processing.

BACKGROUND

Cloud-based services are rapidly expanding and customers are expanding beyond local customer provided equipment (CPE) solutions to include cloud-based services to increase productivity, lower support costs, and reduce up-front investments. Hybrid solutions that combine both CPE with cloud solutions are being adopted rapidly given that a local/cloud approach leverages existing investments and has less risk than a complete migration to the cloud. Conventional cloud services include personal synchronization and sharing services, online backup, and large file transfer services. While these conventional cloud services have focused on online storage offerings, the ever increasing number of connected computers, devices and web sites, has meant that users of these services continue to experience difficultly synchronizing and accessing their latest files across multiple devices or via the web. Further, corporate users require relatively high levels of security to protect their business data, and high-profile security

breaches and long term viability of cloud storage providers has added to the issues associated with migrating storage and services to the cloud. Consequently, there is a need for a cloud service having a streamlined approach to synchronization and access of files across multiple devices, while also focusing on local data storage and peer-to-peer transfer, as well as relatively higher levels of security.

INCORPORATION BY REFERENCE

Each patent, patent application, and/or publication mentioned in this specification is herein incorporated by reference in its entirety to the same extent as if each individual patent, patent application, and/or publication was specifically and individually indicated to be incorporated by reference.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a block diagram of the system including the platform and agents, under an embodiment.

Figure 2 is a block diagram of the system including the platform coupled to a group of independent agents and a group of networked agents, under an embodiment.

Figure 3 is a block diagram of the system including the platform and numerous agents that are included in an organization, under an embodiment.

Figure 4A is a block diagram of an example system including the platform and an agent, under an embodiment.

Figure 4B is another block diagram of an example system including the platform and an agent, under an embodiment.

Figure 5 is a block diagram of an example involving file synchronization between the platform and agents, under an embodiment.

Figure 6 is an example flow diagram involving an agent, the platform, and databases, under an embodiment.

Figure 7 is an example flow diagram involving use of a sync goal to scan and sync a library, under an embodiment.

DETAILED DESCRIPTION

Data management systems and methods are described that include a cloud-based platform or engine coupled to a system of agents or folders hosted on client devices. The platform of an embodiment does not store actual data but instead makes use of metadata provided by the agents to track a location of all data in the system and manage the distributed storage, movement and processing of the actual data among the agents. In so doing, the system of an embodiment pools networked storage into "virtual clusters" using local storage at the agents. The agents collectively monitor, store, and transfer or move data, and perform data processing operations as directed by the platform, as described in detail herein. The agents of an embodiment include agents hosted on or coupled to processor-based devices (e.g., personal computers (PCs), tablet computers, server computers, Network-Attached Storage (NAS) devices, Apple computers, mobile devices, iOS devices, Android devices, etc.), agents hosted on devices of a local area network (LAN), agents hosted on devices of a wide area network (WAN), agents hosted on mobile devices, and agents hosted on cloud-based devices (e.g., servers, storage devices, etc.), but are not so limited.

Although the detailed description herein contains many specifics for the purposes of illustration, anyone of ordinary skill in the art will appreciate that many variations and alterations to the following details are within the scope of the embodiments described herein. Thus, the following illustrative embodiments are set forth without any loss of generality to, and without imposing limitations upon, any claimed invention.

Figure 1 is a block diagram of a system including the platform 110 and agents 120, under an embodiment. In this embodiment, the platform is coupled to and/or includes numerous databases 130, as described in detail herein. A number of client devices 140, each of which include or host an agent, are coupled to the platform and the databases via a network coupling and make use of the databases to receive and execute work tasks as directed by the platform.

Figure 2 is a block diagram of a system including the platform coupled to a group of independent agents and a group of peer agents, under an embodiment. The platform of this embodiment is coupled to and/or includes numerous databases, as described in detail herein. The client devices are coupled to the platform via network couplings. A first group of client devices 210 includes a number of client devices, each of which include or host an agent and is independently coupled to the platform via a network coupling. A second group of client devices 220 includes a number of client devices that each include an agent and independently couple to the platform and, additionally, form a peer network (e.g., LAN, WAN, etc.) in which peer-to-peer communications 230 are supported between the agents.

Figure 3 is a block diagram of a system including the platform and numerous peer agents that are included in an organization, under an embodiment. The platform is coupled to and/or includes numerous databases, as described in detail herein. The organization includes a number of groups (e.g., group 1 to group M, where M is any number), and each group includes some number of users (e.g., user 1 to use N, where N is any number). Each user of the organization includes some number of devices (e.g., device 1 to device P, where P is any number), and each device includes or hosts an agent that is coupled to the platform via a network coupling. The organization also includes an administrator console that couples to the platform and enables an administrator to access the devices according to the hierarchy. The platform includes controls or rules that control access to the data according to the organization, and the console provides access to the controls or rules and thereby allows the administrator to setup and maintain (e.g., add, delete, modify, etc.) the rules.

Under this organizational example, the agents are directed by and execute instructions received from the platform, and each agent is hosted on a device that is registered to a user. Each user is a subset of a group, and each group is a subset of an organization. The agents on each device index the contents of the memory included on or accessible by the corresponding device, and provide the metadata of the memory contents to the platform. The console provides an administrator control over rules that control

access to the data according to the organization. For example, an administrator can generate a rule specifying that an organization cannot include any file that exceeds a size of 10 MB. An administrator can generate another rule specifying that Group 1 cannot include any music file or video file. Yet another example rule specifies that Group 2 cannot include any file marked confidential.

Figure 4A is a block diagram of an example system including the platform and an agent, under an embodiment. **Figure 4B** is another block diagram of an example system including the platform and an agent, under an embodiment. Figures 4A and 4B are collectively referred to herein as "Figure 4." The platform uses knowledge of the data accessible by each agent gained via metadata to create or generate work and assign the work to agents. The agents in turn perform work assigned to them by the platform and provide to the platform information of their data and work performed. Because the agents of an embodiment perform work as directed by the platform, they are not applications and not users.

The platform includes and/or is coupled to a variety of databases. For example, the platform includes an agent database that is a list of all agents available to the platform and the relevant information of each agent. In addition to the agent database, the platform comprises other databases for use by the agents in exchanging information in support of tracking a location of all data in the system and managing the distributed storage, movement and processing of the actual data among the agents. The databases include but are not limited to the agent database, as described above, a libraries database, a goals database, a work database. Each agent can have one or more libraries that represent their local disk or storage, and the libraries database includes a list of the storage available to a corresponding agent, where the storage includes but is not limited to any device to which the agent has been granted read and/or write access (e.g., a disk or disk drive, network mount point, hard drive, flash drive, storage service, etc.). The platform and each agent comprise a libraries database that includes a list of libraries of the system and each corresponding agent of the system that can access each library.

The goals database includes a list of goals which are collections of libraries on which a particular task is to be performed (e.g., synchronized). The goals are used by agents to perform tasks, where goals correspond to one or more libraries and are used to carry out tasks or work on the contents of the library. Furthermore, the platform generates a work database for each agent, and puts the work tasks that are to be accomplished by the corresponding agent in the work database. The work database includes a description of each task that is to be performed by an agent along with all information needed by the agent to perform the work. The databases are described in detail herein.

The platform of an embodiment is a universal synchronization (sync) engine that forms a grid through couplings among the platform and numerous agents, where the platform acts as the master controller for the agents. The platform includes or is coupled to one or more databases in the cloud, and the databases comprise information received from the agents that includes data (e.g., metadata) on the location and state of all data of the grid for which the platform has responsibility. The agents are hosted or installed on particular devices or computers and, as such, are independently functioning entities that can communicate with and work in cooperation with the platform and with other agents to form the grid, where the components of the grid work cooperatively to achieve particular tasks as assigned by the platform.

In operation, an agent is hosted or installed on a device, and the agent functions to perform the work or tasks assigned to it by the platform. The platform knows the identity and location of all agents with which it is associated, and the data to which each agent has access. The agents are not applications, but are instead agents of the platform that communicate via encrypted channels with both the platform and with other agents to perform work as directed by the platform. Generally, the platform generates and includes goals for which it is responsible, and the platform assigns work or tasks to the agents and manages the agents to accomplish the goals. Further, the agents require no knowledge of the objectives that correspond to the tasks assigned the respective agent or any other agent.

The platform of an embodiment does not store actual data in the databases of an embodiment but, instead, stores metadata corresponding to the data for which it is responsible. Consequently, as the platform stores only metadata, the agents serve as distributed data storage entities at which actual data is stored. The platform uses the metadata for a bit-level understanding of the data stored at the agents, thereby providing more than a file-level understanding of the data. The metadata is received from each agent and is generated by the agent from the data for which each respective agent is responsible. In generating metadata, the agents scan files as requested by the platform. For each file, the scanning involves generating fragments or components of data from the data forming the file by splitting the file into variable size fragments referred to herein as blobs. The blobs are generated using a data fingerprinting algorithm that generates portions of data having variable sizes that are between a pre-specified minimum length and maximum length, as described in detail herein. As the embodiments described herein generate blobs from the contents of a file, and the blobs represent the contents of the file, the terms "file(s)" and "blob(s)" are used interchangeably in the description herein.

When generating blobs, a description of the blob is generated that includes a value of the hash at the point where the blob was broken or separated from the remaining portion of the file, a hash of the complete contents of the blob, an offset value based on the break point for the blob, and a size (e.g., length) of the blob. The hash of the complete contents of the blob (blob hash, or "bash") serves as a unique identifier for the blob. Furthermore, a complete list of blobs making up the contents of a file is hashed (cash) to generate a unique identifier for the entire file contents, as is a hash of the combination of the contents hash and the name hash (fash). The hashes of an embodiment further include a hash of the file name and path of a file (pash), and a hash of the file metadata (e.g., file name, size, date, location, etc.) (mash).

These various hashes, all of which are described in detail herein, are generated by each agent for each data file for which the respective agent is responsible, and delivered to the platform. The platform generates a record for each file to include the information of these various hashes. The platform uses the information or state of the data reported to

it by the agents and included in these records to determine the state of the data and to determine which agents have versions of data that are out of date because of modifications to the data, and to reconcile file versions among agents. The platform then issues work to agents needing to update one or more files to a most recent version of the file, as described in detail herein.

Using the metadata, therefore, the platform maintains in the cloud a master index of the data, and uses goals to read master indexes or libraries of the agents to determine the locations of the various pieces of data relating to the respective goals, identify any inconsistencies or discrepancies in the data, and issue or generate work to eliminate any identified inconsistencies or discrepancies. This is in contrast to peer-to-peer systems because agents of an embodiment, instead of being anonymous, are known to and directed by the platform and only exchange content with other agents controlled by and having an identity established at the platform.

The platform finds efficiency in transferring as little data as possible as directly as possible between peers. Also, the embodiments herein use the knowledge gathered by the platform of the location of all data along with cooperative processing among agents to only move data that needs to be moved and to process data at the location (e.g., agent) of the data, thereby providing relatively greater bandwidth than any one agent has when working independently. So, the platform of an embodiment provides data ubiquity by providing efficient storage, delivery, and global data mobility.

The system provides hashing and block-level transfer of data among devices through use of the blobs. The block-level transfer supports efficient synchronizing of files when data changes because there is only a need to update across devices the blocks of data that changed. The block-level transfer further supports non-sequential file transfers because once the data is hashed to form the blobs, the blobs are moved and then reassembled, thereby eliminating the need for sequential file transfer. Similarly, embodiments support a reduction in data upload/download times because the blobs can be simultaneously or near-simultaneously transferred from one or more other devices to any device where they are needed. Moreover, when a file is needed by a particular

device on the grid, and it is present at more than one other device on the grid, different blocks can be simultaneously transferred from these multiple devices in order to place the file at the location (e.g., agent) where it is currently needed.

The platform distributes work to agents in a grid and directs the work by generating tasks or work items for particular agents and delivering the work items to the agents. A work item is delivered to an agent along with conditions of completion for that work item, but the embodiment is not so limited. The agents of an embodiment are flexible clients that execute the tasks or work items assigned to them by the platform.

As an example, a work item assigned to the agent directs the agent to scan or monitor a directory or folder on a memory device accessible by the agent, and continuously report any changes to any file in the directory or folder. This work item is used by the platform to initially receive the metadata of files corresponding to the agent upon registration of the agent with the platform, and is also used to continuously update that metadata in response to changes made to the content of the files. Therefore, in response to any change to any file, the scan task causes the agent to report any changes in blobs or metadata of the file to the platform which subsequently assigns work to one or more other agents as appropriate to the reported changes. Furthermore, the platform of an embodiment is used to distribute logic for new functionality to one or more agents. Therefore, if the platform needs the agents to provide new functionality, the logic to enable that functionality can be delivered to the appropriate agents by the platform.

The tasks or work items of an embodiment that are generated by the platform and assigned to the respective agents for execution include scanning, deleting, writing, and uploading. An agent performs the scan task by monitoring a file system or directory recursively and reporting to the platform any changes (e.g., modifications to a file, additions to a file, deletions to a file, renaming of a file, etc.) to any files. The reporting of the change involves placing the modified file in the local database, and the metadata of the local database is subsequently provided to the platform.

The write task of an embodiment involves the writing of blobs, as described in detail herein, but is not so limited. The agent performs the write task by copying one or

more blobs of a file from a first location (e.g., source device) to a second location (e.g., destination device). Each of the first location and the second location include locations on a storage device (e.g., from a first location on a hard disk of a computer to a second location on the hard disk of the computer), locations on different storage devices in a same domain, and locations on storage device in two or more different domains (e.g., from a file system of an agent to a cloud-based storage device, from a file system of a first agent to a file system of a second agent via peer-to-peer communications, etc.). The source of an embodiment therefore can be one of a file system of any agent on the grid and a cloud-based or network-based storage device.

A work item is delivered to an agent along with conditions of completion for that work item, as described above. The conditions of completion associated with a write work item include retrieving the blobs or component necessary to complete the write, and that the blob or component being overwritten is the correct file. If the condition is not met that the blob being overwritten is the correct file, then a report of this is made to the platform.

The agent performs the upload task by copying a blob from an agent to a cloud-based storage device. Consequently, when a new file is placed into the file system of an agent, the agent reports the addition of the file to the platform, and the platform assigns work to the agent to copy the file blobs to the cloud-based storage device. In response, the agent determines if the new file exists in the cloud-based database and, if not, copies the file to the cloud-based database.

The work items of an embodiment include three phases that include being queued, pending, and completed (successfully or unsuccessfully). The platform tracks the phases of work items assigned to each agent in order to track the state of files corresponding to the respective agents.

In order to maximize efficiency of communications between the platform and the agents, some information is stored at both the platform and the agents. For example, the agent of an embodiment polls the platform to identify new work items assigned to that agent via the work database, and periodically receives and stores locally the assigned

work items of the work database. This reduces the amount of communication between the agent and the server required to download work items to the agent. Likewise, when a new file is identified on an agent, the new file is written to the local library database of the agent, and the local library database is periodically replicated in batches at the platform. This eliminates the requirement for the agent and/or the platform to perform a file transfer upon each occurrence of a new file. Furthermore, and as described in detail herein, the blobs of an embodiment support differential file transfer, and at least some portion of the information used to track blobs and their locations in the system is stored at both the platform and the agents.

Regarding the retrieval of information needed by an agent to perform a work task assigned by the platform, the agent includes a hierarchy to identify the location of information needed to complete the task but not presently in a file possessed by the agent. The hierarchy of an embodiment involves an agent first searching local memory, followed by the use of peer-to-peer communication to retrieve needed information that is not found locally but that is located in a file belonging to a peer. When the needed information is not found locally, and is not located with a peer, the agent retrieves the information from cloud-based storage (e.g., Amazon S3, etc.) or another remote storage entity.

When peer-to-peer communication is used to obtain needed files, the platform provides the agent with a list of peer agents that possess the needed files, but is not so limited. The list of peer agents provided by the platform is an ordered list, thereby enabling the agent to contact peer agents using the order of the list, but the list of alternative embodiments may not be ordered.

Referring to **Figure 4B**, the agent of an embodiment includes numerous threads or components that execute in parallel to provide the functionality of the agent. For example, a provider thread retrieves work items designated for an agent from the work item database of the platform, and provides or stores the work item database locally. A runner thread monitors the local work item database of the agent and individually pulls work items from the local work item database, marks the item as pending in the local

work item database, and provides the item to the appropriate task execution thread (e.g., a write task is provided to the write thread). When a task execution thread completes execution of a task received from the runner thread, it reports to the runner thread the status of the execution (e.g., successful completion, unsuccessful completion, reason for unsuccessful completion, etc.), and the runner thread reports this status information back to the local work item database. An update thread monitors the local work item database for items marked as completed, and reports the status information of the completed work items back to the platform. The platform in response updates platform databases as appropriate to the completed work by any agent.

The platform of an embodiment operates using metadata of the data corresponding to each agent coupled or connected to the platform as reported to it by the respective agents, as described in detail herein. To generate that metadata for the platform, the agents scan all files they have in the folder that is to be synchronized. The agents create or generate fragments or components of data from the data that makes up the file by splitting files into variable size fragments that are the blobs. The blobs are generated using a fingerprinting algorithm and identified by their blob hash (bash), referred to herein as a bash. More particularly, the agent creates the blob by running a fast-windowed checksum over a pre-specified number of bytes. An embodiment generates portions of data having variable sizes that are between a pre-specified minimum length and maximum length.

The procedure that breaks files into component parts, referred to herein as fingerprinting, involves running a hash algorithm over slices of the data, and the hash algorithm is set to identify a particular pattern of data (e.g., a sequence of twelve "0" bits; a sequence of three "0" bits; a sequence of six "1" bits, etc.). The "size" of each piece is determined through the setting of the hash string, because a continuous string of 12 bits is going to occur less often and thus produce longer data slices than a continuous string of three bits. The fingerprints determine boundaries of blocks of data within a file and these boundaries are used for determining common blocks or portions between two or more files.

A hash is generated for each byte in the file using a hash algorithm over a sliding window of a pre-specified size. For example, in an embodiment, the hash is generated for each byte in the file using a BUZ hash algorithm over a 48-byte window, where the result of application of the hash algorithm is a 64-bit number. When the hash has a particular value (e.g., corresponding to the lower 12 bits all being "0" bits), referred to as a match value, then a break is generated at that point in the file. The match value of an embodiment, for example, occurs when the lower 12 bits of the window are "0" bits.

Upon creation of a blob, a description of the blob is generated that includes a value of the hash at the point where the blob was broken or separated from the remaining portion of the file, a hash of the complete contents of the blob, an offset value based on the break point for the blob, and a size (e.g., length) of the blob. The hash function of an embodiment used to hash the complete contents of the blob is the skein cryptographic hash function (internal state size is 512 bits, producing an output of 160 bits), but the embodiment is not so limited. The hash of the complete contents of the blob serves as a unique identifier for the blob. A file is reported or described as a list of all blobs comprising the file, and this list of file hashes serves as a manifest of the corresponding file contents.

The blobs of an embodiment are stored at a central storage entity, but are not so limited. In so doing, the platform instructs each agent to transfer blobs not previously reported by any agent to a central cloud-based storage entity; as an example the central storage entity can include Amazon Simple Storage Service (S3) available from Amazon Web Services. Therefore, all blobs are stored locally on one or more agents and stored in centralized storage. Subsequently, when an agent reports a blob and the platform determines the blob is not present in the central storage, the platform instructs the agent to transfer that blob to the central storage if it has not been previously reported by another agent.

In addition to the bash described above, the system of an embodiment includes a number of other hashes for use in describing data in the system. For example, the complete list of blobs making up the contents of a file, as described above, is hashed to

generate a unique identifier for the complete file. This hash identifier of the complete file, referred to herein as the content hash (cash), allows a quick comparison to determine if two files contain identical content.

Another hash of an embodiment includes a hash of the combination of the contents hash and the name hash, referred to herein as a file hash (fash). The fash thus identifies contents of a file at a specific location.

The hashes of an embodiment further include a path hash and a metadata hash. The path hash or file path hash, referred to herein as the pash, is a hash of the file name and path of a file. The metadata hash, referred to herein as the mash, is a hash of the file metadata (e.g., file name, size, date, location, etc.). The hashes described herein are generated by the agents and delivered to the platform, but are not so limited.

Alternative embodiments use hashing to form blocks of data, as described herein, and store each block with version information. This enables versioning so that subsequently the versioning information is used to reassemble data to create older versions of a file.

Using the hashes of an embodiment a record is generated at the platform for each file, and every record for every file includes the all bashes of the file, as well as the cash, fash, pash, and mash. The library database of an embodiment includes these records for each file, but the embodiment is not limited the inclusion of these records in the library database. The platform uses the information or state of the data reported to it by the agents and included in these records to determine which agents have versions of data that are out of date because of changes or new data, and to reconcile file versions among agents. The platform uses the fash (hash of the file content and file name) to determine the state of a file and whether a file reported by an agent is the most recent version of the file. The platform then issues work to agents needing to update one or more files to a most recent version of the file.

When directed by the platform to locate a particular piece of data or blob each agent first queries a local copy of the file manifest in order to determine if a blob with a specific hash corresponding to the data is available in local storage. When the agent does

not locate the file locally, then it uses a peer-to-peer protocol to try to locate the file on a peer agent. When using the peer-to-peer protocol, the agent uses information of a list of agents having the file to selectively locate the file, where the list is received from the platform. Alternative embodiments can include one or more alternative hierarchies for locating data and as such are not limited to peer-to-peer protocols.

Peer-to-peer communication is supported among agents of an embodiment, and in so doing each agent reports to the platform a private IP address and port on which it is listening. Furthermore, the platform observes the public IP address and port of each agent at the time the agent reports the private information (e.g., during registration of the agent, etc.). The platform generates and maintains a list of all agents along with the private IP address and port and public IP address and port of each agent. Subsequently, when a first agent wants to establish peer-to-peer communications with a second agent, the first agent queries and receives from the platform the public and private IP addresses and ports of the second agent and then attempts connects to both using the public and private information.

Peer-to-peer communications between the agents of an embodiment are encrypted using the RSA algorithm for public-key cryptography, but are not so limited. The encryption of an embodiment retains or stores all keys locally within the grid, so no entity outside of the grid can access the unencrypted data. Further, additional data security is realized because no complete file is ever stored at the platform.

In an example involving encrypted peer-to-peer communication among agents, a second agent B wishing to establish communication with a first agent A generates a random secret S that is used to encrypt data for use in symmetric encryption between agents A and B. The second agent B encrypts the secret S using the public key of the first agent A, which it retrieves from the platform, and provides the encrypted secret S to the platform where it is stored. The second agent B communicates to the first agent A that they are going to establish a communication session using a session ID and, in response, the first agent A fetches the encrypted secret S from the platform and decrypts

the secret S. The secret is subsequently used by the first agent A and the second agent B to form an encrypted communication channel for peer-to-peer communication.

With reference to **Figure 4**, which is a block diagram of an example system including the platform and an agent, the platform creates or generates work and assigns the work to agents, as described in detail herein. The agents in turn perform work assigned to them by the platform and provide to the platform information of their data and work performed. Because the agents of an embodiment perform work as directed by the platform, they are not applications and not users. In operation, each agent registers with the platform using a passport prior to being available for operations. The passport, which in an embodiment is received from the platform, includes keys and information that identifies the corresponding agent. The platform includes an agent database that is a list of all agents available to the platform and the relevant information of each agent, and agents are placed in the agent database by the platform upon registering with the platform.

In addition to the agent database, the platform of an embodiment comprises or is coupled to a number of other databases in the cloud, where the agents and the platform of an embodiment couple to the databases and exchange information using the databases. The databases include but are not limited to the agent database, as described above, a libraries database and a goals database. Each agent can have one or more libraries that represent their local disk or storage, and the libraries database includes a list of the storage available to a corresponding agent. The goals database includes a list of goals which are collections of libraries on which a particular task is to be performed (e.g., synchronized).

Regarding the libraries database, each agent identifies storage mount points, referred to herein as libraries, to the platform. Mounts include but are not limited to any device to which the agent has been granted read and/or write access, for example, a disk or disk drive, network mount point, hard drive, flash drive, and storage service to name a few. The platform and each agent include a global libraries database that includes a list

of libraries of the system and each corresponding agent of the system that can access each library.

Additionally, each agent provides information of the contents of each library through the use of periodic "snapshots" or scans of each library, and the agent and platform tracks the current state of all files in the libraries using the information of the snapshots. The libraries database enables queries to locate each library in a grid and identify each agent that can access the library. This architecture enables the agent and the platform to know that if each has access to a library having a particular identification then each is accessing the same library. Thus, each library is presented individually and separately in the libraries database. While the platform and the agent of an embodiment each include a global libraries database that includes information of all libraries of the grid, the embodiment is not so limited. Alternatively, the platform includes a global libraries database of all libraries in the grid, and each agent includes a local libraries database that includes information of the libraries accessible by that particular agent.

The platform also includes goals that are incorporated or used by agents to perform tasks. Goals deal only with libraries instead of dealing with agents, mounts, work, of the like. Goals correspond to one or more libraries and are used to carry out operations or work on the contents of the library. Further, a library can correspond to multiple goals in an embodiment.

The platform of an embodiment includes a work database corresponding to each agent, and the agent receives the work database from the platform. The platform generates the work database for each agent, and puts work tasks that are to be accomplished by the corresponding agent in the work database. The work database includes a description of each task that is to be performed by an agent along with all information needed by the agent to perform the work. Any task placed in the work database by the platform can be dynamically revised and/or deleted in response to other work performed in the grid. For example, if a task is put in the work database to "rename file A as file B," and subsequently file B is to be renamed to file C, then the platform

revises the work as "rename file A as file C." Additionally, any task that subsequently becomes obsolete is removed from the work database.

In an embodiment, an agent provides information of its capabilities to the platform. Alternatively, the platform includes or generates a capabilities database comprising global information of capabilities of all agents of the grid.

Agents receive work assignments or tasks generated by the platform via their respective work databases, as described above. Each discrete work item in the work database includes a verb that describes action to be taken by the agent. The tasks or work items that can be executed by the agent of an embodiment include scanning, deleting, writing, and uploading, for example. A work item is delivered to an agent along with conditions of completion for that work item.

For example, the scan task involves reporting to the platform a current state of the files in the agent's library, and reporting to the platform changes made to the contents of the library (e.g., addition of file(s), deletion of file(s), etc.) and changes made to any file of the library.

Another task is an upload task in which the agent locates the appropriate file, breaks the file into blobs, encrypts the blobs, and writes or transfers a list of the blobs that form the file to an object store like a cloud-based storage device. When data is written to the object store, the data is content addressed, so a hash of the blob is transferred to the object store and subsequently used to reference the blob/data (i.e., the file name is the hash).

The tasks of an embodiment also include a write task that involves locating one or more blobs of a file and copying the blobs from a first location (e.g., source device) to a second location (e.g., destination device). In response to a received write task, an agent attempts to locate a blob locally in the corresponding agent library. If the blob is not found locally, the agent attempts to locate the blob at a peer agent that has reported having the blob. If the blob cannot be found at a peer agent, then the agent retrieves the file from the object store (e.g., cloud-based storage device). When an entire file is to be written, a list of blobs is reported or written as the manifest of the file.

With reference to **Figure 4**, an example operation involves the synchronization of a document on a device/agent in response to a determination that the version of the document is not the most recent version as found on another device/agent. Following registration of the agent AGENT_X using a passport, the agent is directed to perform a scan to create and provide to the platform a library file that includes the contents of a library LIBRARY_X of the agent. To initiate the scan, the platform adds the scan task to the work database (e.g., WORK_AGENTX) for the particular agent (e.g., AGENT_X), and the platform writes into the work database for the agent all work necessary to synchronize files.

The library LIBRARY_X contains metadata about files that reside on a mount accessible by AGENT_X and includes the metadata that defines a subset of files by library identifiers or filters the files. Thus, the library includes metadata of a set of files present on a mount. The scan is a continuously running job that results in metadata being provided to the platform, where the metadata represents for each library "scanned" the contents of that library (e.g., LIBRARY_X). As a result, generally, a mirror of each library on each agent is created and placed in the library database of the platform such that the library database then includes the filtered metadata based on the application of a particular filter to the mounts accessible by each agent.

As a result of the above operations applied to all agents accessible by the platform, the platform includes metadata of the contents of all files of the agents to which it has access. The platform also includes in the goals database, for purposes of this example, a "sync" goal for LIBRARY_X and LIBRARY_Y (not shown) of another agent (not shown). This operation creates a sync document, and the sync document tracks the current signature or goal state of every file in each of these libraries using an entry that corresponds to every file, and updates the goal states as library files or items change in these respective libraries. Using this example, the sync document tracks the current goal state of LIBRARY_X and LIBRARY_Y, and if either file is not in the correct goal state, then work is created and assigned by the platform to put LIBRARY_X and LIBRARY_Y in the correct (same) goal state. In this example, the goal state is determined by the last

or most current state, and the documents of LIBRARY_Y are the most current file versions so work is assigned to AGENT_X to update the files of LIBRARY_X to the same state as the files of LIBRARY_Y. The system then initiates a session involving the appropriate entities (e.g., peer agent(s), remote storage device(s), etc.) so that the necessary information can be retrieved to exchange data and synchronize the libraries.

Another example involving file synchronization between agents is depicted in **Figure 5** which shows a block diagram of an example involving file synchronization between the platform and agents, under an embodiment. This example includes three agents A 502 , B 504 , and C 506, and each agent has a corresponding library LIB_A 508, LIB_B 510, and LIB_C 512, but the embodiment is not so limited. The platform includes a goal 514 in the goals database to synchronize these three libraries. The platform includes a libraries database that comprises numerous records 516, each of which corresponds to a single file managed by the platform. Each record includes information in the form of an entry for each instance of the file, and the entry includes information of the library that includes the file and a hash of the file path (P).

Assume for purposes of this example that agent A reports for a particular file a different hash (e.g., H2) than agents B and C (e.g., H). The platform determines from information of the libraries database that agent A is reporting different hash information for the file than that of agents B and C and that the file of agent A is the most current version of the file. In response, the platform generates work for agents B 518 and C 520 (e.g., WORK_B and WORK_C, respectively) to update their files to the most current version that is possessed by agent A. The work generated is to locate and fetch the corresponding file and use the retrieved information to update their files, and the work task is placed by the platform into a work database of the platform. The work task generated by the platform of an embodiment includes information of a location (e.g., agent A) in the system where the most current version of the file is stored, but is not so limited. The work task includes information to fetch the corresponding file from the appropriate location(s) (e.g., one or more agents controlling blobs of the file) and use the

retrieved information to update files. The work task is placed by the platform into a work database of the platform.

In this example, agents B and C retrieve their respective work (e.g., WORK_B and WORK_C, respectively) from the work database of the platform. Upon completing the work, each of agents A, B, and C include the most current version of the file, as indicated by the hash H2 in the libraries of agents A, B, and C (not shown). Further, each of the agents B and C report completion of the work task to the platform.

More particularly, **Figure 6** is an example flow diagram involving an agent, the platform, and databases, under an embodiment. In this example, a new agent A registers 602 with the platform and an entry is created in the agent database corresponding to the new agent A. The platform in response to agent A being added to the agent database creates 604 a work database WORK_A for the new agent A. The work database WORK_A is generated to include work or tasks for the corresponding agent A. Additionally, a new library entry is added 606 to the libraries database corresponding to the new agent A and, in response, a new database LIB_LA 608 is created for information of the library of agent A. Further, a new sync goal SYNC(LA) 610 is generated and added to the goals database, and the new sync goal comprises information by which the platform directs synchronization among the files of new agent A and the files of other agents to which the platform has access.

In response to registration of the new agent A, the platform initially requires information of the contents of library A, and subsequently requires knowledge of any changes to the contents of library A. **Figure 7** is an example flow diagram involving use of a sync goal to scan and sync a library, under an embodiment. Following the registration process of agent A 702, the sync goal SYNC(LA) operates to create and add a work task 704 to the work database WORK_A of agent A. Agent A responds to the work task by scanning the contents of library A in order to determine or learn the initial contents of the library, as described in detail herein. In addition to passing to the platform the metadata of all contents of the library A, the information of the contents of

library A is subsequently used to determine when changes occur to the contents of library A.

Using the information of the contents of library A, agent A subsequently monitors the library A for any change to content. The change can include the addition of a new file, deletion of a file, any change to content of any file, and renaming of a file to name a few, but is not so limited. Agent A, in response to detecting a change to the contents of library A, posts a change 706 into the library database LIB_LA. Further, agent A passes to the platform the metadata corresponding to any subsequent changes in the contents of the library A.

The platform of an embodiment includes a sync database. The platform generates the sync database to include and maintain a record of agents and file states and, more particularly, include an entry for each file and/or each agent that includes a content hash corresponding to that file and/or agent. When the platform detects the posting of a change in the database library LIB_LA, the platform updates 708 the sync database to reflect that change in library A. This change to library A means that other libraries (e.g., B and C) are no longer in sync with library A. As a result, the platform generates work for the agents 710 of the other libraries to update the libraries to be in sync with the contents of library A. Therefore, in this example, work tasks are generated for agents B and C to synchronize their libraries.

As another example, the platform and agents of an embodiment form a virtual NAS. So, instead of storing all data at a network storage device, which requires the data to be transferred to the device, the platform and agents can be used to generate a virtual NAS using the index of the data and the corresponding metadata of the platform. In this manner, the NAS is formed using the collection of computing devices that form the grid and is therefore on site and controlled by the owning agency.

As an example, a small office environment includes a Network-Attached Storage (NAS) device for use in backing up all computers in the office. After some period of time, however, the data of the office computers consumes all memory of the NAS device and additional storage is needed to back up the computers of the network. Using

conventional technology, a first NAS would be required to be replaced with a NAS having a larger memory, or each computer in the office would be required to be assigned to a NAS so that all computers are not backed up on the same NAS. Use of the platform of an embodiment, however, eliminates this problem because the platform can be used to communicate with agents hosted on each of the memory devices or, alternatively, hosted on each computer in the office, where each agent has the ability to write to each NAS. In so doing, the platform communicates with all members of the grid that is formed through the agents, and by virtue of the communication the platform has information of the location of all data of the office. Using the metadata of the office data, the platform can direct each agent as to a location to which that agent is to store back up data or to a location to which that agent is to go to retrieve or read data needed by that agent to complete processing operations.

As another example application involving the platform of an embodiment, a user places digital pictures into a folder, and subsequently wishes to tweet all of the pictures contained in the folder using the Twitter application. In an embodiment, an agent is installed on a twitter API, and the platform includes a goal that every picture placed into the folder is provided to twitter. Using this goal, the platform directs that agent to provide the picture to twitter, and the twitter agent can then be directed to generate a tweet using the photos. As yet another example, the twitter agent can be directed by the platform to store every tweet received in an account at a particular location on a NAS.

As yet another example, the platform of an embodiment provides content transcoding and streaming instead of file transfer. Under this scenario, a relatively large file that is to be downloaded to a device can be transcoded and streamed to the device from one or more peer agents instead of downloading the complete file to the device.

Embodiments described herein include a system comprising a platform including a processor coupled to a plurality of databases. The system includes a grid comprising a plurality of agents coupled to the platform. Each agent of the plurality of agents is an agent of the platform running on a client device. The system comprises metadata of content of memory accessible by a plurality of client devices corresponding to the

plurality of agents. Each agent of each client device generates and provides the metadata to the platform. The platform comprises the metadata instead of the content and uses the metadata to determine locations of the content, generate goals representing operations for maintaining a state of the content, and generate a plurality of tasks corresponding to the goals and assign each task to an agent having access to the content that corresponds to the task. Each task is a processing operation directed by the platform to be performed by an agent on content accessible by the agent.

Embodiments described herein include a system comprising: a platform comprising a processor coupled to a plurality of databases; a grid comprising a plurality of agents coupled to the platform, wherein each agent of the plurality of agents is an agent of the platform running on a client device; and metadata of content of memory accessible by a plurality of client devices corresponding to the plurality of agents, wherein each agent of each client device generates and provides the metadata to the platform, wherein the platform comprises the metadata instead of the content and uses the metadata to determine locations of the content, generate goals representing operations for maintaining a state of the content, and generate a plurality of tasks corresponding to the goals and assign each task to an agent having access to the content that corresponds to the task, wherein each task is a processing operation directed by the platform to be performed by an agent on content accessible by the agent.

The platform of an embodiment is a master controller for the plurality of agents, wherein the plurality of agents work in cooperation with the platform as controlled by the plurality of tasks.

Each agent of the plurality of agents of an embodiment functions independently of the platform and any other agent of the plurality of agents.

The platform of an embodiment receives the metadata of the content instead of receiving the content, wherein a plurality of client devices hosting the plurality of agents comprise distributive storage devices that include the content.

The platform of an embodiment uses the metadata to maintain at the plurality of databases a master index of the content of the plurality of agents.

The metadata of an embodiment comprises data on location of the content.

The metadata of an embodiment comprises data on state of the content.

The metadata of an embodiment comprises data on identity of the plurality of agents.

The metadata of an embodiment comprises data on the content to which each agent has access.

The metadata of an embodiment comprises information of the plurality of tasks performed by the plurality of agents.

The task of an embodiment is related to tracking location of the content across the grid.

The task of an embodiment is related to managing storage of the content across the grid.

The task of an embodiment is related to managing movement of the content across the grid.

The task of an embodiment is related to processing the content across the grid.

Each task of an embodiment comprises conditions of completion for the task.

The plurality of agents of an embodiment collectively monitors the content as directed by the platform.

The plurality of agents of an embodiment collectively stores the content as directed by the platform.

The plurality of agents of an embodiment collectively transfers the content as directed by the platform.

The plurality of agents of an embodiment collectively performs processing operations on the content as directed by the platform.

Each agent of each client device of an embodiment indexes content of memory accessible by the client device.

The plurality of databases of an embodiment includes an agent database that comprises agents available to the platform and information of each agent.

Each agent of an embodiment includes at least one library, wherein a library represents a device to which the agent has access including at least one of read access and write access.

The plurality of databases of an embodiment includes a libraries database that comprises a list of libraries corresponding to the plurality of agents, wherein each library of the grid is separately represented in the libraries database.

Each agent of an embodiment has access to each library corresponding to the agent.

The plurality of databases of an embodiment include a goals database that comprises a list of the goals, wherein each goal of the list of goals is generated by the platform and corresponds to at least one library, wherein the goal is used to execute operations on corresponding library contents.

The list of goals of an embodiment comprises a collection of libraries on which tasks are to be performed.

Each goal of an embodiment corresponds to at least one library and is used by an agent to execute at least one task on content of a corresponding library.

The platform of an embodiment uses the goals to read a plurality of libraries of the plurality of agents.

The platform of an embodiment uses the goals to identify at least one of inconsistencies and discrepancies in the content accessible by the plurality of agents.

The platform of an embodiment uses the goals to generate at least one task to eliminate at least one of inconsistencies and discrepancies identified in the content.

The plurality of databases of an embodiment includes a work database.

The platform of an embodiment generates a work database corresponding to each agent, wherein the work database includes tasks that are to be performed by the corresponding agent.

The work database of an embodiment includes a description of each task that is to be performed by a corresponding agent and information necessary for the agent to perform the task.

The plurality of databases of an embodiment includes a capabilities database that comprises information of capabilities of the plurality of agents.

The plurality of databases of an embodiment includes a synchronization database that includes and maintains a record of the plurality of agents.

The synchronization database of an embodiment includes a record of file states of content of the plurality of agents.

The synchronization database of an embodiment includes an entry for each file, wherein the entry includes a content hash corresponding to the file, wherein the content hash comprises a hash of a list of blobs representing the content of the file, wherein each blob of the list of blobs comprises a representation of a fragment of a file in the content, wherein the fragment is a component of the file.

The synchronization database of an embodiment includes an entry for each agent, wherein the entry includes a content hash corresponding to files of the agent, wherein the content hash comprises a hash of a list of blobs representing the content of the files, wherein each blob of the list of blobs comprises a representation of a fragment of a file in the content, wherein the fragment is a component of the file.

The platform of an embodiment controls transfer of content among client devices using the plurality of agents.

The platform of an embodiment controls synchronizing of the content among client devices using the plurality of agents, wherein the synchronizing of the content includes synchronizing the content in response to changes in the content.

The transfer of content of an embodiment comprises block-level, non-sequential transfer of content.

The transfer of content of an embodiment comprises transferring a first block of the content from a second client device to a first client device and transferring a second block of the content from a third client device to a first client device.

The plurality of tasks of an embodiment includes a scan task.

The scan task of an embodiment includes the agent recursively monitoring a library corresponding to the agent and reporting to the platform any changes to the library.

The reporting of an embodiment comprises placing a file of the library that includes the changes in a local database of the client device hosting the agent, and providing the metadata of the local database to the platform.

The plurality of tasks of an embodiment includes a write task.

The write task of an embodiment includes the agent copying at least one blob of a file from a first location to a second location, wherein each of the first location and second location correspond to client devices coupled to the grid.

The at least one blob of an embodiment comprises a representation of a fragment of a file in the content, wherein the fragment is a component of the file.

The write task of an embodiment comprises conditions of completion.

The conditions of completion of an embodiment comprise at least one of retrieving the blob corresponding to the write task and identifying that the blob to be overwritten during the write task corresponds to a correct file.

The plurality of tasks of an embodiment comprises an upload task that includes copying a representation of the content from a device accessible by the agent to a remote storage device.

The agent of an embodiment reports to the platform addition of the file to the content accessible by the agent and, in response the platform assigns a task to the agent to upload the file.

In response to the task the agent of an embodiment determines if the file is present at the remote storage device, and uploads the representation of the file to the remote storage device when the file is determined to be absent.

The upload task of an embodiment includes the agent copying at least one blob of a file from the device accessible by the agent to the remote storage device.

The at least one blob of an embodiment comprises a representation of a fragment of the file, wherein the fragment is a component of the file.

The plurality of tasks of an embodiment includes a delete task.

A task of an embodiment comprises a plurality of phases including at least one of queued, pending, and completed, wherein the platform tracks the phase of each task of the plurality of tasks.

Each agent of an embodiment maintains locally at the client device tasks assigned to the agent.

The agent of an embodiment periodically polls the platform to identify assigned tasks.

An agent of an embodiment comprises a hierarchy for locating task information needed to complete a task and located at a remote device.

The hierarchy of an embodiment comprises the agent searching a local database of the client device hosting the agent.

The hierarchy of an embodiment comprises the agent communicating with at least one peer agent of the plurality of agents to locate the task information.

The agent of an embodiment comprises identity of peer agents that possess the task information.

The hierarchy of an embodiment comprises the agent retrieving the task information from a remote storage device.

The agent of an embodiment receives from the platform the identity of peer agents in an ordered list and searches for the task information in accordance with the ordered list.

Each agent of an embodiment includes a plurality of components executing in parallel.

The plurality of components of an embodiment include a provider component that retrieves tasks designated for the agent from the platform and stores retrieved tasks in a task database local to the client device hosting the agent.

The plurality of components of an embodiment comprises a task execution component.

The plurality of components of an embodiment include a runner component that monitors the task database, retrieves each task from the task database, and provides a retrieved task to the task execution component and designates the retrieved task to have a pending status.

The task execution component of an embodiment executes the task, and reports status of task execution to the runner component.

The runner component of an embodiment reports the status to the task database.

The plurality of components of an embodiment includes an update component that monitors the task database for tasks having a completed status, and reports status information of completed tasks to the platform.

The platform of an embodiment updates the plurality of databases in response to the status information.

The metadata of an embodiment is generated by the plurality of agents, wherein metadata generated by an agent corresponds to the content to which the agent has access.

The agent of an embodiment generates the metadata by scanning content of each file to which the agent has access.

The agent of an embodiment generates the metadata by splitting the content of the file into a plurality of fragments, wherein each fragment comprises a variable size component of the file.

The variable size fragments of an embodiment are between a pre-specified minimum length and maximum length.

The agent of an embodiment generates the metadata by generating a plurality of blobs that represent the plurality of fragments, wherein each blob represents a fragment.

The agent of an embodiment generates the plurality of blobs using a data fingerprinting algorithm that comprises running, for each byte in the content, a hash algorithm over components of the content, wherein the hash algorithm is set to identify a specified pattern of data.

The generating of the blob of an embodiment comprises generating a description of the blob that includes a value of a hash at a point where the fragment represented by the blob was separated from a remaining portion of the file.

The generating of the blob of an embodiment comprises generating an offset value based on a break point of the fragment represented by the blob.

The generating of the blob of an embodiment comprises generating a hash of complete contents of the blob, wherein the hash of the complete contents of the blob is an identifier for the blob.

The generating of the blob of an embodiment comprises generating a list of blobs representing the content of the file.

The generating of the blob of an embodiment includes generating a content hash that comprises a hash of the list of blobs representing the content of the file, wherein the content hash is an identifier for the file.

The generating of the blob of an embodiment includes generating a name hash that comprises a hash of a file name corresponding to the file.

The generating of the blob of an embodiment includes generating a file hash that comprises a hash of a combination of the content hash and a name hash.

The generating of the blob of an embodiment includes generating a path hash that comprises a hash of the file name and a file path corresponding to the content of the file.

The generating of the blob of an embodiment includes generating a metadata hash that comprises a hash of file metadata of the file.

The platform of an embodiment generates a record for the file, and the record comprises blob hashes of the file, the content hash, the file hash, the path hash and the metadata hash.

The plurality of databases of an embodiment includes a libraries database, wherein the libraries database comprises the record.

The generating of the blob of an embodiment comprises generating a size of the blob.

The file of an embodiment is described as a list of blobs comprising the file.

Each agent of an embodiment stores a blob locally at the client device hosting the agent, and transfers a blob that is previously unreported to central storage of the platform.

Embodiments described herein include a system comprising a platform including a processor coupled to a plurality of databases. The system includes a grid comprising a plurality of agents coupled to the platform. Each agent of the plurality of agents is an agent of the platform running on a client device. The system comprises metadata of content of memory accessible by a plurality of client devices corresponding to the plurality of agents. The agents generate and provide to the platform metadata that corresponds to the content by hashing a plurality of fragments of the content. The platform uses the metadata instead of the content to generate and assign to the plurality of agents a plurality of tasks including tasks controlling at least one of storing, transferring and processing of the content. A task is a processing operation performed on content accessible by the agent responsible for the task.

Embodiments described herein include a system comprising: a platform comprising a processor coupled to a plurality of databases; a grid comprising a plurality of agents coupled to the platform, wherein each agent of the plurality of agents is an agent of the platform running on a client device; and metadata of content of memory accessible by a plurality of client devices corresponding to the plurality of agents, wherein the agents generate and provide to the platform metadata that corresponds to the content by hashing a plurality of fragments of the content, wherein the platform uses the metadata instead of the content to generate and assign to the plurality of agents a plurality of tasks including tasks controlling at least one of storing, transferring and processing of the content, wherein a task is a processing operation performed on content accessible by the agent responsible for the task.

Embodiments described herein include a system comprising a platform including a processor coupled to a plurality of databases. The system includes a grid comprising a plurality of agents coupled to the platform. Each agent of the plurality of agents is an agent of the platform running on a client device. Each agent of each client device provides to the platform metadata of content of memory accessible by the client device.

The platform uses the metadata to determine locations of the content, generate goals representing operations for maintaining a state of the content, and generate a plurality of tasks corresponding to the goals and assign each task to an agent having access to the content that corresponds to the task. A task is a processing operation directed by the platform to be performed by an agent on content accessible by the agent. The metadata provided by plurality of agents includes information of the plurality of tasks performed by the plurality of agents.

Embodiments described herein include a system comprising: a platform comprising a processor coupled to a plurality of databases; and a grid comprising a plurality of agents coupled to the platform, wherein each agent of the plurality of agents is an agent of the platform running on a client device; wherein each agent of each client device provides to the platform metadata of content of memory accessible by the client device; wherein the platform uses the metadata to determine locations of the content, generate goals representing operations for maintaining a state of the content, and generate a plurality of tasks corresponding to the goals and assign each task to an agent having access to the content that corresponds to the task, wherein a task is a processing operation directed by the platform to be performed by an agent on content accessible by the agent, wherein the metadata provided by plurality of agents includes information of the plurality of tasks performed by the plurality of agents.

Embodiments described herein include a system comprising a platform including a processor coupled to a plurality of databases. The system includes a grid comprising a plurality of agents coupled to the platform. Each agent of the plurality of agents is an agent of the platform running on a client device. The system comprises metadata of content of memory accessible by a plurality of client devices corresponding to the plurality of agents. Each agent generates and provides to the platform metadata that corresponds to the content to which the agent has access. Each agent generates the metadata by hashing a plurality of fragments of the content to generate a plurality of blobs representing the plurality of fragments. The platform uses the metadata instead of the content to generate and assign to the plurality of agents a plurality of tasks. A task is

a processing operation performed on content accessible by the agent responsible for the task. The plurality of tasks includes tasks that at least one of monitor, store, transfer and process the content.

Embodiments described herein include a system comprising: a platform comprising a processor coupled to a plurality of databases; a grid comprising a plurality of agents coupled to the platform, wherein each agent of the plurality of agents is an agent of the platform running on a client device; metadata of content of memory accessible by a plurality of client devices corresponding to the plurality of agents, wherein each agent generates and provides to the platform metadata that corresponds to the content to which the agent has access, wherein each agent generates the metadata by hashing a plurality of fragments of the content to generate a plurality of blobs representing the plurality of fragments, wherein the platform uses the metadata instead of the content to generate and assign to the plurality of agents a plurality of tasks, wherein a task is a processing operation performed on content accessible by the agent responsible for the task, wherein the plurality of tasks includes tasks that at least one of monitor, store, transfer and process the content.

Embodiments described herein include a method comprising coupling a platform comprising a processor to a plurality of databases. The method comprises forming a grid by coupling a plurality of agents to the platform. Each agent of the plurality of agents is an agent of the platform running on a client device. The method comprises generating metadata at each agent and providing the metadata to the platform instead of content. The metadata corresponds to the content of memory accessible by a plurality of client devices corresponding to the plurality of agents. The method comprises determining with the metadata locations of the content. The method comprises generating goals with the metadata representing operations for maintaining a state of the content. The method comprises generating with the metadata a plurality of tasks corresponding to the goals. The method comprises assigning each task to an agent having access to the content that corresponds to the task. Each task is a processing operation directed by the platform to be performed by an agent on content accessible by the agent.

Embodiments described herein include a method comprising: coupling a platform comprising a processor to a plurality of databases; forming a grid by coupling a plurality of agents to the platform, wherein each agent of the plurality of agents is an agent of the platform running on a client device; generating metadata at each agent and providing the metadata to the platform instead of content, wherein the metadata corresponds to the content of memory accessible by a plurality of client devices corresponding to the plurality of agents; determining with the metadata locations of the content; generating goals with the metadata representing operations for maintaining a state of the content; generating with the metadata a plurality of tasks corresponding to the goals; and assigning each task to an agent having access to the content that corresponds to the task, wherein each task is a processing operation directed by the platform to be performed by an agent on content accessible by the agent.

The method comprises the platform operating as a master controller for the plurality of agents, wherein the plurality of agents work in cooperation with the platform as controlled by the plurality of tasks.

The method comprises each agent of the plurality of agents functioning independently of the platform and any other agent of the plurality of agents.

The method comprises the platform receiving the metadata of the content instead of receiving the content, wherein a plurality of client devices hosting the plurality of agents comprise distributive storage devices that include the content.

The method comprises the platform using the metadata to maintain at the plurality of databases a master index of the content of the plurality of agents.

The metadata of an embodiment comprises data on location of the content.

The metadata of an embodiment comprises data on state of the content.

The metadata of an embodiment comprises data on identity of the plurality of agents.

The metadata of an embodiment comprises data on the content to which each agent has access.

The metadata of an embodiment comprises information of the plurality of tasks performed by the plurality of agents.

The task of an embodiment is related to tracking location of the content across the grid.

The task of an embodiment is related to managing storage of the content across the grid.

The task of an embodiment is related to managing movement of the content across the grid.

The task of an embodiment is related to processing the content across the grid.

Each task of an embodiment comprises conditions of completion for the task.

The method comprises the plurality of agents collectively monitoring the content as directed by the platform.

The method comprises the plurality of agents collectively storing the content as directed by the platform.

The method comprises the plurality of agents collectively transferring the content as directed by the platform.

The method comprises the plurality of agents collectively performing processing operations on the content as directed by the platform.

The method comprises each agent of each client device indexing contents of memory accessible by the client device.

The plurality of databases of an embodiment includes an agent database that comprises agents available to the platform and information of each agent.

Each agent of an embodiment includes at least one library, wherein a library represents a device to which the agent has access including at least one of read access and write access.

The plurality of databases of an embodiment includes a libraries database that comprises a list of libraries corresponding to the plurality of agents, wherein each library of the grid is separately represented in the libraries database.

The method comprises each agent having access to each library corresponding to the agent.

The plurality of databases of an embodiment includes a goals database that comprises a list of the goals, wherein each goal of the list of goals is generated by the platform and corresponds to at least one library, wherein the goal is used to execute operations on corresponding library contents.

The method comprises generating the list of goals to include a collection of libraries on which tasks are to be performed.

Each goal of an embodiment corresponds to at least one library and is used by an agent to execute at least one task on content of a corresponding library.

The method comprises the platform using the goals to read a plurality of libraries of the plurality of agents.

The method comprises the platform using the goals to identify at least one of inconsistencies and discrepancies in the content accessible by the plurality of agents.

The method comprises the platform using the goals to generate at least one task to eliminate at least one of inconsistencies and discrepancies identified in the content.

The plurality of databases of an embodiment includes a work database.

The method comprises the platform generating a work database corresponding to each agent, wherein the work database includes tasks that are to be performed by the corresponding agent.

The work database of an embodiment includes a description of each task that is to be performed by a corresponding agent and information necessary for the agent to perform the task.

The plurality of databases of an embodiment includes a capabilities database that comprises information of capabilities of the plurality of agents.

The plurality of databases of an embodiment includes a synchronization database that includes and maintains a record of the plurality of agents.

The synchronization database of an embodiment includes a record of file states of content of the plurality of agents.

The synchronization database of an embodiment includes an entry for each file, wherein the entry includes a content hash corresponding to the file, wherein the content hash comprises a hash of a list of blobs representing the content of the file, wherein each blob of the list of blobs comprises a representation of a fragment of a file in the content, wherein the fragment is a component of the file.

The synchronization database of an embodiment includes an entry for each agent, wherein the entry includes a content hash corresponding to files of the agent, wherein the content hash comprises a hash of a list of blobs representing the content of the files, wherein each blob of the list of blobs comprises a representation of a fragment of a file in the content, wherein the fragment is a component of the file.

The method comprises the platform controlling transfer of content among client devices using the plurality of agents.

The method comprises the platform controlling synchronizing of the content among client devices using the plurality of agents, wherein the synchronizing of the content includes synchronizing the content in response to changes in the content.

The transfer of content comprises block-level, non-sequential transfer of content.

The transfer of content comprises transferring a first block of the content from a second client device to a first client device and transferring a second block of the content from a third client device to a first client device.

The plurality of tasks of an embodiment includes a scan task.

The scan task of an embodiment includes the agent recursively monitoring a library corresponding to the agent and reporting to the platform any changes to the library.

The reporting comprises placing a file of the library that includes the changes in a local database of the client device hosting the agent, and providing the metadata of the local database to the platform.

The plurality of tasks of an embodiment includes a write task.

The write task of an embodiment includes the agent copying at least one blob of a file from a first location to a second location, wherein each of the first location and second location correspond to client devices coupled to the grid.

The at least one blob of an embodiment comprises a representation of a fragment of a file in the content, wherein the fragment is a component of the file.

The write task of an embodiment comprises conditions of completion.

The conditions of completion of an embodiment comprise at least one of retrieving the blob corresponding to the write task and identifying that the blob to be overwritten during the write task corresponds to a correct file.

The plurality of tasks of an embodiment comprises an upload task that includes copying a representation of a file of the content from a device accessible by the agent to a remote storage device.

The method comprises the agent reporting to the platform addition of the file to the content accessible by the agent and, in response the platform assigns a task to the agent to upload the file.

The method comprises in response to the task the agent determining if the file is present at the remote storage device, and uploads the representation of the file to the remote storage device when the file is determined to be absent.

The upload task of an embodiment includes the agent copying at least one blob of a file from the device accessible by the agent to the remote storage device.

The at least one blob of an embodiment comprises a representation of a fragment of the file, wherein the fragment is a component of the file.

The plurality of tasks of an embodiment includes a delete task.

A task of an embodiment comprises a plurality of phases including at least one of queued, pending, and completed, wherein the platform tracks the phase of each task of the plurality of tasks.

The method comprises each agent maintaining locally at the client device tasks assigned to the agent.

The method comprises the agent periodically polling the platform to identify assigned tasks.

An agent of an embodiment comprises a hierarchy for locating task information needed to complete a task and located at a remote device.

The hierarchy of an embodiment comprises the agent searching a local database of the client device hosting the agent.

The hierarchy of an embodiment comprises the agent communicating with at least one peer agent of the plurality of agents to locate the task information.

The agent of an embodiment comprises identity of peer agents that possess the task information.

The hierarchy of an embodiment comprises the agent retrieving the task information from a remote storage device.

The method comprises the agent receiving from the platform the identity of peer agents in an ordered list and searching for the task information in accordance with the ordered list.

Each agent of an embodiment includes a plurality of components executing in parallel.

The plurality of components of an embodiment include a provider component that retrieves tasks designated for the agent from the platform and stores retrieved tasks in a task database local to the client device hosting the agent.

The plurality of components of an embodiment comprises a task execution component.

The plurality of components of an embodiment include a runner component that monitors the task database, retrieves each task from the task database, and provides a retrieved task to the task execution component and designates the retrieved task to have a pending status.

The method comprises the task execution component executing the task, and reports status of task execution to the runner component.

The method comprises the runner component reporting the status to the task database.

The plurality of components of an embodiment includes an update component that monitors the task database for tasks having a completed status, and reports status information of completed tasks to the platform.

The method comprises the platform updating the plurality of databases in response to the status information.

The method comprises the plurality of agents generating the metadata, wherein metadata generated by an agent corresponds to the content to which the agent has access.

The method comprises the agent generating the metadata by scanning content of each file to which the agent has access.

The method comprises the agent generating the metadata by splitting the content of the file into a plurality of fragments, wherein each fragment comprises a variable size component of the file.

The variable size fragments of an embodiment are between a pre-specified minimum length and maximum length.

The method comprises the agent generating the metadata by generating a plurality of blobs that represent the plurality of fragments, wherein each blob represents a fragment.

The method comprises the agent generating the plurality of blobs using a data fingerprinting algorithm that comprises running, for each byte in the content, a hash algorithm over components of the content, wherein the hash algorithm is set to identify a specified pattern of data.

The generating of the blob of an embodiment comprises generating a description of the blob that includes a value of a hash at a point where the fragment represented by the blob was separated from a remaining portion of the file.

The generating of the blob of an embodiment comprises generating an offset value based on a break point of the fragment represented by the blob.

The generating of the blob of an embodiment comprises generating a hash of complete contents of the blob, wherein the hash of the complete contents of the blob is an identifier for the blob.

The generating of the blob of an embodiment comprises generating a list of blobs representing the content of the file.

The generating of the blob of an embodiment includes generating a content hash that comprises a hash of the list of blobs representing the content of the file, wherein the content hash is an identifier for the file.

The generating of the blob of an embodiment includes generating a name hash that comprises a hash of a file name corresponding to the file.

The generating of the blob of an embodiment includes generating a file hash that comprises a hash of a combination of the content hash and a name hash.

The generating of the blob of an embodiment includes generating a path hash that comprises a hash of the file name and a file path corresponding to the content of the file.

The generating of the blob of an embodiment includes generating a metadata hash that comprises a hash of file metadata of the file.

The method comprises the platform generating a record for the file, wherein the record comprises blob hashes of the file, the content hash, the file hash, the path hash and the metadata hash.

The plurality of databases of an embodiment includes a libraries database, wherein the libraries database comprises the record.

The generating of the blob of an embodiment comprises generating a size of the blob.

The file of an embodiment is described as a list of blobs comprising the file.

The method comprises each agent storing a blob locally at the client device hosting the agent, and transfers a blob that is previously unreported to central storage of the platform.

Embodiments described herein include a method comprising coupling a platform comprising a processor to a plurality of databases. The method comprises forming a grid

comprising a plurality of agents coupled to the platform. Each agent is an agent of the platform running on a client device. The method comprises generating metadata at each agent by hashing a plurality of fragments of content accessible by the agent. The metadata corresponds to the content of memory accessible by a plurality of client devices corresponding to the plurality of agents. The method comprises providing the metadata to the platform instead of the content. The method comprises generating with the metadata a plurality of tasks including tasks controlling at least one of storing, transferring and processing of the content. A task is a processing operation performed on content accessible by the agent responsible for the task. The method comprises assigning the plurality of tasks to the plurality of agents.

Embodiments described herein include a method comprising: coupling a platform comprising a processor to a plurality of databases; forming a grid comprising a plurality of agents coupled to the platform, wherein each agent is an agent of the platform running on a client device; generating metadata at each agent by hashing a plurality of fragments of content accessible by the agent, wherein the metadata corresponds to the content of memory accessible by a plurality of client devices corresponding to the plurality of agents; providing the metadata to the platform instead of the content; generating with the metadata a plurality of tasks including tasks controlling at least one of storing, transferring and processing of the content, wherein a task is a processing operation performed on content accessible by the agent responsible for the task; and assigning the plurality of tasks to the plurality of agents.

Embodiments described herein include a method comprising coupling a platform comprising a processor to a plurality of databases. The method comprises forming a grid comprising a plurality of agents coupled to the platform. Each agent is an agent of the platform running on a client device. The method comprises providing to the platform by each agent of each client device metadata of content of memory accessible by the client device. The method comprises determining with the metadata locations of the content. The method comprises generating goals with the metadata representing operations for maintaining a state of the content. The method comprises generating with the metadata a

plurality of tasks corresponding to the goals. The method comprises assign each task to an agent having access to the content that corresponds to the task. A task is a processing operation directed by the platform to be performed by an agent on content accessible by the agent. The metadata provided by plurality of agents includes information of the plurality of tasks performed by the plurality of agents.

Embodiments described herein include a method comprising: coupling a platform comprising a processor to a plurality of databases; forming a grid comprising a plurality of agents coupled to the platform, wherein each agent is an agent of the platform running on a client device; providing to the platform by each agent of each client device metadata of content of memory accessible by the client device; determining with the metadata locations of the content; generating goals with the metadata representing operations for maintaining a state of the content; generating with the metadata a plurality of tasks corresponding to the goals; and assign each task to an agent having access to the content that corresponds to the task, wherein a task is a processing operation directed by the platform to be performed by an agent on content accessible by the agent, wherein the metadata provided by plurality of agents includes information of the plurality of tasks performed by the plurality of agents.

Embodiments described herein include a method comprising establishing a coupling between a platform comprising a processor and a plurality of databases. The method comprises forming a grid by coupling a plurality of agents to the platform. Each agent of the plurality of agents is an agent of the platform running on a client device. The method comprises generating metadata at each agent representing content of memory accessible by a plurality of client devices corresponding to the plurality of agents. The generating comprises hashing a plurality of fragments of the content to generate a plurality of blobs representing the plurality of fragments. The method comprises providing the metadata to the platform instead of the content. The method comprises generating with the metadata a plurality of tasks. A task is a processing operation performed on content accessible by the agent responsible for the task. The plurality of

tasks include tasks that at least one of monitor, store, transfer and process the content; and assigning the plurality of tasks to the plurality of agents.

Embodiments described herein include a method comprising: establishing a coupling between a platform comprising a processor and a plurality of databases; forming a grid by coupling a plurality of agents to the platform, wherein each agent of the plurality of agents is an agent of the platform running on a client device; generating metadata at each agent representing content of memory accessible by a plurality of client devices corresponding to the plurality of agents, wherein the generating comprises hashing a plurality of fragments of the content to generate a plurality of blobs representing the plurality of fragments; providing the metadata to the platform instead of the content; generating with the metadata a plurality of tasks, wherein a task is a processing operation performed on content accessible by the agent responsible for the task, wherein the plurality of tasks include tasks that at least one of monitor, store, transfer and process the content; and assigning the plurality of tasks to the plurality of agents.

As described above, computer networks suitable for use with the embodiments described herein include local area networks (LAN), wide area networks (WAN), Internet, or other connection services and network variations such as the world wide web, the public internet, a private internet, a private computer network, a public network, a mobile network, a cellular network, a value-added network, and the like. Computing devices coupled or connected to the network may be any microprocessor controlled device that permits access to the network, including terminal devices, such as personal computers, workstations, servers, mini computers, main-frame computers, laptop computers, mobile computers, palm top computers, hand held computers, mobile phones, TV set-top boxes, or combinations thereof. The computer network may include one of more LANs, WANs, Internets, and computers. The computers may serve as servers, clients, or a combination thereof.

The components described herein can be components of a single system, multiple systems, and/or geographically separate systems. The components described herein can

also be a subcomponent or subsystem of a single system, multiple systems, and/or geographically separate systems. The components described herein can be coupled to one or more other components (not shown) of a host system or a system coupled to the host system.

The components described herein include and/or run under and/or in association with a processing system. The processing system includes any collection of processor-based devices or computing devices operating together, or components of processing systems or devices, as is known in the art. For example, the processing system can include one or more of a portable computer, portable communication device operating in a communication network, and/or a network server. The portable computer can be any of a number and/or combination of devices selected from among personal computers, personal digital assistants, portable computing devices, and portable communication devices, but is not so limited. The processing system can include components within a larger computer system.

The processing system of an embodiment includes at least one processor and at least one memory device or subsystem. The processing system can also include or be coupled to at least one database. The term "processor" as generally used herein refers to any logic processing unit, such as one or more central processing units (CPUs), digital signal processors (DSPs), application-specific integrated circuits (ASIC), etc.

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 a sense of "including, but not limited to." Words using the singular or plural number also include the plural or singular number respectively. Additionally, the words "herein," "hereunder," "above," "below," and words of similar import, when used in this application, refer to this application as a whole and not to any particular portions of this application. When the word "or" is used in reference to a list of two or more items, that word 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.

The above description of embodiments and corresponding systems and methods is not intended to be exhaustive or to limit the systems and methods to the precise forms disclosed. While specific embodiments of, and examples for, systems and methods are described herein for illustrative purposes, various equivalent modifications are possible within the scope of the systems and methods, as those skilled in the relevant art will recognize. The teachings of the systems and methods provided herein can be applied to other systems and methods, not only for the systems and methods described above.

The elements and acts of the various embodiments described above can be combined to provide further embodiments. These and other changes can be made to the systems and methods in light of the above detailed description.

CLAIMS

What is claimed is:

1. A system comprising:

a platform comprising a processor coupled to a plurality of databases;

a grid comprising a plurality of agents coupled to the platform, wherein each agent of the plurality of agents is an agent of the platform running on a client device; and

metadata of content of memory accessible by a plurality of client devices corresponding to the plurality of agents, wherein each agent of each client device generates and provides the metadata to the platform, wherein the platform comprises the metadata instead of the content and uses the metadata to determine locations of the content, generate goals representing operations for maintaining a state of the content, and generate a plurality of tasks corresponding to the goals and assign each task to an agent having access to the content that corresponds to the task, wherein each task is a processing operation directed by the platform to be performed by an agent on content accessible by the agent.

2. The system of claim 1, wherein the platform is a master controller for the plurality of agents, wherein the plurality of agents work in cooperation with the platform as controlled by the plurality of tasks.

3. The system of claim 1, wherein each agent of the plurality of agents functions independently of the platform and any other agent of the plurality of agents.

4. The system of claim 1, wherein the platform receives the metadata of the content instead of receiving the content, wherein a plurality of client devices hosting the plurality of agents comprise distributive storage devices that include the content.

5. The system of claim 1, wherein the platform uses the metadata to maintain at the plurality of databases a master index of the content of the plurality of agents.

6. The system of claim 1, wherein the metadata comprises data on location of the content.

7. The system of claim 1, wherein the metadata comprises data on state of the content.

8. The system of claim 1, wherein the metadata comprises data on identity of the plurality of agents.

9. The system of claim 1, wherein the metadata comprises data on the content to which each agent has access.

10. The system of claim 1, wherein the metadata comprises information of the plurality of tasks performed by the plurality of agents.

11. The system of claim 1, wherein the task is related to tracking location of the content across the grid.

12. The system of claim 1, wherein the task is related to managing storage of the content across the grid.

13. The system of claim 1, wherein the task is related to managing movement of the content across the grid.

14. The system of claim 1, wherein the task is related to processing the content across the grid.

15. The system of claim 1, wherein each task comprises conditions of completion for the task.

16. The system of claim 1, wherein the plurality of agents collectively monitor the content as directed by the platform.

17. The system of claim 1, wherein the plurality of agents collectively store the content as directed by the platform.

18. The system of claim 1, wherein the plurality of agents collectively transfer the content as directed by the platform.

19. The system of claim 1, wherein the plurality of agents collectively perform processing operations on the content as directed by the platform.

20. The system of claim 1, wherein each agent of each client device indexes content of memory accessible by the client device.

21. The system of claim 1, wherein the plurality of databases include an agent database that comprises agents available to the platform and information of each agent.

22. The system of claim 1, wherein each agent includes at least one library, wherein a library represents a device to which the agent has access including at least one of read access and write access.

23. The system of claim 22, wherein the plurality of databases includes a libraries database that comprises a list of libraries corresponding to the plurality of agents, wherein each library of the grid is separately represented in the libraries database.

24. The system of claim 23, wherein each agent has access to each library corresponding to the agent.

25. The system of claim 22, wherein the plurality of databases include a goals database that comprises a list of the goals, wherein each goal of the list of goals is generated by the platform and corresponds to at least one library, wherein the goal is used to execute operations on corresponding library contents.

26. The system of claim 25, wherein the list of goals comprises a collection of libraries on which tasks are to be performed.

27. The system of claim 25, wherein each goal corresponds to at least one library and is used by an agent to execute at least one task on content of a corresponding library.

28. The system of claim 25, wherein the platform uses the goals to read a plurality of libraries of the plurality of agents.

29. The system of claim 25, wherein the platform uses the goals to identify at least one of inconsistencies and discrepancies in the content accessible by the plurality of agents.

30. The system of claim 29, wherein the platform uses the goals to generate at least one task to eliminate at least one of inconsistencies and discrepancies identified in the content.

31. The system of claim 1, wherein the plurality of databases include a work database.

32. The system of claim 31, wherein the platform generates a work database corresponding to each agent, wherein the work database includes tasks that are to be performed by the corresponding agent.

33. The system of claim 32, wherein the work database includes a description of each task that is to be performed by a corresponding agent and information necessary for the agent to perform the task.

34. The system of claim 1, wherein the plurality of databases include a capabilities database that comprises information of capabilities of the plurality of agents.

35. The system of claim 1, wherein the plurality of databases include a synchronization database that includes and maintains a record of the plurality of agents.

36. The system of claim 35, wherein the synchronization database includes a record of file states of content of the plurality of agents.

37. The system of claim 36, wherein the synchronization database includes an entry for each file, wherein the entry includes a content hash corresponding to the file, wherein the content hash comprises a hash of a list of blobs representing the content of the file, wherein each blob of the list of blobs comprises a representation of a fragment of a file in the content, wherein the fragment is a component of the file.

38. The system of claim 36, wherein the synchronization database includes an entry for each agent, wherein the entry includes a content hash corresponding to files of the agent, wherein the content hash comprises a hash of a list of blobs representing the content of the files, wherein each blob of the list of blobs comprises a representation of a fragment of a file in the content, wherein the fragment is a component of the file.

39. The system of claim 1, wherein the platform controls transfer of content among client devices using the plurality of agents.

40. The system of claim 39, wherein the platform controls synchronizing of the content among client devices using the plurality of agents, wherein the synchronizing of the content includes synchronizing the content in response to changes in the content.

41. The system of claim 39, wherein the transfer of content comprises block-level, non-sequential transfer of content.

42. The system of claim 41, wherein the transfer of content comprises transferring a first block of the content from a second client device to a first client device and transferring a second block of the content from a third client device to a first client device.

43. The system of claim 1, wherein the plurality of tasks includes a scan task.

44. The system of claim 43, wherein the scan task includes the agent recursively monitoring a library corresponding to the agent and reporting to the platform any changes to the library.

45. The system of claim 44, wherein the reporting comprises placing a file of the library that includes the changes in a local database of the client device hosting the agent, and providing the metadata of the local database to the platform.

46. The system of claim 1, wherein the plurality of tasks includes a write task.

47. The system of claim 46, wherein the write task includes the agent copying at least one blob of a file from a first location to a second location, wherein each of the first location and second location correspond to client devices coupled to the grid.

48. The system of claim 47, wherein the at least one blob comprises a representation of a fragment of a file in the content, wherein the fragment is a component of the file.

49. The system of claim 46, wherein the write task comprises conditions of completion.

50. The system of claim 49, wherein the conditions of completion comprise at least one of retrieving the blob corresponding to the write task and identifying that the blob to be overwritten during the write task corresponds to a correct file.

51. The system of claim 1, wherein the plurality of tasks comprises an upload task that includes copying a representation of a file of the content from a device accessible by the agent to a remote storage device.

52. The system of claim 51, wherein the agent reports to the platform addition of the file to the content accessible by the agent and, in response, the platform assigns a task to the agent to upload the file.

53. The system of claim 52, wherein in response to the task the agent determines if the file is present at the remote storage device, and uploads the representation of the file to the remote storage device when the file is determined to be absent.

54. The system of claim 51, wherein the upload task includes the agent copying at least one blob of a file from the device accessible by the agent to the remote storage device.

55. The system of claim 54, wherein the at least one blob comprises a representation of a fragment of the file, wherein the fragment is a component of the file.

56. The system of claim 1, wherein the plurality of tasks includes a delete task.

57. The system of claim 1, wherein a task comprises a plurality of phases including at least one of queued, pending, and completed, wherein the platform tracks the phase of each task of the plurality of tasks.

58. The system of claim 1, wherein each agent maintains locally at the client device tasks assigned to the agent.

59. The system of claim 58, wherein the agent periodically polls the platform to identify assigned tasks.

60. The system of claim 1, wherein an agent comprises a hierarchy for locating task information needed to complete a task and located at a remote device.

61. The system of claim 60, wherein the hierarchy comprises the agent searching a local database of the client device hosting the agent.

62. The system of claim 61, wherein the hierarchy comprises the agent communicating with at least one peer agent of the plurality of agents to locate the task information.

63. The system of claim 62, wherein the agent comprises identity of peer agents that possess the task information.

64. The system of claim 62, wherein the hierarchy comprises the agent retrieving the task information from a remote storage device.

65. The system of claim 64, wherein the agent receives from the platform the identity of peer agents in an ordered list and searches for the task information in accordance with the ordered list.

66. The system of claim 1, wherein each agent includes a plurality of components executing in parallel.

67. The system of claim 66, wherein the plurality of components include a provider component that retrieves tasks designated for the agent from the platform and stores retrieved tasks in a task database local to the client device hosting the agent.

68. The system of claim 67, wherein the plurality of components comprise a task execution component.

69. The system of claim 68, wherein the plurality of components include a runner component that monitors the task database, retrieves each task from the task database, and provides a retrieved task to the task execution component and designates the retrieved task to have a pending status.

70. The system of claim 69, wherein the task execution component executes the task, and reports status of task execution to the runner component.

71. The system of claim 70, wherein the runner component reports the status to the task database.

72. The system of claim 71, wherein the plurality of components include an update component that monitors the task database for tasks having a completed status, and reports status information of completed tasks to the platform.

73. The system of claim 72, wherein the platform updates the plurality of databases in response to the status information.

74. The system of claim 1, wherein the metadata is generated by the plurality of agents, wherein metadata generated by an agent corresponds to the content to which the agent has access.

75. The system of claim 74, wherein the agent generates the metadata by scanning content of each file to which the agent has access.

76. The system of claim 75, wherein the agent generates the metadata by splitting the content of the file into a plurality of fragments, wherein each fragment comprises a variable size component of the file.

77. The system of claim 76, wherein the variable size fragments are between a pre-specified minimum length and maximum length.

78. The system of claim 76, wherein the agent generates the metadata by generating a plurality of blobs that represent the plurality of fragments, wherein each blob represents a fragment.

79. The system of claim 76, wherein the agent generates the plurality of blobs using a data fingerprinting algorithm that comprises running, for each byte in the content, a hash algorithm over components of the content, wherein the hash algorithm is set to identify a specified pattern of data.

80. The system of claim 79, wherein the generating of the blob comprises generating a description of the blob that includes a value of a hash at a point where the fragment represented by the blob was separated from a remaining portion of the file.

81. The system of claim 80, wherein the generating of the blob comprises generating an offset value based on a break point of the fragment represented by the blob.

82. The system of claim 81, wherein the generating of the blob comprises generating a hash of complete contents of the blob, wherein the hash of the complete contents of the blob is an identifier for the blob.

83. The system of claim 82, wherein the generating of the blob comprises generating a list of blobs representing the content of the file.

84. The system of claim 83, wherein the generating of the blob includes generating a content hash that comprises a hash of the list of blobs representing the content of the file, wherein the content hash is an identifier for the file.

85. The system of claim 84, wherein the generating of the blob includes generating a name hash that comprises a hash of a file name corresponding to the file.

86. The system of claim 85, wherein the generating of the blob includes generating a file hash that comprises a hash of a combination of the content hash and a name hash.

87. The system of claim 86, wherein the generating of the blob includes generating a path hash that comprises a hash of the file name and a file path corresponding to the content of the file.

88. The system of claim 87, wherein the generating of the blob includes generating a metadata hash that comprises a hash of file metadata of the file.

89. The system of claim 88, wherein the platform generates a record for the file, and the record comprises blob hashes of the file, the content hash, the file hash, the path hash and the metadata hash.

90. The system of claim 89, wherein the plurality of databases include a libraries database, wherein the libraries database comprises the record.

91. The system of claim 79, wherein the generating of the blob comprises generating a size of the blob.

92. The system of claim 79, wherein the file is described as a list of blobs comprising the file.

93. The system of claim 79, wherein each agent stores a blob locally at the client device hosting the agent, and transfers a blob that is previously unreported to central storage of the platform.

94. A system comprising:

a platform comprising a processor coupled to a plurality of databases;

a grid comprising a plurality of agents coupled to the platform, wherein each agent of the plurality of agents is an agent of the platform running on a client device; and

metadata of content of memory accessible by a plurality of client devices corresponding to the plurality of agents, wherein the agents generate and provide to the platform metadata that corresponds to the content by hashing a plurality of fragments of the content, wherein the platform uses the metadata instead of the content to generate and assign to the plurality of agents a plurality of tasks including tasks controlling at least one

of storing, transferring and processing of the content, wherein a task is a processing operation performed on content accessible by the agent responsible for the task.

95. A system comprising:

a platform comprising a processor coupled to a plurality of databases; and

a grid comprising a plurality of agents coupled to the platform, wherein each agent of the plurality of agents is an agent of the platform running on a client device;

wherein each agent of each client device provides to the platform metadata of content of memory accessible by the client device;

wherein the platform uses the metadata to determine locations of the content, generate goals representing operations for maintaining a state of the content, and generate a plurality of tasks corresponding to the goals and assign each task to an agent having access to the content that corresponds to the task, wherein a task is a processing operation directed by the platform to be performed by an agent on content accessible by the agent, wherein the metadata provided by plurality of agents includes information of the plurality of tasks performed by the plurality of agents.

96. A system comprising:

a platform comprising a processor coupled to a plurality of databases;

a grid comprising a plurality of agents coupled to the platform, wherein each agent of the plurality of agents is an agent of the platform running on a client device;

metadata of content of memory accessible by a plurality of client devices corresponding to the plurality of agents, wherein each agent generates and provides to the platform metadata that corresponds to the content to which the agent has access, wherein each agent generates the metadata by hashing a plurality of fragments of the content to generate a plurality of blobs representing the plurality of fragments, wherein the platform uses the metadata instead of the content to generate and assign to the plurality of agents a plurality of tasks, wherein a task is a processing operation performed on content

accessible by the agent responsible for the task, wherein the plurality of tasks include tasks that at least one of monitor, store, transfer and process the content.

97. A method comprising:

coupling a platform comprising a processor to a plurality of databases;

forming a grid by coupling a plurality of agents to the platform, wherein each agent of the plurality of agents is an agent of the platform running on a client device;

generating metadata at each agent and providing the metadata to the platform instead of content, wherein the metadata corresponds to the content of memory accessible by a plurality of client devices corresponding to the plurality of agents;

determining with the metadata locations of the content;

generating goals with the metadata representing operations for maintaining a state of the content;

generating with the metadata a plurality of tasks corresponding to the goals; and

assigning each task to an agent having access to the content that corresponds to the task, wherein each task is a processing operation directed by the platform to be performed by an agent on content accessible by the agent.

98. The method of claim 97, comprising the platform operating as a master controller for the plurality of agents, wherein the plurality of agents work in cooperation with the platform as controlled by the plurality of tasks.

99. The method of claim 97, comprising each agent of the plurality of agents functioning independently of the platform and any other agent of the plurality of agents.

100. The method of claim 97, comprising the platform receiving the metadata of the content instead of receiving the content, wherein a plurality of client devices hosting the plurality of agents comprise distributive storage devices that include the content.

101. The method of claim 97, comprising the platform using the metadata to maintain at the plurality of databases a master index of the content of the plurality of agents.

102. The method of claim 97, wherein the metadata comprises data on location of the content.

103. The method of claim 97, wherein the metadata comprises data on state of the content.

104. The method of claim 97, wherein the metadata comprises data on identity of the plurality of agents.

105. The method of claim 97, wherein the metadata comprises data on the content to which each agent has access.

106. The method of claim 97, wherein the metadata comprises information of the plurality of tasks performed by the plurality of agents.

107. The method of claim 97, wherein the task is related to tracking location of the content across the grid.

108. The method of claim 97, wherein the task is related to managing storage of the content across the grid.

109. The method of claim 97, wherein the task is related to managing movement of the content across the grid.

110. The method of claim 97, wherein the task is related to processing the content across the grid.

111. The method of claim 97, wherein each task comprises conditions of completion for the task.

112. The method of claim 97, comprising the plurality of agents collectively monitoring the content as directed by the platform.

113. The method of claim 97, comprising the plurality of agents collectively storing the content as directed by the platform.

114. The method of claim 97, comprising the plurality of agents collectively transferring the content as directed by the platform.

115. The method of claim 97, comprising the plurality of agents collectively performing processing operations on the content as directed by the platform.

116. The method of claim 97, comprising each agent of each client device indexing contents of memory accessible by the client device.

117. The method of claim 97, wherein the plurality of databases include an agent database that comprises agents available to the platform and information of each agent.

118. The method of claim 97, wherein each agent includes at least one library, wherein a library represents a device to which the agent has access including at least one of read access and write access.

119. The method of claim 118, wherein the plurality of databases includes a libraries database that comprises a list of libraries corresponding to the plurality of agents, wherein each library of the grid is separately represented in the libraries database.

120. The method of claim 119, comprising each agent having access to each library corresponding to the agent.

121. The method of claim 118, wherein the plurality of databases includes a goals database that comprises a list of the goals, wherein each goal of the list of goals is generated by the platform and corresponds to at least one library, wherein the goal is used to execute operations on corresponding library contents.

122. The method of claim 121, comprising generating the list of goals to include a collection of libraries on which tasks are to be performed.

123. The method of claim 121, wherein each goal corresponds to at least one library and is used by an agent to execute at least one task on content of a corresponding library.

124. The method of claim 121, comprising the platform using the goals to read a plurality of libraries of the plurality of agents.

125. The method of claim 121, comprising the platform using the goals to identify at least one of inconsistencies and discrepancies in the content accessible by the plurality of agents.

126. The method of claim 125, comprising the platform using the goals to generate at least one task to eliminate at least one of inconsistencies and discrepancies identified in the content.

127. The method of claim 97, wherein the plurality of databases include a work database.

128. The method of claim 127, comprising the platform generating a work database corresponding to each agent, wherein the work database includes tasks that are to be performed by the corresponding agent.

129. The method of claim 128, wherein the work database includes a description of each task that is to be performed by a corresponding agent and information necessary for the agent to perform the task.

130. The method of claim 97, wherein the plurality of databases include a capabilities database that comprises information of capabilities of the plurality of agents.

131. The method of claim 97, wherein the plurality of databases include a synchronization database that includes and maintains a record of the plurality of agents.

132. The method of claim 131, wherein the synchronization database includes a record of file states of content of the plurality of agents.

133. The method of claim 132, wherein the synchronization database includes an entry for each file, wherein the entry includes a content hash corresponding to the file, wherein the content hash comprises a hash of a list of blobs representing the content of the file, wherein each blob of the list of blobs comprises a representation of a fragment of a file in the content, wherein the fragment is a component of the file.

134. The method of claim 132, wherein the synchronization database includes an entry for each agent, wherein the entry includes a content hash corresponding to files of the agent, wherein the content hash comprises a hash of a list of blobs representing the content of the files, wherein each blob of the list of blobs comprises a representation of a fragment of a file in the content, wherein the fragment is a component of the file.

135. The method of claim 97, comprising the platform controlling transfer of content among client devices using the plurality of agents.

136. The method of claim 135, comprising the platform controlling synchronizing of the content among client devices using the plurality of agents, wherein the synchronizing of the content includes synchronizing the content in response to changes in the content.

137. The method of claim 135, wherein the transfer of content comprises block-level, non-sequential transfer of content.

138. The method of claim 137, wherein the transfer of content comprises transferring a first block of the content from a second client device to a first client device and transferring a second block of the content from a third client device to a first client device.

139. The method of claim 97, wherein the plurality of tasks includes a scan task.

140. The method of claim 139, wherein the scan task includes the agent recursively monitoring a library corresponding to the agent and reporting to the platform any changes to the library.

141. The method of claim 140, wherein the reporting comprises placing a file of the library that includes the changes in a local database of the client device hosting the agent, and providing the metadata of the local database to the platform.

142. The method of claim 97, wherein the plurality of tasks includes a write task.

143. The method of claim 142, wherein the write task includes the agent copying at least one blob of a file from a first location to a second location, wherein each of the first location and second location correspond to client devices coupled to the grid.

144. The method of claim 143, wherein the at least one blob comprises a representation of a fragment of a file in the content, wherein the fragment is a component of the file.

145. The method of claim 142, wherein the write task comprises conditions of completion.

146. The method of claim 145, wherein the conditions of completion comprise at least one of retrieving the blob corresponding to the write task and identifying that the blob to be overwritten during the write task corresponds to a correct file.

147. The method of claim 97, wherein the plurality of tasks comprises an upload task that includes copying a representation of a file of the content from a device accessible by the agent to a remote storage device.

148. The method of claim 147, comprising the agent reporting to the platform addition of the file to the content accessible by the agent and, in response, the platform assigns a task to the agent to upload the file.

149. The method of claim 148, comprising in response to the task the agent determining if the file is present at the remote storage device, and uploads the representation of the file to the remote storage device when the file is determined to be absent.

150. The method of claim 147, wherein the upload task includes the agent copying at least one blob of a file from the device accessible by the agent to the remote storage device.

151. The method of claim 150, wherein the at least one blob comprises a representation of a fragment of the file, wherein the fragment is a component of the file.

152. The method of claim 97, wherein the plurality of tasks includes a delete task.

153. The method of claim 97, wherein a task comprises a plurality of phases including at least one of queued, pending, and completed, wherein the platform tracks the phase of each task of the plurality of tasks.

154. The method of claim 97, comprising each agent maintaining locally at the client device tasks assigned to the agent.

155. The method of claim 154, comprising the agent periodically polling the platform to identify assigned tasks.

156. The method of claim 97, wherein an agent comprises a hierarchy for locating task information needed to complete a task and located at a remote device.

157. The method of claim 156, wherein the hierarchy comprises the agent searching a local database of the client device hosting the agent.

158. The method of claim 157, wherein the hierarchy comprises the agent communicating with at least one peer agent of the plurality of agents to locate the task information.

159. The method of claim 158, wherein the agent comprises identity of peer agents that possess the task information.

160. The method of claim 158, wherein the hierarchy comprises the agent retrieving the task information from a remote storage device.

161. The method of claim 160, comprising the agent receiving from the platform the identity of peer agents in an ordered list and searching for the task information in accordance with the ordered list.

162. The method of claim 97, wherein each agent includes a plurality of components executing in parallel.

163. The method of claim 162, wherein the plurality of components include a provider component that retrieves tasks designated for the agent from the platform and stores retrieved tasks in a task database local to the client device hosting the agent.

164. The method of claim 163, wherein the plurality of components comprise a task execution component.

165. The method of claim 164, wherein the plurality of components include a runner component that monitors the task database, retrieves each task from the task database, and provides a retrieved task to the task execution component and designates the retrieved task to have a pending status.

166. The method of claim 165, comprising the task execution component executing the task, and reports status of task execution to the runner component.

167. The method of claim 166, comprising the runner component reporting the status to the task database.

168. The method of claim 167, wherein the plurality of components include an update component that monitors the task database for tasks having a completed status, and reports status information of completed tasks to the platform.

169. The method of claim 168, comprising the platform updating the plurality of databases in response to the status information.

170. The method of claim 97, comprising the plurality of agents generating the metadata, wherein metadata generated by an agent corresponds to the content to which the agent has access.

171. The method of claim 170, comprising the agent generating the metadata by scanning content of each file to which the agent has access.

172. The method of claim 171, comprising the agent generating the metadata by splitting the content of the file into a plurality of fragments, wherein each fragment comprises a variable size component of the file.

173. The method of claim 172, wherein the variable size fragments are between a pre-specified minimum length and maximum length.

174. The method of claim 172, comprising the agent generating the metadata by generating a plurality of blobs that represent the plurality of fragments, wherein each blob represents a fragment.

175. The method of claim 172, comprising the agent generating the plurality of blobs using a data fingerprinting algorithm that comprises running, for each byte in the content, a hash algorithm over components of the content, wherein the hash algorithm is set to identify a specified pattern of data.

176. The method of claim 175, wherein the generating of the blob comprises generating a description of the blob that includes a value of a hash at a point where the fragment represented by the blob was separated from a remaining portion of the file.

177. The method of claim 176, wherein the generating of the blob comprises generating an offset value based on a break point of the fragment represented by the blob.

178. The method of claim 177, wherein the generating of the blob comprises generating a hash of complete contents of the blob, wherein the hash of the complete contents of the blob is an identifier for the blob.

179. The method of claim 178, wherein the generating of the blob comprises generating a list of blobs representing the content of the file.

180. The method of claim 179, wherein the generating of the blob includes generating a content hash that comprises a hash of the list of blobs representing the content of the file, wherein the content hash is an identifier for the file.

181. The method of claim 180, wherein the generating of the blob includes generating a name hash that comprises a hash of a file name corresponding to the file.

182. The method of claim 181, wherein the generating of the blob includes generating a file hash that comprises a hash of a combination of the content hash and a name hash.

183. The method of claim 182, wherein the generating of the blob includes generating a path hash that comprises a hash of the file name and a file path corresponding to the content of the file.

184. The method of claim 183, wherein the generating of the blob includes generating a metadata hash that comprises a hash of file metadata of the file.

185. The method of claim 184, comprising the platform generating a record for the file, wherein the record comprises blob hashes of the file, the content hash, the file hash, the path hash and the metadata hash.

186. The method of claim 185, wherein the plurality of databases include a libraries database, wherein the libraries database comprises the record.

187. The method of claim 175, wherein the generating of the blob comprises generating a size of the blob.

188. The method of claim 175, wherein the file is described as a list of blobs comprising the file.

189. The method of claim 175, comprising each agent storing a blob locally at the client device hosting the agent, and transfers a blob that is previously unreported to central storage of the platform.

190. A method comprising:

coupling a platform comprising a processor to a plurality of databases;

forming a grid comprising a plurality of agents coupled to the platform, wherein each agent is an agent of the platform running on a client device;

generating metadata at each agent by hashing a plurality of fragments of content accessible by the agent, wherein the metadata corresponds to the content of memory accessible by a plurality of client devices corresponding to the plurality of agents;

providing the metadata to the platform instead of the content;

generating with the metadata a plurality of tasks including tasks controlling at least one of storing, transferring and processing of the content, wherein a task is a processing operation performed on content accessible by the agent responsible for the task; and

assigning the plurality of tasks to the plurality of agents.

191. A method comprising:

coupling a platform comprising a processor to a plurality of databases;

forming a grid comprising a plurality of agents coupled to the platform, wherein each agent is an agent of the platform running on a client device;

providing to the platform by each agent of each client device metadata of content of memory accessible by the client device;

determining with the metadata locations of the content;

generating goals with the metadata representing operations for maintaining a state of the content;

generating with the metadata a plurality of tasks corresponding to the goals; and

assign each task to an agent having access to the content that corresponds to the task, wherein a task is a processing operation directed by the platform to be performed by an agent on content accessible by the agent, wherein the metadata provided by plurality of agents includes information of the plurality of tasks performed by the plurality of agents.

192. A method comprising:

establishing a coupling between a platform comprising a processor and a plurality of databases;

forming a grid by coupling a plurality of agents to the platform, wherein each agent of the plurality of agents is an agent of the platform running on a client device;

generating metadata at each agent representing content of memory accessible by a plurality of client devices corresponding to the plurality of agents, wherein the generating comprises hashing a plurality of fragments of the content to generate a plurality of blobs representing the plurality of fragments;

providing the metadata to the platform instead of the content;

generating with the metadata a plurality of tasks, wherein a task is a processing operation performed on content accessible by the agent responsible for the task, wherein the plurality of tasks include tasks that at least one of monitor, store, transfer and process the content; and

assigning the plurality of tasks to the plurality of agents.

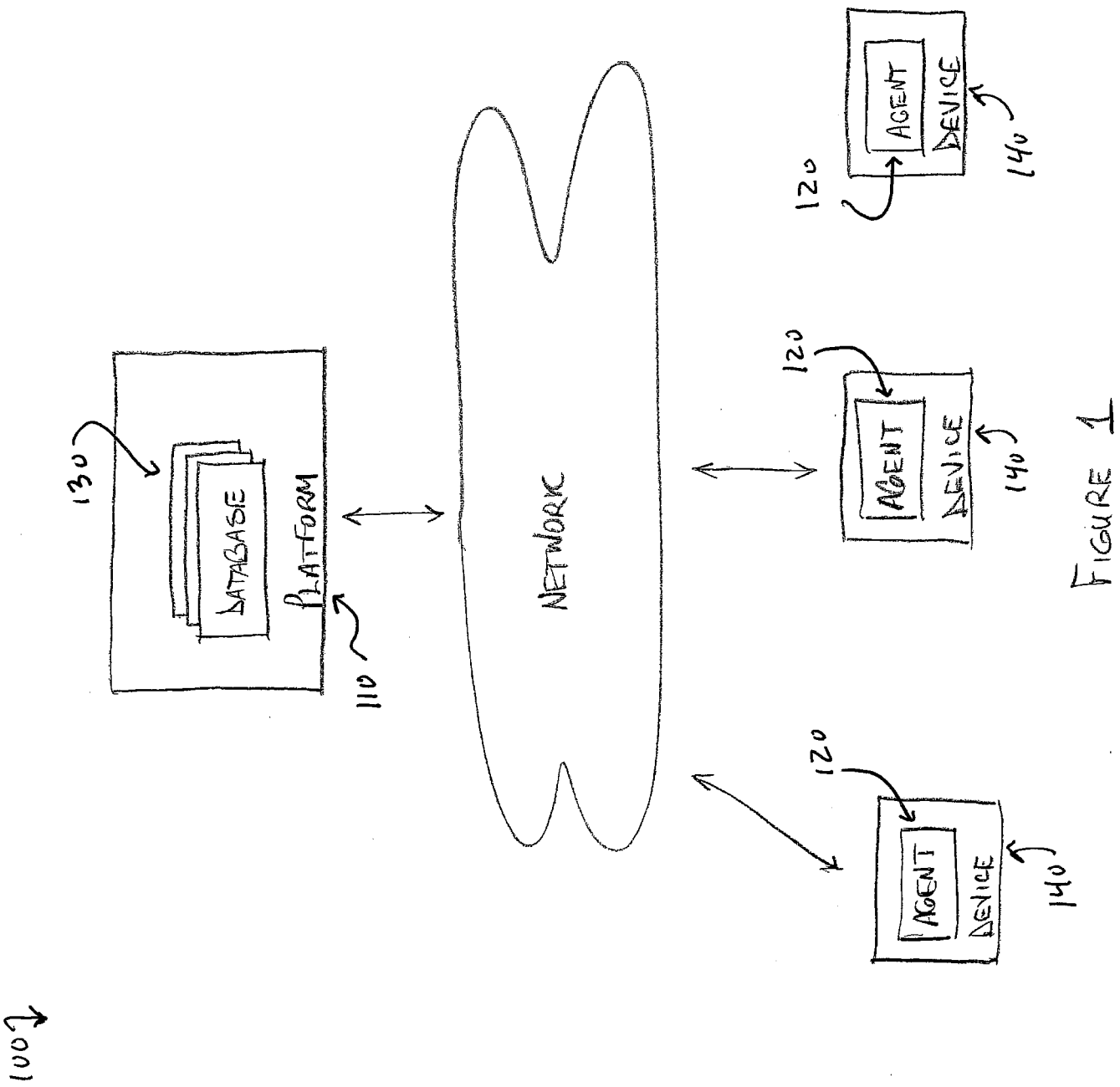

100
130
DATABASE
PLATFORM
110
NETWORK
120
AGENT
DEVICE
140
120
AGENT
DEVICE
140
120
AGENT
DEVICE
140

FIGURE 1

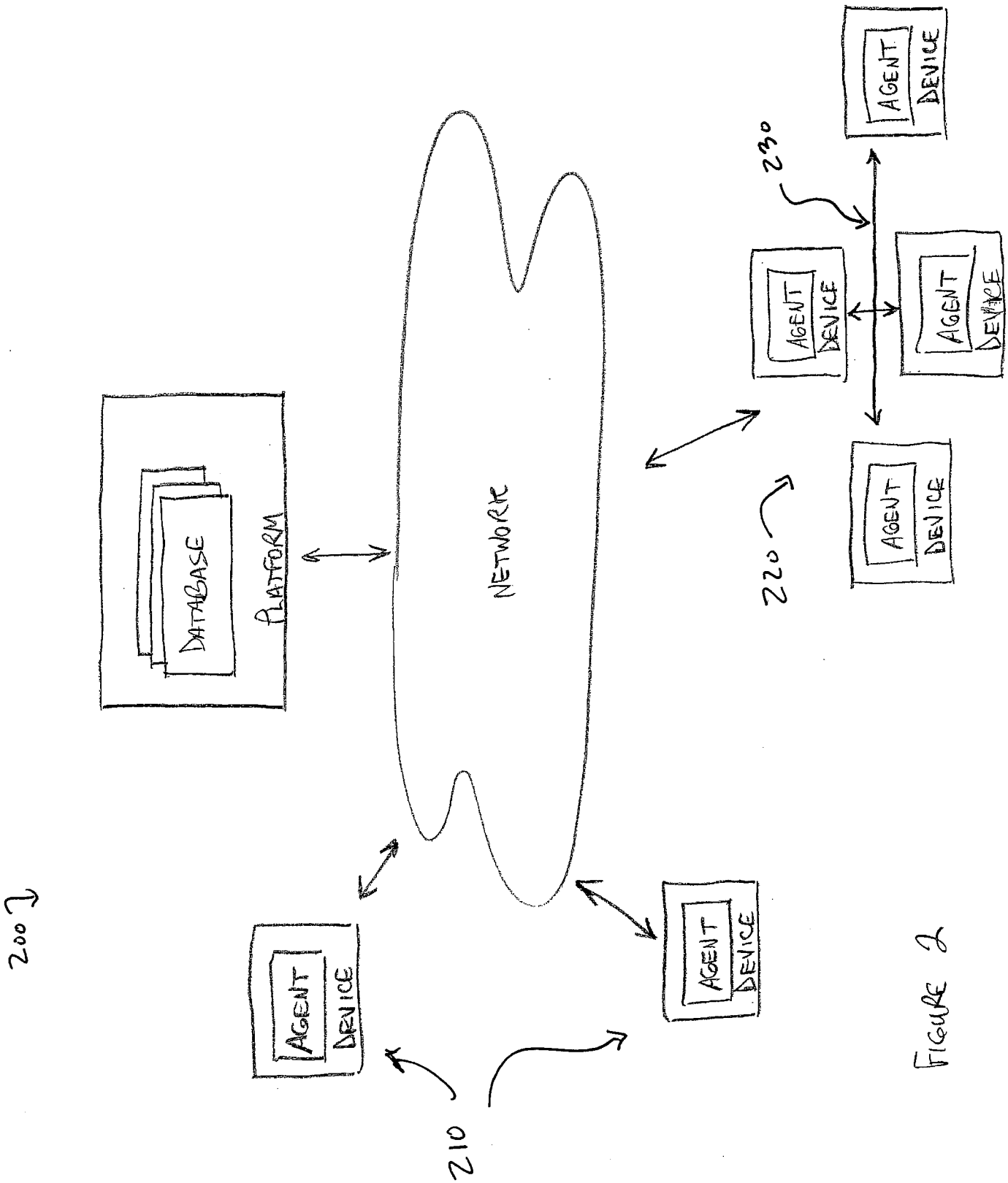
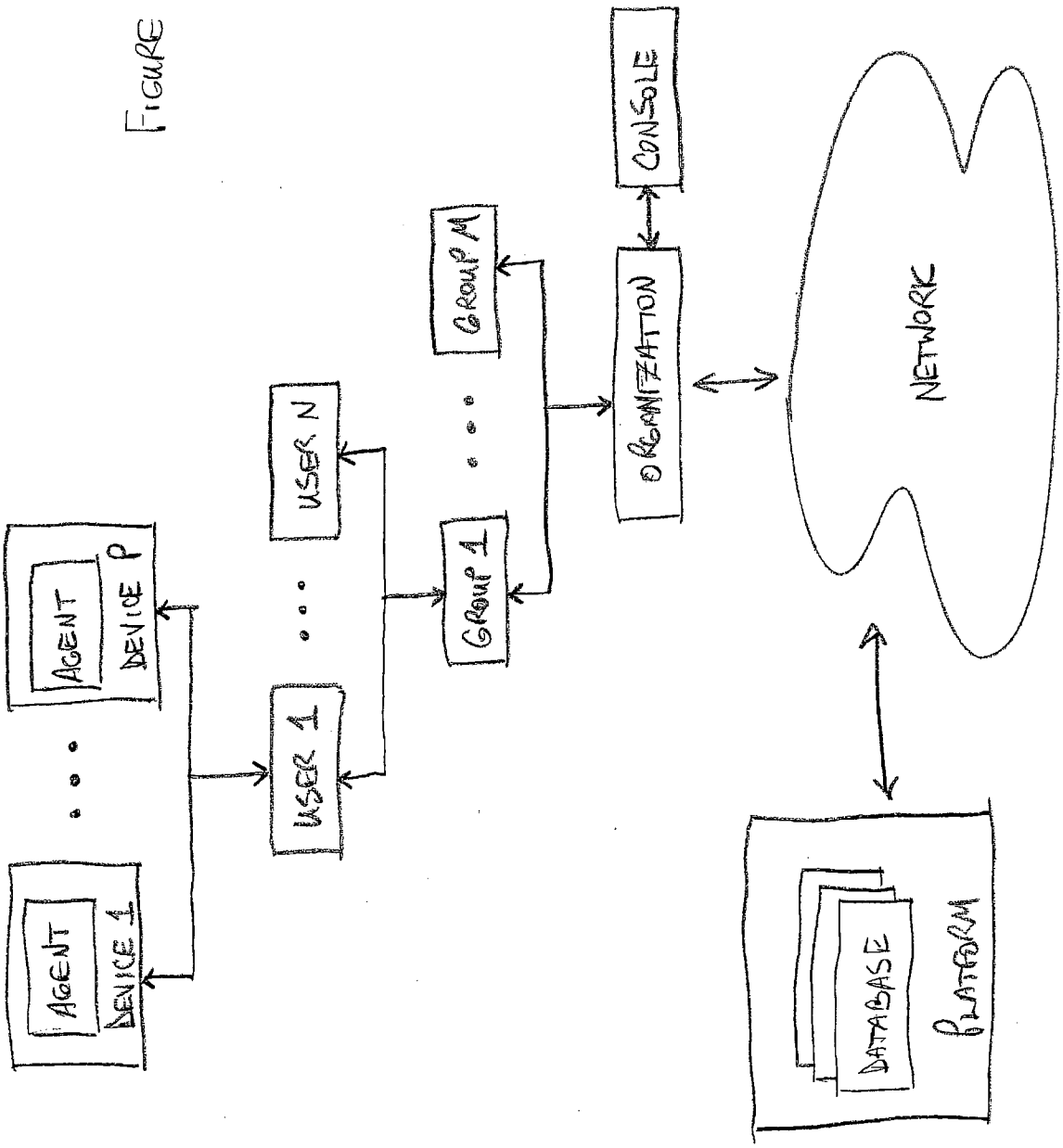

200
DATABASE
PLATFORM
NETWORK
AGENT DEVICE
AGENT DEVICE
AGENT DEVICE
AGENT DEVICE
AGENT DEVICE
AGENT DEVICE
210
220
230

FIGURE 2

FIGURE 3


AGENT
DEVICE 1
AGENT
DEVICE P
USER 1
USER N
GROUP 1
GROUP M
ORGANIZATION
CONSOLE
NETWORK
DATABASE
PLATFORM

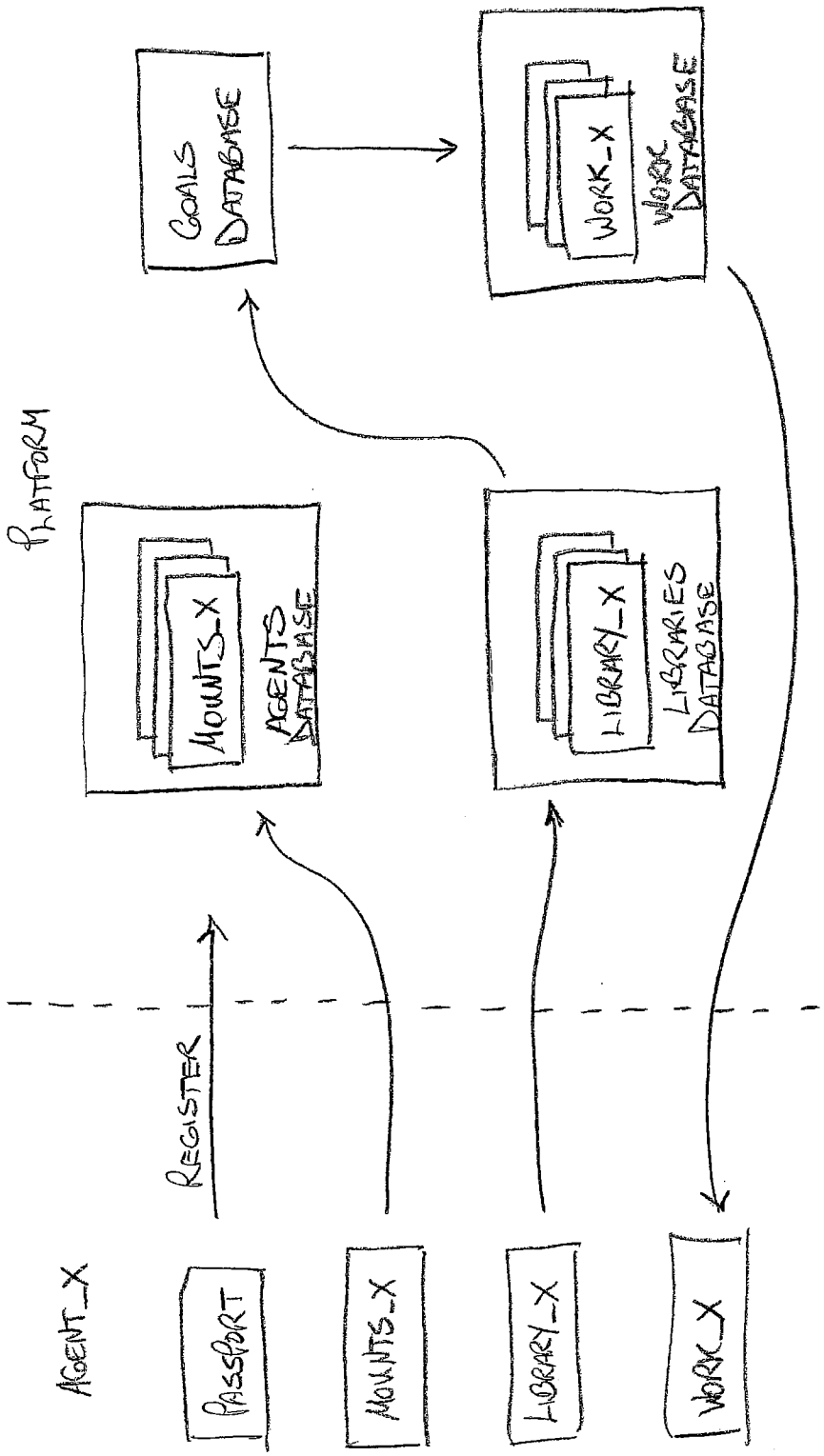

PLATFORM
GOALS DATABASE
WORK_X
WORK DATABASE
MOUNTS_X
AGENTS DATABASE
LIBRARY_X
LIBRARIES DATABASE
REGISTER
AGENT_X
PASSPORT
MOUNTS_X
LIBRARY_X
WORK_X

FIGURE 4A

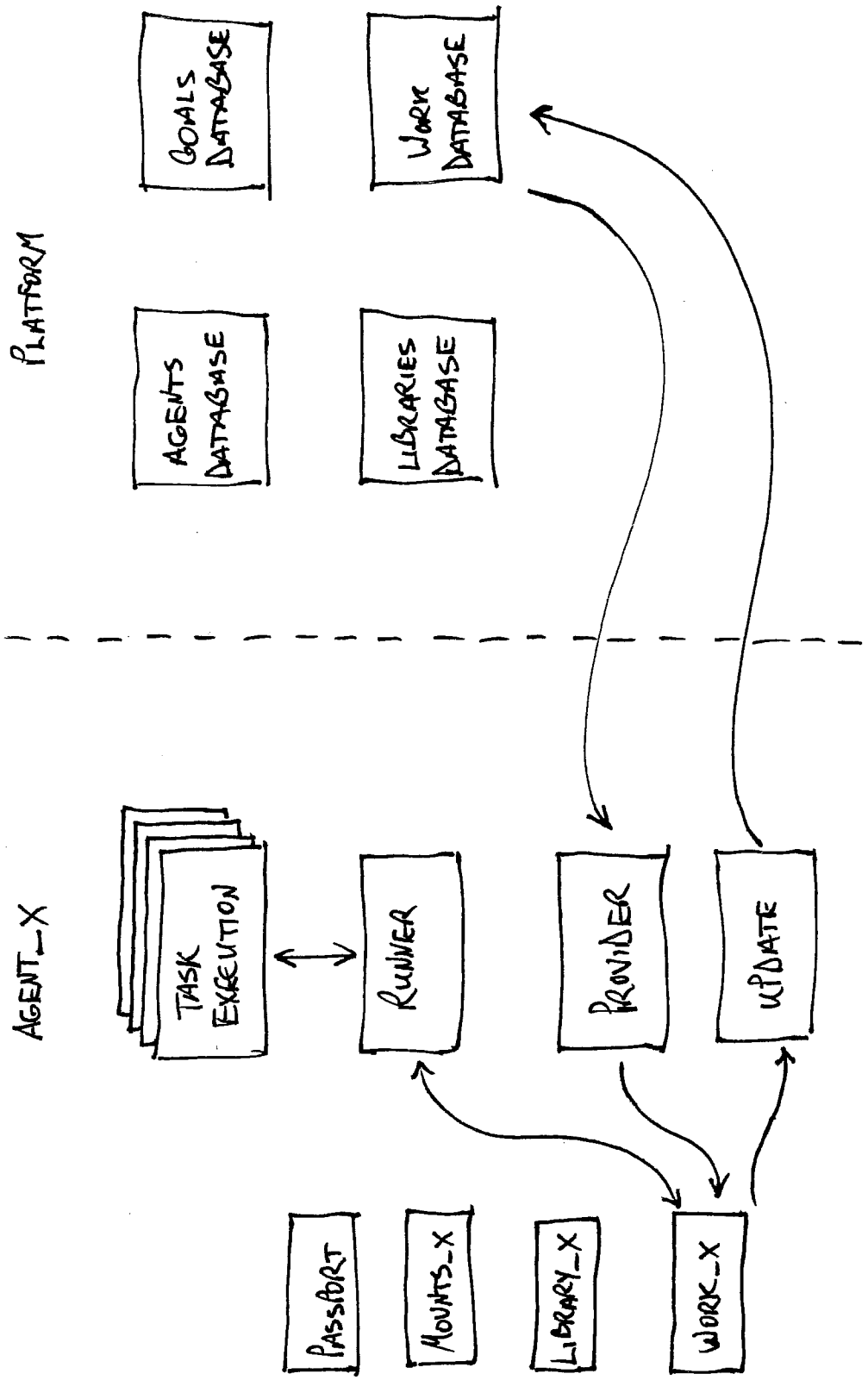

PLATFORM
GOALS DATABASE
WORK DATABASE
AGENTS DATABASE
LIBRARIES DATABASE
AGENT_X
TASK EXECUTION
RUNNER
PROVIDER
UPDATE
PASSPORT
MOUNTS_X
LIBRARY_X
WORK_X

FIGURE 4B

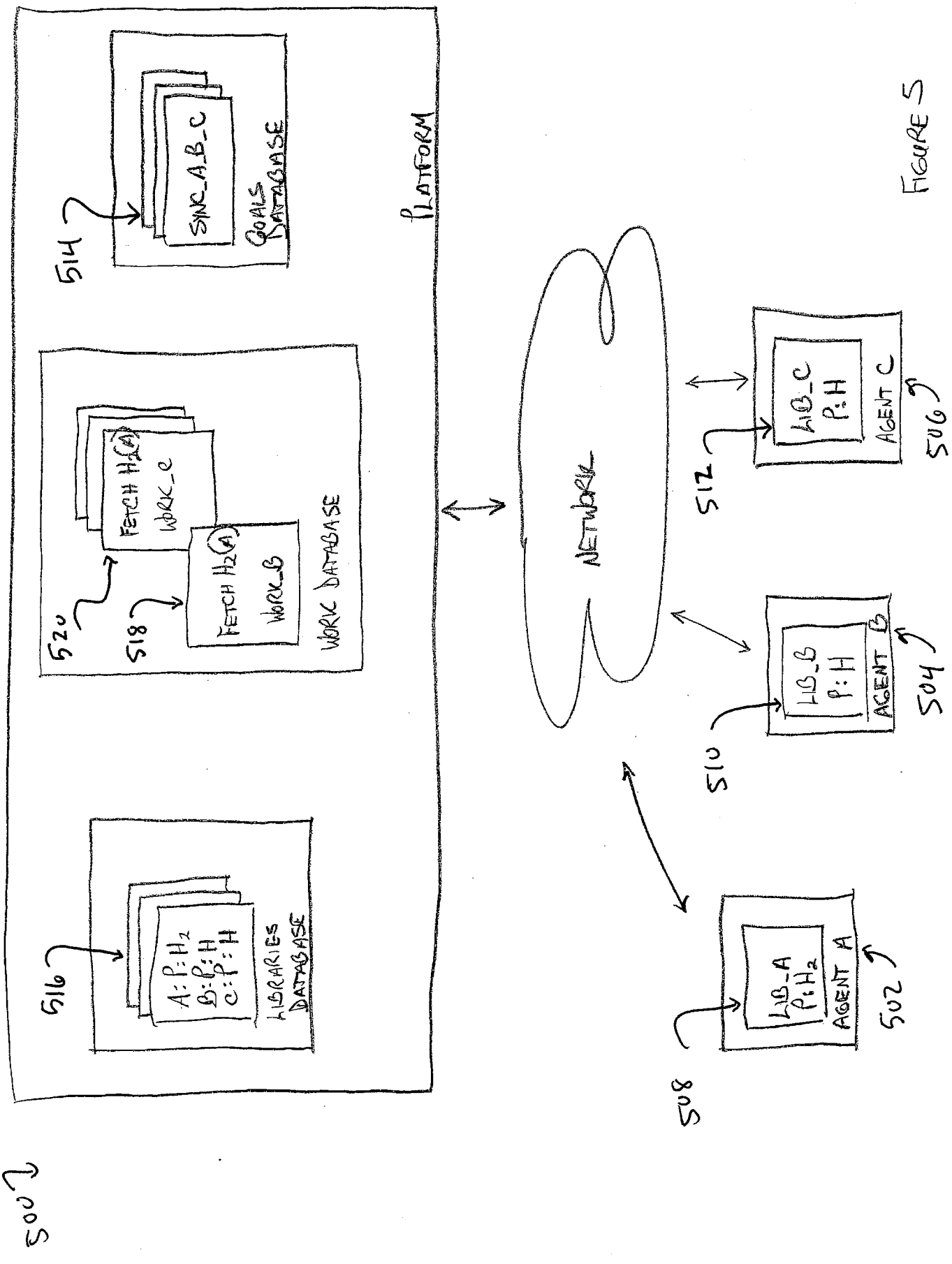

500
PLATFORM
514
SINC_A_B_C
GOALS DATABASE
520
FETCH H2(A)
WORK_C
518
FETCH H2(A)
WORK_B
WORK DATABASE
516
A: P: H2
B: P: H
C: P: H
LIBRARIES DATABASE
NETWORK
512
LIB_C
P: H
AGENT C
506
510
LIB_B
P: H
AGENT B
504
508
LIB_A
P: H2
AGENT A
502

FIGURE 5

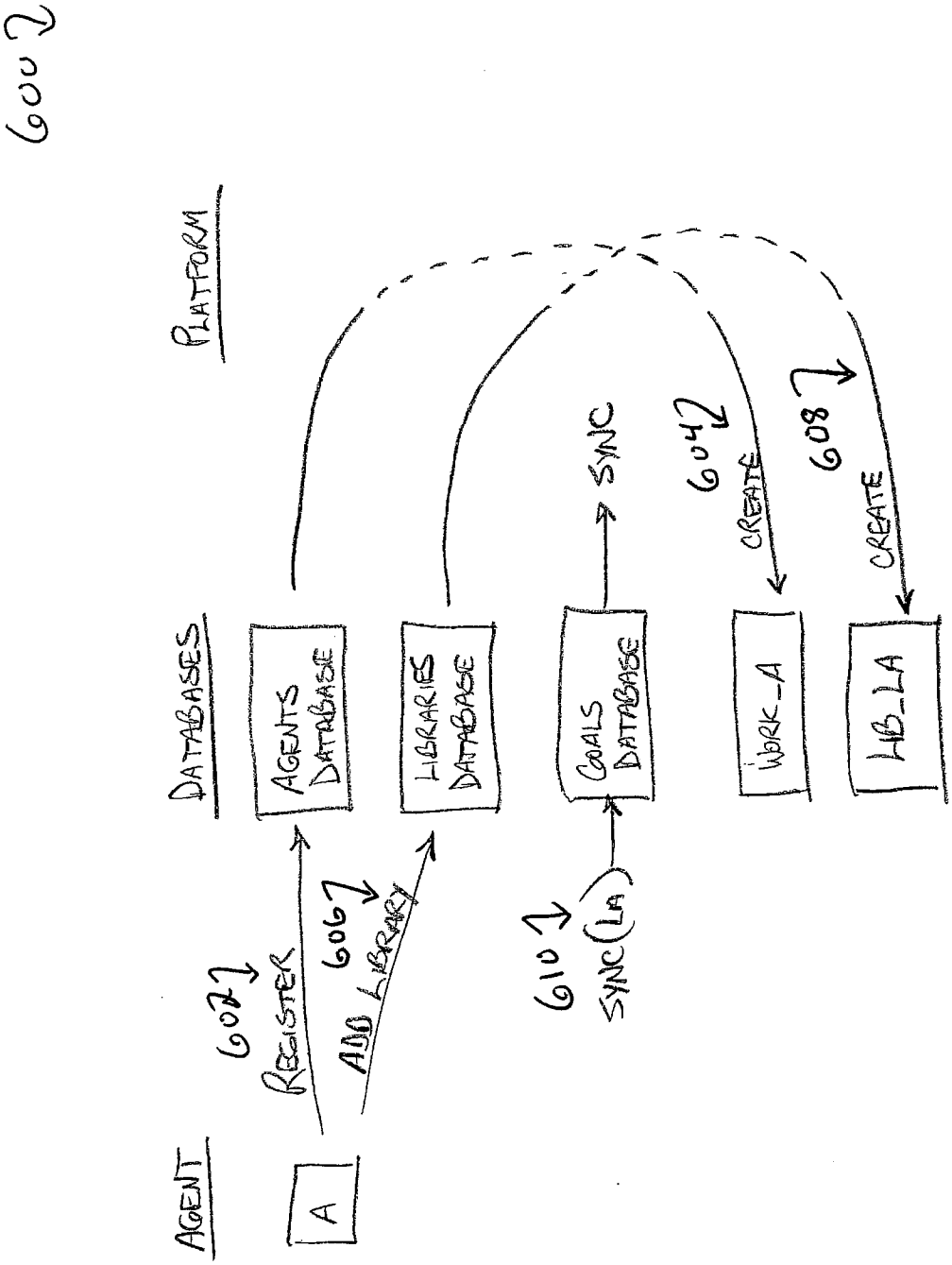

600
PLATFORM
DATABASES
AGENTS DATABASE
LIBRARIES DATABASE
GOALS DATABASE
WORK_A
HB_LA
SYNC
CREATE
CREATE
AGENT
A
REGISTER
ADD LIBRARY
SYNC (LA)
602
606
610
604
608

FIGURE 6

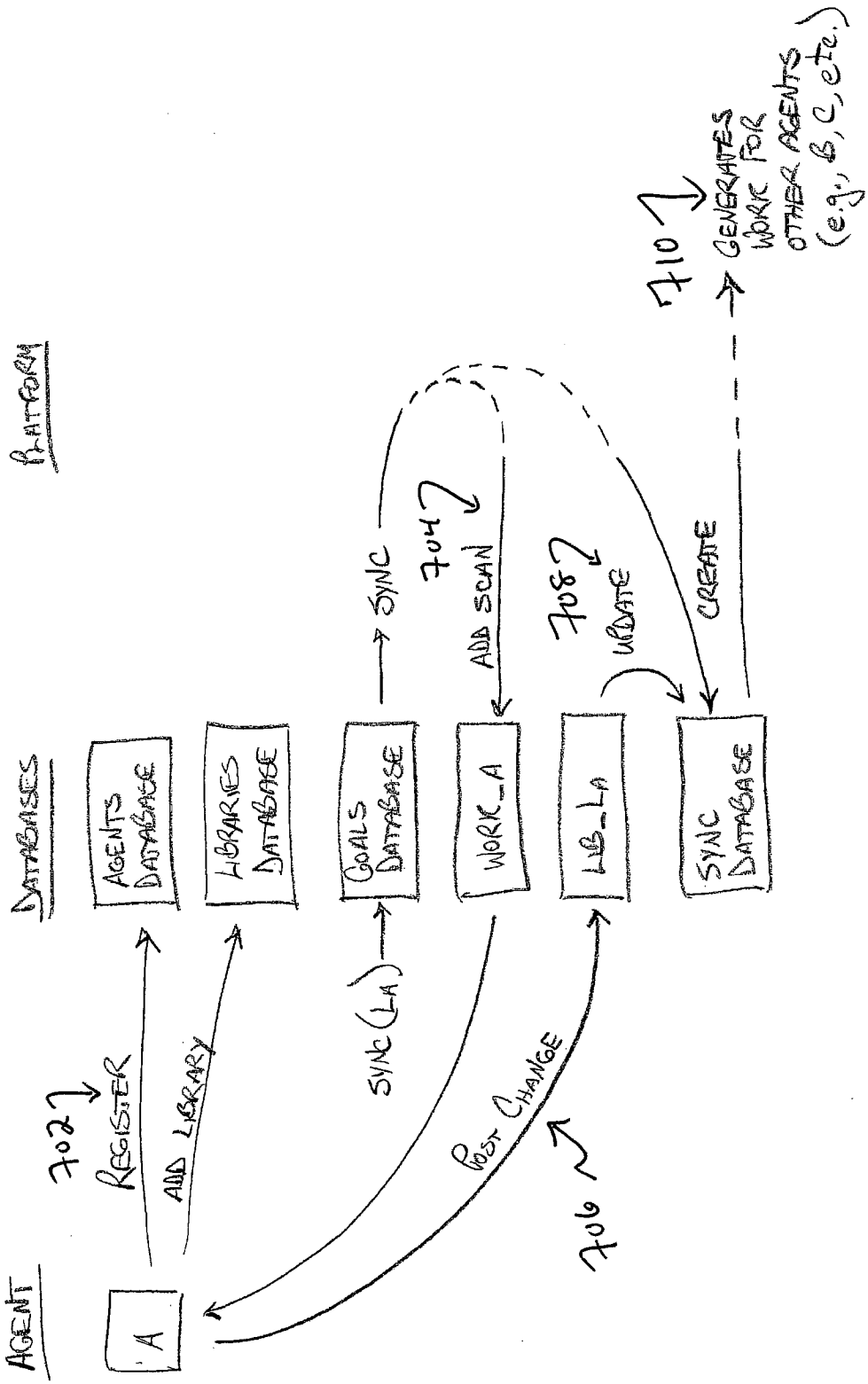

700
AGENT
A
PLATFORM
DATABASES
AGENTS DATABASE
LIBRARIES DATABASE
GOALS DATABASE
WORK_A
LIB_LA
SYNC DATABASE
702
REGISTER
ADD LIBRARY
SYNC (LA)
704
SYNC
ADD SCAN
706
POST CHANGE
708
UPDATE
CREATE
710
GENERATES WORK FOR OTHER AGENTS (e.g., B, C, etc.)

FIGURE 7