



US 20170161297A1

(19) **United States**(12) **Patent Application Publication**
KAISER et al.(10) **Pub. No.: US 2017/0161297 A1**(43) **Pub. Date: Jun. 8, 2017**(54) **DATA SYNCHRONIZATION IN A STORAGE NETWORK****Publication Classification**(71) Applicants: **CHRISTIAN M. KAISER**, Cupertino, CA (US); **Julie A. Pitt**, Livermore, CA (US); **Rick Pasetto**, Cupertino, CA (US); **Stephen Sewerynek**, Cupertino, CA (US)(72) Inventors: **CHRISTIAN M. KAISER**, Cupertino, CA (US); **Julie A. Pitt**, Livermore, CA (US); **Rick Pasetto**, Cupertino, CA (US); **Stephen Sewerynek**, Cupertino, CA (US)(21) Appl. No.: **15/353,658**(22) Filed: **Nov. 16, 2016****Related U.S. Application Data**

(63) Continuation of application No. 14/137,691, filed on Dec. 20, 2013.

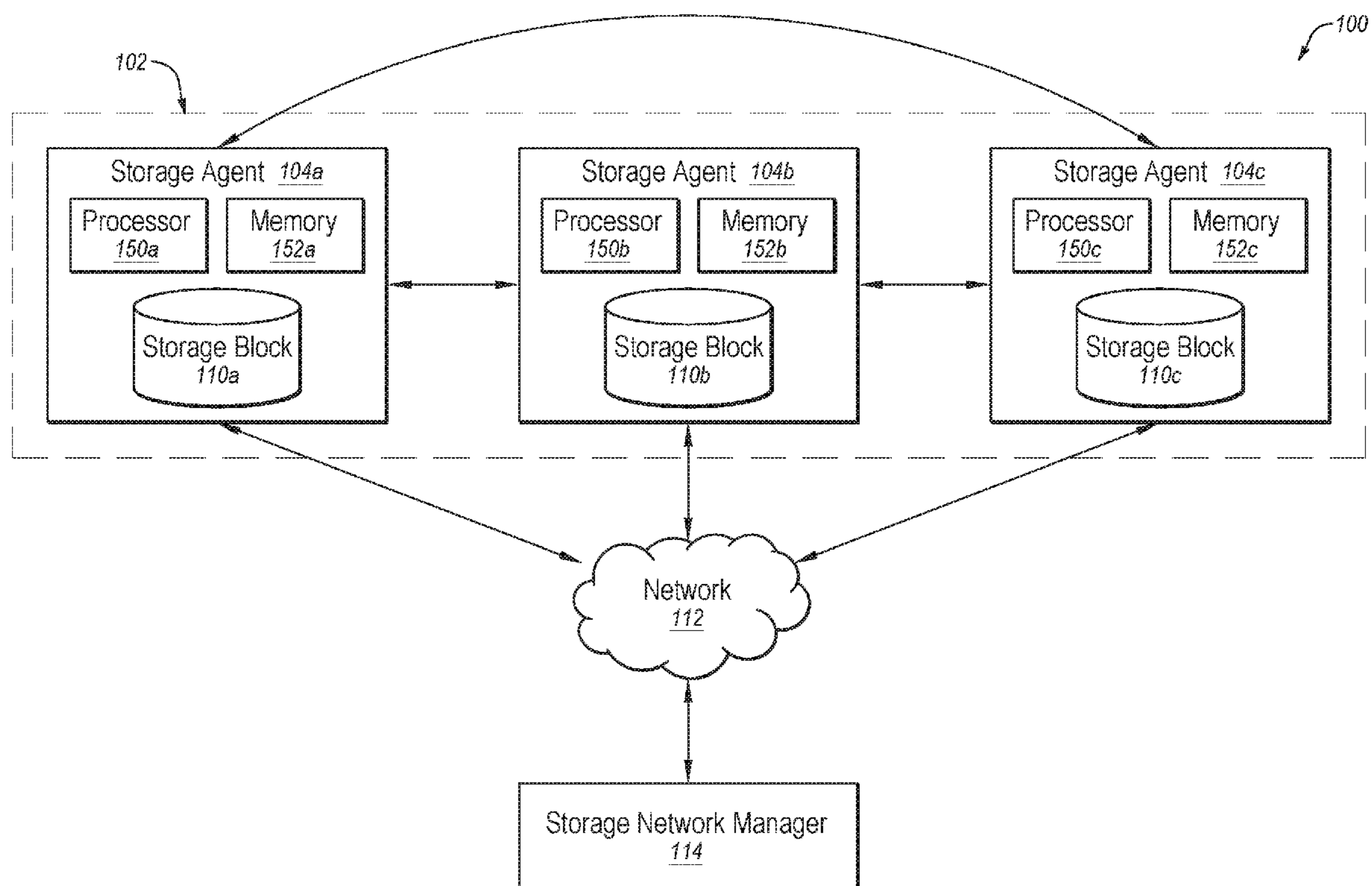
(60) Provisional application No. 61/750,324, filed on Jan. 8, 2013, provisional application No. 61/750,319, filed on Jan. 8, 2013.

(51) **Int. Cl.****G06F 17/30** (2006.01)**G06F 12/02** (2006.01)**G06F 3/06** (2006.01)(52) **U.S. Cl.**CPC .. **G06F 17/30174** (2013.01); **G06F 17/30581** (2013.01); **G06F 3/0604** (2013.01); **G06F 3/067** (2013.01); **G06F 12/023** (2013.01)

(57)

ABSTRACT

A method of synchronizing data may include receiving, by a first storage agent, a recent second-agent synchronization point. The recent second-agent synchronization point may be associated with a recent second-agent synchronization time of a second-agent synchronization between a second storage agent and a third storage agent. The method may further include determining, based on the recent second-agent synchronization point, a first-agent baseline synchronization point. The first-agent baseline synchronization point may be associated with a first-agent baseline synchronization time of a first-agent synchronization between the first storage agent and the third storage agent that occurred before the second-agent synchronization time. Additionally, the method may include establishing a first-agent baseline of synchronization of data between the first storage agent and the second storage agent based on the first-agent baseline synchronization point.



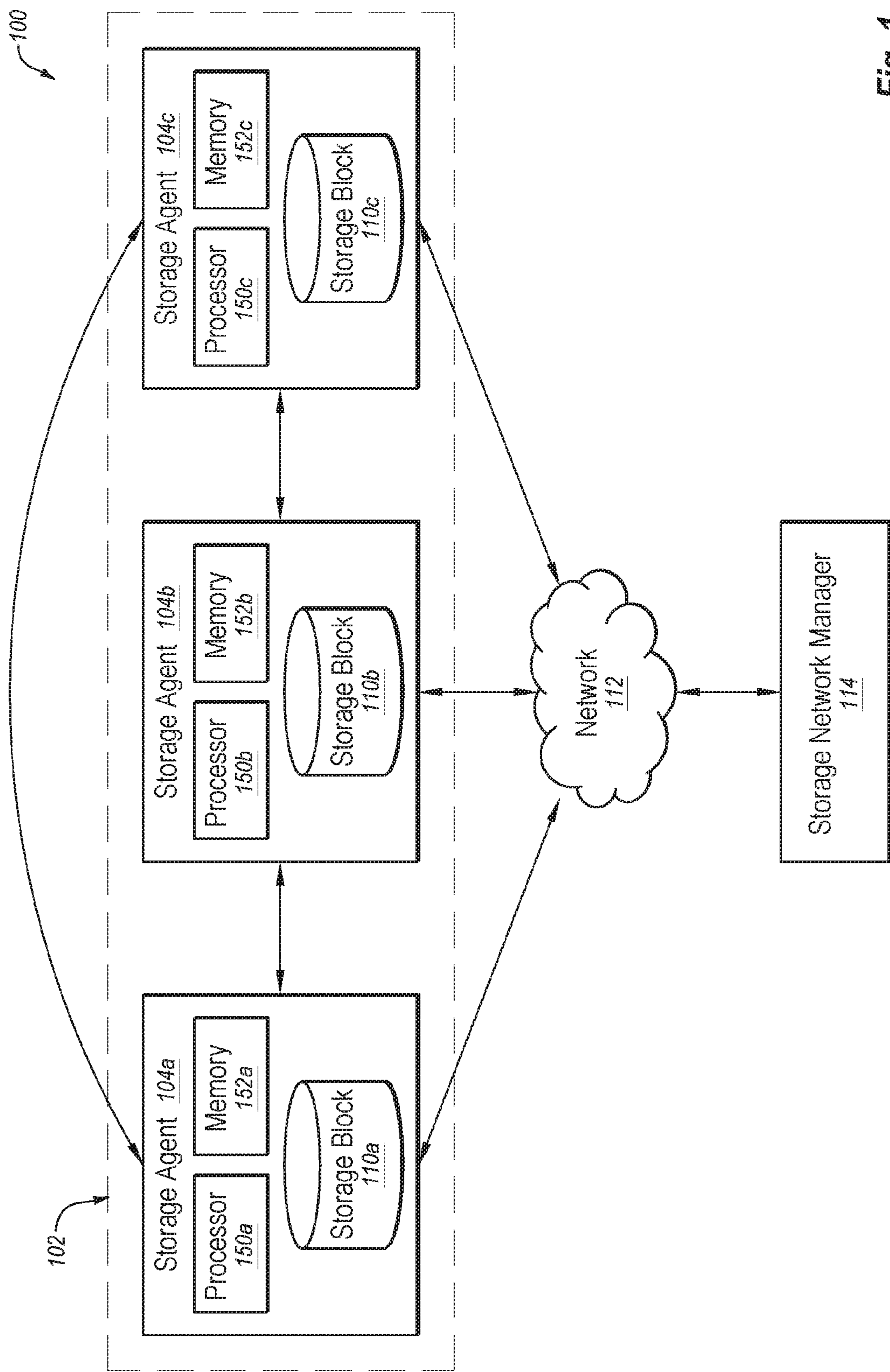


Fig. 1

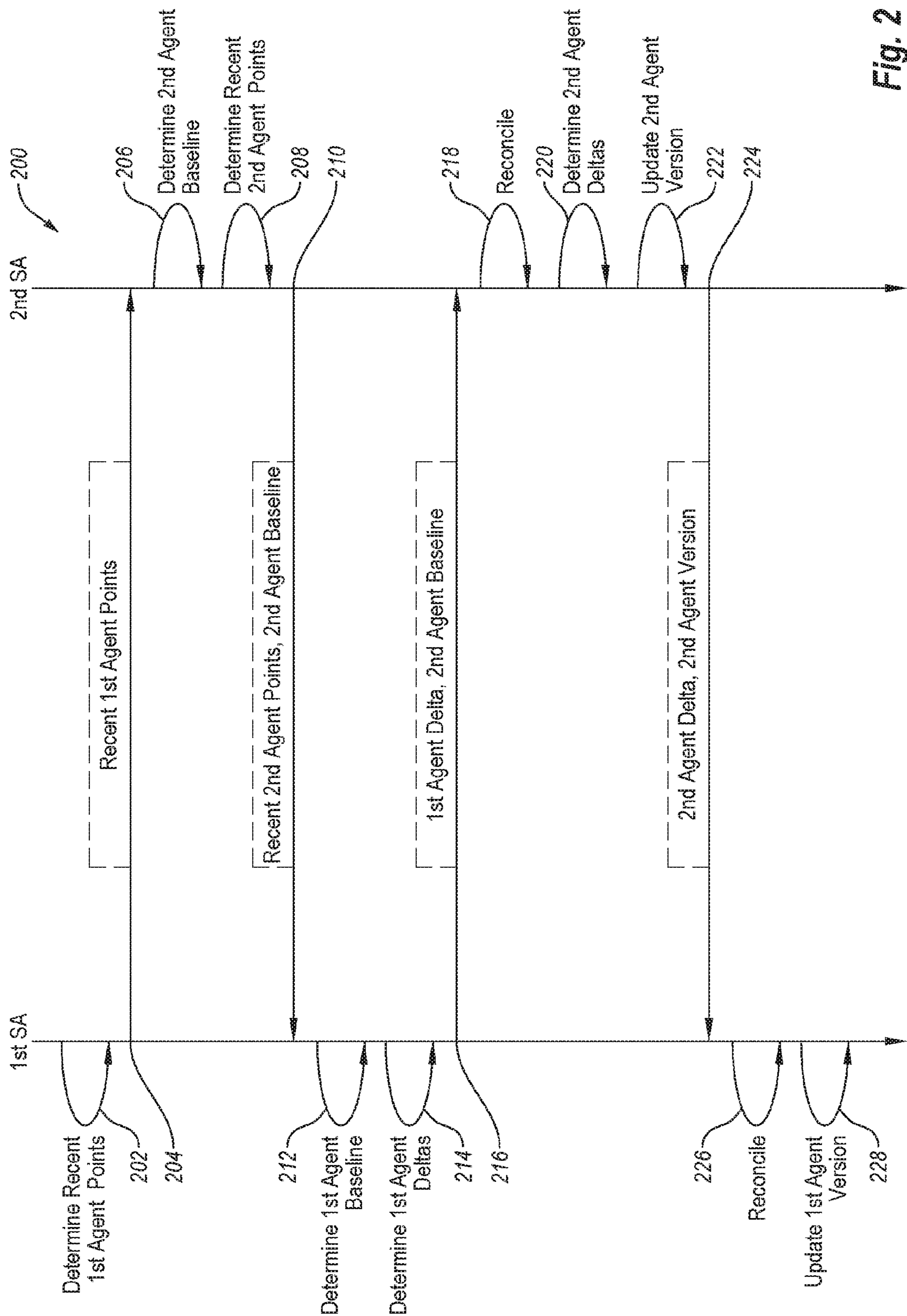
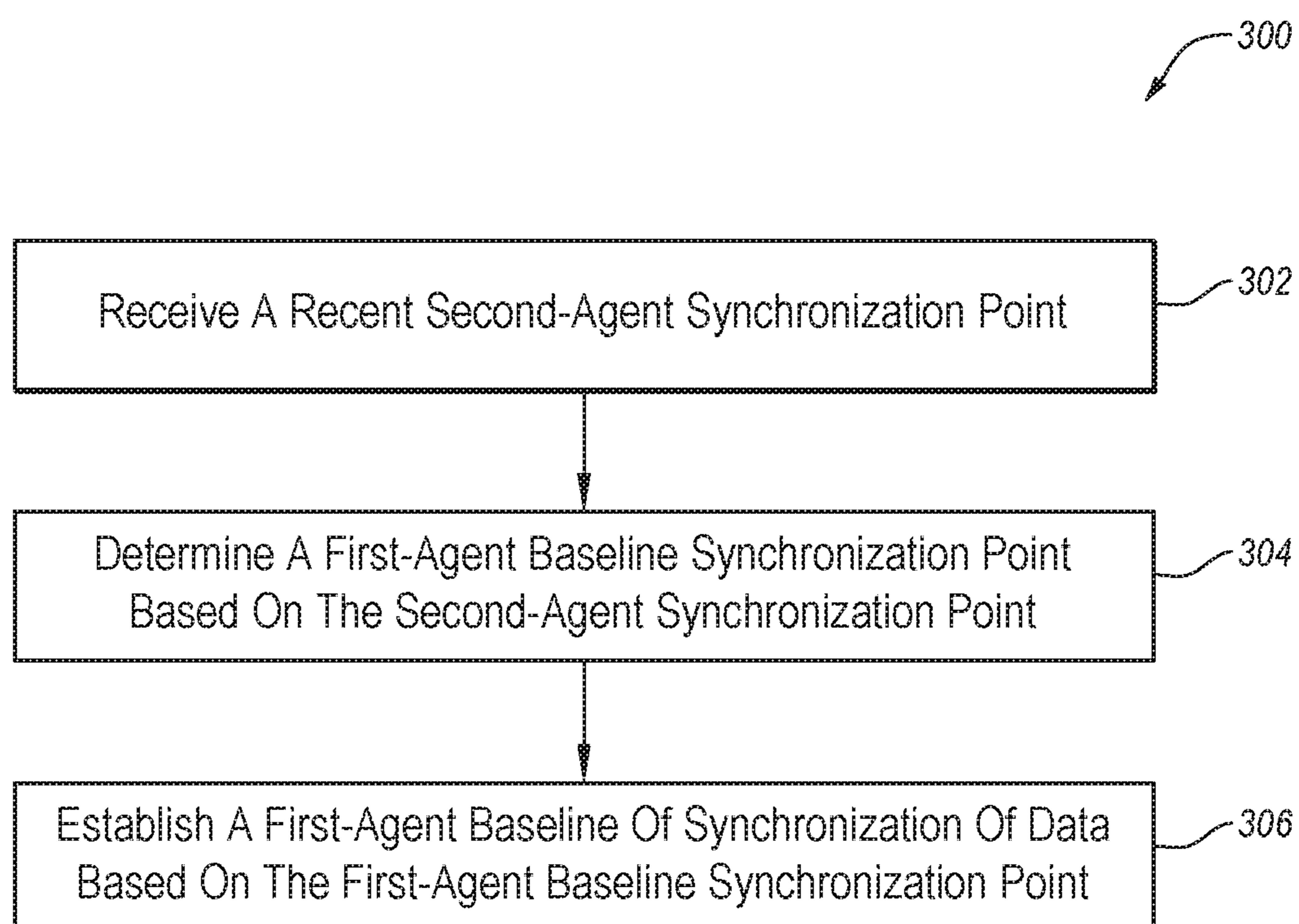


Fig. 2

**Fig. 3**

DATA SYNCHRONIZATION IN A STORAGE NETWORK

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application which is a continuation of U.S. application Ser. No. 14/137,691, filed Dec. 20, 2013, which is based upon and claims the benefit of priority of U.S. Provisional Application No. 61/750,324, filed on Jan. 8, 2013, and of U.S. Provisional Application No. 61/750,319, filed on Jan. 8, 2013. The forgoing applications are incorporated herein by reference in their entirety.

FIELD

[0002] The embodiments discussed herein are related to data synchronization in a storage network.

BACKGROUND

[0003] The amount of personal data (e.g., photos, video, documents, etc.) stored by users is increasing. Additionally, the number and types of devices that store, create, download, etc. personal data are also increasing. However, many methods and systems of storing and managing personal data across multiple devices are cumbersome, time consuming, provide inadequate redundancy, and may not allow for easy accessibility of the personal data on different devices, among other things.

[0004] The subject matter claimed herein is not limited to embodiments that solve any disadvantages or that operate only in environments such as those described above. Rather, this background is only provided to illustrate one example technology area where some embodiments described herein may be practiced.

SUMMARY

[0005] According to an aspect of an embodiment, a method of synchronizing data may include receiving, by a first storage agent, a recent second-agent synchronization point. The recent second-agent synchronization point may be associated with a recent second-agent synchronization time of a second-agent synchronization between a second storage agent and a third storage agent. The method may further include determining, based on the recent second-agent synchronization point, a first-agent baseline synchronization point. The first-agent baseline synchronization point may be associated with a first-agent baseline synchronization time of a first-agent synchronization between the first storage agent and the third storage agent that occurred before the second-agent synchronization time. Additionally, the method may include establishing a first-agent baseline of synchronization of data between the first storage agent and the second storage agent based on the first-agent baseline synchronization point.

[0006] The object and advantages of the embodiments will be realized and achieved at least by the elements, features, and combinations particularly pointed out in the claims.

[0007] It is to be understood that both the foregoing general description and the following detailed description are exemplary and explanatory and are not restrictive of the invention, as claimed.

BRIEF DESCRIPTION OF THE DRAWINGS

[0008] Example embodiments will be described and explained with additional specificity and detail through the use of the accompanying drawings in which:

[0009] FIG. 1 illustrates a block diagram of an example storage system;

[0010] FIG. 2 illustrates an example synchronization process that may be performed between a first storage agent (1st SA) and a second storage agent (2nd SA); and

[0011] FIG. 3 is a flowchart of an example method of synchronizing data.

DESCRIPTION OF EMBODIMENTS

[0012] As described in further detail below, storage agents of a storage network may be configured to perform synchronizations of data with each other based on previous synchronizations of the storage agents with one or more third-party storage agents of the storage network that are not directly part of the synchronization between the storage agents currently synchronizing with each other. Such a synchronization scheme may allow for reducing the amount of data that may be transferred during the synchronization than traditional synchronization schemes, while also allowing for consistent synchronization within the storage network. As such, the storage agents of the storage system may be mobile while also allowing for consistent and more efficient synchronization within the storage network. Embodiments of the present disclosure will be explained with reference to the accompanying drawings.

[0013] FIG. 1 illustrates a block diagram of an example storage system 100 configured according to at least one embodiment of the present disclosure. The storage system 100 may include a storage network 102 that includes storage agents (SA) 104a-104c. Although the storage system 100 is illustrated as including a single storage network 102 with three different storage agents 104 included therein, the system 100 may include any number of storage networks 102 that may each include any number of storage agents 104.

[0014] In some embodiments, the storage system 100 is configured to store, organize, and manage data files such as photos, videos, documents, etc. In some embodiments, the data files may be included in data objects that may also include metadata that may provide information about the data files. The term “data” in the present disclosure may refer to any suitable information that may be stored by the storage agents 104 and may include one or more data files, metadata, or any combination thereof.

[0015] Additionally, the term “shared data” may refer to any data that may be shared by, stored on, and/or synchronized between more than one storage agent 104 of the storage network 102. For example, the shared data may include data files, metadata, data objects, data object root and tree structure information, storage agent status information (explained in further detail below), digital rights management (DRM) license information (explained in further detail below), sharing ticket information (explained in further detail below), any other suitable information, or any combination thereof.

[0016] The storage system 100 may be configured to organize and manage the data stored in storage blocks 110 that are associated with the storage agents 104a-104c in an automated fashion that may reduce the amount of input

required by a user. As such, the storage system **100** may facilitate organization of and access to the data stored by the storage blocks **110** within the storage network **102**.

[0017] The storage agents **104** may each be associated with a processor **150**, memory **152**, and a storage block **110**. For example, in the illustrated embodiment, the storage agent **104a** may include a processor **150a**, memory **152a**, and a storage block **110a**; the storage agent **104b** may include a processor **150b**, memory **152b**, and a storage block **110b**; and the storage agent **104c** may include a processor **150c**, memory **152c**, and a storage block **110c**.

[0018] The processors **150** may include, for example, a microprocessor, microcontroller, digital signal processor (DSP), application specific integrated circuit (ASIC), a Field Programmable Gate Array (FPGA), or any other digital or analog circuitry configured to interpret and/or to execute program instructions and/or to process data. In some embodiments, the processors **150** may interpret and/or execute program instructions and/or process data stored in their associated memories **152** and/or one or more of the storage blocks **110**.

[0019] The memories **152** may include any suitable computer-readable media configured to retain program instructions and/or data for a period of time. By way of example, and not limitation, such computer-readable media may include tangible or non-transitory computer-readable storage media including Random Access Memory (RAM), Read-Only Memory (ROM), Electrically Erasable Programmable Read-Only Memory (EEPROM), Compact Disk Read-Only Memory (CD-ROM) or other optical disk storage, magnetic disk storage or other magnetic storage devices, flash memory devices (e.g., solid state memory devices), a specific molecular sequence (e.g., DNA or RNA), or any other storage medium which may be used to carry or store desired program code in the form of computer-executable instructions or data structures and which may be accessed by the processors **150**. Combinations of the above may also be included within the scope of computer-readable media. Computer-executable instructions may include, for example, instructions and data that cause a general purpose computer, special purpose computer, or special purpose processing device (e.g., the processors **150**) to perform a certain function or group of functions.

[0020] The storage blocks **110** may also be any suitable computer-readable medium configured to store data. The storage blocks **110** may store shared data that may be substantially the same across different storage blocks **110** and may also store data that may only be found on a particular storage block **110**.

[0021] In some embodiments, one or more of the storage agents **104** may be included with any suitable device that may include the components of the storage agents **104**. For example, the storage agents **104** may be included in a cloud storage server, a mobile phone, a tablet computer, a personal computer, a laptop computer, a camera, a personal digital assistant (PDA), a smartphone, a music player, a video player, an external hard drive, etc. Additionally, although the storage agents **104** are explicitly depicted as including the processors **150**, the memories **152** and the storage blocks **110**, different implementations may have different configurations. For example, in some embodiments, the storage agents **104** may be modules included in the memories **152** of their associated devices with computer-executable instructions configured to cause the processors **150** of their asso-

ciated devices to perform operations associated with managing data that may be stored on the storage blocks **110**.

[0022] The devices associated with the storage agents **104** may include any device that may allow for communication of data between the storage agents **104**. Accordingly, the devices may provide some sort of communication capability between the storage agents **104** such as Internet connectivity, Local Area Network (LAN) connectivity, Wide Area Network (WAN) connectivity, Bluetooth connectivity, 3G connectivity, 4G connectivity, LTE connectivity, Wireless Fidelity (Wi-Fi) connectivity, Machine-to-Machine (M2M) connectivity, Device-to-Device (D2D) connectivity, any other suitable communication capability, or any suitable combination thereof.

[0023] In the illustrated embodiment, the storage agents **104** are depicted as performing communications with each other via direct communications between the storage agents **104**. Additionally or alternatively, the storage agents **104** may communicate with each other through a communication network **112**. In some embodiments, the communication network **112** may include, either alone or in any suitable combination, the Internet, an Intranet, a local Wi-Fi network, a wireless LAN, a mobile network (e.g., a 3G, 4G, and/or LTE network), a LAN, a WAN, or any other suitable communication network.

[0024] In some embodiments, the storage agents **104** may act similar to clients or servers included in an object-based file system. For instance, the storage agents **104** may be configured to implement protocols associated with communicating data within the storage network **102** and the storage system **100**. Additionally, some storage blocks **110** managed by the storage agents **104** may be configured to store only metadata included in various data objects, while other storage blocks **110** may be configured to store metadata and data files included in the various data objects.

[0025] In some embodiments, to manage and provide information related to the storage of data in the storage network **102**, a catalog of data may be generated and managed for the storage network **102**. For example, in some embodiments, the catalog may include information such as which storage blocks **110** may be locally storing data objects, individual data files, and/or any other metadata. The catalog may also include any other metadata that may be associated with the data files. In some embodiments, the catalog may include a collection of all the metadata of the data objects stored in the storage network **102**. Accordingly, the catalog may be used to determine which storage block **110** has certain data stored thereon as well as other information about the data stored on the different storage blocks **110**. As such, the storage agents **104** may know from where to access data if the data is not stored locally on their respective storage blocks **110**.

[0026] In some embodiments, the catalog, or any other shared data, may be stored by and synchronized between each of the storage blocks **110** based on synchronizations that may be managed and performed by the storage agents **104**. The synchronization process is described in further detail below. Because the synchronization of data between the storage blocks **110** is managed and performed by the storage agents **104**, use of the terms “synchronization of storage agents,” “synchronization between storage agents,” and the like, may refer to and include the synchronization of storage blocks **110**. Additionally, in the illustrated embodiment, the storage agents **104** are depicted as including the

storage blocks **110** such that data that is described as being stored on the storage agents **104** may actually be stored on the storage blocks **110**. Further, in embodiments where a particular storage agent **104** is a module configured to manage a particular storage block **110**, representations in the present disclosure that describe data being stored on the particular storage agent **104** may refer to data stored on the particular storage block **110** being managed by the particular storage agent **104** at that particular time.

[0027] In some embodiments, the shared data may be updated or changed such that the shared data may have different versions. In these or other embodiments, the storage agents **104** may assign version numbers (referred to as “locally-assigned version numbers”) to the shared data to track which version of the shared data is stored on their associated storage blocks **110**. In some instances, the storage agents **104** may increment the locally-assigned version numbers of shared data when the shared data is updated on the storage blocks **110** managed by the storage agents **104**. Accordingly, the locally-assigned version numbers may indicate and relate to a particular version of the shared data that may be stored on the storage blocks **110**. However, the locally-assigned version numbers of particular shared data on one storage block **110** may differ from the version number of the same version of the particular shared data on another storage block **110**. Accordingly, the locally-assigned version numbers of the shared data may vary between storage agents **104** and their associated storage blocks **110** even though the actual versions of the shared data may be the same on the storage blocks **110**.

[0028] In some embodiments, the storage agents **104** may also have a version number, which may indicate which versions of shared data may be stored on the storage agents **104** (e.g., the storage blocks **110** associated with the synchronization). The storage agent version number may be tied to a particular storage block **110** associated with the synchronization being performed by a particular storage agent **104**. Therefore, in some embodiments, for example in instances where the same storage agent **104** may manage synchronizations of different storage blocks **110** of different storage blocks **110**, the storage agent version number described herein may vary depending on which storage block **110** is associated with the synchronization. In some embodiments, the highest value of locally-assigned version numbers of shared data on a particular storage agent **104** may be used as the version number of the particular storage agent **104**.

[0029] For example, the storage agent **104a** may have stored thereon shared data “A” with a locally-assigned version number “1,” shared data “B” with a locally-assigned version number “2,” shared data “C” with a locally-assigned version number “2,” shared data “D” with a locally-assigned version number “3,” and shared data “E” with a locally-assigned version number “4.” In the illustrated example, the version number of the storage agent **104a** may accordingly have a version number “4” because that is the highest locally-assigned version number of the shared data stored on the storage agent **104a**.

[0030] In some embodiments, when a particular storage agent **104** performs synchronization with another storage agent **104**, the particular storage agent **104** and the other storage agent **104** may each generate a synchronization point, which may indicate a point in time that the particular storage agent **104** synchronized with the other storage agent

104. The synchronization points may also include version numbers of the storage agents **104** involved in the synchronization, identifiers of the storage agents **104**, and/or locally-assigned version numbers of the shared data involved in the synchronization. Similar to the version numbers of the storage agents **104**, the storage agent identifiers may be tied to a particular storage block **110** associated with the synchronization being performed by a particular storage agent **104**. Therefore, in some embodiments, for example in instances where the same storage agent **104** may manage synchronizations of different storage blocks **110** of different storage blocks **110**, the storage agent identifier described herein may vary depending on which storage block **110** is associated with the synchronization.

[0031] For example, in some embodiments, when the storage agent **104a** performs a synchronization with the storage agent **104b**, the storage agent **104a** may generate a synchronization point that may include any one or more of: a timestamp associated with the synchronization (also referred to as a “synchronization time”); the version number of the storage agent **104a** at the time of the synchronization, which may be stored as a local-agent version number; an identifier of the storage agent **104b**, which may be stored as a remote-agent ID; the version number of the storage agent **104b** at the time of the synchronization, which may be stored as a remote-agent version number; the locally-assigned version numbers—as assigned by the storage agent **104b**—of the shared data involved in the synchronization between the storage agents **104a** and **104b**, which may be stored as remote-agent assigned version numbers; and the locally-assigned version numbers—as assigned by the storage agent **104a**—of the shared data involved in the synchronization between the storage agents **104a** and **104b**, which may be stored as local-agent assigned version numbers.

[0032] Similarly, when the storage agent **104a** performs the synchronization with the storage agent **104b**, the storage agent **104b** may also generate a synchronization point that may include any one or more of: the synchronization time; the version number of the storage agent **104b** at the time of the synchronization, which may be stored as a local-agent version number; an identifier of the storage agent **104a**, which may be stored as a remote-agent ID; the version number of the storage agent **104a** at the time of the synchronization, which may be stored as a remote-agent version number; the locally-assigned version numbers—as assigned by the storage agent **104a**—of the shared data involved in the synchronization between the storage agents **104a** and **104b**, which may be stored as remote-agent assigned version numbers; and the locally-assigned version numbers—as assigned by the storage agent **104b**—of the shared data involved in the synchronization between the storage agents **104a** and **104b**, which may be stored as local-agent assigned version numbers.

[0033] In some embodiments, a particular storage agent **104** may store a certain number of synchronization points for synchronizations with each of the different storage agents **104** with which it has performed synchronization. As such, the particular storage agent **104** may store a synchronization history associated with its synchronizations with the different storage agents **104**. As detailed below, the synchronization points and synchronization history may be used during subsequent synchronizations to increase the efficiency of the synchronizations.

[0034] By synchronizing the shared data, the storage agents 104 may be aware of or updated as to the statuses of the other storage agents 104 as well as the data stored within the storage network 102. For example, the storage agent 104a may be synchronized with the storage agent 104b and/or the storage agent 104c such that the storage agent 104a is updated as to the connectivity or operating status of the storage agent 104b and/or the storage agent 104c, as well as the data stored thereon.

[0035] In addition to communicating between each other, in some embodiments, the storage agents 104 may communicate with a service that may be referred to as a storage network manager 114. The storage network manager 114 may act similarly to a central service in a distributed storage system. The storage network manager 114 may perform multiple functions in the storage system 100 such as coordinating actions by the storage agents 104. For example, the functions of the storage network manager 114 may include, but are not limited to, locating data files among the storage blocks 110 of the storage network 102, coordinating synchronization of shared data between the storage agents 104 and storage blocks 110, and allocating data between the storage blocks 110.

[0036] In some embodiments, the storage network manager 114 may be included in the same device as one of the storage agents 104 and, in other embodiments, the storage network manager 114 may be included in a device separate from the storage agents 104. Further, in some embodiments, the storage network manager 114 may perform operations such that the storage network manager 114 may act as and be a storage agent. For example, the storage network manager 114 may store shared data such as the catalog and/or other metadata associated with the storage network 102 and may synchronize its shared data with the storage agents 104 such that the storage network manager 114 may act as a storage agent with respect to such shared data.

[0037] In some embodiments the storage network manager 114 may communicate with the storage agents 104 via the communication network 112 (as illustrated in FIG. 1). The storage network manager 114 may also be configured to communicate with one or more of the storage agents 104 via a direct communication (not expressly illustrated in FIG. 1).

[0038] In some embodiments, the storage network manager 114 may be configured such that data files stored in the storage network 102 are not stored on the storage network manager 114, but metadata related to the data files (e.g., the catalog) of the storage network 102 may be stored on the storage network manager 114. In some embodiments, the storage network manager 114 may communicate instructions to the storage agents 104 regarding storage of the data. The storage agents 104 may act in response to the instructions communicated from the storage network manager 114.

[0039] The storage agents 104 may locate data files within the storage network 102 according to metadata that may be stored on each of the storage agents 104. For example, the storage agent 104a may locate a data file stored on the storage agent 104b using the metadata (e.g., catalog) stored on the storage agent 104a. Some or all the information to locate data files may be communicated during synchronization between the storage agents 104 and/or a storage agent 104 and the storage network manager 114. Additionally or alternatively, the storage agents 104 may communicate with the storage network manager 114 to locate data files stored on the storage network 102.

[0040] Additionally, the storage network manager 114 may communicate with one or more of the storage agents 104 with unreliable or intermittent connectivity. For example, the storage agent 104c may be communicatively coupled to the storage agent 104b and/or the storage agent 104a using an unreliable connection or the storage agent 104c may be included in an external device that intermittently connects to the storage agent 104b and/or the storage agent 104a. The storage network manager 114 may accordingly communicate with the storage agent 104c via the communication network 112, then relay information to the storage agent 104b and/or the storage agent 104a regarding the storage agent 104c.

[0041] As described in further detail below, the storage agents 104 and/or the storage network manager 114 may be configured to perform synchronization in a manner that allows for efficient and consistent synchronization of shared data within the storage network 102 while also allowing for the storage agents 104 to be mobile and have sporadic connectivity with each other. In some embodiments, the storage agents 104 and/or the storage network manager 114 may be configured to perform synchronization in a manner as described by the processes and methods described with respect to FIGS. 2 and 3.

[0042] FIG. 2 illustrates an example synchronization process 200 that may be performed between a first storage agent (1st SA) and a second storage agent (2nd SA), according to at least one embodiment of the present disclosure. The synchronization process 200 may be used to perform synchronization of shared data between the first storage agent and the second storage agent in an efficient manner but also such that the shared data between the first storage agent and the second storage agent may be consistent. For example, the synchronization process 200 may ensure that the versions of the shared data are the same between the first storage agent and the second storage agent. In these or other embodiments, the synchronization process 200 may also ensure that shared data that may have been added to and stored on the first storage agent that may not be on the second storage agent may also be added to and stored on the second storage agent, or vice versa.

[0043] The first storage agent and the second storage agent may be any suitable storage agent, such as the storage agents 104 described above with respect to FIG. 1. Additionally, in some embodiments, the first storage agent and/or the second storage agent may be a storage network manager, such as the storage network manager 114 of FIG. 1, performing operations of and acting as a storage agent. Additionally, in the illustrated embodiment, the first storage agent may initiate the synchronization with the second storage agent; however, in other instances the second storage agent may initiate the synchronization.

[0044] During the description of the synchronization process 200, the term “first-agent synchronizations” may refer to synchronizations in which the first storage agent has participated and the term “second-agent synchronizations” may refer to synchronizations in which the second storage agent has participated. Additionally, in some instances a synchronization may be between the first storage agent and the second storage agent. Therefore, for purposes of clarity, synchronizations between the first storage agent and the second storage agent may be referred to as “common synchronizations.”

[0045] The synchronization process **200** may include a step **202**, where the first storage agent may determine a set of recent first-agent synchronization points (referred to hereinafter as “recent first-agent points”). The recent first-agent points may be associated with synchronizations of the first storage agent with other storage agents included in the same storage network as the first storage agent and the second storage agent. In some embodiments, the recent first-agent points determined at step **202** may be determined based on a first-agent synchronization history and may be related to the most recent synchronizations performed by the first storage agent with respect to each of the other storage agents that are included in the same storage network as the first storage agent and the second storage agent and with which the first storage agent has performed a synchronization. The respective first-agent points may also indicate the times of their associated first-agent synchronizations (referred to hereinafter as “first-agent synchronization times”).

[0046] For example, the recent first-agent points may include a first recent first-agent point, a second recent first-agent point, and a third recent first-agent point. The first recent first-agent point may be associated with the most recent common synchronization between the first storage agent and the second storage agent. The first recent first-agent point may include, as its local-agent version number, the version number of the first storage agent (which may be referred to as a “first-agent version number”) at the time of the first-agent synchronization corresponding to the first recent first-agent point. The first recent first-agent point may also include, as its remote-agent ID, an identifier associated with the second storage agent (which may be referred to as a “second-agent ID”). Further, the first recent first-agent point may include, as its remote-agent version number, the version number of the second storage agent (which may be referred to as a “second-agent version number”) at the time of the first-agent synchronization corresponding to the first recent first-agent point. Additionally, the first recent first-agent point may include, as local-agent assigned version numbers, the locally-assigned version numbers—as assigned by the first storage agent—of the shared data involved in the synchronization corresponding to the first recent first-agent point. Moreover, the first recent first-agent point may include, as remote-agent assigned version numbers, the locally-assigned version numbers—as assigned by the second storage agent—of the shared data involved in the synchronization corresponding to the first recent first-agent point. The first recent first-agent point may also include a first-agent synchronization time associated with when the synchronization corresponding to the first recent first-agent point occurred.

[0047] The second recent first-agent point may be associated with the most recent first-agent synchronization between the first storage agent and a third storage agent. The second recent first-agent point may include, as its local-agent version number, the first-agent version number at the time of the first-agent synchronization corresponding to the second recent first-agent point. The second recent first-agent point may also include, as its remote-agent ID, an identifier associated with the third storage agent (which may be referred to as a “third-agent ID”). Further, the second recent first-agent point may include, as its remote-agent version number, the version number of the third storage agent (which may be referred to as a “third-agent version number”) at the time of the first-agent synchronization corre-

sponding to the second recent first-agent point. Additionally, the second recent first-agent point may include, as local-agent assigned version numbers, the locally-assigned version numbers—as assigned by the first storage agent—of the shared data involved in the synchronization corresponding to the second recent first-agent point. Moreover, the second recent first-agent point may include, as remote-agent assigned version numbers, the locally-assigned version numbers—as assigned by the third storage agent—of the shared data involved in the synchronization corresponding to the second recent first-agent point. The second recent first-agent point may also include a first-agent synchronization time associated with when the synchronization corresponding to the first recent first-agent point occurred.

[0048] The third recent first-agent point may be associated with the most recent first-agent synchronization between the first storage agent and a fourth storage agent. The third recent first-agent point may include, as its local-agent version number, the first-agent version number at the time of the synchronization corresponding to the third recent first-agent point. The third recent first-agent point may also include as its remote-agent ID an identifier associated with the fourth storage agent (which may be referred to as a “fourth-agent ID”). Further, the third recent first-agent point may include, as its remote-agent version number, the version number of the fourth storage agent (which may be referred to as a “fourth-agent version number”) at the time of the synchronization corresponding to the third recent first-agent point. Additionally, the third recent first-agent point may include, as local-agent assigned version numbers, the locally-assigned version numbers—as assigned by the first storage agent—of the shared data involved in the synchronization corresponding to the third recent first-agent point. Moreover, the third recent first-agent point may include, as remote-agent assigned version numbers, the locally-assigned version numbers—as assigned by the fourth storage agent—of the shared data involved in the synchronization corresponding to the third recent first-agent point. The third recent first-agent point may also include a first-agent synchronization time associated with when the synchronization corresponding to the third recent first-agent point occurred.

[0049] At step **204**, the first storage agent may communicate the recent first-agent points to the second storage agent. At step **206**, in response to receiving the recent first-agent points, the second storage agent may determine a second-agent baseline based on the recent first-agent points received at step **204**.

[0050] The second-agent baseline may be associated with a point in time that the versions of the shared data stored on the second storage agent were either the same as, or earlier versions of, the corresponding shared data stored on the first storage agent. Accordingly, as described in further detail below, the second-agent baseline may be used as a reference point for the synchronization process **200** to establish when the second storage agent had shared data stored thereon with versions that were the same as or earlier than the versions of corresponding shared data stored on the first storage agent.

[0051] In some instances, the second-agent baseline may be determined based on synchronizations between the first storage agent and one or more third-party storage agents (i.e., storage agents that are not actively participating in the synchronization process **200**) as well as synchronizations between the second storage agent and the same one or more third-party storage agents. Third-party storage agents that

have performed synchronizations with both the first storage agent and the second storage agent may be referred to as “common third-party agents.” The use of synchronizations with common third-party storage agents to establish the second-agent baseline may allow for improved efficiency of the synchronization as compared to merely using a previous synchronization between the first and second storage agents as the second-agent baseline because less data may need to be communicated in the synchronization. As detailed below, in some embodiments, to determine the second-agent baseline, the second storage agent may determine a set of second-agent baseline synchronization points (referred to hereinafter as “second-agent baseline points”) from a second-agent synchronization history and from the recent first-agent points.

[0052] In determining the second-agent baseline points, the second storage agent may select as second-agent baseline points second-agent synchronization points from the second-agent storage history that include common third-party agents indicated by the recent first-agent points. The second storage agent may also select as the second-agent baseline points those second-agent synchronization points that may be related to synchronizations that occurred before the first-agent synchronizations associated with the first-agent points. Therefore, the second-agent baseline points may each have a second-agent synchronization time that occurred before a respective first-agent synchronization time of the recent first-agent point corresponding to the same third-party storage agent.

[0053] In some embodiments, one or more of the second-agent baseline points may be related to the most recent second-agent synchronizations between the second storage agent and the common third-party agents that occurred before the first-agent synchronization between the first storage agent and the common third-party agents, as indicated by the recent first-agent points. Additionally, the second storage agent may select as a second-agent baseline point a second-agent synchronization point associated with a previous common synchronization that may correspond with one of the recent first-agent points.

[0054] By way of example, the second storage agent may select the second-agent baseline points based on the first recent first-agent point, the second recent first-agent point, and the third recent first-agent point described above, as follows. The first recent first-agent point may be associated with a common synchronization, the second recent first-agent point may be associated with a first-agent synchronization between the first storage agent and the third storage agent, and the third recent first-agent point may be associated with a first-agent synchronization between the first storage agent and the fourth storage agent. Accordingly, the second storage agent may select as the second-agent baseline points second-agent synchronization points that correspond to second-agent synchronizations between the second storage agent and the first storage agent, the third storage agent, and the fourth storage agent. In some embodiments, the second storage agent may select the relevant second-agent synchronization points with the most recent synchronization times that occurred before the first-agent synchronization times indicated by the recent first-agent points.

[0055] By way of example, the first recent first-agent point may include a first-agent synchronization time “A” indicating the time of the last common synchronization between the first storage agent and the second storage agent. The second

storage agent may accordingly select as a first second-agent baseline point the second-agent synchronization point associated with the last common synchronization that also has time “A” as the second-agent synchronization time.

[0056] The second recent first-agent point may include a first-agent synchronization time “B” indicating the time of the last synchronization between the first storage agent and the third storage agent. The second storage agent may accordingly select as a second second-agent baseline point the second-agent synchronization point associated with the last synchronization between the second storage agent and the third storage agent that occurred before the first-agent synchronization time “B.” Accordingly, a second-agent synchronization time “C” of the second second-agent baseline point may be before the first-agent synchronization time “B” and may also be closer in time to the first-agent synchronization time “B” than other second-agent synchronizations between the second storage agent and the third storage agent that also occurred before the first-agent synchronization time “B.”

[0057] Further, the third recent first-agent point may include a first-agent synchronization time “D” indicating the time of the last synchronization between the first storage agent and the fourth storage agent. The second storage agent may accordingly select as a third second-agent baseline point the second-agent synchronization point associated with the last synchronization between the second storage agent and the fourth storage agent that occurred before the first-agent synchronization time “D.” Accordingly, a second-agent synchronization time “E” of the third second-agent baseline point may be before the first-agent synchronization time “D” and may also be closer in time to the first-agent synchronization time “D” than other synchronizations between the second storage agent and the fourth storage agent that also occurred before the first-agent synchronization time “D.”

[0058] In some embodiments, the second storage agent may select the second-agent baseline points based on the version numbers of the storage agents included in the recent first-agent points and the second-agent synchronization points. For example, for the second-agent synchronization points associated with second-agent synchronizations between the first storage agent and the second storage agent, the second-agent synchronization point that includes as its local-agent version number the same second-agent version number as the second-agent version number included in the remote-agent version number of the recent first-agent point associated with the last synchronization between the first and second storage agents may be selected as a second-agent baseline point.

[0059] Additionally, for the second-agent synchronization points associated with synchronizations between the second storage agent and the common third-party storage agents, the second-agent synchronization points having the highest remote-agent version numbers that are also lower than the remote-agent version numbers of the recent first-agent points corresponding to first-agent synchronizations with corresponding third-party storage agents may be selected. Such a selection process based on version numbers of the storage agents may allow for achieving the above-described relationships between synchronization times of the second-agent baseline points and the recent first-agent points.

[0060] By way of example, in some embodiments the first, second, and third second-agent baseline points of the

example above may be determined in the following manner. The first recent first-agent point may include as its remote-agent version number a second-agent version number “5.” Accordingly, a second-agent synchronization point corresponding to a synchronization of the second storage agent with the first storage agent that includes as its local-agent version number the second-agent version number “5” may be selected as the first second-agent baseline point.

[0061] Further, the second recent first-agent point may include as its remote-agent version number a third-agent version number “6.” Additionally, second-agent synchronization points corresponding to synchronizations of the second storage agent with the third storage agent may include as their remote-agent version numbers third-agent version numbers “2,” “4,” and “7.” Accordingly, the second-agent synchronization point that includes the third-agent version number “4” may be selected as the second second-agent baseline point because the third-agent version number “4” is the highest third-agent version number of the relevant second-agent synchronization points that is also less than the third-agent version number “6” included in the second recent first-agent point.

[0062] Also, the third recent first-agent point may include as its remote-agent version number a third-agent version number “9.” Additionally, second-agent synchronization points corresponding to synchronizations of the second storage agent with the fourth storage agent may include as their remote-agent version numbers fourth-agent version numbers “5,” “8,” and “11.” Accordingly, the second-agent synchronization point that includes the fourth-agent version number “8” may be selected as the third second-agent baseline point because the fourth-agent version number “8” is the highest fourth-agent version number of the relevant second-agent synchronization points that is also less than the fourth-agent version number “9” included in the third recent first-agent point.

[0063] The second-agent baseline may be determined based on the second-agent baseline point that may have the latest second-agent synchronization time. For example, with respect to a continuation of the example referenced above with respect to steps 202, 204, and 206, the second-agent synchronization time “E” of the third second-agent baseline point may be later than the second-agent synchronization times “A” and “C.” Therefore, the second storage agent may base the second-agent baseline on the third second-agent baseline point.

[0064] In some embodiments, the second storage agent may determine the second-agent baseline by analyzing the local-agent version numbers of the second-agent baseline points (which may be second-agent version numbers) and determining which of the second-agent baseline points includes the highest local-agent version number—which may indicate the second-agent baseline point with the latest second-agent synchronization time. In these or other embodiments, the second storage agent may select as the second-agent baseline the determined highest local-agent version number.

[0065] As mentioned above, the second-agent baseline points may be selected based on the most recent second-agent synchronizations that occurred before the first-agent synchronizations indicated in the recent first-agent points. Additionally, as mentioned above, the local-agent version numbers included in the second-agent synchronization points—and accordingly included in the second-agent base-

line points—may indicate versions of the shared data stored on the second storage agent at the time of their corresponding synchronizations. As such, the local-agent version number used for the second-agent baseline may indicate the latest versions of the shared data stored on the second storage agent that can be verified as also having been stored on the first storage agent. The second-agent baseline may thus be used as a reference point for the synchronization process 200 to establish when the second storage agent had shared data stored thereon with versions that were the same as or earlier than the versions of corresponding shared data stored on the first storage agent.

[0066] At step 208, the second storage agent may determine a set of recent second-agent synchronization points (referred to hereinafter as “recent second-agent points”) in a manner analogous to that used by the first storage agent to determine the recent first-agent points at step 202. In some embodiments, the second storage agent may determine the recent second-agent points in response to receiving the recent first-agent points. Further, in some embodiments, the second storage agent may determine the recent second-agent points before, after, or while determining the second-agent baseline.

[0067] At step 210, the second storage agent may communicate the recent second-agent points to the first storage agent. Additionally, in some embodiments, the second storage agent may communicate the second-agent baseline to the first storage agent at step 210. The second storage agent may communicate the second-agent baseline to the first storage agent such that the first storage agent may keep track of the second-agent baseline for the second storage agent during the synchronization process 200, which in some instances may facilitate implementation of the synchronization process 200.

[0068] At step 212, the first storage agent may determine a first-agent baseline based on the recent second-agent points in a manner analogous to the second storage agent determining the second-agent baseline based on the recent first-agent points, as described above with respect to step 206. Accordingly, in some embodiments, to determine the first-agent baseline, the first storage agent may determine a set of first-agent baseline synchronization points (referred to hereinafter as “first-agent baseline points”) that may be associated with first-agent synchronizations that occurred before the second-agent synchronizations associated with the recent second-agent points.

[0069] In some embodiments, the first storage agent may consequently select as the first-agent baseline the local-agent version number—which may be a version number of the first storage agent—included in the first-agent baseline point having the latest first-agent synchronization time. As such, the first-agent baseline may indicate the latest versions of the shared data stored on the first storage agent that can be verified as also having been stored on the second storage agent. As described in further detail below and similar to the second-agent baseline with respect to the second storage agent, the first-agent baseline may thus be used as a reference point for the synchronization process 200 to establish when the first storage agent had shared data stored thereon with versions that were the same as or earlier than the versions of corresponding shared data stored on the second storage agent.

[0070] At step 214, the first storage agent may determine a set of first-agent deltas based on the first-agent baseline. As

mentioned above, the first-agent baseline may be used as a reference point for the synchronization process 200 to establish when the first storage agent had shared data stored thereon with versions that were the same as or earlier than the versions of corresponding shared data stored on the second storage agent. Accordingly, the set of first-agent deltas may include the shared data that may have been updated on the first storage agent after the time corresponding to the first-agent baseline such that the same updates may not have been received by the second storage agent.

[0071] In some embodiments the set of first-agent deltas may be determined by comparing the locally-assigned version numbers of the shared data on the first storage agent with the first-agent baseline. As mentioned above, the first-agent baseline may be based on a version number of the first storage agent that indicates the latest versions of the shared data stored on the first storage agent that can be verified as also having been stored on the second storage agent. Additionally, as mentioned above, in some embodiments the latest locally-assigned version number of the shared data may be used as the version number of the first storage agent.

[0072] Therefore, in some embodiments, the set of first-agent deltas may be determined by identifying the shared data with locally-assigned version numbers that are higher than the first-agent baseline, which indicates that the identified shared data stored on the first storage agent may have been updated on the first storage agent but not the second storage agent as far as the first storage agent knows. Additionally, in some embodiments, when shared data is added to or stored on the first storage agent for the first time, the newly-added shared data may be given the current first-agent version number as a locally-assigned version number. Therefore, shared data added to and stored on the first storage agent after the time corresponding to the first-agent baseline may be added to the first-agent deltas because the locally-assigned version numbers of such data may be higher than the first-agent baseline.

[0073] At step 216, the first storage agent may communicate the set of first-agent deltas to the second storage agent. Additionally, in some embodiments, the first storage agent may communicate the second-agent baseline back to the second storage agent. As indicated above, the first storage agent may receive the second-agent baseline from the second storage agent and then communicate the second-agent baseline back to the second storage agent such that the first storage agent remembers the second-agent baseline for the second storage agent to ease some implementations of the first and second storage agents.

[0074] At step 218, the second storage agent may reconcile the shared data of the first-agent deltas with corresponding shared data stored on the second storage agent. During the reconciliation, the second storage agent may determine whether the shared data of the first-agent deltas is older than the corresponding shared data stored on the second storage agent. When the shared data of a particular first-agent delta is older, the second storage agent may ignore the shared data of the particular first-agent delta. When the shared data of the particular first-agent delta is newer, the second storage agent may replace the corresponding shared data stored on the second storage agent with the shared data of the particular first-agent delta.

[0075] In some embodiments, the second storage agent may determine which shared data to replace based on the locally-assigned version numbers of the shared data corre-

sponding to the first-agent deltas. For example, in some embodiments, the first storage agent may communicate the locally-assigned version numbers—as assigned by the first storage agent and referred to hereinafter as “first-agent assigned version numbers”—of the shared data included in the first-agent deltas. The second storage agent may compare the first-agent assigned version numbers of the shared data of the first-agent deltas with the remote-agent assigned version numbers of corresponding shared data that may be included in an applicable second-agent synchronization point associated with the most recent previous common synchronization. As indicated above, the remote-agent assigned version numbers of the corresponding shared data included in the applicable second-agent synchronization point may be the first-agent version numbers at the time of the most recent previous common synchronization between the first storage agent and the second storage agent. Therefore, a first-agent assigned version number of first shared data of the first-agent deltas that is greater than the remote-agent assigned version number of corresponding first shared data in the particular second-agent synchronization point may indicate that the first storage agent updated the first shared data since the most recent previous common synchronization. As such, in some instances the second storage agent may update the first shared data stored on the second storage agent with the first shared data included in the first-agent deltas.

[0076] In contrast, a first-agent assigned version number of second shared data of the first-agent deltas that is less than or equal to the remote-agent assigned version number of corresponding second shared data in the particular second-agent synchronization point may indicate that the first storage agent has not updated the second shared data since the most recent previous common synchronization. As such, the second storage agent may not update the second shared data stored on the second storage agent with that included in the first-agent deltas. The second storage agent may also determine whether there is shared data included in the first-agent deltas that is not stored on the second storage agent at all and may store such shared data.

[0077] By way of example, the shared data of the received first-agent deltas may include shared data “A” with a first-agent assigned version number of “5,” shared data “B” with a first-agent assigned version number of “6,” shared data “C” with a first-agent assigned version number of “7,” shared data “D” with a first-agent assigned version number of “8,” and shared data “E” with a first-agent assigned version number of “0.” Additionally, the second storage agent may include stored thereon an applicable second-agent synchronization point associated with the most recent previous common synchronization. The applicable second-agent synchronization point may include remote-agent assigned version numbers—which as mentioned above may be first-agent assigned version numbers—of shared data “A,” “B,” “C,” and “D” at the time of the most recent previous common synchronization. In the applicable second-agent synchronization point, shared data “A” may have a remote-agent assigned version number of “6,” shared data “B” may have a remote-agent assigned version number of “5,” shared data “C” may have a remote-agent assigned version number of “4,” and shared data “D” may have a remote-agent assigned version number of “8.” Additionally, the applicable second-agent synchronization point may not have any remote-agent assigned version number of shared

data “E” indicating that the shared data “E” was not stored on the first storage agent or the second storage agent at the time of the most recent previous common synchronization. In other embodiments, the second storage agent may store the first-agent assigned version numbers in a location other than the applicable second-agent synchronization point.

[0078] The second storage agent may replace the shared data “B” and “C” stored thereon with the shared data “B” and “C” of the first-agent deltas because the shared data “B” and “C” of the first-agent deltas may be later versions (as indicated by the respective higher first-agent assigned version numbers included in the first agent deltas) of the shared data “B” and “C” stored on the second storage agent. Additionally, the second storage agent may store the shared data “E” because it may not have previously had the shared data “E” stored thereon, as indicated by the shared data “E” not being included in the applicable second-agent synchronization point.

[0079] During the reconciliation at step **218**, the second storage agent may also update the locally-assigned version numbers of the shared data stored on the second storage agent that was updated. For example, the second storage agent may update the locally-assigned version numbers of the shared data “B,” “C,” and “E” because that shared data was updated during the reconciliation.

[0080] As indicated above, the above process determines whether or not to update shared data included in the first-agent deltas based on whether or not the shared data was updated by the first storage agent since the most recent previous common synchronization. However, in some instances shared data included in the first-agent deltas that may have been updated by the first storage agent since the most recent previous common synchronization may have also been updated by the second storage agent since the most recent previous common synchronization. Accordingly, in some embodiments, the reconciliation may also include determining whether or not to replace such shared data with that included in the first-agent deltas.

[0081] In some embodiments, the second storage agent may determine whether shared data stored on the second storage agent corresponding to shared data of the first-agent deltas has been updated since the most recent previous common synchronization based on the second-agent baseline and the locally-assigned version numbers of the shared data as assigned by the second storage agent—which may be referred to as “second-agent assigned version numbers.” As mentioned above, the second-agent baseline may establish a point in time in which the shared data stored on the first storage agent and the second storage agent was the same. Accordingly, particular shared data with a second-agent assigned version number greater than the second-agent baseline may indicate that the particular shared data has been updated since the most recent previous common synchronization. The second storage agent may therefore compare the second-agent baseline with the second-agent assigned version numbers of the shared data stored on the second storage agent that corresponds to the shared data of the first-agent deltas to determine whether or not the shared data had been updated since the most recent previous common synchronization. In some embodiments, the second storage agent may also determine whether shared data has also been updated by the first storage agent since the most recent previous synchronization between the first and second

storage agents based on version numbers of the shared data as assigned by the first storage agent in a manner as discussed above.

[0082] When it is determined that particular shared data has been updated by both the first storage agent and the second storage agent since the most recent previous common synchronization, the second storage agent may determine whether or not to replace the particular shared data with that included in the first-agent deltas. For example, in some embodiments, the second storage agent may refer to timestamps of the particular shared data as indicated in a corresponding first-agent delta and as stored on the second storage agent.

[0083] The second storage agent may be configured to update the particular shared data stored on the second storage agent with that included in the corresponding first-agent delta when the timestamp indicated in the corresponding first-agent delta is later than the timestamp stored on the second storage agent that is associated with the particular shared data. In contrast, the second storage agent may be configured to not update the particular shared data stored on the second storage agent with that included in the corresponding first-agent delta when the timestamp indicated in the corresponding first-agent delta is earlier than the timestamp stored on the second storage agent that is associated with the particular shared data.

[0084] In some embodiments, when it is determined that particular shared data has been updated by both the first storage agent and the second storage agent since the most recent previous common synchronization, the reconciliation of the shared data may include some sort of combination of the shared data. For example, if one of the fields in a shared data element is a list of labels, and both the first and second storage agents have updated the labels since the most recent previous common synchronization, the reconcile process may choose to take the union of the two lists of labels as a reconciled data element of the shared data element.

[0085] At step **220**, the second storage agent may determine a set of second-agent deltas based on the second-agent baseline in a manner analogous to the first storage agent determining the set of first-agent deltas based on the first-agent baseline described above with respect to step **214**. At step **222**, the second storage agent may update the version number of the second storage agent based on one or more of the newly-assigned version numbers of the shared data that were updated during the reconciliation of step **218**.

[0086] At step **224**, the second storage agent may communicate the set of second-agent deltas to the first storage agent. Additionally, in some embodiments, the second storage agent may communicate the updated version number of the second storage agent to the first storage agent. Therefore, the first storage agent may have the most recent second-agent version number at the time of the synchronization performed with respect to the synchronization process **200**. In some embodiments, the first storage agent may store the most recent second-agent version number in a first-agent synchronization point that may be created for the synchronization performed with respect to the synchronization process **200**.

[0087] At step **226**, the first storage agent may reconcile the shared data of the set of second-agent deltas with corresponding shared data stored on the first storage agent in a manner analogous to the reconciliation performed by the second storage agent described above with respect to step

218. At step **228**, the first storage agent may update the version number of the first storage agent based on one or more of the new version numbers of the shared data that were updated during the reconciliation of step **226**. Additionally, in some embodiments, the first storage agent may communicate the updated version number of the first storage agent to the second storage agent at or after step **228**. Therefore, the second storage agent may have the most recent first-agent version number at the time of the synchronization performed with respect to the synchronization process **200**. In some embodiments, the second storage agent may store the most recent first-agent version number in a second-agent synchronization point that may be created for the synchronization performed with respect to the synchronization process **200**.

[0088] Accordingly, the synchronization process **200** may be performed to synchronize storage agents. One skilled in the art will appreciate that, for this and other processes and methods disclosed herein, the functions performed in the processes and methods may be implemented in differing order. Furthermore, the outlined steps and operations are only provided as examples, and some of the steps and operations may be optional, combined into fewer steps and operations, or expanded into additional steps and operations without detracting from the essence of the disclosed embodiments.

[0089] For example, the order and/or timing of determining synchronization points, baselines, and deltas may vary from that described depending on various implementations and embodiments. Additionally, the order and/or timing of communication of the synchronization points, baselines (if applicable), and agent versions (if applicable) may also vary depending on various implementations and embodiments.

[0090] FIG. 3 is a flow chart of an example method **300** of synchronizing data, according to at least one embodiment described herein. One or more steps of the method **300** may be implemented, in some embodiments, by a storage agent operating in a storage system, such as the example storage agents **104** operating in the example storage system **100** of FIG. 1. For example, one of the storage agents **104** of FIG. 1 may be configured to execute computer instructions to perform operations of managing storage allocation on a storage agent as represented by one or more of the blocks of the method **300**. Although illustrated as discrete blocks, various blocks may be divided into additional blocks, combined into fewer blocks, or eliminated, depending on the desired implementation. The method **300** will now be discussed with reference to FIG. 3.

[0091] The method **300** may begin and at block **302**, a first storage agent may receive a recent second-agent synchronization point associated with a recent second-agent synchronization time of a second-agent synchronization between a second storage agent and a third storage agent. In some embodiments, the recent second-agent synchronization point may be associated with a most recent second-agent synchronization between the second storage agent and the third storage agent.

[0092] At block **304**, a first-agent baseline synchronization point may be determined based on the recent second-agent synchronization point. The first-agent baseline synchronization point may be associated with a first-agent baseline synchronization time of a first-agent synchronization of the first storage agent with the third storage agent that occurred before the second-agent synchronization time. In some

embodiments, the first-agent baseline synchronization point may be associated with a latest first agent synchronization between the first storage agent and the third storage agent that occurred before the recent second-agent synchronization time.

[0093] At block **306**, a first-agent baseline of synchronization of data between the first storage agent and the second storage agent may be established. The first-agent baseline may be determined based on the first-agent baseline synchronization point and may indicate a time when the first storage agent and the second storage agent had the same versions of shared data stored thereon.

[0094] Accordingly, the method **300** may be used to establish a common point in time that may be used as a baseline for synchronizing shared data between the first storage agent and the second storage agent. One skilled in the art will appreciate that, for the method **300** and other processes and methods disclosed herein, the functions performed in the processes and methods may be implemented in differing order. Furthermore, the outlined steps and operations are only provided as examples, and some of the steps and operations may be optional, combined into fewer steps and operations, or expanded into additional steps and operations without detracting from the essence of the disclosed embodiments.

[0095] For example, in some embodiments, the method **300** may include steps associated with determining data that was updated by the first storage agent after the first-agent baseline synchronization time based on the first-agent baseline. In some embodiments, this data may be transmitted to the second storage agent such that the first and second storage agents may synchronize this data.

[0096] Additionally, in some embodiments, the method **300** may include steps associated with transmitting, by the first storage agent to the second storage agent, a recent first-agent synchronization point. The recent first-agent synchronization point may be associated with a recent first-agent synchronization time of a recent first-agent synchronization of the first storage agent with the third storage agent. The method **300** may further include receiving, by the first storage agent from the second storage agent, data that was updated by the second storage agent after the recent first-agent synchronization time as determined based on a second-agent baseline. The second-agent baseline may be related to a second-agent baseline synchronization point, which may be associated with a second-agent baseline synchronization time of the second storage agent with the third storage agent that occurred before the recent first-agent synchronization time.

[0097] In these and other embodiments, the method **300** may further include reconciling the data received from the second storage agent with corresponding data stored on the first storage agent. In some embodiments, the reconciling may include determining whether the received data is older than the corresponding data stored on the first storage agent. When the received data is older than the corresponding data, the received data may be ignored. When the received data is newer than the corresponding data the corresponding data may be replaced with the received data.

[0098] As indicated above, the embodiments described herein may include the use of a special purpose or general purpose computer (e.g., the processors **150** of FIG. 1) including various computer hardware or software modules, as discussed in greater detail below.

[0099] Embodiments described herein may be implemented using computer-readable media (e.g., the memories **152** of FIG. **1**) for carrying or having computer-executable instructions or data structures stored thereon. Such computer-readable media may be any available media that may be accessed by a general purpose or special purpose computer. By way of example, and not limitation, such computer-readable media may include tangible computer-readable storage media including Random Access Memory (RAM), Read-Only Memory (ROM), Electrically Erasable Programmable Read-Only Memory (EEPROM), Compact Disc Read-Only Memory (CD-ROM) or other optical disk storage, magnetic disk storage or other magnetic storage devices, flash memory devices (e.g., solid state memory devices), or any other storage medium which may be used to carry or store desired program code in the form of computer-executable instructions or data structures and which may be accessed by a general purpose or special purpose computer. Combinations of the above may also be included within the scope of computer-readable media.

[0100] Computer-executable instructions comprise, for example, instructions and data which cause a general purpose computer, special purpose computer, or special purpose processing device (e.g., one or more processors) to perform a certain function or group of functions. Although the subject matter has been described in language specific to structural features and/or methodological acts, it is to be understood that the subject matter defined in the appended claims is not necessarily limited to the specific features or acts described above. Rather, the specific features and acts described above are disclosed as example forms of implementing the claims.

[0101] As used herein, the terms “module” or “component” may refer to specific hardware implementations configured to perform the operations of the module or component and/or software objects or software routines that may be stored on and/or executed by general purpose hardware (e.g., computer-readable media, processing devices, etc.) of the computing system. In some embodiments, the different components, modules, engines, and services described herein may be implemented as objects or processes that

execute on the computing system (e.g., as separate threads). While some of the systems and methods described herein are generally described as being implemented in software (stored on and/or executed by general purpose hardware), specific hardware implementations or a combination of software and specific hardware implementations are also possible and contemplated. In this description, a “computing entity” may be any computing system as previously defined herein, or any module or combination of modules running on a computing system.

[0102] All examples and conditional language recited herein are intended for pedagogical objects to aid the reader in understanding the invention and the concepts contributed by the inventor to furthering the art, and are to be construed as being without limitation to such specifically recited examples and conditions. Although embodiments of the present inventions have been described in detail, it should be understood that the various changes, substitutions, and alterations could be made hereto without departing from the spirit and scope of the disclosure.

What is claimed is:

1. A method of synchronizing data, the method comprising:
 - receiving, by a first storage agent, a recent second-agent synchronization point associated with a recent second-agent synchronization time of a second-agent synchronization between a second storage agent and a third storage agent;
 - determining, based on the recent second-agent synchronization point, a first-agent baseline synchronization point associated with a first-agent baseline synchronization time of a first-agent synchronization between the first storage agent and the third storage agent that occurred before the second-agent synchronization time; and
 - establishing a first-agent baseline of synchronization of data between the first storage agent and the second storage agent based on the first-agent baseline synchronization point.

* * * * *