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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
03 January 2019 (03.01.2019)

WIPO | PCT

(10) International Publication Number
WO 2019/005240 A1

(51) International Patent Classification:
H04L 29/08 (2006.01) *G06F 17/30* (2006.01)
G06F 3/06 (2006.01)

(21) International Application Number:
PCT/US2018/023988

(22) International Filing Date:
23 March 2018 (23.03.2018)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
15/635,056 27 June 2017 (27.06.2017) US

(71) Applicant: **WESTERN DIGITAL TECHNOLOGIES, INC.** [US/US]; 5601 Great Oaks Parkway, San Jose, California 95119 (US).

(72) Inventors: **XU, Jun**; 5601 Great Oaks Parkway, San Jose, California 95119 (US). **XI, Wei**; 5601 Great Oaks Parkway, San Jose, California 95119 (US). **GUO, Guoxiao**; 5601 Great Oaks Parkway, San Jose, California 95119 (US). **BJORNSON, Eric**; 5601 Great Oaks Parkway, San Jose, California 95119 (US). **YU, Jie**; 5601 Great Oaks Parkway, San Jose, California 95119 (US). **WEI, Ling Chih**; 5601 Great Oaks Parkway, San Jose, California 95119 (US).

(74) Agent: **PUA, Don Neil C.**; 4199 Campus Drive, Suite 550, Irvine, California 92612 (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, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, 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, SA, 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, ST, 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, 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, KM, ML, MR, NE, SN, TD, TG).

(54) Title: HYBRID DATA STORAGE SYSTEM WITH PRIVATE STORAGE CLOUD AND PUBLIC STORAGE CLOUD

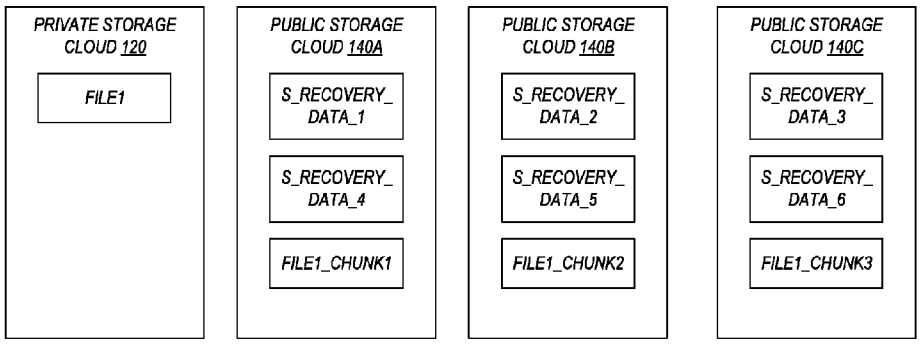


FIG. 2B

(57) Abstract: Systems and methods are disclosed for accessing data on a storage system. An apparatus, such as a data storage device or a computing device, may include a memory configured to store data. The apparatus is configured to receive a request for a file stored in a storage system. The storage system includes a private storage cloud and a set of public storage clouds. The apparatus is also configured to determine whether the file is retrievable from the set of public storage clouds. If the file is retrievable from the set of public storage clouds the apparatus may determine whether access to the set of public storage clouds is faster than access to the private storage cloud and may retrieve the file from one or more of the private storage cloud and the set of public storage clouds.

WO 2019/005240 A1

[Continued on next page]

Declarations under Rule 4.17:

— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

Published:

— *with international search report (Art. 21(3))*

HYBRID DATA STORAGE SYSTEM WITH PRIVATE STORAGE CLOUD AND PUBLIC STORAGE CLOUD

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application claims priority to U.S. Patent Application No. 15/635,056 filed June 27, 2017, entitled HYBRID DATA STORAGE SYSTEM WITH PRIVATE STORAGE CLOUD AND PUBLIC STORAGE CLOUD, the disclosure of which is hereby incorporated by reference in its entirety.

BACKGROUND

Field

[0002] The present disclosure relates to data storage systems. In particular, the present disclosure relates to data storage systems that may store data, such as files.

Description of Related Art

[0003] Users may store data, such as files in various types of data storage systems. One type of data storage system may be a private storage cloud. A private storage cloud may be a storage network (e.g., data storage devices, server computers, networks, etc.) that provides certain users (e.g., authorized users, employees, a family/household) with access to data stored in (the storage space) of the private storage cloud. A private storage cloud may be used by a household, a company/corporation, etc. Another type of data storage system may be a public storage cloud. A public storage cloud may be a storage network (e.g., data storage devices, server computers, networks, etc.) that provides the general public with access to data stored in (the storage space) of the public storage cloud. A public storage cloud may be maintained and/or provided by a storage provider.

BRIEF DESCRIPTION OF THE DRAWINGS

[0004] Various embodiments are depicted in the accompanying drawings for illustrative purposes, and should in no way be interpreted as limiting the scope of this disclosure. In addition, various features of different

disclosed embodiments can be combined to form additional embodiments, which are part of this disclosure.

[0005] Figure 1 is a diagram illustrating a storage system, in accordance with one or more embodiments.

[0006] Figure 2A is a diagram illustrating an example distribution of a file in a storage system, in accordance with one or more embodiments.

[0007] Figure 2B is a diagram illustrating an example distribution of a file in a storage system, in accordance with one or more embodiments.

[0008] Figure 2C is a diagram illustrating an example distribution of a file in a storage system, in accordance with one or more embodiments.

[0009] Figure 2D is a diagram illustrating an example distribution of a file in a storage system, in accordance with one or more embodiments.

[0010] Figure 2E is a diagram illustrating an example distribution of a file in a storage system, in accordance with one or more embodiments.

[0011] Figure 2F is a diagram illustrating an example distribution of a file in a storage system, in accordance with one or more embodiments.

[0012] Figure 3 is a table illustrating example metadata, in accordance with one or more embodiments.

[0013] Figure 4 is a flow diagram illustrating an example process for writing data to a storage system, in accordance with one or more embodiments.

[0014] Figure 5 is a flow diagram illustrating an example process for writing data to a storage system, in accordance with one or more embodiments.

[0015] Figure 6 is a flow diagram illustrating an example process for writing data to a storage system, in accordance with one or more embodiments.

[0016] Figure 7 is a flow diagram illustrating an example process for reading data from a storage system, in accordance with one or more embodiments.

[0017] Figure 8 is a diagram of a computing device, in accordance with one or more embodiments.

DETAILED DESCRIPTION

[0018] While certain embodiments are described, these embodiments are presented by way of example only, and are not intended to limit the scope of protection. Indeed, the novel methods and systems described herein may be embodied in a variety of other forms. Furthermore, various omissions, substitutions and changes in the form of the methods and systems described herein may be made without departing from the scope of protection.

Storage Clouds

[0019] As discussed above, users may store data in a private storage cloud or a public storage cloud. A private storage cloud may be a storage network that provides certain users with access to data stored in (the storage space) of the private storage cloud, as discussed above. A public storage cloud may be a storage network that provides the general public with access to data stored in (the storage space) of the public storage cloud, as discussed above.

[0020] A private storage cloud may provide users with a higher level of privacy and/or security because access to the private storage cloud may restricted to certain users (e.g., members of a household, employees of a company, etc.) and may be restricted by geographical location (e.g., within a building, a house, etc.). However, access to the private storage cloud may be slower when users are not within the private storage cloud (e.g., are not on a local-area network (LAN) provided by the private storage cloud) and/or geographical location. In addition, a private storage cloud may be less reliable because the private storage cloud may not protect against data loss as well when compared to a public storage cloud. A public storage cloud may provide users with faster access when users are not within the private storage cloud (e.g., when users are on a wide-area network (WAN), a cellular network, etc.). However, a public storage cloud may be easier to compromise because the general public may have access to the public storage cloud. For example, the username and password for a user's account on the public storage cloud may be compromised.

[0021] Certain embodiments described herein provide the ability to distribute (e.g., store) chunks of a file (e.g., file chunks, recovery data chunks,

etc.) across one or more private storage clouds and one or more public storage clouds. The chunks of the file may optionally be encrypted prior to distributing the chunks of the file to the private storage clouds and/or public storage clouds. In some embodiments, distributing the file chunks (for a file) and/or recovery data chunks (e.g., systematic recovery data chunks, non-systematic recovery data chunks) across the private storage clouds and the public storage clouds may allow a user faster access to the file, regardless of the location of the user (e.g., whether the user is on a LAN of a private storage cloud or using a WAN), as discussed in more detail below. In other embodiments, distributing the set of chunks across the private storage clouds and the public storage clouds may also help increase the security of the storage system as discussed in more detail below. In further embodiments, encrypting the file chunks and/or the recovery data chunks may also increase the security of the storage system as discussed in more detail below.

Hybrid Storage System

[0022] Figure 1 is a diagram of a storage system 100, in accordance with one or more embodiments. The system architecture includes a network 105, a private storage cloud 120, a private storage cloud 130, public storage clouds 140A though 140X, and computing device 110 (e.g., a client computing device). Private storage cloud 120, private storage cloud 130, and public storage clouds 140A through 140X may be coupled to the network 105. The network 105 may include one or more of an ad hoc network, a peer to peer communication link, an intranet, an extranet, a virtual private network (VPN), a public network (e.g., the Internet), a private network (e.g., a local area network (LAN)), or wide area network (WAN) such as the Internet, a wired network (e.g., Ethernet network), a wireless network (e.g., an 802.11 network, a Wi-Fi network, a wireless LAN (WLAN), a wireless WAN (WWAN), etc.), a cellular network (e.g., a Long Term Evolution (LTE) network), a metropolitan area network (MAN), a portion of the Internet, a portion of the Public Switched Telephone Network (PSTN), routers, hubs, switches, server computers, other types of computer network, and/or a combination thereof.

[0023] In one embodiment, each of the private storage clouds 120 through 130 and the public storage clouds 140A through 140X may include a network. For example, private storage cloud 120 may include a local-area network (LAN) that may allow computing devices coupled to the LAN to communicate with each other and/or the NAS device 125. The computing devices on the LAN may access the network 105 via the LAN. In another example, the public storage cloud 140A may include a LAN and/or a wide area network (WAN).

[0024] Computing device 110 includes storage application 111. The storage application 111 may be an application (e.g., software, an app, etc.) that allows a user to access data (e.g., files) stored in the storage system 100 (e.g., stored in one or more of the private storage clouds 120 through 130 and public storage clouds 140A through 140X). The storage application 111 may communicate with the private storage clouds 120 through 130 and the public storage clouds 140A through 140X using a representation state transfer (REST) based interface. For example, one or more of the private storage clouds 120 through 130 and the public storage clouds 140A through 140X may provide application programming interfaces (APIs) that may be used by the storage application 111 to communicate (e.g., transmit and/or receive data such as messages, files, etc.) with the private storage clouds 120 through 130 and the public storage clouds 140A through 140X. The computing device 110 may access a files in the storage system 100 via the network 105 and/or via one or more of the private storage clouds 120 through 130. Examples of computing devices may include, but are not limited to, phones (e.g., smart phones, cellular phones, etc.), cable set-top boxes, smart TV's, game consoles, laptop computers, tablet computers, desktop computers, wearable computers, and/or other network-connected computing devices.

[0025] Private storage cloud 120 includes network-access storage (NAS) device 125 and private storage cloud 130 includes NAS device 135. NAS device 125 includes storage module 126 and metadata 127. NAS device 135 includes storage module 136 and metadata 137. Storage module 136, storage module 126, metadata 127, and metadata 137 are discussed in more detail below. NAS devices 125 and 135 may include one or more data storage devices (e.g., multiple storage drives such as hard drives, solid state

drives (SSDs), etc.). A storage drive may comprise magnetic media, hard disk media, and/or solid-state media. While certain description herein may refer to solid state memory or flash memory generally, it is understood that solid state memory and/or flash memory may comprise one or more of various types of solid state non-volatile memory devices such as flash integrated circuits, Chalcogenide RAM (C-RAM), Phase Change Memory (PC-RAM or PRAM), Programmable Metallization Cell RAM (PMC-RAM or PMCm), Ovonic Unified Memory (OUM), Resistance RAM (RRAM), NAND memory (e.g., single-level cell (SLC) memory, multi-level cell (MLC) memory, or any combination thereof), NOR memory, EEPROM, Ferroelectric Memory (FeRAM), Magnetoresistive RAM (MRAM), other discrete NVM (non-volatile memory) chips, or any combination thereof.

[0026] In some embodiments, computing device 110 may be coupled to the network 105. The computing device 110 may access the private storage clouds 120 through 130 and the public storage clouds 140A through 140X via the network 104. In other embodiments, the computing device 110 may be coupled to the private storage cloud 120. The computing device 110 may access the network 105 and the public storage clouds 140A through private storage cloud 120 (e.g., through a LAN provided by the private storage cloud 120).

[0027] The storage module 126 may receive a file to be stored in the storage system 100. For example, a user of the computing device 110 may write (e.g., upload, transmit, etc.) a file to the storage system 100 using the storage application 111. In one embodiment, the storage module 126 (and/or storage module 136) may determine importance levels for files to be stored in the storage system 100 (e.g., store on a hybrid storage cloud). For example, the storage module 126 may analyze the file and/or the contents of the file to determine the importance level of the file. In another example, the storage module 126 may determine the importance level of the file based on data/input received from a user of the storage system 100 (e.g., a user may provide the importance level for the file using storage application 111). In a further example the storage module 126 may determine the importance level of the file based on a configuration file/setting. In some embodiments, an importance level for a file may be data (e.g., a number, a value, an

alphanumeric string, etc.) that may indicate the importance of the file to a user of the storage system 100. For example, the importance level of a file may indicate how valuable the file is to a user. In another example, the importance level may be used to determine how the file should be distributed among the private storage clouds 120 through 130 and the public storage clouds 140A through 140X (e.g., whether the file should be distributed among the public storage clouds 140A through 140X). One having ordinary skill in the art understands that various representations may be used to represent the importance levels and various numbers of importance levels may be used. For example, the storage system 100 may have three, five, ten, etc., importance levels. In another example, a numerical representation may be used such that the higher the number, the higher the importance level (or vice versa).

[0028] In one embodiment, the storage module 126 (and/or storage module 136) may generate a set of recovery data chunks based on the file and/or the importance level for the file (e.g., the importance level associated with the file). The recovery data chunks may be used to recover (e.g., regenerate, reconstruct, rebuild, etc.) the file if the file (or portions of the file) are lost, deleted, corrupted, damaged, inaccessible, etc. The recovery data chunks may prevent the file from being lost and may also provide a computing device and/or a user with faster access to the file, as discussed in more detail below. The set of recovery data chunks may also be referred to as recovery data.

[0029] In one embodiment, the set of recovery data chunks may include systematic recovery data. Systematic recovery data may be error recovery data that may be used to recover the file when the file (or portions of the file) is lost, deleted, corrupted, damaged, inaccessible, etc. Systematic recovery data may be error recovery data that is generated in addition to the file. For example, systematic recovery data may be error recovery data that is separate from the file. Examples of systematic recovery data may include redundant array of independent disks (RAID) parity data, data generated using error correction codes (ECCs), data generated using a low-density parity-check (LDPC) code, etc. In another embodiment, the set of recovery data chunks may include non-systematic recovery data. Non-systematic

recovery data may be error recovery data that may also be used to recover the file when the file (or portions of the file) is lost, deleted, corrupted, damaged, inaccessible, etc. Non-systematic recovery data may be error recovery data where the file is included in the non-systematic recovery data. For example, the file and the non-systematic recovery data may be combined such that the file is no longer distinguishable from the non-systematic recovery data. Examples of non-systematic recovery data may include data generated using erasure codes.

[0030] In one embodiment, a threshold number of recovery data chunks (e.g., systematic recovery data chunks, non-systematic recovery data chunks, etc.) may be used to recover, reconstruct, regenerate, etc., a file using the recovery data chunks. For example, ten recovery data chunks may be generated for a file. Any five of the ten recovery data chunks may be used to reconstruct the file. The threshold (e.g., minimum) number of recovery data chunks used to reconstruct the file may vary based on the data recovery (or error recovery) schemes/codes used to generate the recovery data chunks.

[0031] In one embodiment, the storage module 126 (and/or storage module 136) may store the set of recovery data chunks in one or more of the public storage clouds 140A through 140X, as discussed in more detail below. For example, the storage module 126 may evenly distribute the set of recovery data chunks across one or more of the public storage clouds 140A through 140X. In another example, the storage module 126 may prefer certain public storage clouds. This may help prevent the data from being lost and may allow faster access to the file, as discussed in more detail below.

[0032] In one embodiment, the storage module 126 (and/or storage module 136) may store the set of recovery data chunks on the public storage clouds 140A through 140X by transmitting the set of recovery data chunks to the public storage clouds 140A through 140X. The storage module 126 may optionally wait for acknowledgements that the set of recovery data chunks was received by the public storage clouds 140A through 140X. For example, the storage module 126 may transmit one recovery data chunk to each of the public storage clouds 140A through 140X and may wait for acknowledgements from each of the public storage clouds 140A through 140X. An acknowledgement from a public storage cloud may indicate that the

public storage cloud has successfully received and/or stored a chunk of data. The storage module 126 may wait for acknowledgements based on a setting/configuration, user input (e.g., an option selected by a user), etc.

[0033] In one embodiment, the storage module 126 (and/or storage module 136) may store at least a portion of the file in the private storage cloud 120 and/or private storage cloud 130, as discussed in more detail below. For example, the storage module 126 may store the file (e.g., the whole file, the complete file) in the private storage cloud 120 (e.g., in the NAS device 125). In another example, the storage module 126 may divide the file into a set of file chunks (e.g., into different chunks, portions, pieces, etc.) and may store one or more of the chunks (e.g., file chunks) in the private storage cloud 120 (e.g., in the NAS device 125).

[0034] In one embodiment, the storage module 126 (and/or storage module 136) may divide the file into the set of chunks (e.g., a set of file chunks) based on an importance level of the file. For example, if the importance level of the file is above a first importance threshold (e.g., a first threshold importance level), the storage module 126 may divide the file into the set of chunks and may store the set of chunks in one or more of the public storage clouds 140A through 140X. For example, the storage module may distribute the set of chunks across the public storage clouds 140A through 140X. The storage module 126 may also store the set of chunks in the private storage cloud 120, as discussed above.

[0035] In one embodiment, the storage module 126 (and/or storage module 136) may also determine a privacy level for the file. For example, the storage module 126 may analyze the file and/or the contents of the file to determine the privacy level of the file. In another example, the storage module 126 may determine the privacy level of the file based on data/input received from a user of the storage system 100 (e.g., a user may provide the privacy level for the file using storage application 111). In a further example the storage module 126 may determine the privacy level of the file based on a configuration file/setting. In some embodiments, a privacy level for a file may be data (e.g., a number, a value, an alphanumeric string, etc.) that may indicate how private and/or public the file is to a user of the storage system 100. For example, the privacy level of a file may indicate how much the user

would like the file to remain private (e.g., how much the user would like to prevent others from accessing the file). One having ordinary skill in the art understands that various representations may be used to represent the privacy levels and various numbers of privacy levels may be used. For example, the storage system 100 may have three, five, ten, etc., privacy levels. In another example, a numerical representation may be used such that the higher the number, the higher the privacy level (or vice versa).

[0036] In one embodiment, the storage module 126 (and/or storage module 136) may encrypt the set of chunks (of the file) before storing the set of chunks in one or more of the public storage clouds 140A through 140X, based on the privacy level. For example, the storage module 126 may encrypt the set of chunks (of the file) if the privacy level of the file is above a privacy threshold (e.g., is greater than a threshold privacy level). The storage module 126 may use different types, methods, algorithm, operations, functions, etc., for encrypting the set of chunks based on the privacy level. For example, the storage module 126 may use a 256-bit encryption algorithm (e.g., a 256-bit Advanced Encryption Standard (AES)) if the file has the highest privacy level. In another example, the storage module may use a 128-bit encryption algorithm if the file has the second highest privacy level.

[0037] In one embodiment, the storage module 126 may also encrypt the recovery data chunks (e.g., the systematic recovery data chunks and/or non-systematic recovery data chunks) for a file, based on the privacy level. For example, the storage module 126 may encrypt the recovery data chunks (of the file) if the privacy level of the file is above the privacy threshold. The storage module 126 may use different types, methods, algorithm, operations, functions, etc., for encrypting the set of chunks based on the privacy level, as discussed above.

[0038] In one embodiment, the storage module 126 (and/or storage module 136) may determine a trust level for the private storage cloud 120 (and/or private storage cloud 130). For example, the storage module 126 may analyze the security settings, parameters, protocols, encryption, etc., used by the private storage cloud to determine the trust level for the private storage cloud. In another example, the storage module 126 may determine the trust level for the private storage cloud 120 based on data/input received

from a user of the storage system 100 (e.g., a user may provide the trust level for the private storage cloud 120 and/or the trust level for the private storage cloud 130 using storage application 111). In a further example the storage module 126 may determine the trust level for the private storage cloud 120 based on a configuration file/setting. In some embodiments, a trust level for a private storage cloud may be data (e.g., a number, a value, an alphanumeric string, etc.) that may indicate how much the user trusts the NAS 125. For example, the trust level may indicate how secure the NAS 125 (or the private storage cloud 120) seems to the user. In another example, the trust level may indicate how reliable the NAS 125 (and/or the private storage cloud 120) seems to the user. One having ordinary skill in the art understands that various representations may be used to represent the trust levels and various numbers of trust levels may be used. For example, the storage system 100 may have three, five, ten, etc., trust levels. In another example, a numerical representation may be used such that the higher the number, the higher the trust level (or vice versa).

[0039] In one embodiment, the storage module 126 (and/or storage module 136) may store all portions of the file in the private storage cloud 120 if the trust level of the private storage cloud 120 is above a threshold trust level. For example, the storage module 126 may store the complete file (e.g., all chunks, portions, pieces, etc.) of the file on the NAS device 125 if the trust level of the private storage cloud 120 is above a threshold trust level. If the trust level of the private storage cloud 120 is not above the threshold trust level (e.g., is less than or equal to the threshold trust level), the storage module 126 may determine whether the importance level of the file is below a first importance level. If the importance level of the file is below the first importance level (and the trust level of the private storage cloud 120 is not above the threshold trust level), the storage module 126 may store the file in the private storage cloud 120.

[0040] In one embodiment, the storage module 126 (and/or storage module 136) may also determine a hotness level for the file. For example, the storage module 126 may analyze the file and/or the contents of the file to determine the hotness level of the file. In another example, the storage module 126 may determine the hotness level of the file based on data/input

received from a user of the storage system 100 (e.g., a user may provide the hotness level for the file using storage application 111). In a further example the storage module 126 may determine the hotness level of the file based on a configuration file/setting. In some embodiments, a hotness level for a file may be data (e.g., a number, a value, an alphanumeric string, etc.) that may indicate how often a file may be accessed. For example, the hotness level may indicated how often the file may be read, written to, updated/modified, etc. One having ordinary skill in the art understands that various representations may be used to represent the hotness levels and various numbers of hotness levels may be used. For example, the storage system 100 may have two, three, five, ten, etc., hotness levels. In another example, a numerical representation may be used such that the higher the number, the higher the hotness level (or vice versa).

[0041] In one embodiment, the storage module 126 (and/or storage module 136) may divide the set of file into a set of chunks (e.g., a set of file chunks) based on the hotness level of the file and the privacy level of the file. For example, if the hotness level of the file is above a threshold hotness level (e.g., a hotness threshold) and the privacy level for the file is below the threshold privacy level, the storage module 126 may divide the file into the set of chunks and may store the set of chunks in one or more of the public storage clouds 140A through 140X. For example, the storage module may distribute across the set of chunks across the public storage clouds 140A through 140X.

[0042] In one embodiment, the storage module 126 (and/or storage module 136) may determine that a file has been updated. For example, the storage module 126 may determine that a user has changed, added, and/or removed the content of the file. The storage module 126 may receive data indicating that a file has been updated from one or more of the public storage clouds 140A through 140X, and/or one or more of the private storage clouds 120 through 130. For example, when a user updates a file, the storage application 111 may transmit a message to the storage module 126 indicating that the file has been updated and/or may transmit the updated file to the storage module 126. The storage module 126 may generate a second set of recovery data chunks based on the updated file and/or the importance level

for updated the file, as discussed above. For example, the storage module 126 may generate a second set of systematic recovery data chunks or a second set of non-systematic recovery data chunks. The storage module 126 may also store the second set of recovery data chunks in one or more of the public storage clouds 140A through 140X, as discussed above. The storage module 126 may also store at least a portion of the updated filed in the private storage cloud 120, as discussed above.

[0043] In one embodiment, the storage module 126 (and/or storage module 136) may generate metadata 127 indicative of which recovery data chunks (in a set of recovery data chunks for the file) are stored in which public storage clouds, as discussed in more detail below. For example, the storage module 126 may generate a table that indicates where each recovery data chunk (in the set of recovery data chunks for the file) is stored (e.g., which public storage cloud stores which recovery data chunk). In another example, the storage module may generate a table that indicates where each chunk of the file (e.g., each file chunk) is stored (e.g., which public storage cloud stores which file chunk). The storage module 126 may also update the metadata 127 when a file is updated. For example, the second (e.g., updated) set of recovery data chunks may be stored on different public storage clouds than the previous set of recovery data chunks. The storage module 126 may update the metadata 127 to indicate which public storage clouds stored second set of recovery data chunks. The storage module 126 may also update the metadata 127 to indicate that a new version of the file was created.

[0044] In one embodiment, the protection level of the recovery data chunks (e.g., the systematic recovery data chunks and/or the non-systematic recovery data chunks) may be based on the importance level. The protection level of the recovery data chunks may indicate how easy it is to recover a file using the recovery data chunks. For example, the higher the protection level, the fewer non-systematic data recovery chunks may be used to reconstruct the file (e.g., for high protection level, two recovery data chunks may be used to reconstruct the file but for a lower protection level, four recovery data chunks may be used to reconstruct the file). The protection level of the recovery data chunks may also indicate how resistant the recovery data

chunks are to loss. For example, different data recovery (or error recovery) schemes/codes may still allow recovery/reconstruction of a file even when number of recovery data chunks are lost or inaccessible. A higher protection level may allow a higher number of recovery data chunks to become lost/inaccessible while still allowing recovery/reconstruction of the file, and vice versa.

[0045] In one embodiment, the storage application 111 may receive a request for a file stored in the storage system 100. For example, the storage application 111 may present a graphical user interface (or other interface, such as a command line interface) to a user of the computing device 110. The GUI may allow the user to read, write, modify, upload, download, files, etc. The storage application 111 may receive a request to read (e.g., access) a file stored in the storage system 100 (e.g., which may be stored and/or distributed among the private storage clouds 130 through 120 and public storage clouds 140A though 140X) via the GUI.

[0046] In one embodiment, the storage application may determine whether the file is retrievable from one or more of the public storage clouds 140A through 140X. For example, the storage application 111 may determine whether there are enough recovery data chunks stored in the public storage clouds 140A though 140X to reconstruct the file (from the recovery data chunks). In another example, the storage application 111 may determine whether file chunks (e.g., all of the portions, pieces, chunks, etc.) of the file have been stored (e.g., distributed) across the public storage clouds 140A through 140X. If the file is not retrievable from public storage clouds 140A through 140X, the storage application 111 may retrieve the file from private storage cloud 120. For example, if there are not enough recovery data chunks for the file in the public storage clouds 140A through 140X to reconstruct/regenerate the file, the storage application 111 may retrieve the file from the private storage cloud 120. In another example, if file chunks (for the file) are not stored in the public storage clouds 140A through 140X, the storage application 111 may retrieve the file from the private storage cloud 120.

[0047] If the file is retrievable from public storage clouds 140A through 140X (e.g., if there are file chunks for the file or if there are enough

recovery data chunks for the file stored in the public storage clouds 140A through 140X), the storage application 111 may determine whether access to the public storage clouds 140A though 140X is faster than access to the private storage cloud 120 (and/or private storage cloud 130). For example, the storage application 111 may determine the access latency (e.g., round-trip delay (RTD), round-trip time (RTT), ping time, etc.) for the private storage cloud 120 through 130 and the public storage clouds 140A through 140X. In another example, the storage application 111 may determine the data throughput for the private storage cloud 120 through 130 and the public storage clouds 140A through 140X (e.g., the amount of data that may be accessed/downloaded over a period of time from a public/private storage cloud).

[0048] In one embodiment, the storage application 111 may retrieve (e.g., download, access, read, etc.) the file from one or more of the private storage clouds 120 through 130, and public storage clouds 140A through 140X based on whether access to the public storage clouds 140A though 140X is faster than access to the private storage cloud 120 (and/or private storage cloud 130), if the file is retrievable from public storage clouds 140A through 140X. For example, the storage application 111 may retrieve the file from the private storage cloud 120 (e.g., may retrieve, download, access, etc., recovery data chunks and/or file chunks) when throughput to the private storage cloud 120 is faster than the throughput to the public storage clouds 140A through 140X. Throughput to the private storage cloud 120 may be faster when the computing device 110 is on a network (e.g., a local-area network (LAN)) of the private storage cloud 120. In another example, the storage application 111 may retrieve the file from one or more of the public storage clouds 140A through 140X (e.g., may retrieve, download, access, etc., recovery data chunks and/or file chunks) when throughput to one or more of the public storage clouds 140A through 140X is faster than the throughput to the private storage cloud 120.

[0049] In one embodiment, the storage application 111 may determine whether the private storage cloud 120 and the public storage clouds 140A through 140X may be accessed in parallel. For example, the storage application 111 may determine whether the computing device 110 is

able to access the private storage cloud 120 and one or more of public storage clouds 140A through 140X, simultaneously (e.g., by pinging and/or transmitting messages to the private storage cloud 120 and one or more of public storage clouds 140A through 140X and determining whether responses to the messages/pings are received). If the private storage cloud 120 and the public storage clouds 140A through 140X may be accessed in parallel, the storage application 111 may retrieve one or more portions of the file from the private storage cloud 120 and may retrieve one or more portions from the public storage clouds 140A through 140X. For example, the storage application 111 may retrieve one file chunk (of the file) from the private storage cloud 120 and may retrieve three file chunks (of the file) from the public storage clouds 140A through 140X. In another example, the storage application 111 may retrieve two recovery data chunks (of the file) from the private storage cloud 120 and may retrieve seven recovery data chunks (of the file) from the public storage clouds 140A through 140X.

[0050] If the private storage cloud 120 and the public storage clouds 140A through 140X may not be accessed in parallel, the storage application 111 may retrieve the file from the public storage clouds 140A through 140X. For example, the storage application 111 may download all of the file chunks (of the file) from one or more of the public storage clouds 140A through 140X and may combine the file chunks (e.g., concatenate the file chunks) to generate/obtain the file. In another example, the storage application 111 may download recovery data chunks from one or more of the public storage clouds 140A through 140X and may reconstruct (e.g., generate, obtain, etc.) the file using the recovery data chunks. The storage application 111 may download the minimum number of recovery data chunks that can be used to reconstruct (e.g., regenerate, rebuild, etc.) the file, based on the data recovery (or error recovery) scheme/code used to generate the recovery data chunks.

[0051] In one embodiment, the storage application 111 (and/or the storage module 126) may determine a hotness level for the file if the file is not retrievable from the public storage clouds 140A through 140X. For example, the storage application 111 may analyze the file and/or the contents of the file to determine the hotness level of the file. In another example, the storage application 111 may determine the hotness level of the file based on

data/input received from a user of the storage system 100 (e.g., a user may provide the hotness level for the file using storage application 111). In a further example the storage application 111 may determine the hotness level of the file based on a configuration file/setting. In one example, the storage application 111 may receive the hotness level for the file from the storage module 126 (and/or storage module 136). For example, the storage module 126 may determine the hotness level of the file and may provide the hotness level to the storage application 111.

[0052] In one embodiment, the storage application 111 may store one or more file chunks of the file (e.g., one or more portions of the file) and/or one or more recovery data chunks for the file on one or more of the public storage clouds 140A through 140X if the hotness level of the file is above a threshold hotness level. For example, the storage application 111 may divide the file into a set of file chunks (e.g., potions, pieces, etc.) and may store one or more of the file chunks on one or more of the public storage clouds 140A through 140X, if the hotness level of the file is above the threshold hotness level. In another example, the storage application 111 may generate recovery data chunks for the file and may store one or more of the recovery data chunks on one or more of the public storage clouds 140A through 140X, if the hotness level of the file is above the threshold hotness level.

[0053] In another embodiment, the storage application 111 may provide the file to the storage module 126 and the storage module 126 may store one or more file chunks of the file (e.g., one or more portions of the file) and/or one or more recovery data chunks for the file on one or more of the public storage clouds 140A through 140X if the hotness level of the file is above a threshold hotness level. For example, the storage application 111 may transmit the file to the storage module 126 and the storage module 126 may divide the file into chunks and/or generate recovery data chunks if the hotness level of the file is above the threshold hotness level. The storage module 126 may store the file chunks and/or the recovery data chunks on one or more of the public storage clouds 140A through 140X.

[0054] In one embodiment, the storage application 111 may determine whether the latest version of the file is retrievable from one or more of the public storage clouds 140A through 140X. For example, the storage

application 111 may communicate with the storage module 126 (and/or the storage module 136) to determine whether the latest version of the file (e.g., file chunks for the latest version of the file, recovery data chunks for the latest version of the file, etc.) is retrievable from one or more of the public storage clouds 140A through 140X. In another example, the storage application 111 may access the metadata 127 stored on the NAS device 125 to determine whether the latest version of the file is retrievable from one or more the public storage clouds 140A through 140X, as discussed in more detail below. If the latest version of the file is not retrievable from the public storage clouds 140A through 140X, the storage application 111 may retrieve the file from the private storage cloud 120 (and/or private storage cloud 130). For example, the storage application 111 may download the file from the NAS device 125. The storage application 111 may also request that the latest version of the file be stored in the public storage clouds 140A through 140X when the latest version of the file is not retrievable from the public storage clouds 140A through 140X. For example, the storage application 111 may transmit a message (e.g., a request) to the storage module 126 (and/or storage module 136) indicating that the storage module 126 should divide the file in to file chunks and/or should generate recovery data chunks (e.g., generate systematic recovery data chunks, generate non-systematic recovery data chunks, etc.)

[0055] In one embodiment, the storage module 126 (and/or the storage application 111) may track the version of file that is stored in the storage system 100. For example, the version of the file may be changed each time a user updates the file (e.g., the user edits the file). The storage module 126 (and/or the storage application 111) may maintain version information (e.g., data indicating the different version of the file, the latest version of the file, etc.) in the metadata 127. For example, the version may be a numerical value that is incremented each time the file is changed/updated by the user. Other types of information that may be use to track the version of the file may be a timestamp indicating the last time a file was updated/modified, the size of the file, etc. In another embodiment, the storage module 126 (and/or the storage application 111) may periodically distribute an updated file through the storage system 100. For example, if a file (that is

stored in the private storage cloud 120) is modified by a user, the storage module 126 may regenerate new file chunks and/or recovery data chunks and may distribute the chunks among the public storage clouds 140A through 140X.

[0056] One having ordinary skill in the art understands that the operations, methods, actions, processes, etc., described herein may be performed by one or more of the storage module 126, the storage module 136, and/or the storage application 111. For example, the storage module 126 may perform some of the operations described herein (e.g., generating file chunks) while the storage application 111 may perform other operations described herein (e.g., retrieving file chunks). In another example, the storage application 111 may perform all of the operations described herein (e.g., generating file chunks, transmitting the file chunks to the public storage clouds, and retrieving the file chunks).

[0057] In the embodiments, examples, implementations, etc., disclosed herein, the storage system (e.g., the hybrid storage system) may use three importance levels: importance level 0, importance level 1, and importance level 2. A higher importance level may indicate that a file is more important to a user, and vice versa. The storage system may also use three privacy levels: privacy level 0, privacy level 1, and privacy level 2. A higher privacy level may indicate that the user wants to keep the file more private, and vice versa. The storage system may further use two hotness levels: hotness level 0, and hotness level 1. A higher hotness level may indicate that a file may be more frequently accessed within a period of time (within a shorter period or more recently), and vice versa. One having ordinary skill in the art understands that different numbers of importance levels, privacy levels, and/or hotness levels may be used in other embodiments.

[0058] In some embodiments, distributing the file chunks (for a file) and/or recovery data chunks (e.g., systematic recovery data chunks, non-systematic recovery data chunks) across the private storage clouds 120 through 130 and the public storage clouds 140A through 140X may allow a user faster access to the file. For example, the user may be able to access (e.g., download) the different file chunks from the public storage clouds 140A through 140X simultaneously which may allow the user to download the file

from the public storage clouds 140A through 140X more quickly (e.g., by downloading different portions/chunks of the file from different public storage clouds). In another example, the user may be able to access (e.g., download) the different recovery data chunks from the public storage clouds 140A through 140X simultaneously. This may allow the user to download the recovery data chunks from the public storage clouds 140A through 140X more quickly (e.g., by downloading different recovery data chunks of the file from different public storage clouds) and reconstruct, rebuild, regenerate, etc., the file from the recovery data chunks.

[0059] In some embodiments, distributing the file chunks (for a file) and/or recovery data chunks (e.g., systematic recovery data chunks, non-systematic recovery data chunks) across the private storage clouds 120 through 130 and the public storage clouds 140A through 140X may allow a user faster access to the file based on their location. For example, storing the file and/or chunks (e.g., file chunks, recovery data chunks, etc.) of the file on the private storage cloud may allow the user to access the file more quickly if the user is on the LAN provided by the private storage cloud (e.g., if the user is at home, at work, etc.). In another example, storing the file and/or chunks (e.g., file chunks, recovery data chunks, etc.) of the file on the public storage clouds may allow the user to access the file more quickly if the user is outside of the private storage clouds (e.g., outside of home/work, at a coffee shop, etc.).

[0060] In some embodiments, distributing the set of chunks across the private storage clouds 120 through 130 and the public storage clouds 140A through 140X may also help increase the security of the storage system 100. For example, if one public storage cloud is compromised by another user (e.g., a malicious user), the other user may still be unable to obtain the file because other file chunks and/or other recovery data chunks are stored on other public storage clouds and the other user would be unable to reconstruct, rebuild, regenerate, etc., the file without comprising the other public storage clouds.

[0061] In some embodiments, encrypting the file chunks and/or the recovery data chunks may also increase the security of the storage system 100. For example, if one or more public storage cloud are compromised by

another user (e.g., a malicious user), the user may still not be able to reconstruct, rebuild, regenerate, etc., the file because the file chunks and/or the recovery data chunks are encrypted prior to storing the file chunks and/or the recovery data chunks on the private storage clouds 120 through 130 and the public storage clouds 140A through 140X.

[0062] Figure 2A is a diagram illustrating an example distribution of a file in a storage system, in accordance with one or more embodiments. As discussed above, a file, such as FILE1, may be stored in a storage system (e.g., storage system 100 illustrated in Figure 1). The storage system may be a hybrid storage system that includes a private storage cloud 120, public storage cloud 140A, public storage cloud 140B, and public storage cloud 140C. The private storage cloud 120 may include a NAS device that may store the FILE1. Each of the public storage cloud 140A, 140B, and 140C may be separate/different storage service providers (e.g., a company, entity, service provider, etc., that provides storage space/capacity for storing data).

[0063] As illustrated in Figure 2A, FILE1 is stored on the private storage cloud 120. For example, the complete file FILE1 is stored in a memory (e.g., magnetic media, flash media, etc.) on a NAS device of the private storage cloud 120 (e.g., NAS device 125 illustrated in Figure 1). Also as illustrated in Figure 2A, recovery data chunks are distributed among the public storage clouds 140A, 140B, and 140C. The recovery data chunks may be systematic recovery data chunks (e.g., systematic recovery data, etc.) S_RECOVERY_DATA_1 and S_RECOVERY_DATA_4 are stored on public storage cloud 140A, S_RECOVERY_DATA_2 and S_RECOVERY_DATA_5 are stored on public storage cloud 140B, and S_RECOVERY_DATA_3 and S_RECOVERY_DATA_6 are stored on public storage cloud 140C. The protection level of the recovery data S_RECOVERY_DATA_1 through S_RECOVERY_DATA_6 may be based on the importance level of the file, as discussed above. For example, for a lower importance level, the systematic recovery data (e.g., S_RECOVERY_DATA_1 through S_RECOVERY_DATA_6) may be generated using exclusive OR (XOR) operations. In another example, for a higher importance level, the systematic recovery data (e.g., S_RECOVERY_DATA_1 through S_RECOVERY_DATA_6) may be generated using RAID

functions/operations. In one embodiment, the recovery data may be encrypted based on the privacy level of the file, as discussed above. For example, S_RECOVERY_DATA_1 through S_RECOVERY_DATA_6 may be encrypted before they are stored on the public storage clouds 140A through 140C if the privacy level for FILE1 is greater than a privacy threshold.

[0064] As discussed above, a client device (e.g., a storage application on the client device) may download FILE1 from the private storage cloud 120 and/or one or more of the public storage clouds 140A through 140C based on whether the client device is able to access the private storage cloud 120 and public storage clouds 140A through 140C simultaneously. The client device may also determine whether access to the public storage clouds 140A though 140C is faster than access to the private storage cloud 120, as discussed above. For example, if access to the private storage cloud 120 is faster (e.g., the client device is on a LAN of the private storage cloud 120), the client device may download FILE1 from the private storage cloud 120. In another example, if access to the public storage clouds 140A through 140C is faster, the client device may download one or more of the systematic recovery data (e.g., one or more of S_RECOVERY_DATA_1 through S_RECOVERY_DATA_6) and may reconstruct, regenerate, etc., the file using the systematic recovery data. If the systematic recovery data is encrypted, the client device may also decrypt the systematic recovery data.

[0065] Figure 2B is a diagram illustrating an example distribution of a file in a storage system, in accordance with one or more embodiments. As discussed above, a file, such as FILE1, may be stored in a storage system (e.g., storage system 100 illustrated in Figure 1). The storage system may be a hybrid storage system that includes a private storage cloud 120, public storage cloud 140A, public storage cloud 140B, and public storage cloud 140C. The private storage cloud 120 may include a NAS device that may store the FILE1. Each of the public storage cloud 140A, 140B, and 140C may be separate/different storage service providers (e.g., a company, entity, service provider, etc., that provides storage space/capacity for storing data).

[0066] As illustrated in Figure 2B, FILE1 (e.g., all of FILE1) is stored on the private storage cloud 120, as discussed above. Also as illustrated in Figure 2A, FILE1 is divided into three file chunks, FILE1_CHUNK1,

FILE1_CHUNK2, and FILE1_CHUNK3. FILE1_CHUNK1 is stored on public storage cloud 140A, FILE1_CHUNK2 is stored on public storage cloud 140B, and FILE1_CHUNK3 is stored on public storage cloud 140C. In addition, recovery data chunks are distributed among the public storage clouds 140A, 140B, and 140C. The recovery data chunks may be systematic recovery data chunks (e.g., systematic recovery data, etc.) S_RECOVERY_DATA_1 and S_RECOVERY_DATA_4 are stored on public storage cloud 140A, S_RECOVERY_DATA_2 and S_RECOVERY_DATA_5 are stored on public storage cloud 140B, and S_RECOVERY_DATA_3 and S_RECOVERY_DATA_6 are stored on public storage cloud 140C. The protection level of the recovery data S_RECOVERY_DATA_1 through S_RECOVERY_DATA_6 may be based on the importance level of the file, as discussed above. In one embodiment, the recovery data may be encrypted based on the privacy level of the file, as discussed above.

[0067] As discussed above, a client device (e.g., a storage application on the client device) may download FILE1 from the private storage cloud 120 and/or one or more of the public storage clouds 140A through 140C based on whether the client device is able to access the private storage cloud 120 and public storage clouds 140A through 140C simultaneously. The client device may also determine whether access to the public storage clouds 140A though 140C is faster than access to the private storage cloud 120, as discussed above. For example, if access to the private storage cloud 120 is faster (e.g., the client device is on a LAN of the private storage cloud 120), the client device may download FILE1 from the private storage cloud 120. In another example, if access to the public storage clouds 140A through 140C is faster, the client device may download one or more of the systematic recovery data (e.g., one or more of S_RECOVERY_DATA_1 through S_RECOVERY_DATA_6) and may reconstruct, regenerate, etc., the file using the systematic recovery data. If the systematic recovery data is encrypted, the client device may also decrypt the systematic recovery data. In a further example, if access to the public storage clouds 140A through 140C is faster, the client device may download the file chunks (e.g., FILE1_CHUNK1 through FILE1_CHUNK3) and may obtain the file using the file chunks (e.g., may

append, concatenate, combine, etc., the file chunks together). If the file chunks are encrypted, the client device may also decrypt the file chunks.

[0068] Figure 2C is a diagram illustrating an example distribution of a file in a storage system, in accordance with one or more embodiments. As discussed above, a file, such as FILE2 (not illustrated in Figure 2C), may be stored in a storage system (e.g., storage system 100 illustrated in Figure 1). The storage system may be a hybrid storage system that includes a private storage cloud 120, public storage cloud 140A, public storage cloud 140B, and public storage cloud 140C. The private storage cloud 120 may include a NAS device. Each of the public storage cloud 140A, 140B, and 140C may be separate/different storage service providers (e.g., a company, entity, service provider, etc., that provides storage space/capacity for storing data).

[0069] As illustrated in Figure 2C, recovery data chunks are distributed among the public storage clouds 140A, 140B, and 140C. The recovery data chunks may be non-systematic recovery data chunks (e.g., non-systematic recovery data, etc.) NS_RECOVERY_DATA_1 is stored in the private storage cloud 120, NS_RECOVERY_DATA_2 and NS_RECOVERY_DATA_5 are stored on public storage cloud 140A, NS_RECOVERY_DATA_3 is stored on public storage cloud 140B, and NS_RECOVERY_DATA_4 and NS_RECOVERY_DATA_6 are stored on public storage cloud 140C. The protection level of the recovery data NS_RECOVERY_DATA_1 through NS_RECOVERY_DATA_6 may be based on the importance level of the file, as discussed above. In one embodiment, the recovery data may be encrypted based on the privacy level of the file, as discussed above.

[0070] As discussed above, a client device (e.g., a storage application on the client device) may download FILE2 from the private storage cloud 120 and/or one or more of the public storage clouds 140A through 140C based on whether the client device is able to access the private storage cloud 120 and public storage clouds 140A through 140C simultaneously. The client device may also determine whether access to the public storage clouds 140A though 140C is faster than access to the private storage cloud 120, as discussed above. If access to the private storage cloud 120 is faster, the client device may download NS_RECOVERY_DATA_1 from the private

storage cloud 120 and may download one or more of the remaining recovery data from the public storage clouds 140A through 140C. For example, if at least three non-systematic recovery data chunks out of the six non-systematic recovery data chunks are be used to reconstruct FILE2, the client device may download NS_RECOVERY_DATA_1 from the private storage cloud 120 and may download two other non-systematic recovery data chunks (e.g., NS_RECOVERY_DATA_2 and NS_RECOVERY_DATA_4) one or more of the public storage clouds 140A through 140C. If access to the public storage clouds 140A through 140C is faster, the client device may download one or more of the non-systematic recovery data chunks (e.g., one or more of NS_RECOVERY_DATA_1 through NS_RECOVERY_DATA_6) from the public storage clouds 140A through 140C and may reconstruct, regenerate, etc., the file using the non-systematic recovery data chunks. For example, if at least three non-systematic recovery data chunks out of the six non-systematic recovery data chunks are be used to reconstruct FILE2, the client device may download NS_RECOVERY_DATA_2 and NS_RECOVERY_DATA_4 from public storage cloud 140A and may download NS_RECOVERY_DATA_3 from public storage cloud 140B. If the non-systematic recovery data chunks are encrypted, the client device may also decrypt the non-systematic recovery data chunks.

[0071] Figure 2D is a diagram illustrating an example distribution of a file in a storage system, in accordance with one or more embodiments. As discussed above, a file, such as FILE2, may be stored in a storage system (e.g., storage system 100 illustrated in Figure 1). The storage system may be a hybrid storage system that includes a private storage cloud 120, public storage cloud 140A, public storage cloud 140B, and public storage cloud 140C. Each of the public storage cloud 140A, 140B, and 140C may be separate/different storage service providers (e.g., a company, entity, service provider, etc., that provides storage space/capacity for storing data).

[0072] As illustrated in Figure 2D, FILE2 is stored on the private storage cloud 120. For example, the complete file FILE2 is stored in a memory (e.g., magnetic media, flash media, etc.) on a NAS device of the private storage cloud 120 (e.g., NAS device 125 illustrated in Figure 1). Also as illustrated in Figure 2D, recovery data chunks are distributed among the

public storage clouds 140A, 140B, and 140C. The recovery data chunks may be non-systematic recovery data chunks (e.g., non-systematic recovery data, etc.) NS_RECOVERY_DATA_1 is stored in the private storage cloud 120, NS_RECOVERY_DATA_2 and NS_RECOVERY_DATA_5 are stored on public storage cloud 140A, NS_RECOVERY_DATA_3 is stored on public storage cloud 140B, and NS_RECOVERY_DATA_4 and NS_RECOVERY_DATA_6 are stored on public storage cloud 140C. The protection level of the recovery data NS_RECOVERY_DATA_1 through NS_RECOVERY_DATA_6 may be based on the importance level of the file, as discussed above. In one embodiment, the recovery data may be encrypted based on the privacy level of the file, as discussed above.

[0073] As discussed above, a client device (e.g., a storage application on the client device) may download FILE2 from the private storage cloud 120 and/or one or more of the public storage clouds 140A through 140C based on whether the client device is able to access the private storage cloud 120 and public storage clouds 140A through 140C simultaneously. The client device may also determine whether access to the public storage clouds 140A though 140C is faster than access to the private storage cloud 120, as discussed above. If access to the private storage cloud 120 is faster, the client device may download FILE2 from the private storage cloud 120. If access to the public storage clouds 140A through 140C is faster, the client device may download one or more of the non-systematic recovery data chunks (e.g., one or more of NS_RECOVERY_DATA_1 through NS_RECOVERY_DATA_6) from the public storage clouds 140A through 140C and may reconstruct, regenerate, etc., the file using the non-systematic recovery data chunks, as discussed above. If the non-systematic recovery data chunks are encrypted, the client device may also decrypt the non-systematic recovery data chunks.

[0074] Figure 2E is a diagram illustrating an example distribution of a file in a storage system, in accordance with one or more embodiments. As discussed above, a file, such as FILE2 (not illustrated in Figure 2E), may be stored in a storage system (e.g., storage system 100 illustrated in Figure 1). The storage system may be a hybrid storage system that includes a private storage cloud 120, public storage cloud 140A, public storage cloud 140B, and

public storage cloud 140C. The private storage cloud 120 may include a NAS device. Each of the public storage cloud 140A, 140B, and 140C may be separate/different storage service providers (e.g., a company, entity, service provider, etc., that provides storage space/capacity for storing data).

[0075] As illustrated in Figure 2E, FILE2 is divided into three file chunks, FILE2_CHUNK1, FILE2_CHUNK2, and FILE2_CHUNK3. FILE2_CHUNK1 is stored on public storage cloud 140A, FILE2_CHUNK2 is stored on public storage cloud 140B, and FILE2_CHUNK3 is stored on public storage cloud 140C. Also as illustrated in Figure 2D, recovery data chunks are distributed among the public storage clouds 140A, 140B, and 140C. The recovery data chunks may be non-systematic recovery data chunks (e.g., non-systematic recovery data, etc.) NS_RECOVERY_DATA_1 is stored in the private storage cloud 120, NS_RECOVERY_DATA_2 and NS_RECOVERY_DATA_5 are stored on public storage cloud 140A, NS_RECOVERY_DATA_3 is stored on public storage cloud 140B, and NS_RECOVERY_DATA_4 and NS_RECOVERY_DATA_6 are stored on public storage cloud 140C. The protection level of the recovery data NS_RECOVERY_DATA_1 through NS_RECOVERY_DATA_6 may be based on the importance level of the file, as discussed above. In one embodiment, the recovery data may be encrypted based on the privacy level of the file, as discussed above.

[0076] As discussed above, a client device (e.g., a storage application on the client device) may download FILE2 from the private storage cloud 120 and/or one or more of the public storage clouds 140A through 140C based on whether the client device is able to access the private storage cloud 120 and public storage clouds 140A through 140C simultaneously. The client device may also determine whether access to the public storage clouds 140A though 140C is faster than access to the private storage cloud 120, as discussed above. If access to the private storage cloud 120 is faster, the client device may download NS_RECOVERY_DATA_1 from the private storage cloud 120 and may download one or more of the remaining recovery data from the public storage clouds 140A through 140C. For example, if at least three non-systematic recovery data chunks out of the six non-systematic recovery data chunks are be used to reconstruct FILE2, the client device may

download NS_RECOVERY_DATA_1 from the private storage cloud 120 and may download two other non-systematic recovery data chunks (e.g., NS_RECOVERY_DATA_2 and NS_RECOVERY_DATA_4) one or more of the public storage clouds 140A through 140C. If access to the public storage clouds 140A through 140C is faster, the client device may download one or more of the non-systematic recovery data chunks (e.g., one or more of S_RECOVERY_DATA_1 through S_RECOVERY_DATA_6) from the public storage clouds 140A through 140C and may reconstruct, regenerate, etc., the file using the non-systematic recovery data chunks. For example, if at least three non-systematic recovery data chunks out of the six non-systematic recovery data chunks are be used to reconstruct FILE2, the client device may download NS_RECOVERY_DATA_2 and NS_RECOVERY_DATA_4 from public storage cloud 140A and may download NS_RECOVERY_DATA_3 from public storage cloud 140B. If the non-systematic recovery data chunks are encrypted, the client device may also decrypt the non-systematic recovery data chunks. The client device may also download file chunks (e.g., FILE2_CHUNK1 through FILE2_CHUNK3) from public storage clouds 140A through 140C and may obtain the file using the file chunks (e.g., may append, concatenate, combine, etc., the file chunks together), if access to the public storage clouds 140A through 140C is faster. If the file chunks are encrypted, the client device may also decrypt the file chunks.

[0077] Figure 2F is a diagram illustrating an example distribution of a file in a storage system, in accordance with one or more embodiments. As discussed above, a file, such as FILE2, may be stored in a storage system (e.g., storage system 100 illustrated in Figure 1). The storage system may be a hybrid storage system that includes a private storage cloud 120, public storage cloud 140A, public storage cloud 140B, and public storage cloud 140C. The private storage cloud 120 may include a NAS device. Each of the public storage cloud 140A, 140B, and 140C may be separate/different storage service providers (e.g., a company, entity, service provider, etc., that provides storage space/capacity for storing data).

[0078] As illustrated in Figure 2F, FILE2 is stored in the private storage cloud 120. For example, the complete file FILE2 is stored in a memory (e.g., magnetic media, flash media, etc.) on a NAS device of the

private storage cloud 120 (e.g., NAS device 125 illustrated in Figure 1). Also as illustrated in Figure 2F, FILE2 is divided into three file chunks, FILE2_CHUNK1, FILE2_CHUNK2, and FILE2_CHUNK3. FILE2_CHUNK1 is stored on public storage cloud 140A, FILE2_CHUNK2 is stored on public storage cloud 140B, and FILE2_CHUNK3 is stored on public storage cloud 140C.

[0079] As discussed above, a client device (e.g., a storage application on the client device) may download FILE2 from the private storage cloud 120 and/or one or more of the public storage clouds 140A through 140C based on whether the client device is able to access the private storage cloud 120 and public storage clouds 140A through 140C simultaneously. The client device may also determine whether access to the public storage clouds 140A though 140C is faster than access to the private storage cloud 120, as discussed above. If access to the private storage cloud 120 is faster, the client device may download FILE2 from the private storage cloud 120. If access to the public storage clouds 140A through 140C is faster, the client device may download file chunks (e.g., FILE2_CHUNK1 through FILE2_CHUNK3) from public storage clouds 140A through 140C and may obtain the file using the file chunks (e.g., may append, concatenate, combine, etc., the file chunks together). If the file chunks are encrypted, the client device may also decrypt the file chunks.

[0080] Figure 3 is a table 300 illustrating example metadata, in accordance with one or more embodiments. The table 300 includes five columns: File Name, Chunk ID, Chunk Location, Recovery Data ID, and Recovery Data Location. The File Name column may indicate file names (e.g., or some other identifier such as a hashes) for files that are stored on the data storage system. The Chunk ID column may indicate names (or other identifiers) for file chunks (e.g., pieces, portions, chunks of a file) which are generated for files that are stored on the data storage system. The Chunk Location column may indicate which public storage clouds store which file chunks. The Recovery Data ID column may indicate names (or other identifiers) for recovery data chunks (e.g., systematic recovery data chunks, non-systematic recovery data chunks, etc.) which are generated for files that

are stored on the data storage system. The Recovery Data Location column may indicate which public storage clouds store which recovery data chunks.

[0081] As illustrated in Figure 3, table 300 indicates that there are two files stored in the storage system (e.g., the hybrid storage system), FILE_A and FILE_B. As illustrated by table 300, FILE_A is divided into three file chunks CHUNK_A1, CHUNK_A2, and CHUNK_A3. CHUNK_A1 is stored on PUBLIC_STORAGE_CLOUD1, CHUNK_A2 is stored on PUBLIC_STORAGE_CLOUD2, and CHUNK_A3 is stored on PUBLIC_STORAGE_CLOUD3. Systematic recovery data S_RECOVERY_DATA_A1 through S_RECOVERY_DATA_A6 were generated for FILE_A and are stored in the public storage clouds. S_RECOVERY_DATA_A1 and S_RECOVERY_DATA_A4 are stored in PUBLIC_STORAGE_CLOUD1, S_RECOVERY_DATA_A2 and S_RECOVERY_DATA_A5 are stored in PUBLIC_STORAGE_CLOUD2, and S_RECOVERY_DATA_A3 and S_RECOVERY_DATA_A6 are stored in PUBLIC_STORAGE_CLOUD3.

[0082] As indicated by the blank entries in the Chunk ID and Chunk Location columns for FILE_B, FILE_B was not divided into file chunks and no file chunks for FILE_B are stored in the public storage clouds. Non-systematic recovery data NS_RECOVERY_DATA_A1 through S_RECOVERY_DATA_A5 were generated for FILE_B and are stored in the public storage clouds. NS_RECOVERY_DATA_B1 is stored in PRIVATE_STORAGE_CLOUD1. NS_RECOVERY_DATA_B2 and NS_RECOVERY_DATA_B5 are stored in PUBLIC_STORAGE_CLOUD1. NS_RECOVERY_DATA_B3 is stored in PUBLIC_STORAGE_CLOUD2. NS_RECOVERY_DATA_B4 is stored in PUBLIC_STORAGE_CLOUD3.

[0083] Figure 4 is a flow diagram illustrating an example process 400 for writing data (e.g., a file) to a storage system, in accordance with one or more embodiments. The storage system may include a private storage cloud (or multiple storage clouds) and may include one or more public storage clouds, as discussed above. The storage system may be referred to as a hybrid storage system. The process 400 may be performed by a storage module (e.g., storage module 126 illustrated in Figure 1), a storage application (e.g., storage application 111 illustrated in Figure 1), a processing

device (e.g., a processor, a central processing unit (CPU)), and/or a computing device (e.g., a NAS device). The storage module, storage application, processing device, and/or computing device may be processing logic that includes hardware (e.g., circuitry, dedicated logic, programmable logic, microcode, etc.), software (e.g., instructions run on a processor to perform hardware simulation), firmware, or a combination thereof.

[0084] The process 400 begins at block 405 where the process 400 may receive a file to be stored in the storage system. For example, the file may be received from a client device and/or a storage application (on the client device). At block 410, the process 400 determines whether the trust level of the private storage cloud is greater than a trust threshold (e.g., is greater than a threshold trust level). For example, the data storage system may use two trust levels for private storage clouds, trust level 0 and trust level 1. The process 400 may determine whether the trust level of the private storage cloud is greater than trust level 0. If the trust level of the private storage cloud is greater than trust level 0 (e.g. the trust level is equal to trust level 1), the process 400 may proceed to block A, which is illustrated and discussed in more detail below in conjunction with Figure 5. If the trust level of the private storage cloud is not greater than trust level 0 (e.g. the trust level is equal to trust level 0), the process 400 may proceed to block B, which is illustrated and discussed in more detail below in conjunction with Figure 6.

[0085] Figure 5 is a flow diagram illustrating an example process 500 for writing data (e.g., a file) to a storage system, in accordance with one or more embodiments. The storage system may include a private storage cloud (or multiple storage clouds) and may include one or more public storage clouds, as discussed above. The storage system may be referred to as a hybrid storage system. The process 500 may be performed by a storage module (e.g., storage module 126 illustrated in Figure 1), a storage application (e.g., storage application 111 illustrated in Figure 1), a processing device (e.g., a processor, a central processing unit (CPU)), and/or a computing device (e.g., a NAS device). The storage module, storage application, processing device, and/or computing device may be processing logic that includes hardware (e.g., circuitry, dedicated logic, programmable

logic, microcode, etc.), software (e.g., instructions run on a processor to perform hardware simulation), firmware, or a combination thereof.

[0086] As discussed above, Figure 5 illustrates the processes, operations, actions, etc., that may be performed in block A of Figure 4. In the embodiment illustrated in Figure 5, the storage system may use three importance levels: importance level 0, importance level 1, and importance level 2. A higher importance level may indicate that a file is more important to a user, and vice versa. The storage system may also use three privacy levels: privacy level 0, privacy level 1, and privacy level 2. A higher privacy level may indicate that the user wants to keep the file more private, and vice versa. The storage system may further use two hotness levels: hotness level 0, and hotness level 1. A higher hotness level may indicate that a file may be more frequently accessed, and vice versa. One having ordinary skill in the art understands that different numbers of importance levels, privacy levels, and/or hotness levels may be used in other embodiments.

[0087] The process 500 begins at block 505 where the process 500 may store the file in a private storage cloud, as discussed above. For example, the process 500 may store the file in a NAS device of the private storage cloud. At block 510, the process 500 determines whether the importance level of the file is greater than a first importance threshold, as discussed above. For example, the process 500 may determine whether the importance level of the file is greater than importance level 1 (e.g., whether the importance level of the file is equal to importance level 2). If the importance level of the file is greater than the first importance threshold, the process 500 may divide the file into file chunks (e.g., pieces, portions, chunks, etc.) at block 515. As discussed above, the size of each file chunk may vary in different embodiments (e.g., the size of each file chunk may be in bits, bytes, kilobytes, megabytes, etc.). The process 500 may generate systematic recovery data (e.g., systematic recovery data chunks) with a first protection level at block 520, as discussed above. The first protection level may indicate how easy it is to recover a file using the recovery data chunks and/or may indicate how resistant the recovery data chunks are to loss, as discussed above. At block 525, the process 500 determines whether the privacy level for the file is greater than a first privacy threshold, as discussed above. For

example, the process 500 may determine whether the privacy level of the file is greater than privacy level 1 (e.g., whether the privacy level of the file is equal to privacy level 2). If the privacy level of the file is greater than the first privacy level (e.g., the privacy level of the file is equal to privacy level 2), the process 500 may encrypt the file chunks at block 530 (as discussed above) and may proceed to block 535. If the privacy level of the file is not greater than the first privacy level (e.g., the privacy level of the file is equal to privacy level 1 or privacy level 0), the process 500 may proceed to block 535. At block 535, the process 500 may transmit the systematic recovery data and the file chunks (which may be encrypted) to the one or more public storage clouds, as discussed above. After block 535, the process may proceed to block 555.

[0088] If the importance level of the file is not greater than the first importance threshold, the process 500 may determine whether the importance level is greater than a second importance threshold at block 540, as discussed above. For example, the process 500 may determine whether the importance level of the file is greater than importance level 0 (e.g., whether the importance level of the file is equal to importance level 1). If the importance level of the file is greater than the second importance threshold, the process 500 may generate systematic recovery data (e.g., systematic recovery data chunks) with a second protection level at block 545, as discussed above. The second protection level may indicate how easy it is to recover a file using the recovery data chunks and/or may indicate how resistant the recovery data chunks are to loss, as discussed above. The second protection level may be lower than the first protection level (e.g., the recovery data chunks with the second protection level may be less resistant to loss and/or it may be harder to recover the file using the recovery data chunks with the second protection level). At block 550, the process may transmit the systematic recovery data to the one or more public storage clouds, as discussed above. After block 550, the process may proceed to block 555.

[0089] If the importance level of the file is not greater than the second importance threshold, the process 500 may determine whether hotness level of the file is greater than a threshold hotness (e.g., hotness level 0) and if the privacy level of the file is less than a second privacy threshold

(e.g., if the privacy level of the file is equal to privacy level 0) at block 565. If the hotness level is not greater than the threshold hotness and if the privacy level of the file is not less than the second privacy threshold, the process 500 may end. If the hotness level of the file is greater than the threshold hotness (e.g., hotness level 0) and if the privacy level of the file is less than the second privacy threshold, the process 500 may divided the file into chunks at block 570, as discussed above. At block 575, the process 500 may transmit the file chunks to the one or more public storage clouds, as discussed above.

[0090] At block 555, the process 500 may determine whether to wait for acknowledgements (e.g., acks) for the systematic recovery data and/or file chunks that were transmitted to the one or more public storage clouds, as discussed above. For example, the process 500 may determine whether to wait acknowledgements (from the one more public storage clouds) indicating that file chunks were received/stored. In another example, the process 500 may determine whether to wait acknowledgements (from the one more public storage clouds) indicating that recovery data chunks were received/stored. If the process 500 should wait for acknowledgments, the process 500 may proceed to block 560 where the process 500 may wait for acknowledgments from the one or more public storage clouds (e.g., may wait until the acknowledgements are received). For example, the process 500 may periodically check whether the acknowledgments were received. In one embodiment, the process 500 may retransmit the systematic recovery data and/or file chunks if acknowledgments are not received after a period of time, at block 560. After block 560, the process 500 ends.

[0091] Figure 6 is a flow diagram illustrating an example process 600 for writing data (e.g., a file) to a storage system, in accordance with one or more embodiments. The storage system may include a private storage cloud (or multiple storage clouds) and may include one or more public storage clouds, as discussed above. The storage system may be referred to as a hybrid storage system. The process 600 may be performed by a storage module (e.g., storage module 126 illustrated in Figure 1), a storage application (e.g., storage application 111 illustrated in Figure 1), a processing device (e.g., a processor, a central processing unit (CPU)), and/or a computing device (e.g., a NAS device). The storage module, storage

application, processing device, and/or computing device may be processing logic that includes hardware (e.g., circuitry, dedicated logic, programmable logic, microcode, etc.), software (e.g., instructions run on a processor to perform hardware simulation), firmware, or a combination thereof.

[0092] As discussed above, Figure 6 illustrates the processes, operations, actions, etc., that may be performed in block B of Figure 4. In the embodiment illustrated in Figure 6, the storage system may use three importance levels: importance level 0, importance level 1, and importance level 2. A higher importance level may indicate that a file is more important to a user, and vice versa. The storage system may also use three privacy levels: privacy level 0, privacy level 1, and privacy level 2. A higher privacy level may indicate that the user wants to keep the file more private, and vice versa. The storage system may further use two hotness levels: hotness level 0, and hotness level 1. A higher hotness level may indicate that a file may be more frequently accessed, and vice versa. One having ordinary skill in the art understands that different numbers of importance levels, privacy levels, and/or hotness levels may be used in other embodiments.

[0093] The process 600 begins at block 605 where the process 600 determines whether the importance level of the file is greater than a first importance threshold, as discussed above. For example, the process 600 may determine whether the importance level of the file is greater than importance level 1 (e.g., whether the importance level of the file is equal to importance level 2). If the importance level of the file is greater than the first importance threshold, the process 600 may divide the file into file chunks (e.g., pieces, portions, chunks, etc.) at block 610. As discussed above, the size of each file chunk may vary in different embodiments (e.g., the size of each file chunk may be in bits, bytes, kilobytes, megabytes, etc.). The process 600 may generate non-systematic recovery data (e.g., non-systematic recovery data chunks) with a first protection level at block 615, as discussed above. The first protection level may indicate how easy it is to recover a file using the recovery data chunks and/or may indicate how resistant the recovery data chunks are to loss, as discussed above. At block 620, the process 600 determines whether the privacy level for the file is greater than a first privacy threshold, as discussed above. For example, the

process 600 may determine whether the privacy level of the file is greater than privacy level 1 (e.g., whether the privacy level of the file is equal to privacy level 2). If the privacy level of the file is greater than the first privacy level (e.g., the privacy level of the file is equal to privacy level 2), the process 600 may encrypt the file chunks at block 625 (as discussed above) and may proceed to block 630. If the privacy level of the file is not greater than the first privacy level (e.g., the privacy level of the file is equal to privacy level 1 or privacy level 0), the process 600 may proceed to block 630. At block 630, the process 600 may transmit the non-systematic recovery data and the file chunks (which may be encrypted) to the one or more public storage clouds, as discussed above. After block 635, the process may proceed to block 655.

[0094] If the importance level of the file is not greater than the first importance threshold, the process 600 may determine whether the importance level is greater than a second importance threshold at block 635, as discussed above. For example, the process 600 may determine whether the importance level of the file is greater than importance level 0 (e.g., whether the importance level of the file is equal to importance level 1). If the importance level of the file is greater than the second importance threshold, the process 600 may store the file in the private storage cloud at block 640. For example, the process 600 may store the file in a NAS device of the private storage cloud. At block 645, the process 600 may generate non-systematic recovery data (e.g., non-systematic recovery data chunks) with a second protection level at block 640, as discussed above. The second protection level may indicate how easy it is to recover a file using the recovery data chunks and/or may indicate how resistant the recovery data chunks are to loss, as discussed above. The second protection level may be lower than the first protection level (e.g., the recovery data chunks with the second protection level may be less resistant to loss and/or it may be harder to recover the file using the recovery data chunks with the second protection level). At block 650, the process may transmit the non-systematic recovery data to the one or more public storage clouds, as discussed above. After block 650, the process may proceed to block 655.

[0095] If the importance level of the file is not greater than the second importance threshold, the process 600 may store the file in the private

storage cloud at block 665, as discussed above. At block 670, the process 600 may determine whether hotness level of the file is greater than a threshold hotness (e.g., hotness level 0) and if the privacy level of the file is less than a second privacy threshold (e.g., if the privacy level of the file is equal to privacy level 0). If the hotness level is not greater than the threshold hotness and if the privacy level of the file is not less than the second privacy threshold, the process 600 may end. If the hotness level of the file is greater than the threshold hotness (e.g., hotness level 0) and if the privacy level of the file is less than the second privacy threshold, the process 600 may divided the file into chunks at block 675, as discussed above. At block 680, the process 600 may transmit the file chunks to the one or more public storage clouds, as discussed above.

[0096] At block 655, the process 600 may determine whether to wait for acknowledgements (e.g., acks) for the non-systematic recovery data and/or file chunks that were transmitted to the one or more public storage clouds, as discussed above. For example, the process 600 may determine whether to wait acknowledgements (from the one more public storage clouds) indicating that file chunks were received/stored. In another example, the process 600 may determine whether to wait acknowledgements (from the one more public storage clouds) indicating that recovery data chunks were received/stored. If the process 600 should wait for acknowledgments, the process 600 may proceed to block 660 where the process 600 may wait for acknowledgments from the one or more public storage clouds (e.g., may wait until the acknowledgements are received). For example, the process 600 may periodically check whether the acknowledgments were received. In one embodiment, the process 600 may retransmit the non-systematic recovery data and/or file chunks if acknowledgments are not received after a period of time, at block 660. After block 660, the process 600 ends. In one embodiment, the process 600 may delete a cached or temporary copy of the file at block 660. For example, if the file is not stored on the private storage cloud based on the importance level of the file, the process 600 may generate the non-systematic data chunks and the file chunks using the file. After generating the non-systematic data chunks and the file chunks, the process 600 may delete the file. The process 600 may also delete the non-systematic

data chunks and/or the file chunks which were generated (to be transmitted to the public storage clouds) but are not stored on the private storage cloud.

[0097] Figure 7 is a flow diagram illustrating an example process 700 for reading data (e.g., downloading a file) in a storage system, in accordance with one or more embodiments. The storage system may include a private storage cloud (or multiple storage clouds) and may include one or more public storage clouds, as discussed above. The storage system may be referred to as a hybrid storage system. The process 700 may be performed by a storage module (e.g., storage module 126 illustrated in Figure 1), a storage application (e.g., storage application 111 illustrated in Figure 1), a processing device (e.g., a processor, a central processing unit (CPU)), and/or a computing device (e.g., a NAS device). The storage module, storage application, processing device, and/or computing device may be processing logic that includes hardware (e.g., circuitry, dedicated logic, programmable logic, microcode, etc.), software (e.g., instructions run on a processor to perform hardware simulation), firmware, or a combination thereof.

[0098] The process 700 begins at block 705 where the process 700 may determine whether the file is retrievable from the public storage clouds, as discussed above. For example, the process 700 may access metadata (illustrated in Figures 1 and 3) to determine whether file chunks and/or recovery data for the file is stored in the one or more public storage clouds. If the file is not retrievable from the one or more public storage clouds, the process 700 may proceed to block 710. If the file is retrievable from the one or more public storage clouds, the process 700 may determine whether access to the public storage clouds is faster than access to the private storage cloud at block 725, as discussed above. For example, the process 700 may ping the private storage cloud and the one or more public storage clouds. In another example, the process 700 may determine the throughputs of the private storage cloud and the one or more public storage clouds. If access to the one or more public storage clouds is not faster than access to the private storage cloud, the process 700 may proceed to block 720.

[0099] If access to one or more public store clouds is faster, the process may determine whether the latest version of the file (e.g., file chunks of the file, recovery data chunks of the file, etc.) are stored on the one or more

public storage clouds. For example, the process 700 may access metadata (illustrated in Figures 1 and 3) to determine whether the latest version of the file is stored on the one or more public storage clouds. If the latest version of the file is stored on the one or more public storage clouds, the process 700 may determine whether it is possible to access the private storage cloud and the one or more public storage clouds in parallel at block 735, as discussed above. If the private storage cloud and the one or more public storage clouds can be accessed in parallel, the process 700 may retrieve portions of the file from the file from the private storage cloud and the one or more public storage clouds at block 740, as discussed above. For example, the process 700 may download one or more file chunks from the private storage cloud and may download one or more file chunks from the public storage clouds. In another example, the process 700 may download one or more file recovery data chunks from the private storage cloud and may download one or more recovery data chunks from the public storage clouds. After block 740, the process 700 ends. If the private storage cloud and the one or more public storage clouds cannot be accessed in parallel, the process 700 may retrieve the portions of the file from the one or more storage clouds at block 745, as discussed above. For example, the process 700 may download file chunks (for the file) from the public storage clouds. In another example, the process 700 may download recovery data chunks from the public storage clouds. After block 745, the process 700 ends.

[0100] As discussed above, the process 700 may proceed to block 710 if the file is not retrievable from the one or more public storage clouds. The process 700 may determine whether the hotness level of the file is greater than a threshold hotness, as discussed above. For example, the process 700 may analyze the file to determine a hotness level for the file and may compare the hotness level of the file to the threshold hotness (e.g., a threshold hotness level). In another example, the process 700 may receive user input indicating the hotness level of the file and may compare the hotness level of the file to the threshold hotness. If the hotness level for the file is greater than the threshold hotness, the process 700 may store the file on the one or more public storage clouds at block 715. For example, the process 700 may generate file chunks and may transmit the file chunks to the

public storage clouds. In another example, the process 700 may generate recovery data chunks and may transmit the recovery data chunks to the public storage clouds. If the hotness level of the files is not greater than the threshold hotness, the process 700 may proceed to block 720. The process 700 may retrieve the file from the private storage cloud at block 720. For example, the process 700 may download the file from the private storage cloud. After block 720, the process 700 ends.

[0101] Figure 8 is a diagram of a computing device 800, in accordance with one or more embodiments. The computing device 800 may execute instructions that may cause the computing device 800 to perform any one or more of the methodologies (e.g., operations, methods, functions, etc.) discussed herein, may be executed. The computing device 800 may be a mobile phone, a smart phone, a netbook computer, a rackmount server, a router computer, a server computer, a personal computer, a mainframe computer, a laptop computer, a tablet computer, a desktop computer, a NAS device, a set-top box (STB), etc., within which a set of instructions, for causing the machine to perform any one or more of the methodologies discussed herein, may be executed. The machine may also be any machine/device capable of executing a set of instructions (sequential or otherwise) that specify actions to be taken by that machine/device. In alternative embodiments, the machine may be connected (e.g., networked) to other machines in a LAN, an intranet, an extranet, or the Internet. The machine may operate in the capacity of a server machine in client-server network environment. Further, while only a single machine is illustrated, the term "machine" shall also be taken to include any collection of machines that individually or jointly execute a set (or multiple sets) of instructions to perform any one or more of the functions, operations, methods, algorithms, etc., discussed herein.

[0102] The example computing device 800 includes a processing device (e.g., a processor, a controller, a central processing unit (CPU), etc.) 802, a main memory 804 (e.g., read-only memory (ROM), flash memory, dynamic random access memory (DRAM) such as synchronous DRAM (SDRAM)), a network-access interface 808, a direct-access interface 809, an

output device, 810, an input device 812, and a data storage device 818, which communicate with each other via a bus 830.

[0103] Processing device 802 represents one or more general-purpose processing devices such as a microprocessor, central processing unit, or the like. More particularly, the processing device 802 may be a complex instruction set computing (CISC) microprocessor, reduced instruction set computing (RISC) microprocessor, very long instruction word (VLIW) microprocessor, or a processor implementing other instruction sets or processors implementing a combination of instruction sets. The processing device 802 may also be one or more special-purpose processing devices such as an application specific integrated circuit (ASIC), a field programmable gate array (FPGA), a digital signal processor (DSP), network processor, or the like. The processing device 802 is configured to execute storage module/application instructions 835 (e.g., instructions for the storage module 126, storage module 136, and/or storage application 111 illustrated in Figure 1) for performing the operations and steps discussed herein.

[0104] The computing device 800 may include a network-access interface 808 (e.g., a network interface card, a Wi-Fi interface, etc.) which may communicate with a network (e.g., network 170 illustrated in Figure 1). The computing device may also include a direct-access interface 809 (e.g., a USB interface, an eSATA interface, a Thunderbolt interface, etc.). The computing device 800 also may include an output device 810 (e.g., a liquid crystal display (LCD) or a cathode ray tube (CRT)), and an input device 812 (e.g., a mouse, a keyboard, etc.). In one embodiment, the output device 810 and the input device 812 may be combined into a single component or device (e.g., an LCD touch screen).

[0105] The data storage device 818 may include a computer-readable storage medium 828 on which is stored one or more sets of instructions (e.g., storage module/application instructions 835) embodying any one or more of the methodologies or functions described herein. The storage module/application instructions 835 may also reside, completely or at least partially, within the main memory 804 and/or within the processing device 802 during execution thereof by the computing device 800. The main memory 804 and the processing device 802 may also constitute computer-readable media.

The instructions may further be transmitted or received over via the network-access interface 808 and/or direct-access interface 809.

[0106] While the computer-readable storage medium 828 is shown in an example embodiment to be a single medium, the term "computer-readable storage medium" should be taken to include a single medium or multiple media (e.g., a centralized or distributed database and/or associated caches and servers) that store the one or more sets of instructions. The term "computer-readable storage medium" shall also be taken to include any medium that is capable of storing, encoding or carrying a set of instructions for execution by the machine and that cause the machine to perform any one or more of the methodologies of the present disclosure. The term "computer-readable storage medium" shall accordingly be taken to include, but not be limited to, solid-state memories, optical media and magnetic media.

General Comments

[0107] Those skilled in the art will appreciate that in some embodiments, other types of distributed data storage systems can be implemented while remaining within the scope of the present disclosure. In addition, the actual steps taken in the processes discussed herein may differ from those described or shown in the figures. Depending on the embodiment, certain of the steps described above may be removed, others may be added.

[0108] While certain embodiments have been described, these embodiments have been presented by way of example only, and are not intended to limit the scope of protection. Indeed, the novel methods and systems described herein may be embodied in a variety of other forms. Furthermore, various omissions, substitutions and changes in the form of the methods and systems described herein may be made. The accompanying claims and their equivalents are intended to cover such forms or modifications as would fall within the scope and spirit of the protection. For example, the various components illustrated in the figures may be implemented as software and/or firmware on a processor, ASIC/FPGA, or dedicated hardware. Also, the features and attributes of the specific embodiments disclosed above may be combined in different ways to form additional embodiments, all of which fall within the scope of the present disclosure. Although the present disclosure

provides certain preferred embodiments and applications, other embodiments that are apparent to those of ordinary skill in the art, including embodiments which do not provide all of the features and advantages set forth herein, are also within the scope of this disclosure. Accordingly, the scope of the present disclosure is intended to be defined only by reference to the appended claims.

[0109] The words "example" or "exemplary" are used herein to mean serving as an example, instance, or illustration. Any aspect or design described herein as "example' or "exemplary" is not necessarily to be construed as preferred or advantageous over other aspects or designs. Rather, use of the words "example" or "exemplary" is intended to present concepts in a concrete fashion. As used in this application, the term "or" is intended to mean an inclusive "or" rather than an exclusive "or". That is, unless specified otherwise, or clear from context, "X includes A or B" is intended to mean any of the natural inclusive permutations. That is, if X includes A; X includes B; or X includes both A and B, then "X includes A or B" is satisfied under any of the foregoing instances. In addition, the articles "a" and "an" as used in this application and the appended claims should generally be construed to mean "one or more" unless specified otherwise or clear from context to be directed to a singular form. Moreover, use of the term "an embodiment" or "one embodiment" or "an implementation" or "one implementation" throughout is not intended to mean the same embodiment or implementation unless described as such. Furthermore, the terms "first," "second," "third," "fourth," etc., as used herein are meant as labels to distinguish among different elements and may not necessarily have an ordinal meaning according to their numerical designation.

[0110] All of the processes described above may be embodied in, and fully automated via, software code modules executed by one or more general purpose or special purpose computers or processors. The code modules may be stored on any type of computer-readable medium (e.g., a non-transitory computer-readable medium) or other computer storage device or collection of storage devices. Some or all of the methods may alternatively be embodied in specialized computer hardware.

WHAT IS CLAIMED IS:

1. An apparatus, comprising:

a memory configured to store data; and

a processor coupled to the memory, the processor configured to:

receive a request for a file stored in a storage system comprising a private storage cloud and a set of public storage clouds;

determine whether the file is retrievable from the set of public storage clouds; and

responsive to determining that the file is retrievable from the set of public storage clouds:

determine whether access to the set of public storage clouds is faster than access to the private storage cloud; and

retrieve the file from one or more of the private storage cloud and the set of public storage clouds.

2. The apparatus of claim 1, wherein the processor is further configured to:

retrieve the file from the private storage cloud responsive to determining that the file is not retrievable from the set of public storage clouds.

3. The apparatus of claim 2, wherein the processor is further configured to:

determine a hotness level for the file, responsive to determining that the file is not retrievable from the set of public storage clouds; and

store one or more of portions of the file or a set of recovery data chunks for the file in the set of public storage clouds, responsive to determining that the hotness level exceeds a hotness threshold.

4. The apparatus of claim 1, wherein the processor is configured to retrieve the file by:

retrieving the file from the private storage cloud responsive to determining that access to the set of public storage clouds is not faster than access to the private storage cloud.

5. The apparatus of claim 1, wherein the processor is further configured to:

determine whether the private storage cloud and the set of public storage clouds can be accessed in parallel.

6. The apparatus of claim 5, wherein retrieving the file comprises:

retrieving a first set of portions of the file from the private storage cloud and a second set of portions of the file from the set of public storage clouds, responsive to determining that the private storage cloud and the set of public storage clouds can be accessed in parallel.

7. The apparatus of claim 5, wherein retrieving the file comprises:

retrieving the file from the set of public storage clouds, responsive to determining that the private storage cloud and the set of public storage clouds cannot be accessed in parallel.

8. The apparatus of claim 1, wherein the processor is further configured to:

determine whether a latest version of the file is retrievable from the set of public storage clouds; and

retrieve the file from the private storage cloud, responsive to determining that the latest version of the file is not retrievable from the set of public storage clouds.

9. The apparatus of claim 8, wherein the processor is further configured to:

request that the latest version of the file be stored in the set of public storage clouds, responsive to determining that the latest version of the file is not retrievable from the set of public storage clouds.

10. The apparatus of claim 1, wherein the processor is configured to retrieve the file by:

retrieving portions of the file from one or more of the private storage cloud or the set of public storage clouds.

11. The apparatus of claim 1, wherein the processor is configured to retrieve the file by:

retrieving a set of recovery data chunks for the file from one or more of the private storage cloud or the set of public storage clouds; and

generating the file based on the set of recovery data chunks.

12. A method, comprising:

receiving a request for a file stored in a storage system comprising a private storage cloud and a set of public storage clouds;

determining whether the file is retrievable from the set of public storage clouds; and

responsive to determining that the file is retrievable from the set of public storage clouds:

determining whether access to the set of public storage clouds is faster than access to the private storage cloud; and

retrieving the file from one or more of the private storage cloud and the set of public storage clouds.

13. The method of claim 12, further comprising:

retrieving the file from the private storage cloud responsive to determining that the file is not retrievable from the set of public storage clouds.

14. The method of claim 13, further comprising:

determining a hotness level for the file, responsive to determining that the file is not retrievable from the set of public storage clouds; and

storing one or more of portions of the file or a set of recovery data chunks for the file in the set of public storage clouds, responsive to determining that the hotness level exceeds a hotness threshold.

15. The method of claim 12, wherein retrieving the file comprises:

retrieving the file from the private storage cloud responsive to determining that access to the set of public storage clouds is not faster than access to the private storage cloud.

16. The method of claim 12, further comprising:

determining whether the private storage cloud and the set of public storage clouds can be accessed in parallel.

17. The method of claim 16, wherein retrieving the file comprises:

retrieving a first set of portions of the file from the private storage cloud and a second set of portions of the file from the set of public storage clouds, responsive to determining that the private storage cloud and the set of public storage clouds can be accessed in parallel.

18. The method of claim 16, wherein retrieving the file comprises:

retrieving the file from the set of public storage clouds, responsive to determining that the private storage cloud and the set of public storage clouds cannot be accessed in parallel.

19. The method of claim 12, further comprising:

determining whether a latest version of the file is retrievable from the set of public storage clouds; and

retrieving the file from the private storage cloud, responsive to determining that the latest version of the file is not retrievable from the set of public storage clouds.

20. A non-transitory computer-readable medium having stored thereon instructions that, when executed by a processor, cause the processor to perform operations comprising:

receiving a request for a file stored in a storage system comprising a private storage cloud and a set of public storage clouds;

determining whether the file is retrievable from the set of public storage clouds; and

responsive to determining that the file is retrievable from the set of public storage clouds:

determining whether access to the set of public storage clouds is faster than access to the private storage cloud; and

retrieving the file from one or more of the private storage cloud and the set of public storage clouds.

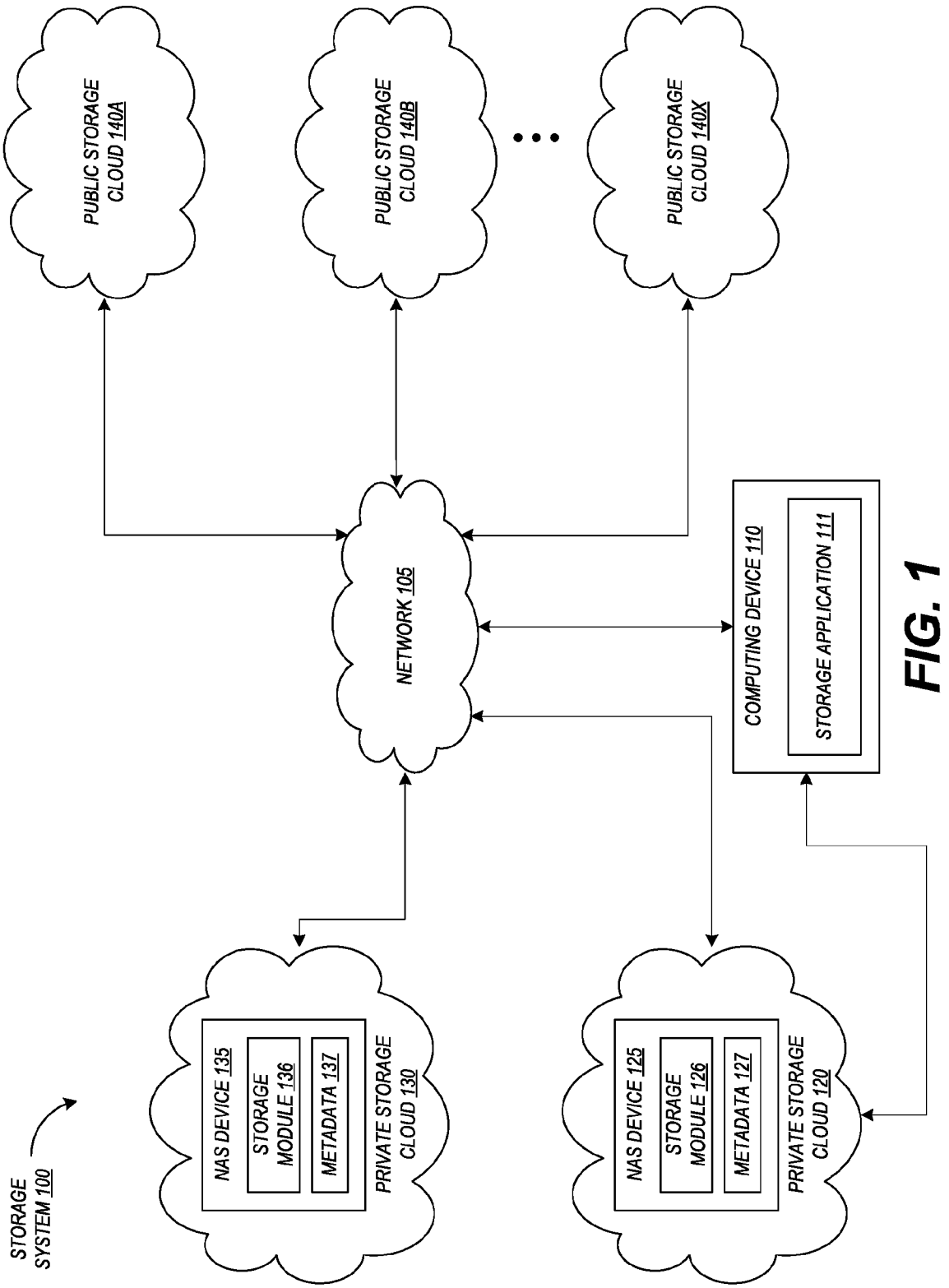

PUBLIC STORAGE CLOUD 140A
PUBLIC STORAGE CLOUD 140B
PUBLIC STORAGE CLOUD 140X
NETWORK 105
COMPUTING DEVICE 110
STORAGE APPLICATION 111
NAS DEVICE 135
STORAGE MODULE 136
METADATA 137
PRIVATE STORAGE CLOUD 130
NAS DEVICE 125
STORAGE MODULE 126
METADATA 127
PRIVATE STORAGE CLOUD 120
STORAGE SYSTEM 100

FIG. 1

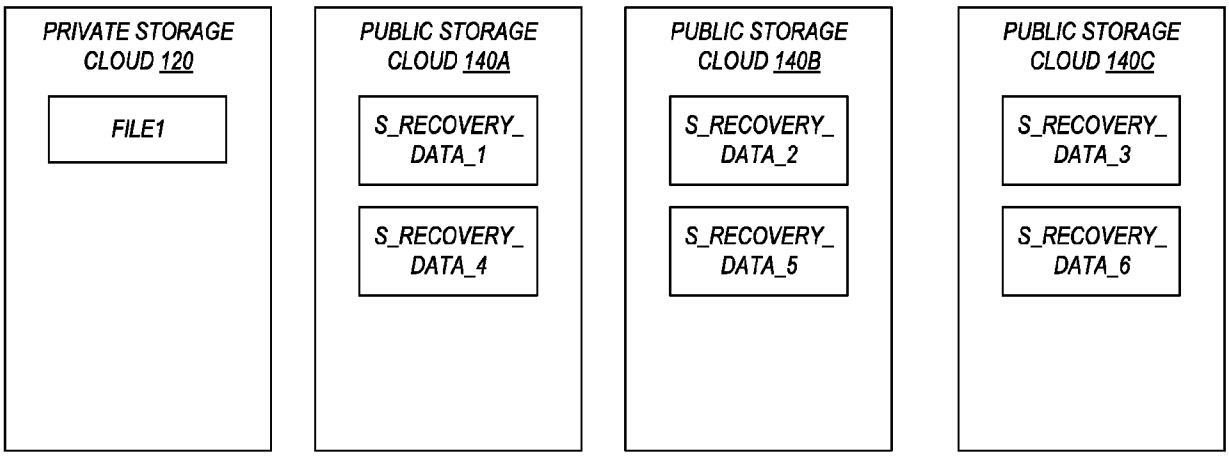

PRIVATE STORAGE CLOUD 120
FILE1
PUBLIC STORAGE CLOUD 140A
S_RECOVERY_ DATA_1
S_RECOVERY_ DATA_4
PUBLIC STORAGE CLOUD 140B
S_RECOVERY_ DATA_2
S_RECOVERY_ DATA_5
PUBLIC STORAGE CLOUD 140C
S_RECOVERY_ DATA_3
S_RECOVERY_ DATA_6

FIG. 2A

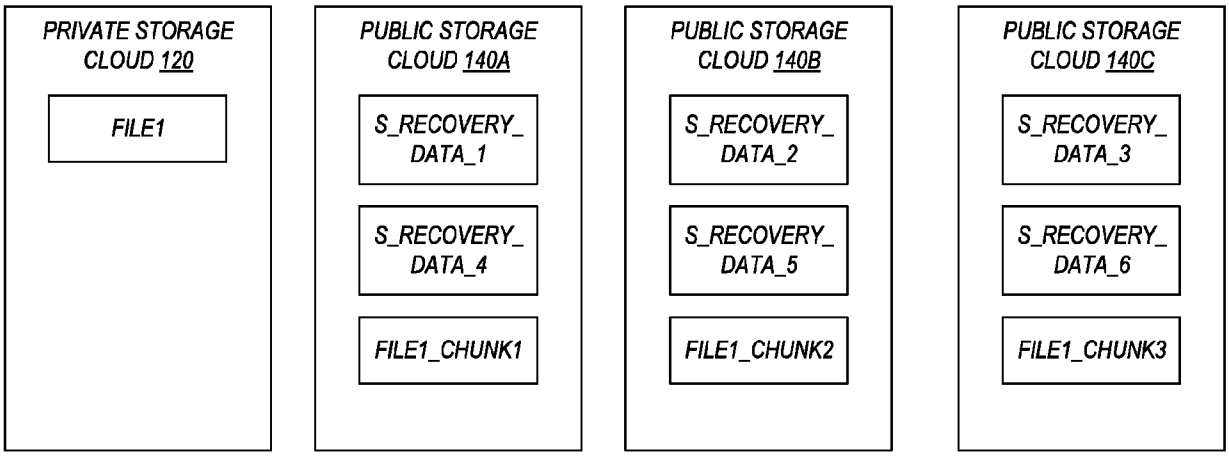

PRIVATE STORAGE CLOUD 120
FILE1
PUBLIC STORAGE CLOUD 140A
S_RECOVERY_ DATA_1
S_RECOVERY_ DATA_4
FILE1_CHUNK1
PUBLIC STORAGE CLOUD 140B
S_RECOVERY_ DATA_2
S_RECOVERY_ DATA_5
FILE1_CHUNK2
PUBLIC STORAGE CLOUD 140C
S_RECOVERY_ DATA_3
S_RECOVERY_ DATA_6
FILE1_CHUNK3

FIG. 2B

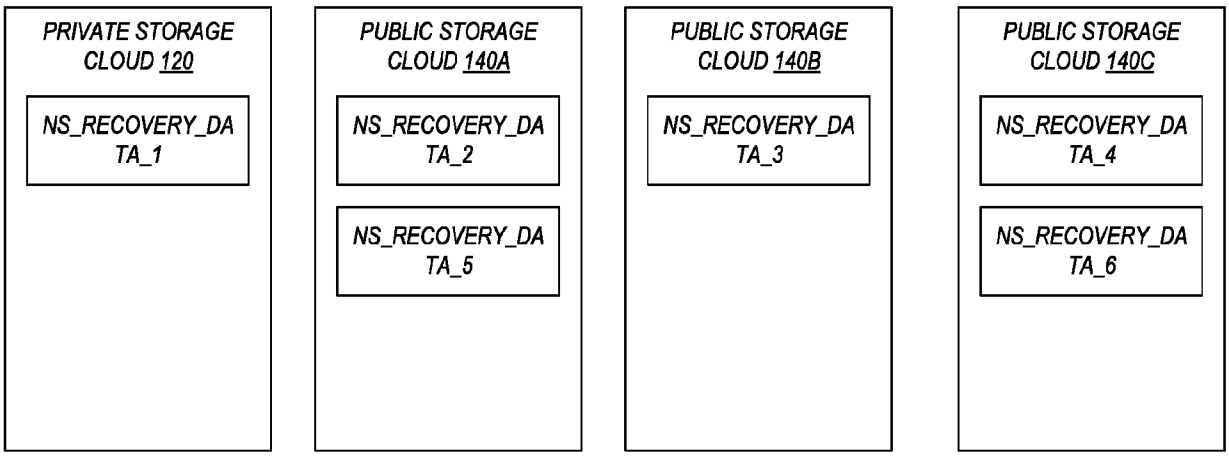

PRIVATE STORAGE CLOUD 120
NS_RECOVERY_DATA_1
PUBLIC STORAGE CLOUD 140A
NS_RECOVERY_DATA_2
NS_RECOVERY_DATA_5
PUBLIC STORAGE CLOUD 140B
NS_RECOVERY_DATA_3
PUBLIC STORAGE CLOUD 140C
NS_RECOVERY_DATA_4
NS_RECOVERY_DATA_6

FIG. 2C

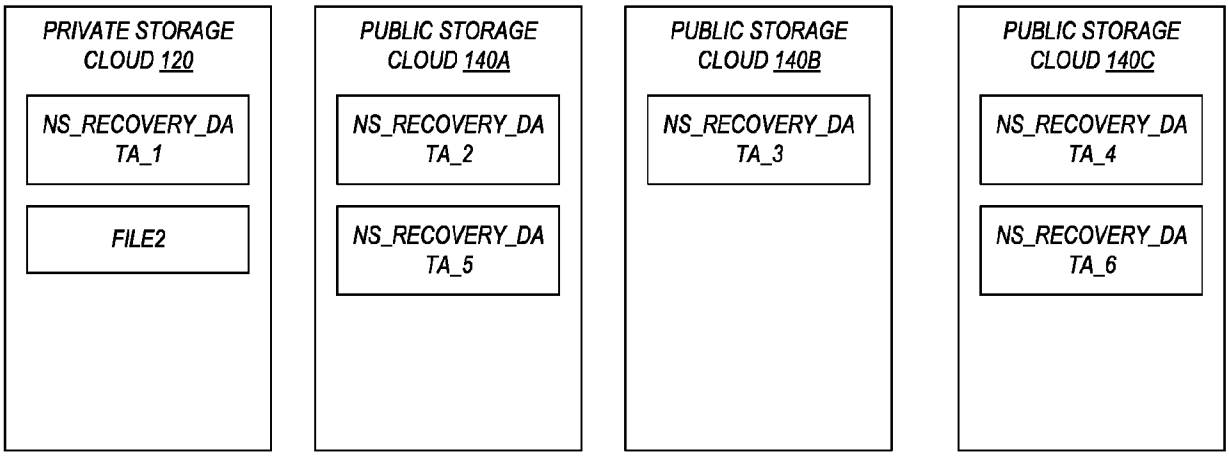

PRIVATE STORAGE CLOUD 120
NS_RECOVERY_DATA_1
FILE2
PUBLIC STORAGE CLOUD 140A
NS_RECOVERY_DATA_2
NS_RECOVERY_DATA_5
PUBLIC STORAGE CLOUD 140B
NS_RECOVERY_DATA_3
PUBLIC STORAGE CLOUD 140C
NS_RECOVERY_DATA_4
NS_RECOVERY_DATA_6

FIG. 2D

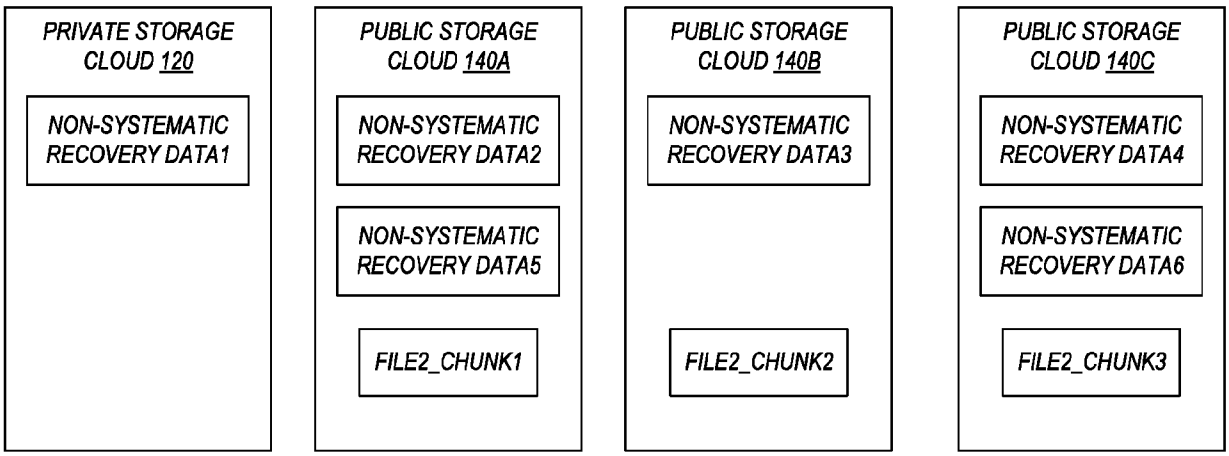

PRIVATE STORAGE CLOUD 120
NON-SYSTEMATIC RECOVERY DATA1
PUBLIC STORAGE CLOUD 140A
NON-SYSTEMATIC RECOVERY DATA2
NON-SYSTEMATIC RECOVERY DATA5
FILE2_CHUNK1
PUBLIC STORAGE CLOUD 140B
NON-SYSTEMATIC RECOVERY DATA3
FILE2_CHUNK2
PUBLIC STORAGE CLOUD 140C
NON-SYSTEMATIC RECOVERY DATA4
NON-SYSTEMATIC RECOVERY DATA6
FILE2_CHUNK3

FIG. 2E

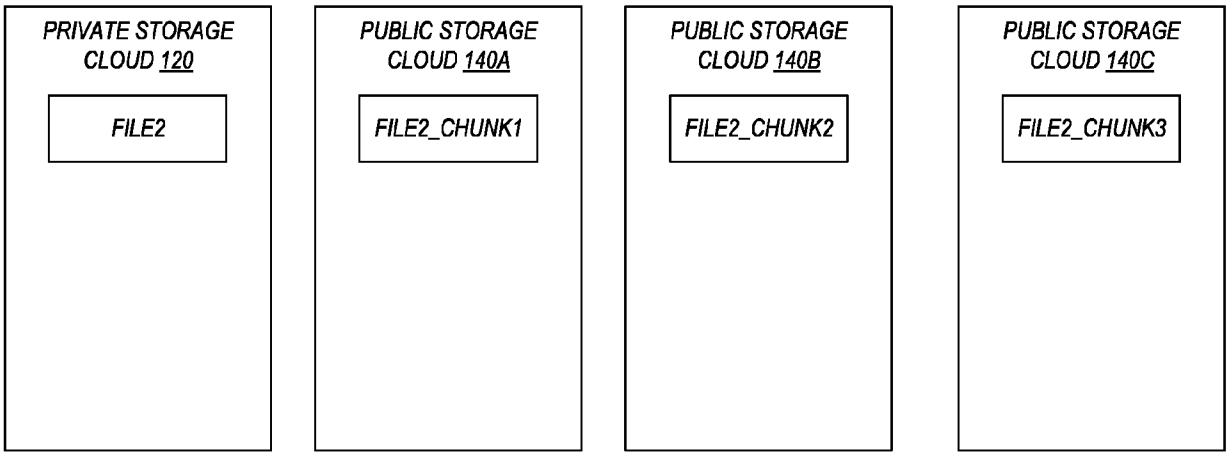
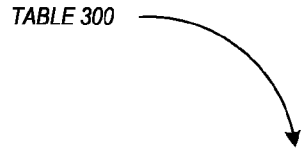

PRIVATE STORAGE CLOUD 120
FILE2
PUBLIC STORAGE CLOUD 140A
FILE2_CHUNK1
PUBLIC STORAGE CLOUD 140B
FILE2_CHUNK2
PUBLIC STORAGE CLOUD 140C
FILE2_CHUNK3

FIG. 2F



File Name	Chunk ID	Chunk Location	Recovery Data ID	Recovery Data Location
FILE A	CHUNK A1	PUBLIC STORAGE CLOUD1	S RECOVERY DATA A1	PUBLIC STORAGE CLOUD1
	CHUNK A2	PUBLIC STORAGE CLOUD2	S RECOVERY DATA A2	PUBLIC STORAGE CLOUD2
	CHUNK A3	PUBLIC STORAGE CLOUD3	S RECOVERY DATA A3	PUBLIC STORAGE CLOUD3
			S RECOVERY DATA A4	PUBLIC STORAGE CLOUD1
			S RECOVERY DATA A5	PUBLIC STORAGE CLOUD2
			S RECOVERY DATA A6	PUBLIC STORAGE CLOUD3
FILE B			NS RECOVERY DATA B1	PRIVATE STORAGE CLOUD1
			NS RECOVERY DATA B2	PUBLIC STORAGE CLOUD1
			NS RECOVERY DATA B3	PUBLIC STORAGE CLOUD2
			NS RECOVERY DATA B4	PUBLIC STORAGE CLOUD3
			NS RECOVERY DATA B5	PUBLIC STORAGE CLOUD1

FIG. 3

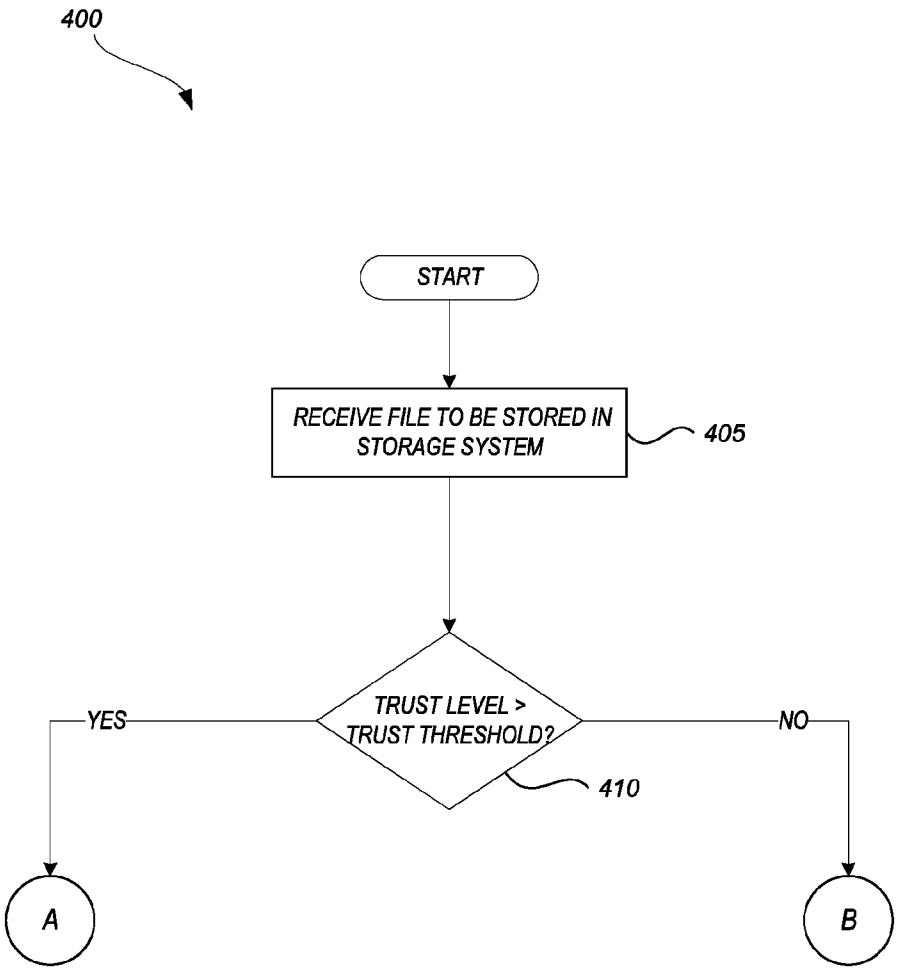

400
START
RECEIVE FILE TO BE STORED IN STORAGE SYSTEM
405
TRUST LEVEL > TRUST THRESHOLD?
410
YES
NO
A
B

FIG. 4

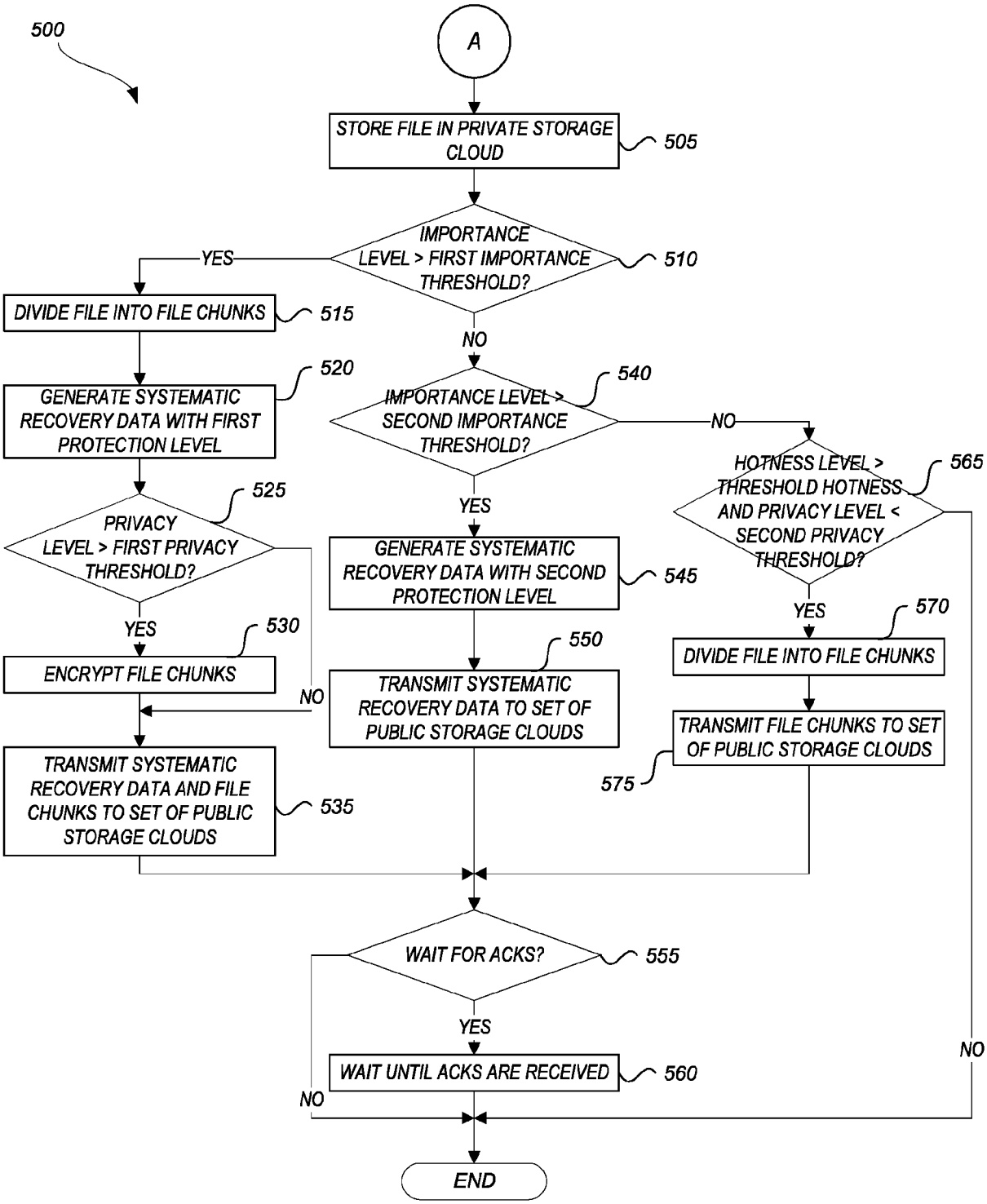

500
A
STORE FILE IN PRIVATE STORAGE CLOUD
505
IMPORTANCE LEVEL > FIRST IMPORTANCE THRESHOLD?
510
YES
NO
DIVIDE FILE INTO FILE CHUNKS
515
GENERATE SYSTEMATIC RECOVERY DATA WITH FIRST PROTECTION LEVEL
520
PRIVACY LEVEL > FIRST PRIVACY THRESHOLD?
525
YES
NO
ENCRYPT FILE CHUNKS
530
TRANSMIT SYSTEMATIC RECOVERY DATA AND FILE CHUNKS TO SET OF PUBLIC STORAGE CLOUDS
535
IMPORTANCE LEVEL > SECOND IMPORTANCE THRESHOLD?
540
YES
NO
GENERATE SYSTEMATIC RECOVERY DATA WITH SECOND PROTECTION LEVEL
545
TRANSMIT SYSTEMATIC RECOVERY DATA TO SET OF PUBLIC STORAGE CLOUDS
550
HOTNESS LEVEL > THRESHOLD HOTNESS AND PRIVACY LEVEL < SECOND PRIVACY THRESHOLD?
565
YES
NO
DIVIDE FILE INTO FILE CHUNKS
570
TRANSMIT FILE CHUNKS TO SET OF PUBLIC STORAGE CLOUDS
575
WAIT FOR ACKS?
555
YES
NO
WAIT UNTIL ACKS ARE RECEIVED
560
END

FIG. 5

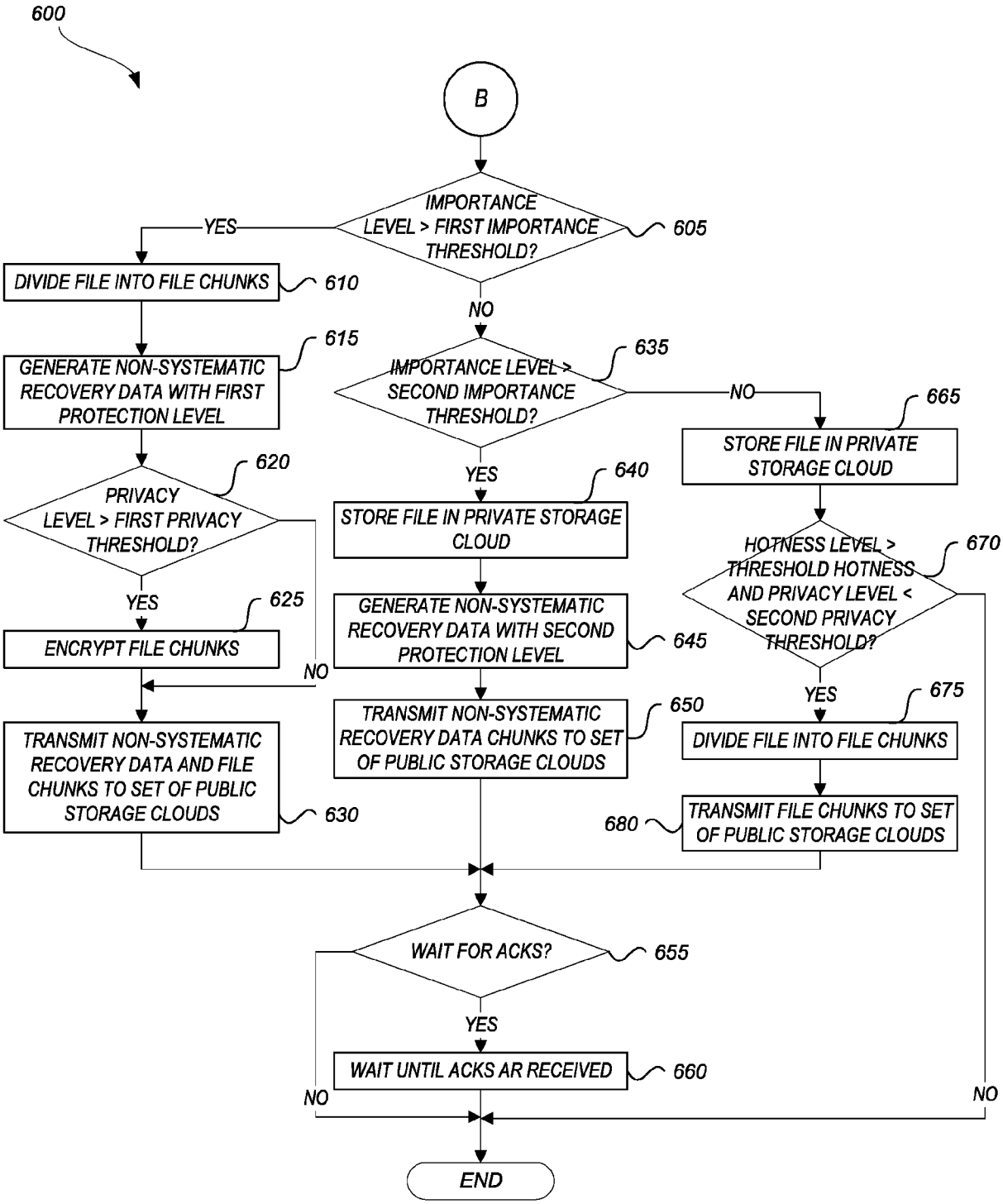

600
B
IMPORTANCE LEVEL > FIRST IMPORTANCE THRESHOLD?
605
YES
DIVIDE FILE INTO FILE CHUNKS
610
NO
GENERATE NON-SYSTEMATIC RECOVERY DATA WITH FIRST PROTECTION LEVEL
615
IMPORTANCE LEVEL > SECOND IMPORTANCE THRESHOLD?
635
NO
665
STORE FILE IN PRIVATE STORAGE CLOUD
PRIVACY LEVEL > FIRST PRIVACY THRESHOLD?
620
YES
640
STORE FILE IN PRIVATE STORAGE CLOUD
HOTNESS LEVEL > THRESHOLD HOTNESS AND PRIVACY LEVEL < SECOND PRIVACY THRESHOLD?
670
YES
625
ENCRYPT FILE CHUNKS
NO
GENERATE NON-SYSTEMATIC RECOVERY DATA WITH SECOND PROTECTION LEVEL
645
NO
YES
675
TRANSMIT NON-SYSTEMATIC RECOVERY DATA AND FILE CHUNKS TO SET OF PUBLIC STORAGE CLOUDS
630
TRANSMIT NON-SYSTEMATIC RECOVERY DATA CHUNKS TO SET OF PUBLIC STORAGE CLOUDS
650
DIVIDE FILE INTO FILE CHUNKS
680
TRANSMIT FILE CHUNKS TO SET OF PUBLIC STORAGE CLOUDS
WAIT FOR ACKS?
655
YES
WAIT UNTIL ACKS AR RECEIVED
660
NO
NO
END

FIG. 6

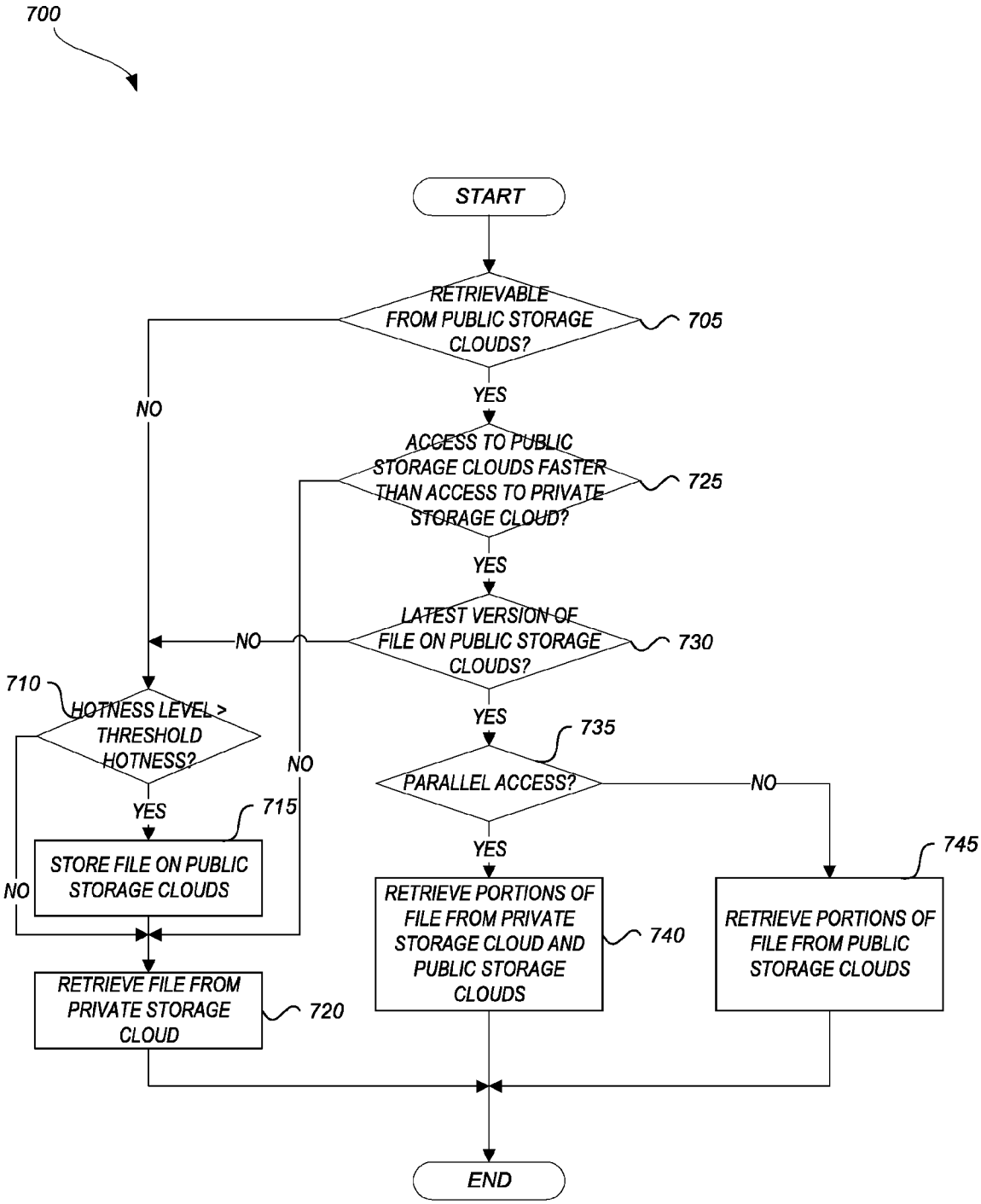

700
START
RETRIEVABLE FROM PUBLIC STORAGE CLOUDS?
705
YES
NO
ACCESS TO PUBLIC STORAGE CLOUDS FASTER THAN ACCESS TO PRIVATE STORAGE CLOUD?
725
YES
NO
LATEST VERSION OF FILE ON PUBLIC STORAGE CLOUDS?
730
YES
710
HOTNESS LEVEL > THRESHOLD HOTNESS?
NO
735
PARALLEL ACCESS?
NO
YES
715
YES
745
STORE FILE ON PUBLIC STORAGE CLOUDS
NO
RETRIEVE PORTIONS OF FILE FROM PRIVATE STORAGE CLOUD AND PUBLIC STORAGE CLOUDS
740
RETRIEVE PORTIONS OF FILE FROM PUBLIC STORAGE CLOUDS
RETRIEVE FILE FROM PRIVATE STORAGE CLOUD
720
END

FIG. 7

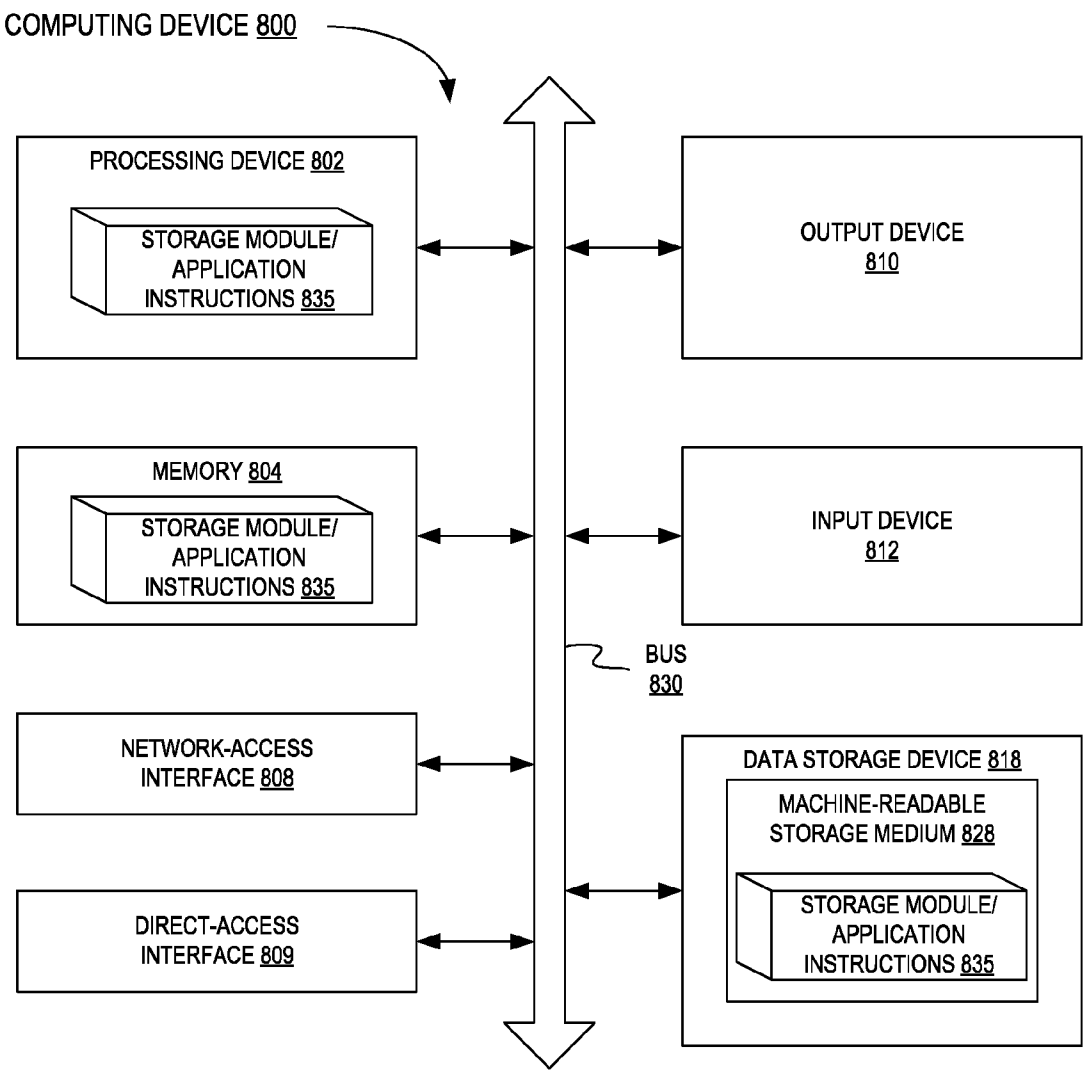

COMPUTING DEVICE 800
PROCESSING DEVICE 802
STORAGE MODULE/ APPLICATION INSTRUCTIONS 835
MEMORY 804
STORAGE MODULE/ APPLICATION INSTRUCTIONS 835
NETWORK-ACCESS INTERFACE 808
DIRECT-ACCESS INTERFACE 809
OUTPUT DEVICE 810
INPUT DEVICE 812
BUS 830
DATA STORAGE DEVICE 818
MACHINE-READABLE STORAGE MEDIUM 828
STORAGE MODULE/ APPLICATION INSTRUCTIONS 835

FIG. 8

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2018/023988

A. CLASSIFICATION OF SUBJECT MATTER
INV. H04L29/08 G06F3/06 G06F17/30
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
H04L G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
EPO-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 2014/108182 A1 (QATAR FOUNDATION [QA]; HOARTON LLOYD [GB]) 17 July 2014 (2014-07-17)	1,2, 4-13, 15-20
Y	abstract page 1, line 29 - page 2, line 10 page 4, line 2 - page 4, line 6 page 4, line 15 - page 4, line 21 page 6, line 6 - page 6, line 17 page 8, line 1 - page 8, line 3 page 8, line 14 - page 8, line 18 page 7, line 10 - page 7, line 20 page 16, line 29 - page 17, line 10 page 17, line 25 - page 18, line 12 page 19, line 16 - page 20, line 14 page 20, line 15 - page 20, line 31 page 22, line 5 - page 23, line 2 page 30, line 9 - page 31, line 15	3,14

-/--

[X] Further documents are listed in the continuation of Box C.

[X] See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
6 June 2018	14/06/2018

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Pereira, Mafalda

1

Form PCT/ISA/210 (second sheet) (April 2005)

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2018/023988

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 2014/289492 A1 (RANJITH REDDY BASIREDDY [IN] ET AL) 25 September 2014 (2014-09-25) the whole document	3,14
A	WO 2013/142008 A1 (ALCATEL LUCENT [FR]) 26 September 2013 (2013-09-26) the whole document	1-20

1

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/US2018/023988

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 2014108182 A1	17-07-2014	GB 2522372 A	22-07-2015
		US 2015312243 A1	29-10-2015
		WO 2014108182 A1	17-07-2014
US 2014289492 A1	25-09-2014	KR 20140114618 A	29-09-2014
		US 2014289492 A1	25-09-2014
WO 2013142008 A1	26-09-2013	CN 104254838 A	31-12-2014
		EP 2828749 A1	28-01-2015
		JP 2015512534 A	27-04-2015
		KR 20140129245 A	06-11-2014
		US 2013254248 A1	26-09-2013
		WO 2013142008 A1	26-09-2013