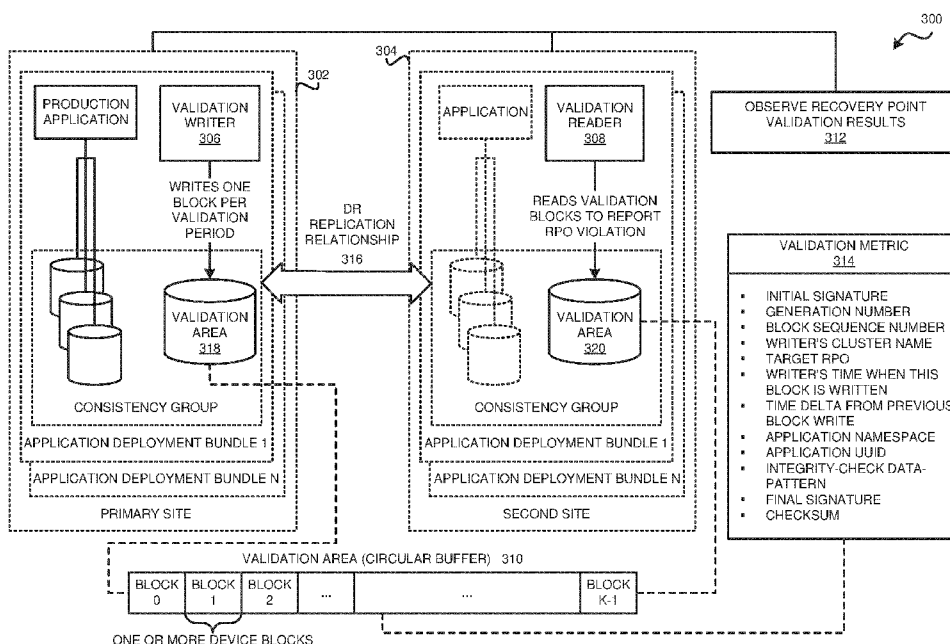




- (51) **International Patent Classification:**
H04L 67/1095 (2022.01) *H04L 69/40* (2022.01)
- (21) **International Application Number:**
PCT/EP2024/083260
- (22) **International Filing Date:**
22 November 2024 (22.11.2024)
- (25) **Filing Language:** English
- (26) **Publication Language:** English
- (30) **Priority Data:**
18/533,541 08 December 2023 (08.12.2023) US
- (71) **Applicant:** **INTERNATIONAL BUSINESS MACHINES CORPORATION** [US/US]; New Orchard Road, Armonk, New York 10504 (US).
- (71) **Applicant (for MG only):** **IBM DEUTSCHLAND GMBH** [DE/DE]; Schönaicher Str. 220, 71032 Böblingen (DE).
- (72) **Inventors:** **DEENADHAYALAN, Veeraraghavan**; c/o IBM Almaden Research Center, 650 Harry Road, San Jose, California 95120-6099 (US). **HATFIELD, Brian**; c/o IBM CORPORATION, 9000 S Rita Road, Tucson, Arizona 85744-0002 (US). **MUENCH, Paul**; c/o IBM Almaden Research Center, 650 Harry Road, San Jose, California 95120-6099 (US).
- (74) **Agent:** **VETTER, Svenja**; c/o IBM Deutschland GmbH, Patentwesen und Urheberrecht, Schönaicher Str. 220, 71032 Böblingen (DE).
- (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, CV, CZ, DE, DJ, DK, DM,

(54) **Title:** A RECOVERY POINT OBJECTIVE VALIDATOR

Fig. 3



(57) **Abstract:** An embodiment provisions a Validation Area within a consistency group of a primary site and a Validation Area within a consistency group of a secondary site. The embodiment provisions a Validation Writer in the primary site and provisions a Validation Reader in the secondary site; the Validation Writer writes a validation metric into the Validation Area in the primary site at an interval determined by a recovery point objective. The embodiment also reads the validation metric by the Validation Reader in the secondary site, computes a validation result based on the validation metric, the recovery point objective and a recovery point time delta where the validation result comprises an outcome of a replication of the primary site to the secondary site where the Validation Area of the primary site, the Validation Area of the secondary site, the Validation Writer and the Validation Reader are storage and application agnostic.

DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IQ, IR, IS, IT, JM, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, MG, MK, MN, MU, 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, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

(84) Designated States (*unless otherwise indicated, for every kind of regional protection available*): ARIPO (BW, CV, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SC, 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, ME, 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).

Published:

- *with international search report (Art. 21(3))*
- *in black and white; the international application as filed contained color or greyscale and is available for download from PATENTSCOPE*

A RECOVERY POINT OBJECTIVE VALIDATOR

BACKGROUND

[0001] The present invention relates generally to disaster recovery. More particularly, the present invention relates to a method, system, and computer program for A Recovery Point Objective Validator.

[0002] Disaster recovery (DR) consists of IT technologies and best practices designed to prevent or minimize data loss and business disruption resulting from catastrophic events—everything from equipment failures to localized power outages, civil emergencies, criminal or military attacks, and natural disasters. Infrastructure failure can cost as much as USD 100,000 per hour and critical application failure costs can range from USD 500,000 to USD 1 million per hour. Many businesses cannot recover from such losses.

SUMMARY

[0003] The illustrative embodiments provide for A Recovery Point Objective Validator. An embodiment includes provisioning a Validation Area within a consistency group of a primary site and a Validation Area within a consistency group of a secondary site. The embodiment includes provisioning a Validation Writer with access to the Validation Area in the primary site and provisioning a Validation Reader with access to the Validation Area in the secondary site.

The embodiment includes writing a validation metric by the Validation Writer into the Validation Area in the primary site at an interval determined by a recovery point objective. The embodiment also includes reading the validation metric by the Validation Reader from the Validation Area in the secondary site, computing a validation result by the Validation Reader based in part on the validation metric, the recovery point objective and a recovery point time delta wherein the validation result comprises an outcome of a replication of the consistency group in the primary site to the consistency group in the secondary site wherein the Validation Area within the consistency group of the primary site, the Validation Area within the consistency group of the secondary site, the Validation Writer and the Validation Reader are storage and application agnostic. Other embodiments of this aspect include corresponding computer systems, apparatus, and computer programs recorded on one or more computer storage devices, each configured to perform the actions of the embodiment.

[0004] An embodiment includes a computer usable program product. The computer usable program product includes a computer-readable storage medium, and program instructions stored on the storage medium.

[0005] An embodiment includes a computer system. The computer system includes a processor, a computer-readable memory, and a computer-readable storage medium, and program instructions stored on the storage medium for execution by the processor via the memory.

BRIEF DESCRIPTION OF THE DRAWINGS

[0006] The novel features believed characteristic of the invention are set forth in the appended claims. The invention itself, however, as well as a preferred mode of use, further objectives, and advantages thereof, will best be understood by reference to the following detailed description of the illustrative embodiments when read in conjunction with the accompanying drawings, wherein:

[0007] Fig. 1 depicts a block diagram of a computing environment in accordance with an illustrative embodiment;

[0008] Fig. 2 depicts a diagram in accordance with an illustrative embodiment;

[0009] Fig. 3 depicts a system diagram in accordance with an illustrative embodiment;

[0010] Fig. 4 depicts a flowchart diagram in accordance with an illustrative embodiment;

[0011] Fig. 5 depicts a diagram in accordance with an illustrative embodiment;

[0012] Fig. 6 depicts a diagram in accordance with an illustrative embodiment;

[0013] Fig. 7 depicts a diagram in accordance with an illustrative embodiment; and

[0014] Fig. 8 depicts a flowchart diagram of an example On Demand process in accordance with an illustrative embodiment.

DETAILED DESCRIPTION

[0015] Disaster recovery (DR) consists of IT technologies and best practices designed to prevent or minimize data loss and business disruption resulting from catastrophic events—everything from equipment failures to localized power outages, civil emergencies, criminal or military attacks, and natural disasters. Infrastructure failure can cost as much as USD 100,000

per hour and critical application failure costs can range from USD 500,000 to USD 1 million per hour. Many businesses cannot recover from such losses.

[0016] The present disclosure addresses the deficiencies described above by providing a method, a machine-readable medium, and a system for A Recovery Point Objective Validator. Embodiments include provisioning a Validation Area within a consistency group of a primary site and a Validation Area within a consistency group of a secondary site. The embodiment includes provisioning a Validation Writer with access to the Validation Area in the primary site and provisioning a Validation Reader with access to the Validation Area in the secondary site. The embodiment includes writing a validation metric by the Validation Writer into the Validation Area in the primary site at an interval determined by a recovery point objective. The embodiment also includes reading the validation metric by the Validation Reader from the Validation Area in the secondary site, computing a validation result by the Validation Reader based in part on the validation metric, the recovery point objective and a recovery point time delta wherein the validation result comprises an outcome of a replication of the consistency group in the primary site to the consistency group in the secondary site wherein the Validation Area within the consistency group of the primary site, the Validation Area within the consistency group of the secondary site, the Validation Writer and the Validation Reader are storage and application agnostic. Other embodiments of this aspect include corresponding computer systems, apparatus, and computer programs recorded on one or more computer storage devices, each configured to perform the actions of the embodiment.

[0017] Illustrative embodiments include wherein computing the validation result comprises comparing a current generation number of the validation metric with a previous generation number of the validation metric.

[0018] Illustrative embodiments include wherein a replication event is computed based on the validation result.

[0019] Illustrative embodiments include wherein the Validation Writer and the Validation Reader are automatically aligned based on a recovery point objective metric.

[0020] Illustrative embodiments include wherein aligning the Validation Writer and the Validation Reader comprises performing a slow search comprising an optimization of a system impact relative to an alignment precision.

[0021] Illustrative embodiments include wherein aligning the Validation Writer and the Validation Reader comprises performing a fast search comprising an optimization of an alignment precision relative to a system impact.

[0022] Illustrative embodiments also include wherein an application on the primary site senses the validation result computed by the Validation Reader.

[0023] For the sake of clarity of the description, and without implying any limitation thereto, the illustrative embodiments are described using some example configurations. From this disclosure, those of ordinary skill in the art will be able to conceive many alterations, adaptations, and modifications of a described configuration for achieving a described purpose, and the same are contemplated within the scope of the illustrative embodiments.

[0024] Furthermore, simplified diagrams of the data processing environments are used in the figures and the illustrative embodiments. In an actual computing environment, additional structures or components that are not shown or described herein, or structures or components different from those shown but for a similar function as described herein may be present without departing the scope of the illustrative embodiments.

[0025] Furthermore, the illustrative embodiments are described with respect to specific actual or hypothetical components only as examples. Any specific manifestations of these and other similar artifacts are not intended to be limiting to the invention. Any suitable manifestation of these and other similar artifacts can be selected within the scope of the illustrative embodiments.

[0026] The examples in this disclosure are used only for the clarity of the description and are not limiting to the illustrative embodiments. Any advantages listed herein are only examples and are not intended to be limiting to the illustrative embodiments. Additional or different advantages may be realized by specific illustrative embodiments. Furthermore, a particular illustrative embodiment may have some, all, or none of the advantages listed above.

[0027] Furthermore, the illustrative embodiments may be implemented with respect to any type of data, data source, or access to a data source over a data network. Any type of data storage device may provide the data to an embodiment of the invention, either locally at a data processing system or over a data network, within the scope of the invention. Where an embodiment is described using a mobile device, any type of data storage device suitable for use with the mobile device may provide the data to such embodiment, either locally at the mobile device or over a data network, within the scope of the illustrative embodiments.

[0028] The illustrative embodiments are described using specific code, computer readable storage media, high-level features, designs, architectures, protocols, layouts, schematics, and tools only as examples and are not limiting to the illustrative embodiments. Furthermore, the illustrative embodiments are described in some instances using particular software, tools, and data processing environments only as an example for the clarity of the description. The illustrative embodiments may be used in conjunction with other comparable or similarly purposed structures, systems, applications, or architectures. For example, other comparable mobile devices, structures, systems, applications, or architectures therefor, may be used in conjunction with such embodiment of the invention within the scope of the invention. An illustrative embodiment may be implemented in hardware, software, or a combination thereof.

[0029] The examples in this disclosure are used only for the clarity of the description and are not limiting to the illustrative embodiments. Additional data, operations, actions, tasks, activities, and manipulations will be conceivable from this disclosure and the same are contemplated within the scope of the illustrative embodiments.

[0030] Various aspects of the present disclosure are described by narrative text, flowcharts, block diagrams of computer systems and/or block diagrams of the machine logic included in computer program product (CPP) embodiments. With respect to any flowcharts, depending upon the technology involved, the operations can be performed in a different order than what is shown in a given flowchart. For example, again depending upon the technology involved, two operations shown in successive flowchart blocks may be performed in reverse order, as a single integrated step, concurrently, or in a manner at least partially overlapping in time.

[0031] A computer program product embodiment ("CPP embodiment" or "CPP") is a term used in the present disclosure to describe any set of one, or more, storage media (also called "mediums") collectively included in a set of one, or more, storage devices that collectively include machine readable code corresponding to instructions and/or data for performing computer operations specified in a given CPP claim. A "storage device" is any tangible device that can retain and store instructions for use by a computer processor. Without limitation, the computer readable storage medium may be an electronic storage medium, a magnetic storage medium, an optical storage medium, an electromagnetic storage medium, a semiconductor storage medium, a mechanical storage medium, or any suitable combination of the foregoing. Some known types of storage devices that include these mediums include: diskette, hard disk, random access memory (RAM), read-only memory (ROM), erasable programmable read-only memory (EPROM or Flash memory), static random-access memory (SRAM), compact disc

read-only memory (CD-ROM), digital versatile disk (DVD), memory stick, floppy disk, mechanically encoded device (such as punch cards or pits / lands formed in a major surface of a disc) or any suitable combination of the foregoing. A computer readable storage medium, as that term is used in the present disclosure, is not to be construed as storage in the form of transitory signals per se, such as radio waves or other freely propagating electromagnetic waves, electromagnetic waves propagating through a waveguide, light pulses passing through a fiber optic cable, electrical signals communicated through a wire, and/or other transmission media. As will be understood by those of skill in the art, data is typically moved at some occasional points in time during normal operations of a storage device, such as during access, de-fragmentation or garbage collection, but this does not render the storage device as transitory because the data is not transitory while it is stored.

[0032] With reference to Fig. 1, this figure depicts a block diagram of a computing environment 100. Data center environment 100 contains an example of an environment for the execution of at least some of the computer code involved in performing the inventive methods, such as an Application module 200 that provides A Recovery Point Objective Validator. In addition to block 200, computing environment 100 includes, for example, computer 101, wide area network (WAN) 102, end user device (EUD) 103, remote server 104, public cloud 105, and private cloud 106. In this embodiment, computer 101 includes processor set 110 (including processing circuitry 120 and cache 121), communication fabric 111, volatile memory 112, persistent storage 113 (including operating system 122 and block 200, as identified above), peripheral device set 114 (including user interface (UI) device set 123, storage 124, and Internet of Things (IoT) sensor set 125), and network module 115. Remote server 104 includes remote database 130. Public cloud 105 includes gateway 140, cloud orchestration module 141, host physical machine set 142, virtual machine set 143, and container set 144.

[0033] COMPUTER 101 may take the form of a desktop computer, laptop computer, tablet computer, smart phone, smart watch or other wearable computer, mainframe computer, quantum computer or any other form of computer or mobile device now known or to be developed in the future that is capable of running a program, accessing a network or querying a database, such as remote database 130. As is well understood in the art of computer technology, and depending upon the technology, performance of a computer-implemented method may be distributed among multiple computers and/or between multiple locations. On the other hand, in this presentation of computing environment 100, detailed discussion is focused on a single computer, specifically computer 101, to keep the presentation as simple

as possible. Computer 101 may be located in a cloud, even though it is not shown in a cloud in Figure 1. On the other hand, computer 101 is not required to be in a cloud except to any extent as may be affirmatively indicated.

[0034] PROCESSOR SET 110 includes one, or more, computer processors of any type now known or to be developed in the future. Processing circuitry 120 may be distributed over multiple packages, for example, multiple, coordinated integrated circuit chips. Processing circuitry 120 may implement multiple processor threads and/or multiple processor cores. Cache 121 is memory that is located in the processor chip package(s) and is typically used for data or code that should be available for rapid access by the threads or cores running on processor set 110. Cache memories are typically organized into multiple levels depending upon relative proximity to the processing circuitry. Alternatively, some, or all, of the cache for the processor set may be located “off chip.” In some computing environments, processor set 110 may be designed for working with qubits and performing quantum computing.

[0035] Computer readable program instructions are typically loaded onto computer 101 to cause a series of operational steps to be performed by processor set 110 of computer 101 and thereby effect a computer-implemented method, such that the instructions thus executed will instantiate the methods specified in flowcharts and/or narrative descriptions of computer-implemented methods included in this document (collectively referred to as “the inventive methods”). These computer readable program instructions are stored in various types of computer readable storage media, such as cache 121 and the other storage media discussed below. The program instructions, and associated data, are accessed by processor set 110 to control and direct performance of the inventive methods. In computing environment 100, at least some of the instructions for performing the inventive methods may be stored in block 200 in persistent storage 113.

[0036] COMMUNICATION FABRIC 111 is the signal conduction path that allows the various components of computer 101 to communicate with each other. Typically, this fabric is made of switches and electrically conductive paths, such as the switches and electrically conductive paths that make up buses, bridges, physical input / output ports and the like. Other types of signal communication paths may be used, such as fiber optic communication paths and/or wireless communication paths.

[0037] VOLATILE MEMORY 112 is any type of volatile memory now known or to be developed in the future. Examples include dynamic type random access memory (RAM) or

static type RAM. Typically, volatile memory 112 is characterized by random access, but this is not required unless affirmatively indicated. In computer 101, the volatile memory 112 is located in a single package and is internal to computer 101, but, alternatively or additionally, the volatile memory may be distributed over multiple packages and/or located externally with respect to computer 101.

[0038] PERSISTENT STORAGE 113 is any form of non-volatile storage for computers that is now known or to be developed in the future. The non-volatility of this storage means that the stored data is maintained regardless of whether power is being supplied to computer 101 and/or directly to persistent storage 113. Persistent storage 113 may be a read only memory (ROM), but typically at least a portion of the persistent storage allows writing of data, deletion of data and re-writing of data. Some familiar forms of persistent storage include magnetic disks and solid state storage devices. Operating system 122 may take several forms, such as various known proprietary operating systems or open source Portable Operating System Interface-type operating systems that employ a kernel. The code included in block 200 typically includes at least some of the computer code involved in performing the inventive methods.

[0039] PERIPHERAL DEVICE SET 114 includes the set of peripheral devices of computer 101. Data communication connections between the peripheral devices and the other components of computer 101 may be implemented in various ways, such as Bluetooth connections, Near-Field Communication (NFC) connections, connections made by cables (such as universal serial bus (USB) type cables), insertion-type connections (for example, secure digital (SD) card), connections made through local area communication networks and even connections made through wide area networks such as the internet. In various embodiments, UI device set 123 may include components such as a display screen, speaker, microphone, wearable devices (such as goggles and smart watches), keyboard, mouse, printer, touchpad, game controllers, and haptic devices. Storage 124 is external storage, such as an external hard drive, or insertable storage, such as an SD card. Storage 124 may be persistent and/or volatile. In some embodiments, storage 124 may take the form of a quantum computing storage device for storing data in the form of qubits. In embodiments where computer 101 is required to have a large amount of storage (for example, where computer 101 locally stores and manages a large database) then this storage may be provided by peripheral storage devices designed for storing very large amounts of data, such as a storage area network (SAN) that is shared by multiple, geographically distributed computers. IoT sensor set 125 is made up of

sensors that can be used in Internet of Things applications. For example, one sensor may be a thermometer and another sensor may be a motion detector.

[0040] NETWORK MODULE 115 is the collection of computer software, hardware, and firmware that allows computer 101 to communicate with other computers through WAN 102. Network module 115 may include hardware, such as modems or Wi-Fi signal transceivers, software for packetizing and/or de-packetizing data for communication network transmission, and/or web browser software for communicating data over the internet. In some embodiments, network control functions and network forwarding functions of network module 115 are performed on the same physical hardware device. In other embodiments (for example, embodiments that utilize software-defined networking (SDN)), the control functions and the forwarding functions of network module 115 are performed on physically separate devices, such that the control functions manage several different network hardware devices. Computer readable program instructions for performing the inventive methods can typically be downloaded to computer 101 from an external computer or external storage device through a network adapter card or network interface included in network module 115.

[0041] WAN 102 is any wide area network (for example, the internet) capable of communicating computer data over non-local distances by any technology for communicating computer data, now known or to be developed in the future. In some embodiments, the WAN 102 may be replaced and/or supplemented by local area networks (LANs) designed to communicate data between devices located in a local area, such as a Wi-Fi network. The WAN and/or LANs typically include computer hardware such as copper transmission cables, optical transmission fibers, wireless transmission, routers, firewalls, switches, gateway computers and edge servers.

[0042] END USER DEVICE (EUD) 103 is any computer system that is used and controlled by an end user (for example, a customer of an enterprise that operates computer 101), and may take any of the forms discussed above in connection with computer 101. EUD 103 typically receives helpful and useful data from the operations of computer 101. For example, in a hypothetical case where computer 101 is designed to provide a recommendation to an end user, this recommendation would typically be communicated from network module 115 of computer 101 through WAN 102 to EUD 103. In this way, EUD 103 can display, or otherwise present, the recommendation to an end user. In some embodiments, EUD 103 may be a client device, such as thin client, heavy client, mainframe computer, desktop computer and so on.

[0043] REMOTE SERVER 104 is any computer system that serves at least some data and/or functionality to computer 101. Remote server 104 may be controlled and used by the same entity that operates computer 101. Remote server 104 represents the machine(s) that collect and store helpful and useful data for use by other computers, such as computer 101. For example, in a hypothetical case where computer 101 is designed and programmed to provide a recommendation based on historical data, then this historical data may be provided to computer 101 from remote database 130 of remote server 104.

[0044] PUBLIC CLOUD 105 is any computer system available for use by multiple entities that provides on-demand availability of computer system resources and/or other computer capabilities, especially data storage (cloud storage) and computing power, without direct active management by the user. Cloud computing typically leverages sharing of resources to achieve coherence and economies of scale. The direct and active management of the computing resources of public cloud 105 is performed by the computer hardware and/or software of cloud orchestration module 141. The computing resources provided by public cloud 105 are typically implemented by virtual computing environments that run on various computers making up the computers of host physical machine set 142, which is the universe of physical computers in and/or available to public cloud 105. The virtual computing environments (VCEs) typically take the form of virtual machines from virtual machine set 143 and/or containers from container set 144. It is understood that these VCEs may be stored as images and may be transferred among and between the various physical machine hosts, either as images or after instantiation of the VCE. Cloud orchestration module 141 manages the transfer and storage of images, deploys new instantiations of VCEs and manages active instantiations of VCE deployments. Gateway 140 is the collection of computer software, hardware, and firmware that allows public cloud 105 to communicate through WAN 102.

[0045] Some further explanation of virtualized computing environments (VCEs) will now be provided. VCEs can be stored as “images.” A new active instance of the VCE can be instantiated from the image. Two familiar types of VCEs are virtual machines and containers. A container is a VCE that uses operating-system-level virtualization. This refers to an operating system feature in which the kernel allows the existence of multiple isolated user-space instances, called containers. These isolated user-space instances typically behave as real computers from the point of view of programs running in them. A computer program running on an ordinary operating system can utilize all resources of that computer, such as connected devices, files and folders, network shares, CPU power, and quantifiable hardware capabilities.

However, programs running inside a container can only use the contents of the container and devices assigned to the container, a feature which is known as containerization.

[0046] PRIVATE CLOUD 106 is similar to public cloud 105, except that the computing resources are only available for use by a single enterprise. While private cloud 106 is depicted as being in communication with WAN 102, in other embodiments a private cloud may be disconnected from the internet entirely and only accessible through a local/private network. A hybrid cloud is a composition of multiple clouds of different types (for example, private, community or public cloud types), often respectively implemented by different vendors. Each of the multiple clouds remains a separate and discrete entity, but the larger hybrid cloud architecture is bound together by standardized or proprietary technology that enables orchestration, management, and/or data/application portability between the multiple constituent clouds. In this embodiment, public cloud 105 and private cloud 106 are both part of a larger hybrid cloud.

[0047] Measured service: cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth, and active user accounts). Resource usage can be monitored, controlled, reported, and invoiced, providing transparency for both the provider and consumer of the utilized service.

[0048] The process software A Recovery Point Objective Validator is shared, simultaneously serving multiple customers in a flexible, automated fashion. It is standardized, requiring little customization, and it is scalable, providing capacity on demand in a pay-as-you-go model.

[0049] The process software can be stored on a shared file system accessible from one or more servers. The process software is executed via transactions that contain data and server processing requests that use CPU units on the accessed server. CPU units are units of time, such as minutes, seconds, and hours, on the central processor of the server. Additionally, the accessed server may make requests of other servers that require CPU units. CPU units are an example that represents but one measurement of use. Other measurements of use include, but are not limited to, network bandwidth, memory usage, storage usage, packet transfers, complete transactions, etc.

[0050] When multiple customers use the same process software application, their transactions are differentiated by the parameters included in the transactions that identify the

unique customer and the type of service for that customer. All of the CPU units and other measurements of use that are used for the services for each customer are recorded. When the number of transactions to any one server reaches a number that begins to affect the performance of that server, other servers are accessed to increase the capacity and to share the workload. Likewise, when other measurements of use, such as network bandwidth, memory usage, storage usage, etc., approach a capacity so as to affect performance, additional network bandwidth, memory usage, storage, etc. are added to share the workload.

[0051] The measurements of use employed for each service and customer are sent to a collecting server that sums the measurements of use for each customer for each service that was processed anywhere in the network of servers that provide the shared execution of the process software. The summed measurements of use units are periodically multiplied by unit costs, and the resulting total process software application service costs are alternatively sent to the customer and/or indicated on a web site accessed by the customer, who may then remit payment to the service provider.

[0052] In another embodiment, the service provider requests payment directly from a customer account at a banking or financial institution.

[0053] In another embodiment, if the service provider is also a customer of the customer that uses the process software application, the payment owed to the service provider is reconciled to the payment owed by the service provider to minimize the transfer of payments.

[0054] Fig. 2 depicts a diagram 220 in accordance with an illustrative embodiment. In a particular embodiment, the components 220 are representative of aspects of the application 200 of Fig. 1.

[0055] In the illustrated embodiment, a slow detection of a recovery point objective (RPO) violation 222 is shown in the top left where a long RPO of over an hour and the replication has not occurred within the RPO interval. The violation is only detected after a long duration from the first replication, and close to the next anticipated replication. In the illustrated embodiment shown in the bottom left, the detection is performed much sooner.

[0056] Typically, replication of the consistency group replicates the data in the underlying volumes. In practice, it is preferable if the data is replicated together with the application configuration during the same interval. The diagram in the top right shows an example of the storage RPO inconsistent with other critical data sources 224 such as application meta data when a snapshot of the underlying volumes is taken but a long duration elapses before the

other critical data is replicated. In the illustrated embodiment, this is addressed with a fast detection of the snapshot as shown in the bottom right.

[0057] Fig. 3 depicts a system diagram 300 in accordance with an illustrative embodiment. In a particular embodiment, the components of the system 300 are representative of aspects of the application 200 of Fig. 1.

[0058] In the illustrated embodiment, a primary site 302 comprises a consistency group which contains production applications. A Validation Writer 306 with access to the Validation Area 318 within a consistency group of the primary site writes a validation metric 314 to the Validation Area 318. In an embodiment, the Validation Writer writes a validation metric for each of a plurality of applications in the primary consistency group at an interval determined by the RPO. A secondary site 304 comprises a consistency group which contains applications, a Validation Reader 308 with access to the Validation Area 320 within a consistency group of the secondary site, reads the validation metric 314 from the Validation Area. In some embodiments, a Validation Area 310 for example may be a circular buffer comprising of validation metrics that may comprise of persistent versions across the primary and secondary sites: Validation Area 318 and Validation Area 320. In yet another embodiment, the Validation Area may be replicated from the primary site to the secondary site for example when the consistency group of the primary site is replicated to the consistency group of the secondary site. Each validation metric 314 may comprise a time stamp, generation number, writer's time when the block was written, application universal unique identifier (UUID), and target RPO. In some embodiments, the validation results are observed 312 from the observe recovery point validation results component, for example an orchestrator or a controller. In embodiments, it may be seen that the various components described herein including the Validation Writer, Validation Reader, and the Validation Area within the consistency group of the primary site, the Validation Area within the consistency group of the secondary site are storage and application agnostic and may independently validate the results of the replication.

[0059] In the illustrated embodiment, a disaster recovery (DR) replication relationship 316 is configured to perform replication of the consistency group in the primary area at the RPO interval based on a snapshot-based replication which captures the current state of the consistency group at a moment in time. Snapshot-based replication can be used for backup or disaster recovery, but data is only as complete as the most recent snapshot.

[0060] Fig. 4 depicts a flowchart diagram 400 in accordance with an illustrative embodiment. In a particular embodiment, the components 400 are representative of aspects of the application 200 of Fig. 1.

[0061] In the illustrated embodiment, the start validation configuration 402 begins the steps, followed by provision validation area in consistency group 404. The start validation writer 406 starts the operation of the Validation Writer by filling validation data in memory 408 which writes a validation metric to validation area 410. The Validation Writer next waits for RPO seconds 412 or a RPO interval. The process repeats from 408.

[0062] The start Validation Reader 414 begins the operation of the Validation Reader. The values are first initialized, the Now value is assigned the current clock time of the secondary site 416, the violated value is set to false, and the previous value is set to now 418. After the initial run, new values may be entered if a new validation metric is available 428. In an embodiment, the Validation Reader waits for an interval for instance half the RPO seconds (interval) 420. In another embodiment, the Validation Reader detects the new validation metric. The values are recomputed 422 with Now assigned to current clock time of the secondary site, the recovery point time delta computed by subtracting previous from Now. The computation of the recovery point time delta may also comprise comparing the validation metric. For example, the generation number of the validation metric from the previous reading may be compared with the generation number of the validation metric from the current reading. In an embodiment, a replication event may also be computed from the generation number where the generation number is incremented when a replication occurs. A decision is made 424, if the recovery point time delta exceeds the RPO, report RPO violation 430 and set the violated value to true. Next, proceed to updating the validation area and reading the validation area into memory 426, and continue to 428. In embodiments, the validation result may comprise of the recovery point time delta value and the decision, which may be a Boolean value or equivalent. The outcome of the replication may be inferred from the validation result, for example, success if the delta does not exceed the RPO or failure, if the recovery point time delta exceeds the RPO. In an embodiment, the updated validation area is detected by the applications in the primary area.

[0063] Fig. 5 depicts a diagram 500 in accordance with an illustrative embodiment. In a particular embodiment, the components 500 are representative of aspects of the application 200 of Fig. 1.

[0064] Starting from the left, the first depiction shows the correct aligned 502 when the writer (Validation Writer) writes before the storage is replicated and the reader (Validation Reader) reads after the storage is replicated. In an embodiment, the storage replication event may be computed from the generation number of the validation metric written by the Validation Writer compared to the generation number of the validation metric read by the Validation Reader. The middle depiction shows unaligned, reader (Validation Reader) too early 504, and the right depiction shows unaligned writer (Validation Writer) too late 506. In both 504 and 506, the unaligned state may be detected, for example, when the reader fails to detect a new generation number on expected schedule.

[0065] Fig. 6 depicts a diagram 600 in accordance with an illustrative embodiment. In a particular embodiment, the components 600 are representative of aspects of the application 200 of Fig. 1.

[0066] The diagram on the left depicts the state diagram of an overall 602 operation to align unaligned Validation Reader and Validation Writer. In an embodiment, the Validation Reader may signal a hint to the Validation Writer comprising the last read generation number and last write timestamp. In another embodiment, the Validation Reader and Validation Writer may be automatically aligned based for example on a failed or delayed replication or an updated RPO. In some embodiments, the Validation Writer and the Validation Reader are automatically aligned based on a recovery point objective metric. For example, a recovery point objective metric may comprise a failed or delayed replication or an updated RPO. A policy slow will perform a slow search until alignment is found or the slow search duration is exceeded, in which case the fast search is performed. A policy fast search will perform a fast search until alignment is found. If alignment is lost, repeat from the start. The diagram on the right depicts a slow search 604 where both the Validation Reader and Validation Writer make adjustments.

[0067] Fig. 7 depicts a diagram 700 in accordance with an illustrative embodiment. In a particular embodiment, the components 700 are representative of aspects of the application 200 of Fig. 1.

[0068] For a fast search 702 frequently perform writes and reads to align the Validation Writer and the Validation Reader within one replication cycle. In some embodiments, aligning the Validation Writer and the Validation Reader comprises performing a fast search comprising an optimization of alignment precision relative to system impact. For example, the

fast search may enable the alignment precision of the Validation Writer and the Validation Reader to be achieved sooner but at the expense of higher impact to system resources such as CPU, or disk usage. A slow search 704 may be performed comprising for example one read and one write per replication cycle where alignment may take multiple replication cycles. In some embodiments, aligning the Validation Writer and the Validation Reader comprises performing a slow search comprising an optimization of system impact relative to alignment precision. For example, the slow search may optimize i.e. minimize the impact to system resources relative to the fast alignment of the Validation Writer and the Validation Reader. In an embodiment, the Validation Writer may shift right on the time scale for example, writing later to the Validation Area and then shifting left to adjust. The Validation Reader may shift left to read earlier from the Validation Area and shifting right to adjust.

[0069] Fig. 8 depicts a flowchart diagram of an example On Demand process in accordance with an illustrative embodiment.

[0070] Step 840 begins the On Demand process. A transaction is created that contains the unique customer identification, the requested service type, and any service parameters that further specify the type of service (841). The transaction is then sent to the main server (842). In an On Demand environment, the main server can initially be the only server, and then as capacity is consumed other servers are added to the On Demand environment.

[0071] The server central processing unit (CPU) capacities in the On Demand environment are queried (843). The CPU requirement of the transaction is estimated, and then the server's available CPU capacity in the On Demand environment is compared to the transaction CPU requirement to see if there is sufficient CPU available capacity in any server to process the transaction (844). If there is not sufficient server CPU available capacity, then additional server CPU capacity is allocated to process the transaction (848). If there was already sufficient available CPU capacity, then the transaction is sent to a selected server (845).

[0072] Before executing the transaction, a check is made of the remaining On Demand environment to determine if the environment has sufficient available capacity for processing the transaction. This environment capacity consists of such things as, but not limited to, network bandwidth, processor memory, storage etc. (846). If there is not sufficient available capacity, then capacity will be added to the On Demand environment (847). Next the required software to process the transaction is accessed, loaded into memory, and then the transaction is executed (849).

[0073] The usage measurements are recorded (850). The usage measurements consist of the portions of those functions in the On Demand environment that are used to process the transaction. The usage of such functions as, but not limited to, network bandwidth, processor memory, storage and CPU cycles are what is recorded. The usage measurements are summed, multiplied by unit costs, and then recorded as a charge to the requesting customer (851).

[0074] If the customer has requested that the On Demand costs be posted to a web site (852), then they are posted thereto (853). If the customer has requested that the On Demand costs be sent via e-mail to a customer address (854), then they are sent (855). If the customer has requested that the On Demand costs be paid directly from a customer account (856), then payment is received directly from the customer account (857). On Demand process proceeds to 858 and exits.

[0075] While it is understood that the process software A Recovery Point Objective Validator may be deployed by manually loading it directly in the client, server, and proxy computers via loading a storage medium such as a CD, DVD, etc., the process software may also be automatically or semi-automatically deployed into a computer system by sending the process software to a central server or a group of central servers. The process software is then downloaded into the client computers that will execute the process software. Alternatively, the process software is sent directly to the client system via e-mail. The process software is then either detached to a directory or loaded into a directory by executing a set of program instructions that detaches the process software into a directory. Another alternative is to send the process software directly to a directory on the client computer hard drive. When there are proxy servers, the process will select the proxy server code, determine on which computers to place the proxy servers' code, transmit the proxy server code, and then install the proxy server code on the proxy computer. The process software will be transmitted to the proxy server, and then it will be stored on the proxy server.

[0076] Introduction

[0077] • Loading the process software directly on the client, server, and proxy computers by loading a storage medium such as a CD, DVD, etc.

[0078] • Automatically or semi-automatically deploying the process software into a computer system by sending it to a central server or a group of central servers. The process software is then downloaded into the client computers that will execute the process software.

[0079] • Sending the process software directly to the client system via e-mail. The process software is then either detached to a directory or loaded into a directory by executing a set of program instructions that detaches the process software into a directory.

[0080] • Sending the process software directly to a directory on the client computer hard drive.

[0081] • When there are proxy servers, the process will select the proxy server code, determine on which computers to place the proxy servers' code, transmit the proxy server code, and then install the proxy server code on the proxy computer. The process software will be transmitted to the proxy server and then it will be stored on the proxy server.

[0082] General Description

[0083] While it is understood that the process software A Recovery Point Objective Validator may be deployed by manually loading it directly in the client, server, and proxy computers via loading a storage medium such as a CD, DVD, etc., the process software may also be automatically or semi-automatically deployed into a computer system by sending the process software to a central server or a group of central servers. The process software is then downloaded into the client computers that will execute the process software. Alternatively, the process software is sent directly to the client system via e-mail. The process software is then either detached to a directory or loaded into a directory by executing a set of program instructions that detaches the process software into a directory. Another alternative is to send the process software directly to a directory on the client computer hard drive. When there are proxy servers, the process will select the proxy server code, determine on which computers to place the proxy servers' code, transmit the proxy server code, and then install the proxy server code on the proxy computer. The process software will be transmitted to the proxy server, and then it will be stored on the proxy server.

[0084] The following definitions and abbreviations are to be used for the interpretation of the claims and the specification. As used herein, the terms “comprises,” “comprising,” “includes,” “including,” “has,” “having,” “contains” or “containing,” or any other variation thereof, are intended to cover a non-exclusive inclusion. For example, a composition, a mixture, process, method, article, or apparatus that comprises a list of elements is not necessarily limited to only those elements but can include other elements not expressly listed or inherent to such composition, mixture, process, method, article, or apparatus.

[0085] Additionally, the term “illustrative” is used herein to mean “serving as an example, instance or illustration.” Any embodiment or design described herein as “illustrative” is not necessarily to be construed as preferred or advantageous over other embodiments or designs. The terms “at least one” and “one or more” are understood to include any integer number greater than or equal to one, i.e., one, two, three, four, etc. The terms “a plurality” are understood to include any integer number greater than or equal to two, i.e., two, three, four, five, etc. The term “connection” can include an indirect “connection” and a direct “connection.”

[0086] References in the specification to “one embodiment,” “an embodiment,” “an example embodiment,” etc., indicate that the embodiment described can include a particular feature, structure, or characteristic, but every embodiment may or may not include the particular feature, structure, or characteristic. Moreover, such phrases are not necessarily referring to the same embodiment. Further, when a particular feature, structure, or characteristic is described in connection with an embodiment, it is submitted that it is within the knowledge of one skilled in the art to affect such feature, structure, or characteristic in connection with other embodiments whether or not explicitly described.

[0087] The terms “about,” “substantially,” “approximately,” and variations thereof, are intended to include the degree of error associated with measurement of the particular quantity based upon the equipment available at the time of filing the application. For example, “about” can include a range of $\pm 8\%$ or 5% , or 2% of a given value.

[0088] The descriptions of the various embodiments of the present invention have been presented for purposes of illustration but are not intended to be exhaustive or limited to the embodiments disclosed. Many modifications and variations will be apparent to those of ordinary skill in the art without departing from the scope and spirit of the described embodiments. The terminology used herein was chosen to best explain the principles of the embodiments, the practical application or technical improvement over technologies found in the marketplace, or to enable others of ordinary skill in the art to understand the embodiments described herein.

[0089] The descriptions of the various embodiments of the present invention have been presented for purposes of illustration but are not intended to be exhaustive or limited to the embodiments disclosed. Many modifications and variations will be apparent to those of ordinary skill in the art without departing from the scope and spirit of the described embodiments. The terminology used herein was chosen to best explain the principles of the

embodiments, the practical application or technical improvement over technologies found in the marketplace, or to enable others of ordinary skill in the art to understand the embodiments described herein.

[0090] Thus, a computer implemented method, system or apparatus, and computer program product are provided in the illustrative embodiments for managing participation in online communities and other related features, functions, or operations. Where an embodiment or a portion thereof is described with respect to a type of device, the computer implemented method, system or apparatus, the computer program product, or a portion thereof, are adapted or configured for use with a suitable and comparable manifestation of that type of device.

[0091] Where an embodiment is described as implemented in an application, the delivery of the application in a Software as a Service (SaaS) model is contemplated within the scope of the illustrative embodiments. In a SaaS model, the capability of the application implementing an embodiment is provided to a user by executing the application in a cloud infrastructure. The user can access the application using a variety of client devices through a thin client interface such as a web browser (e.g., web-based e-mail), or other light-weight client-applications. The user does not manage or control the underlying cloud infrastructure including the network, servers, operating systems, or the storage of the cloud infrastructure. In some cases, the user may not even manage or control the capabilities of the SaaS application. In some other cases, the SaaS implementation of the application may permit a possible exception of limited user-specific application configuration settings.

[0092] Embodiments of the present invention may also be delivered as part of a service engagement with a client corporation, nonprofit organization, government entity, internal organizational structure, or the like. Aspects of these embodiments may include configuring a computer system to perform, and deploying software, hardware, and web services that implement, some or all of the methods described herein. Aspects of these embodiments may also include analyzing the client's operations, creating recommendations responsive to the analysis, building systems that implement portions of the recommendations, integrating the systems into existing processes and infrastructure, metering use of the systems, allocating expenses to users of the systems, and billing for use of the systems. Although the above embodiments of present invention each have been described by stating their individual advantages, respectively, present invention is not limited to a particular combination thereof. To the contrary, such embodiments may also be combined in any way and number according to the intended deployment of present invention without losing their beneficial effects.

CLAIMS

1. A computer-implemented method comprising:

provisioning a Validation Area within a consistency group of a primary site and a Validation Area within a consistency group of a secondary site;

provisioning a Validation Writer with access to the Validation Area in the primary site and provisioning a Validation Reader with access to the Validation Area in the secondary site;

writing a validation metric by the Validation Writer into the Validation Area in the primary site at an interval determined by a recovery point objective; and

reading the validation metric by the Validation Reader from the Validation Area in the secondary site, computing a validation result by the Validation Reader based in part on the validation metric, the recovery point objective and a recovery point time delta wherein the validation result comprises an outcome of a replication of the consistency group in the primary site to the consistency group in the secondary site wherein the Validation Area within the consistency group of the primary site, the Validation Area within the consistency group of the secondary site, the Validation Writer and the Validation Reader are storage and application agnostic.

2. The computer-implemented method of claim 1, wherein computing the validation result comprises comparing a current generation number of the validation metric with a previous generation number of the validation metric.

3. The computer-implemented method of claims 1 or 2, wherein a replication event is computed based on the validation result.

4. The computer-implemented method of any one of the claims 1 to 3, wherein the Validation Writer and the Validation Reader are automatically aligned based on a recovery point objective metric.

5. The computer-implemented method of any one of the claims 1 to 4, wherein aligning the Validation Writer and the Validation Reader comprises performing a slow search comprising an optimization of a system impact relative to an alignment precision.

6. The computer-implemented method of any one of the claims 1 to 5, wherein aligning the Validation Writer and the Validation Reader comprises performing a fast search comprising an optimization of an alignment precision relative to a system impact.
7. The computer-implemented method of any one of the claims 1 to 6, wherein an application on the primary site detects the validation result computed by the Validation Reader.
8. A computer program product comprising one or more computer readable storage media, and program instructions collectively stored on the one or more computer readable storage media, the program instructions executable by a processor to cause the processor to perform operations comprising:
- provisioning a Validation Area within a consistency group of a primary site and a Validation Area within a consistency group of a secondary site;
 - provisioning a Validation Writer with access to the Validation Area in the primary site and provisioning a Validation Reader with access to the Validation Area in the secondary site;
 - writing a validation metric by the Validation Writer into the Validation Area in the primary site at an interval determined by a recovery point objective; and
 - reading the validation metric by the Validation Reader from the Validation Area in the secondary site, computing a validation result by the Validation Reader based in part on the validation metric, the recovery point objective and a recovery point time delta wherein the validation result comprises an outcome of a replication of the consistency group in the primary site to the consistency group in the secondary site wherein the Validation Area within the consistency group of the primary site, the Validation Area within the consistency group of the secondary site, the Validation Writer and the Validation Reader are storage and application agnostic.
9. The computer program product of claim 8, wherein computing the validation result comprises comparing a current generation number of the validation metric with a previous generation number of the validation metric.
10. The computer program product of claims 8 or 9, wherein a replication event is computed based on the validation result.

11. The computer program product of any one of the claims 8 to 10, wherein the Validation Writer and the Validation Reader are automatically aligned based on a recovery point objective metric.
12. The computer program product of any one of the claims 8 to 11, wherein aligning the Validation Writer and the Validation Reader comprises performing a slow search comprising an optimization of a system impact relative to an alignment precision.
13. The computer program product of any one of the claims 8 to 12, wherein aligning the Validation Writer and the Validation Reader comprises performing a fast search comprising an optimization of an alignment precision relative to a system impact.
14. The computer program product of any one of the claims 8 to 13, wherein an application on the primary site senses the validation result computed by the Validation Reader.
15. A computer system comprising a processor and one or more computer readable storage media, and program instructions collectively stored on the one or more computer readable storage media, the program instructions executable by the processor to cause the processor to perform operations comprising:
- provisioning a Validation Area within a consistency group of a primary site and a Validation Area within a consistency group of a secondary site;
 - provisioning a Validation Writer with access to the Validation Area in the primary site and provisioning a Validation Reader with access to the Validation Area in the secondary site;
 - writing a validation metric by the Validation Writer into the Validation Area in the primary site at an interval determined by a recovery point objective; and
 - reading the validation metric by the Validation Reader from the Validation Area in the secondary site, computing a validation result by the Validation Reader based in part on the validation metric, the recovery point objective and a recovery point time delta wherein the validation result comprises an outcome of a replication of the consistency group in the primary site to the consistency group in the secondary site wherein the Validation Area within the consistency group of the primary site, the Validation Area within the consistency group of the secondary site, the Validation Writer and the Validation Reader are storage and application agnostic.

16. The computer system of claim 15, wherein computing the validation result comprises comparing a current generation number of the validation metric with a previous generation number of the validation metric.

17. The computer system of claims 15 or 16, wherein the Validation Writer and the Validation Reader are automatically aligned based on a recovery point objective metric.

18. The computer system of any one of the claims 15 to 17, wherein aligning the Validation Writer and the Validation Reader comprises performing a slow search comprising an optimization of a system impact relative to an alignment precision.

19. The computer system of any one of the claims 15 to 18, wherein aligning the Validation Writer and the Validation Reader comprises performing a fast search comprising an optimization of an alignment precision relative to a system impact.

20. The computer system of any one of the claims 15 to 19, wherein an application on the primary site senses the validation result computed by the Validation Reader.

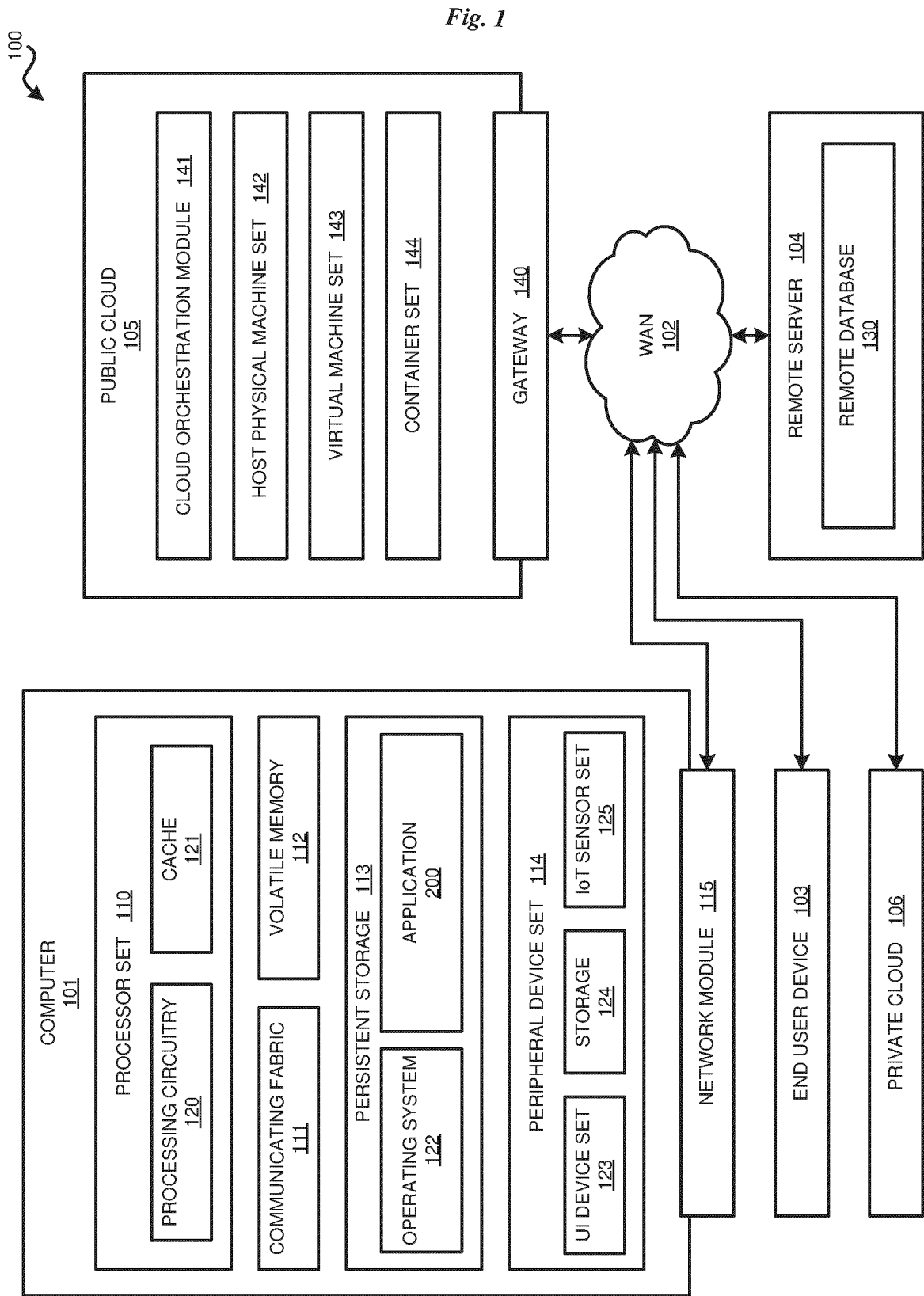


Fig. 2

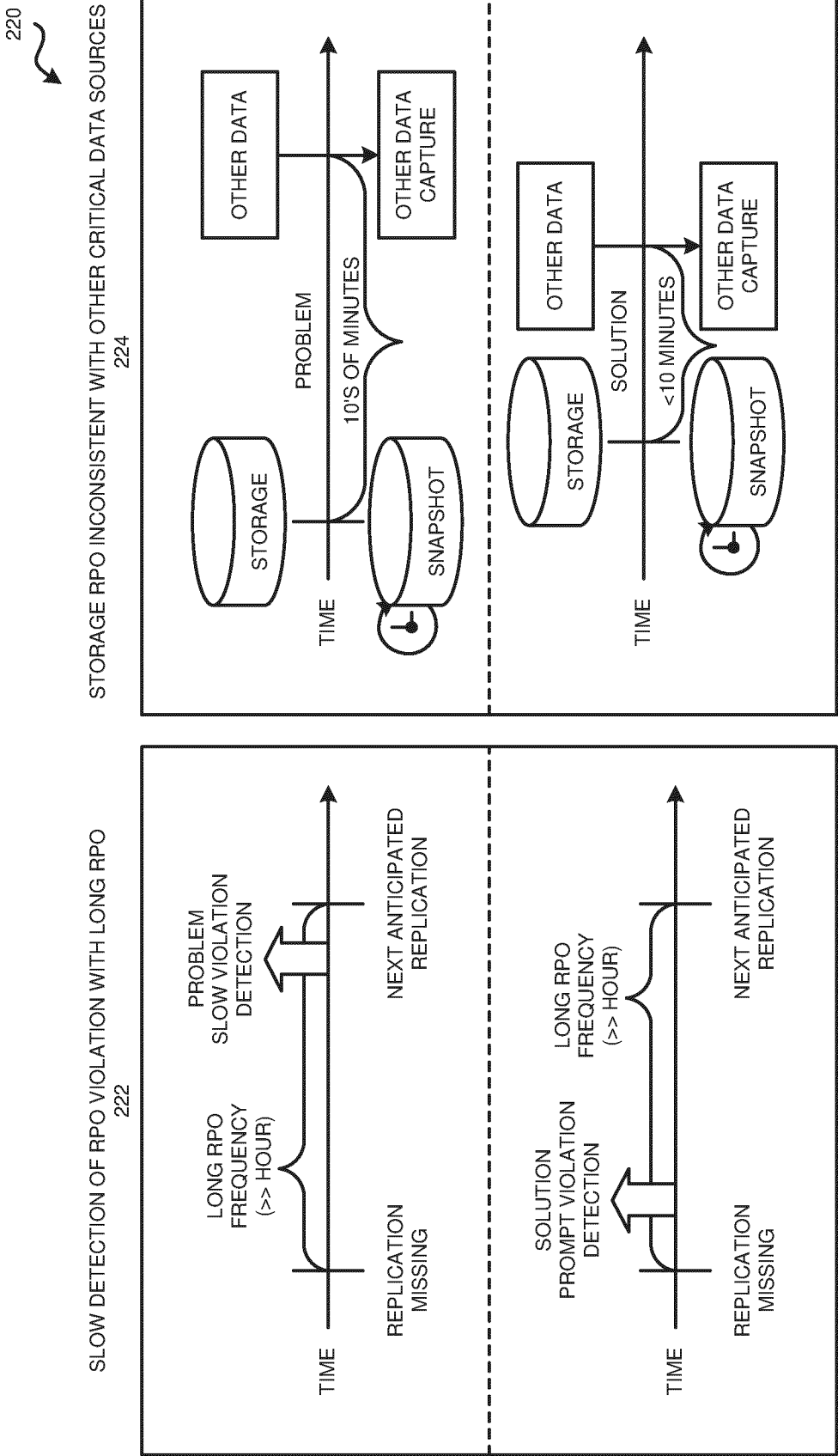


Fig. 3

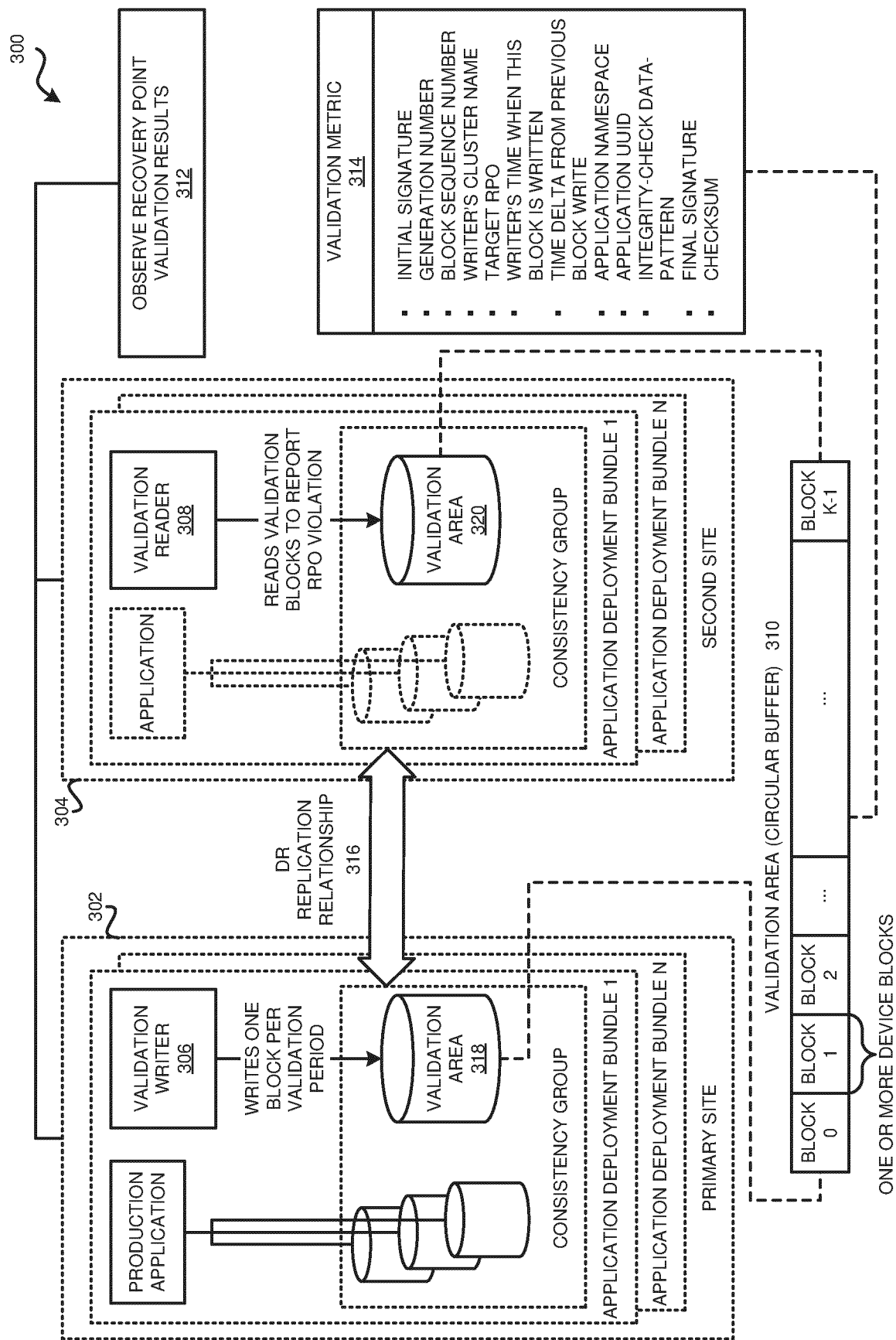


Fig. 4

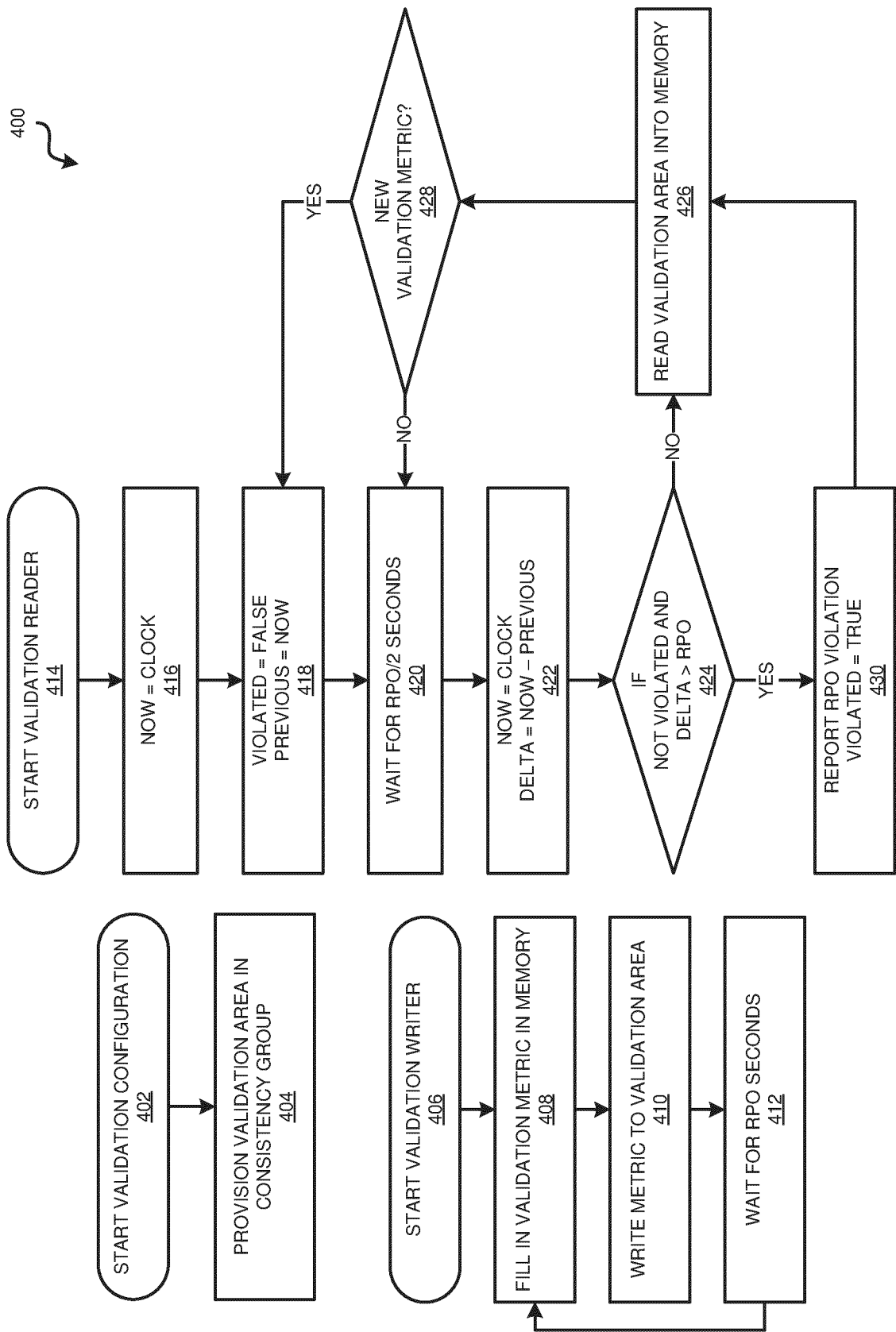
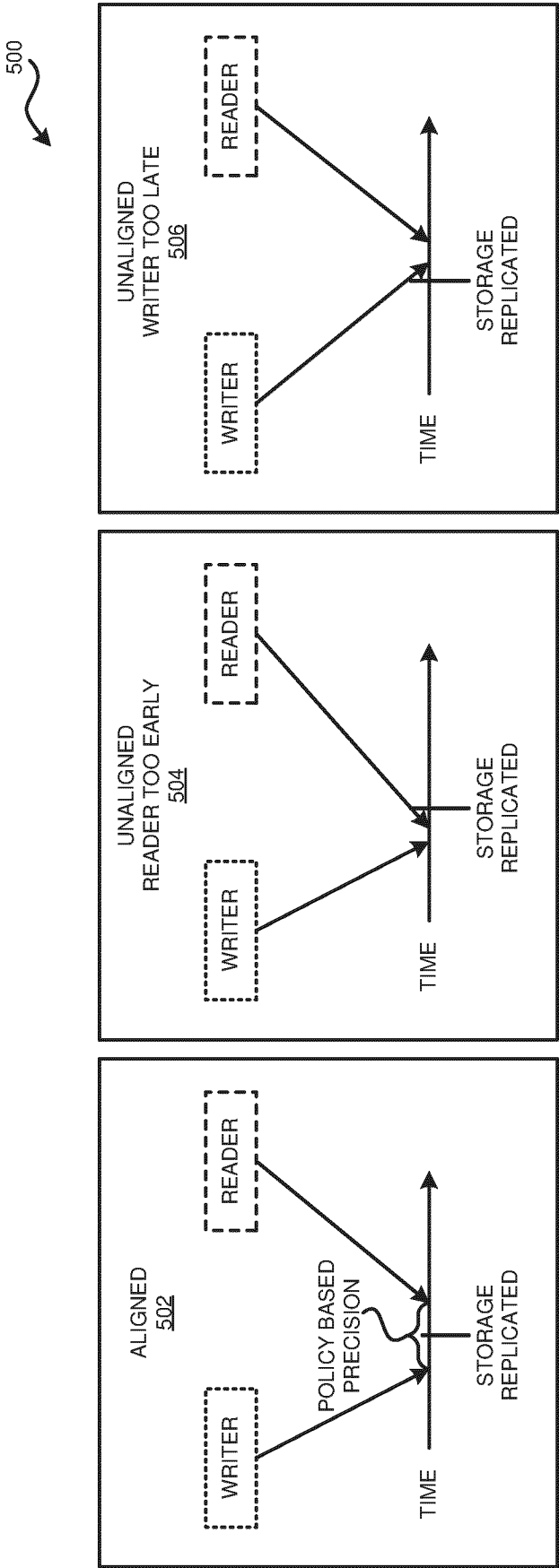
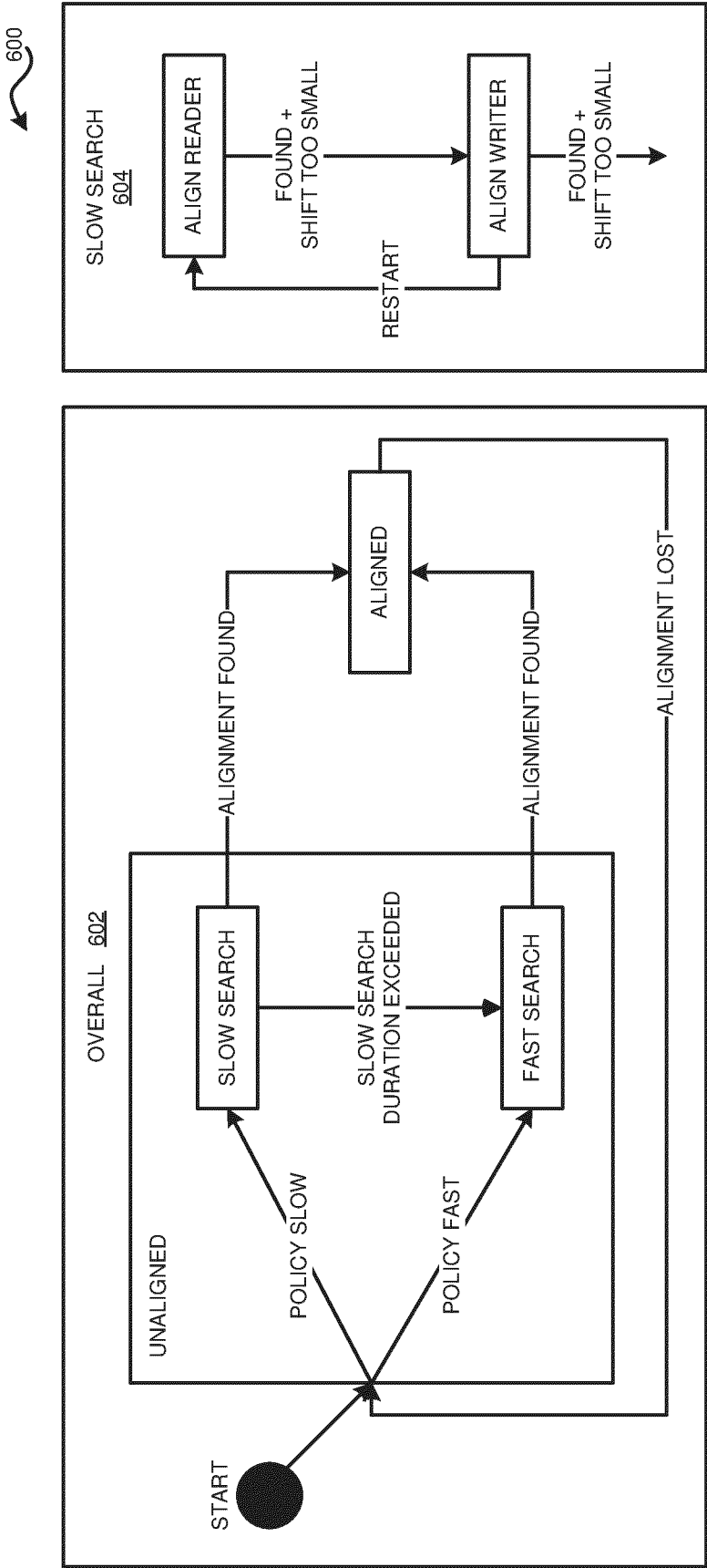


Fig. 5



UNALIGNED == READER FAILS TO DETECT NEW GENERATION ON EXPECTED SCHEDULE

Fig. 6



7/8
Fig. 7

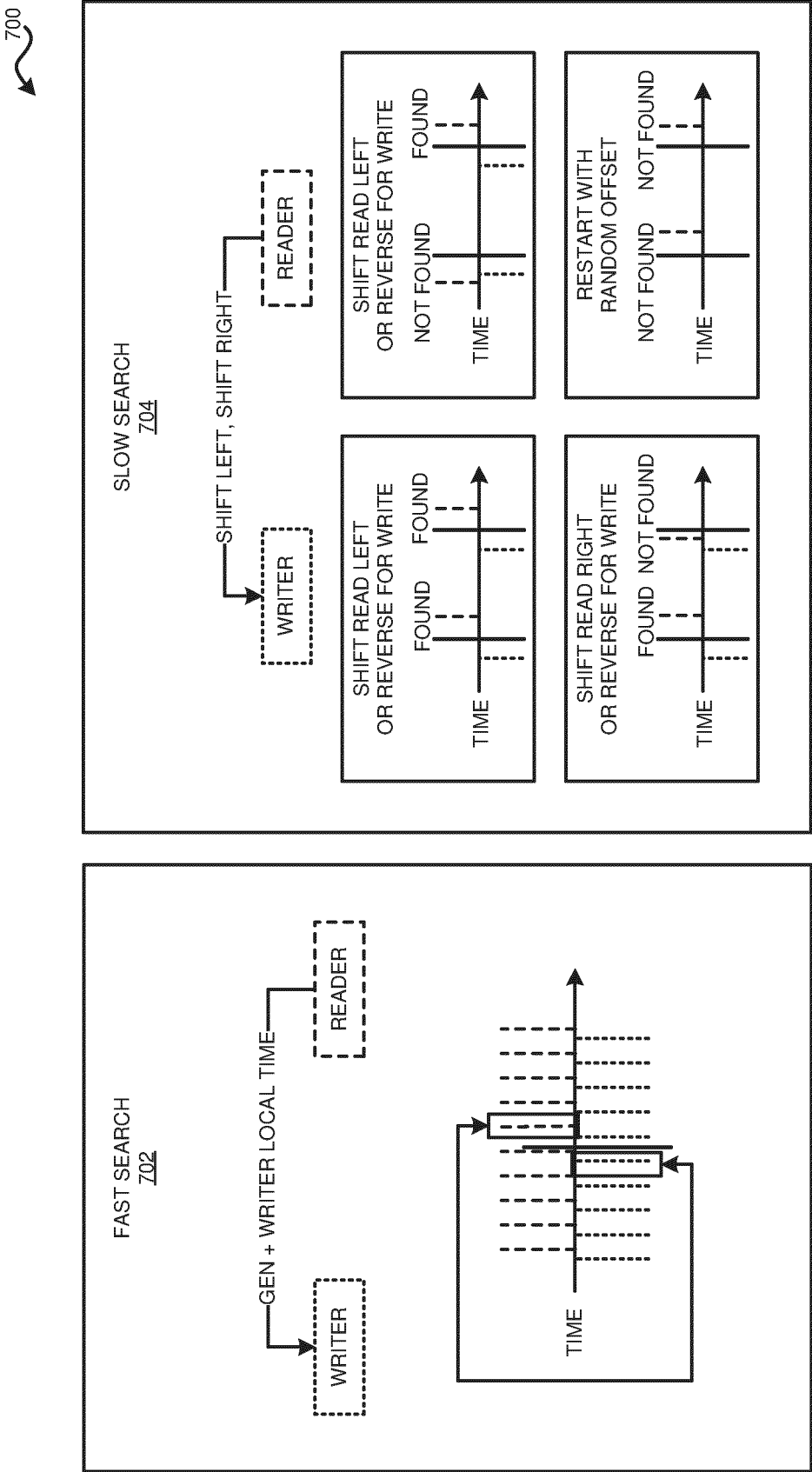
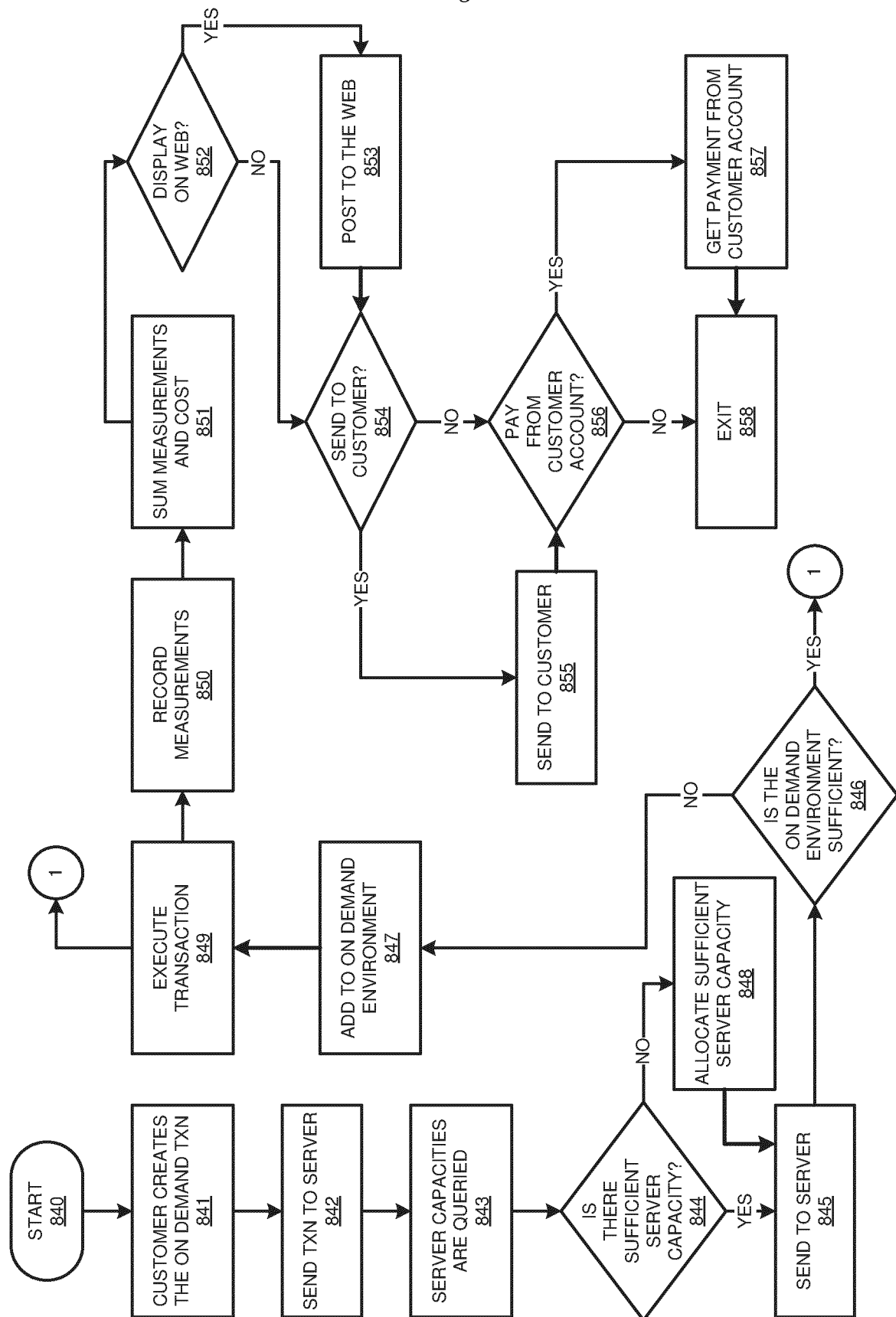


Fig. 8



INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2024/083260

A. CLASSIFICATION OF SUBJECT MATTER INV. H04L67/1095 H04L69/40 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	US 2022/318264 A1 (JAIN ABHISHEK [US] ET AL) 6 October 2022 (2022-10-06) paragraph [0236] paragraph [0245] - paragraph [0247] paragraphs [0225], [0238], [0222] -----	1 - 20
☐ Further documents are listed in the continuation of Box C. ☒ See patent family annex.		
<p>* Special categories of cited documents :</p> <p>"A" document defining the general state of the art which is not considered to be of particular relevance</p> <p>"E" earlier application or patent but published on or after the international filing date</p> <p>"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)</p> <p>"O" document referring to an oral disclosure, use, exhibition or other means</p> <p>"P" document published prior to the international filing date but later than the priority date claimed</p> <p>"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</p> <p>"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</p> <p>"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</p> <p>"&" document member of the same patent family</p>		
Date of the actual completion of the international search 31 January 2025		Date of mailing of the international search report 14/02/2025
Name and mailing address of the ISA/ European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016		Authorized officer Manea, Anda

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/EP2024/083260

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2022318264 A1	06-10-2022	US 2022318264 A1	06-10-2022
		US 2023087089 A1	23-03-2023
		US 2024394277 A1	28-11-2024
