US 20200233881A1

(19) United States
(12) Patent Application Publication
Harduf et al.
(10) Pub. No.: US 2020/0233881 A1
(43) Pub. Date: Jul. 23, 2020

(54) STORAGE SYSTEM WITH DATA CONSISTENCY CHECKING IN SYNCHRONOUS REPLICATION USING ACTIVE SNAPSHOT SET

(71) Applicant: **EMC IP Holding Company LLC**, Hopkinton, MA (US)

(72) Inventors: **Yuval Harduf**, Yahud (IL); **Xiangping Chen**, Sherborn, MA (US); **David Meiri**, Somerville, MA (US)

(21) Appl. No.: **16/253,793**

(22) Filed: **Jan. 22, 2019**

Publication Classification

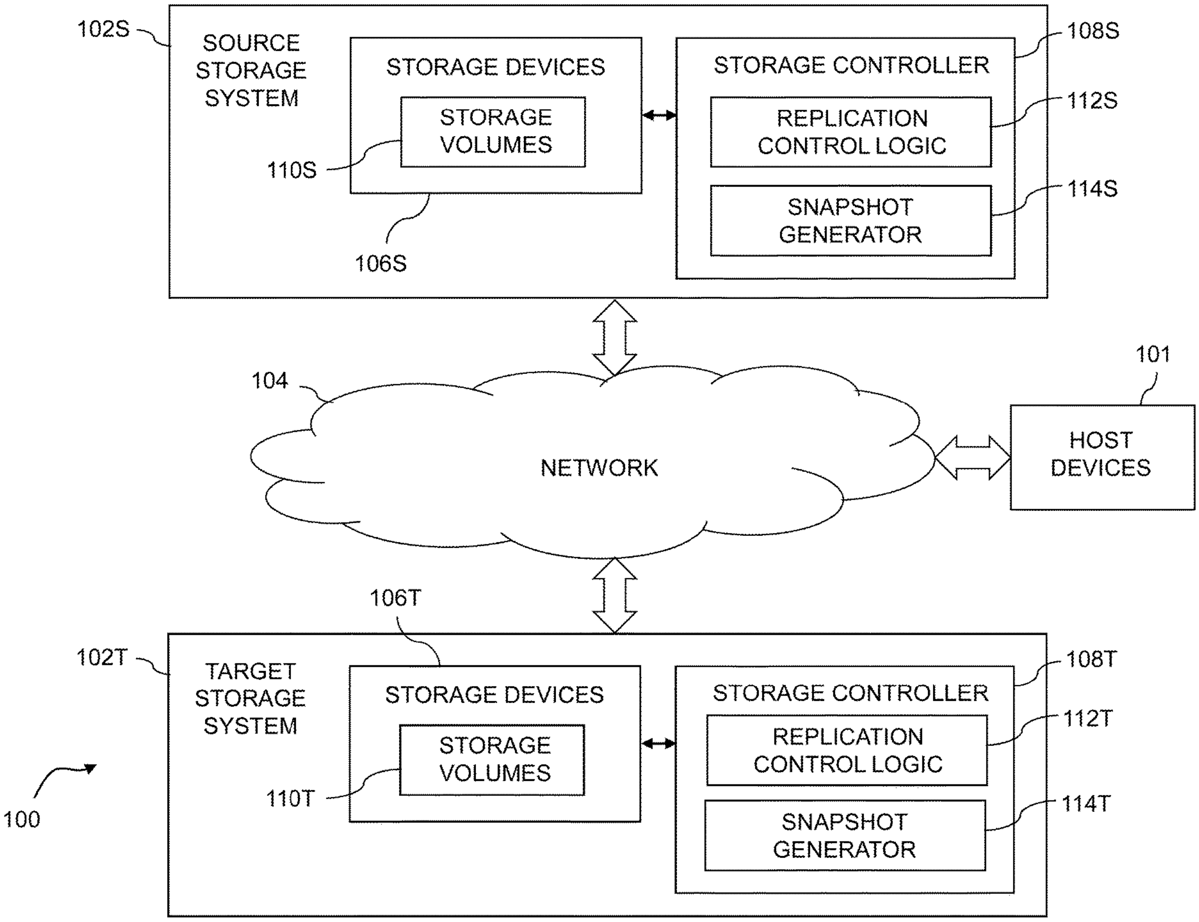
(51) **Int. Cl.**

G06F 16/27	(2006.01)
G06F 16/23	(2006.01)
G06F 16/22	(2006.01)

(52) **U.S. Cl.**
CPC ***G06F 16/275*** (2019.01); ***G06F 16/2246*** (2019.01); ***G06F 16/2365*** (2019.01)

(57) ABSTRACT

An apparatus comprises at least one processing device configured to define a consistency group comprising a plurality of storage volumes subject to replication from a source storage system to a target storage system. In conjunction with a transition from asynchronous replication of the consistency group to synchronous replication of the consistency group, the at least one processing device completes a current asynchronous replication cycle, utilizes a current asynchronous snapshot set on the target storage system to establish an active snapshot set on the target storage system, transfers differential data of a subsequent asynchronous replication cycle to the active snapshot set on the target storage system, specifies the active snapshot set on the target storage system as a destination of the synchronous replication, and performs a data consistency check on the target storage system utilizing an additional snapshot set generated from the active snapshot set.



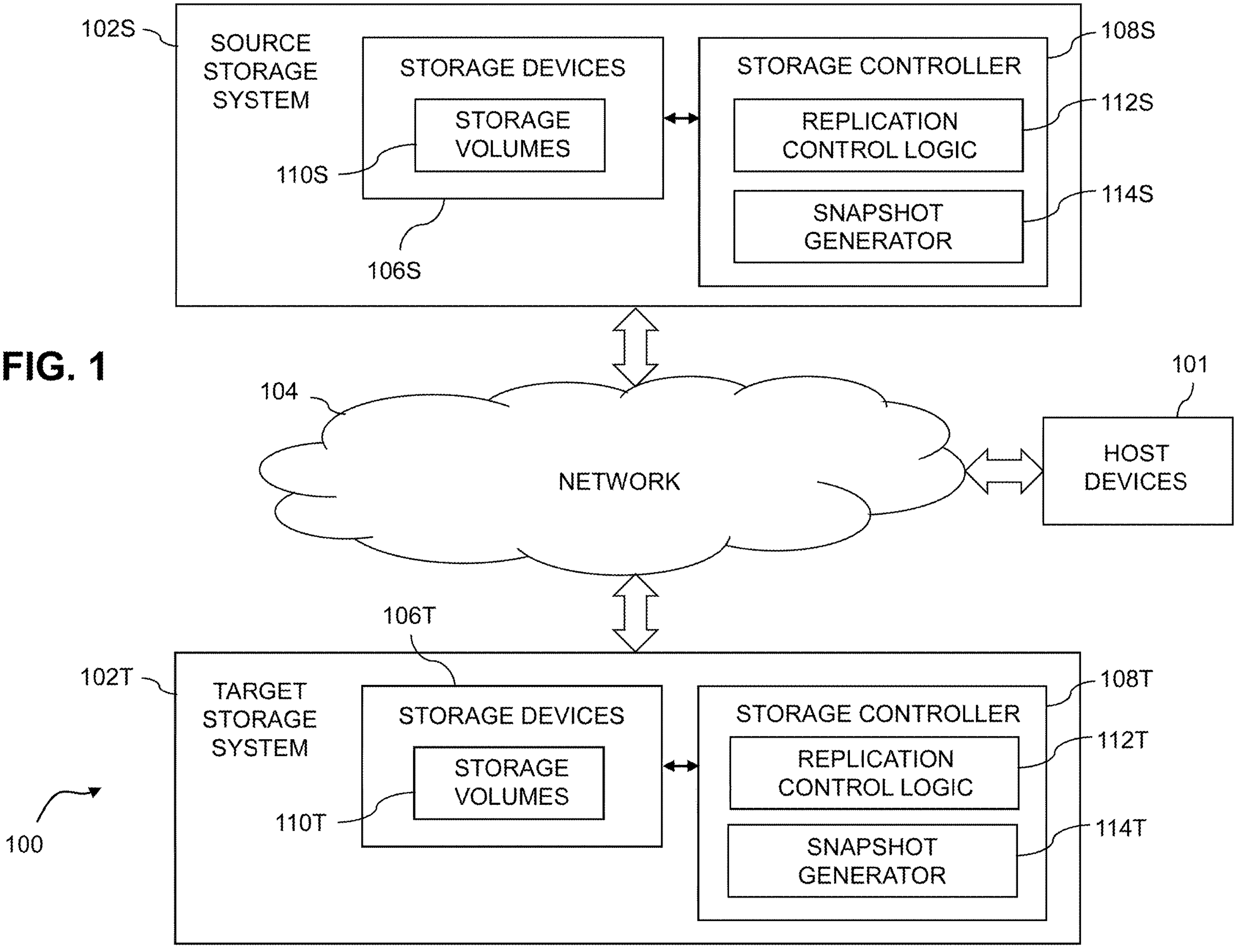


FIG. 1

102S
SOURCE STORAGE SYSTEM
STORAGE DEVICES
STORAGE VOLUMES
110S
106S
STORAGE CONTROLLER
108S
REPLICATION CONTROL LOGIC
112S
SNAPSHOT GENERATOR
114S
104
NETWORK
101
HOST DEVICES
106T
102T
TARGET STORAGE SYSTEM
STORAGE DEVICES
STORAGE VOLUMES
110T
STORAGE CONTROLLER
108T
REPLICATION CONTROL LOGIC
112T
SNAPSHOT GENERATOR
114T
100

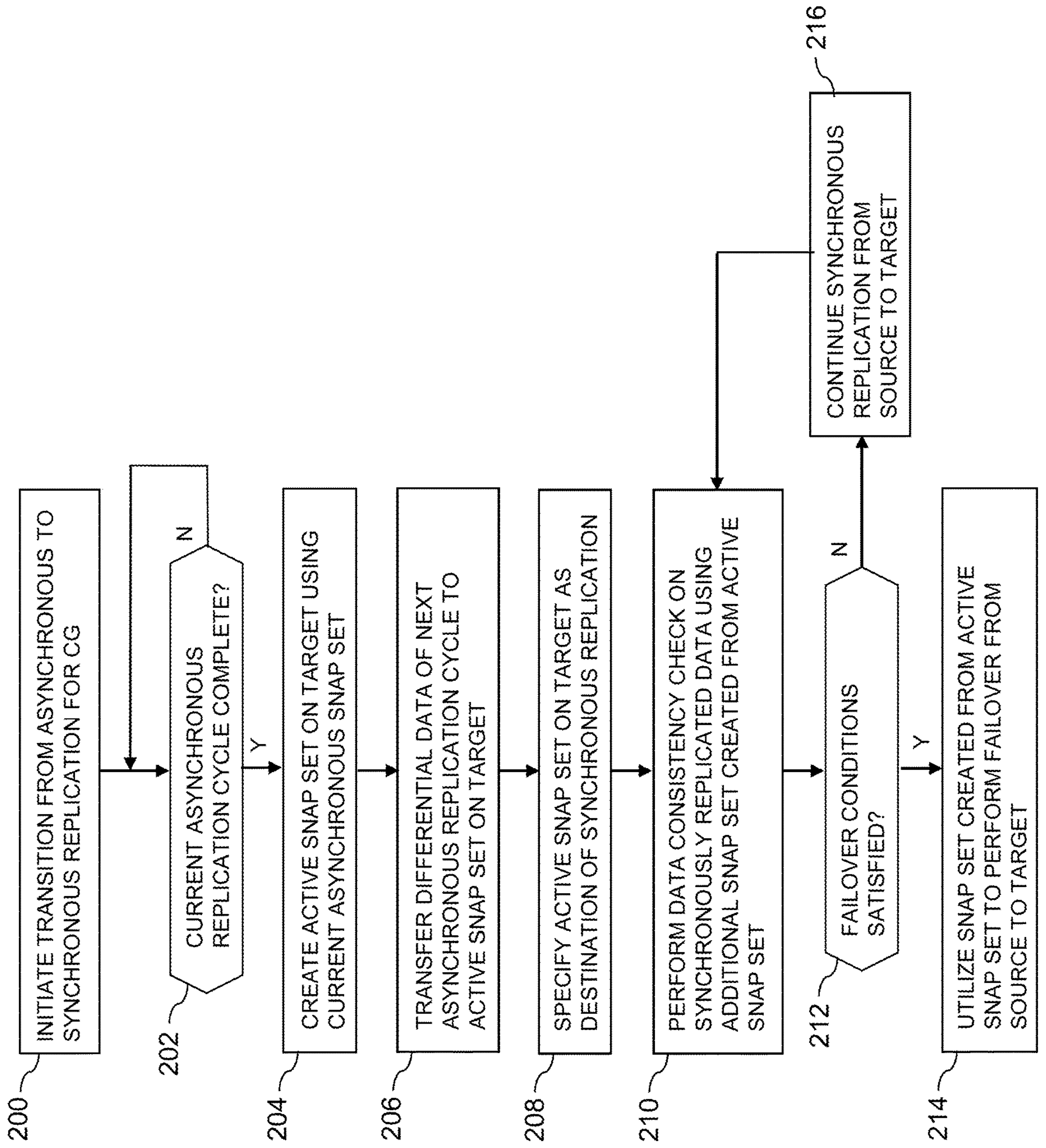

200
INITIATE TRANSITION FROM ASYNCHRONOUS TO SYNCHRONOUS REPLICATION FOR CG
202
CURRENT ASYNCHRONOUS REPLICATION CYCLE COMPLETE?
N
Y
204
CREATE ACTIVE SNAP SET ON TARGET USING CURRENT ASYNCHRONOUS SNAP SET
206
TRANSFER DIFFERENTIAL DATA OF NEXT ASYNCHRONOUS REPLICATION CYCLE TO ACTIVE SNAP SET ON TARGET
208
SPECIFY ACTIVE SNAP SET ON TARGET AS DESTINATION OF SYNCHRONOUS REPLICATION
210
PERFORM DATA CONSISTENCY CHECK ON SYNCHRONOUSLY REPLICATED DATA USING ADDITIONAL SNAP SET CREATED FROM ACTIVE SNAP SET
212
FAILOVER CONDITIONS SATISFIED?
N
Y
214
UTILIZE SNAP SET CREATED FROM ACTIVE SNAP SET TO PERFORM FAILOVER FROM SOURCE TO TARGET
216
CONTINUE SYNCHRONOUS REPLICATION FROM SOURCE TO TARGET

FIG. 2

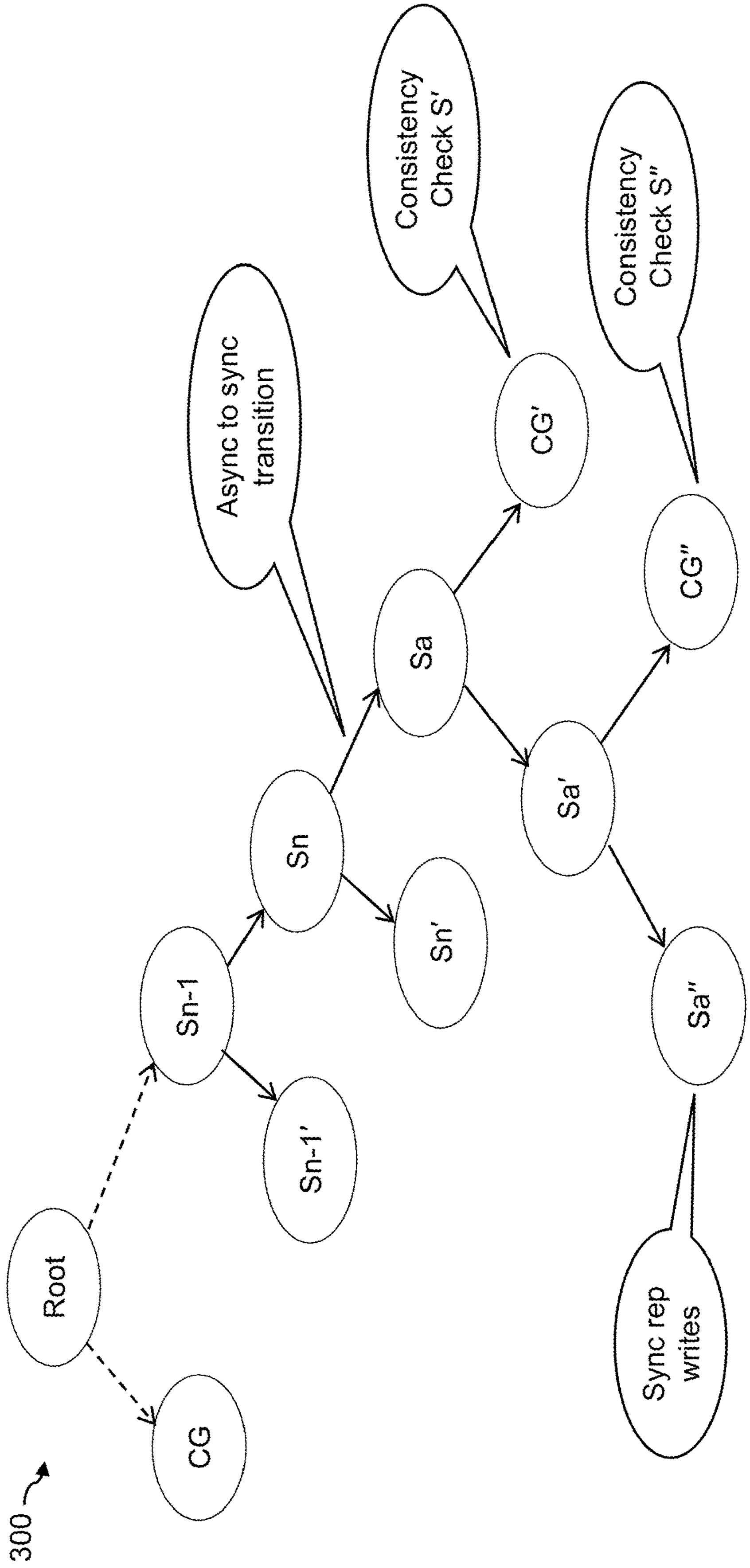

300
Root
CG
Sn-1
Sn-1'
Sn
Sn'
Async to sync transition
Sa
Sa'
Sa''
Sync rep writes
CG'
Consistency Check S'
CG''
Consistency Check S''

FIG. 3

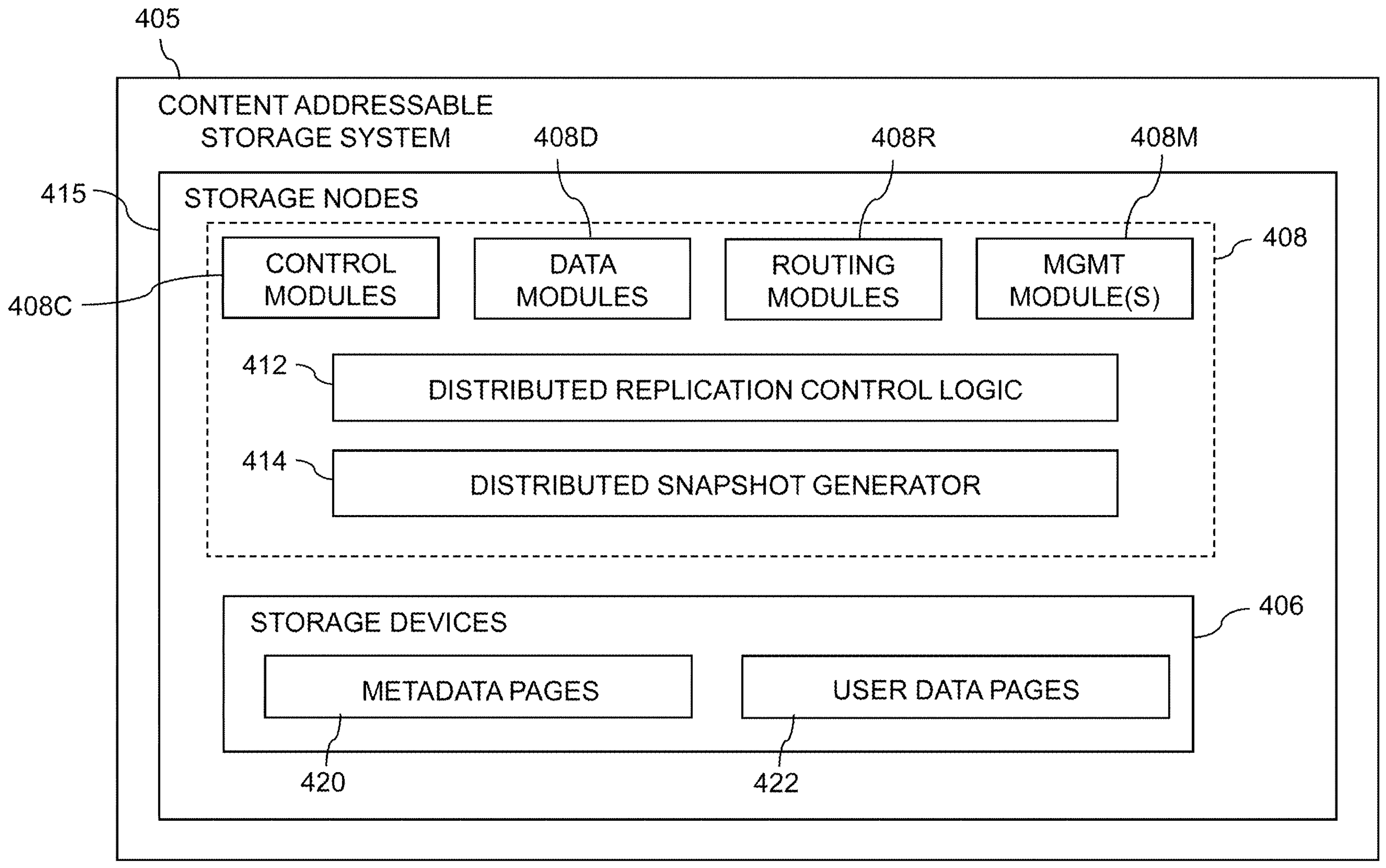

405
CONTENT ADDRESSABLE STORAGE SYSTEM
415
STORAGE NODES
408D
408R
408M
408
408C
CONTROL MODULES
DATA MODULES
ROUTING MODULES
MGMT MODULE(S)
412
DISTRIBUTED REPLICATION CONTROL LOGIC
414
DISTRIBUTED SNAPSHOT GENERATOR
406
STORAGE DEVICES
METADATA PAGES
USER DATA PAGES
420
422

FIG. 4

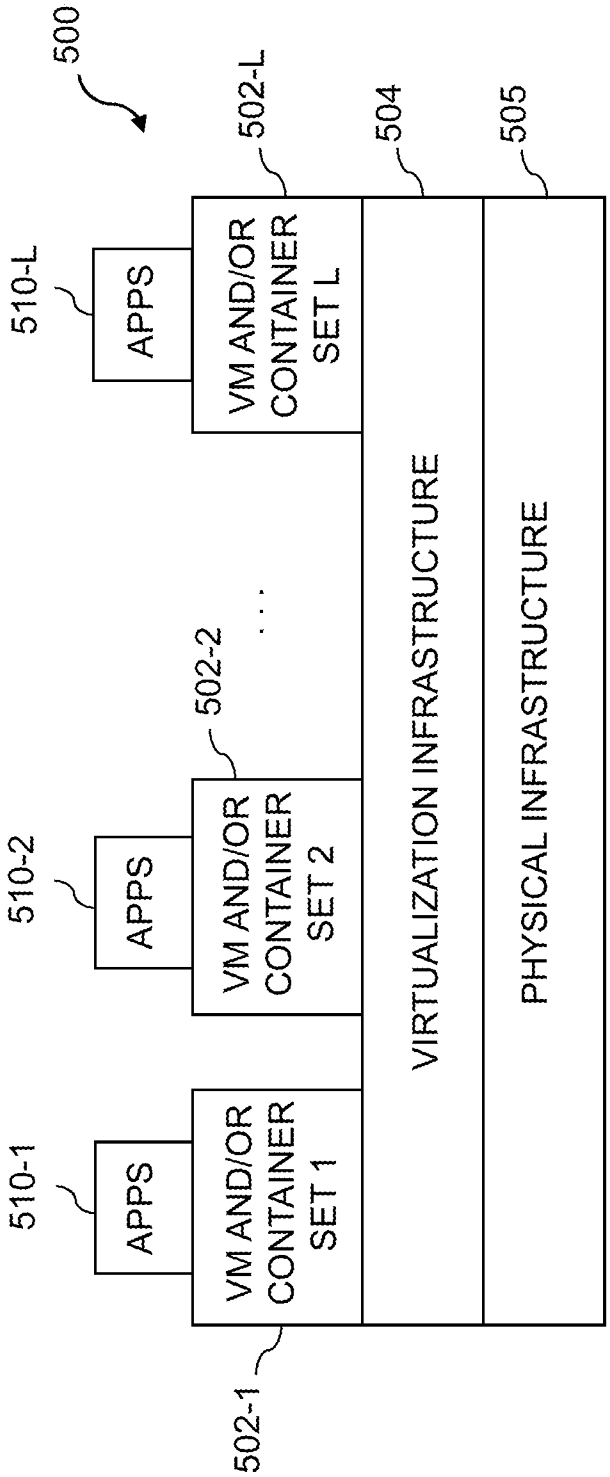

500
510-1
APPS
VM AND/OR CONTAINER SET 1
502-1
510-2
APPS
VM AND/OR CONTAINER SET 2
502-2
. . .
510-L
APPS
VM AND/OR CONTAINER SET L
502-L
VIRTUALIZATION INFRASTRUCTURE
504
PHYSICAL INFRASTRUCTURE
505

FIG. 5

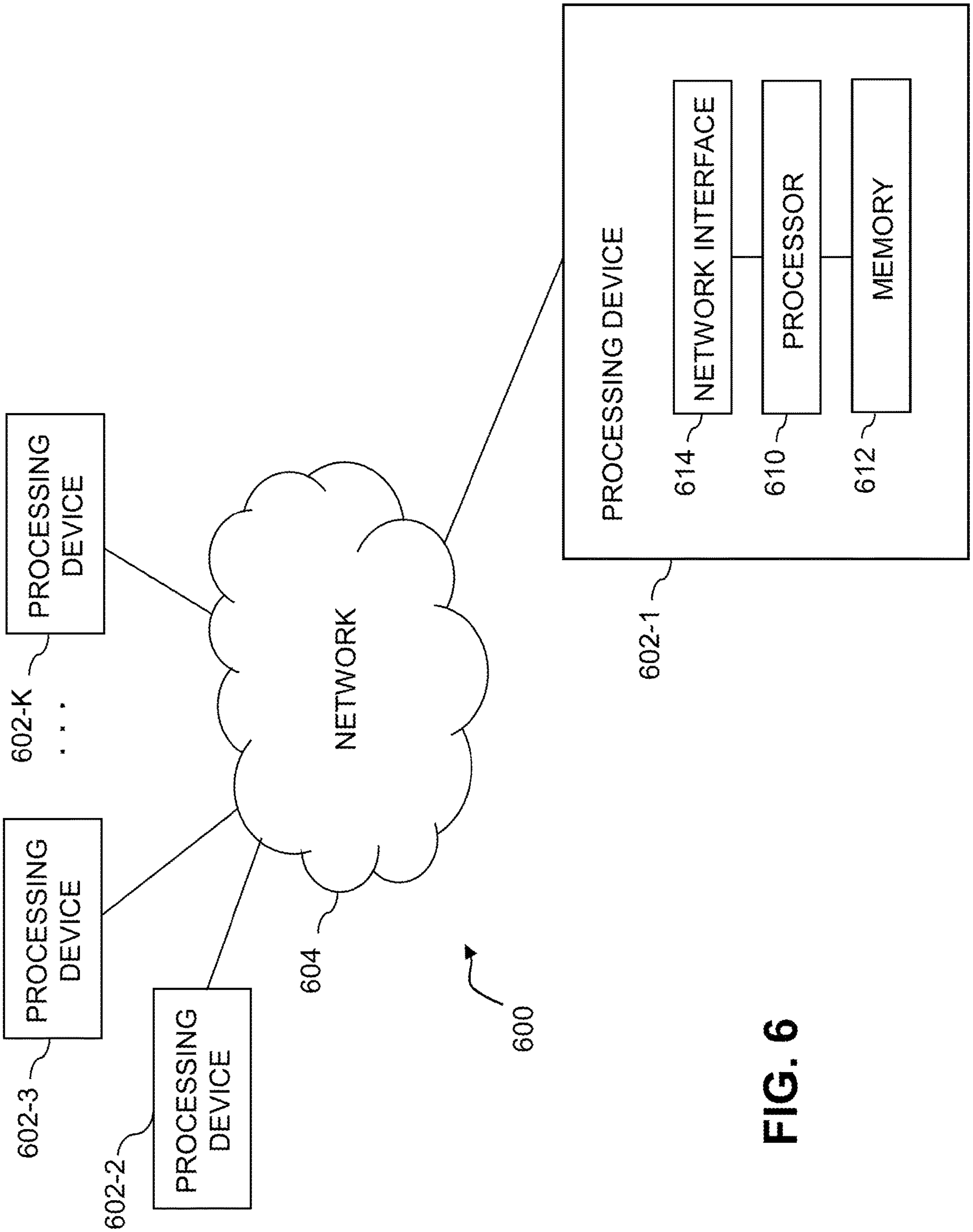

PROCESSING DEVICE
NETWORK INTERFACE
PROCESSOR
MEMORY
614
610
612
602-1
NETWORK
604
PROCESSING DEVICE
602-K
. . .
PROCESSING DEVICE
602-3
PROCESSING DEVICE
602-2
600

FIG. 6

STORAGE SYSTEM WITH DATA CONSISTENCY CHECKING IN SYNCHRONOUS REPLICATION USING ACTIVE SNAPSHOT SET

FIELD

[0001] The field relates generally to information processing systems, and more particularly to storage in information processing systems.

BACKGROUND

[0002] Many information processing systems are configured to replicate data from one storage system to another storage system, possibly at different physical sites. In some cases, such arrangements are utilized to support disaster recovery functionality within the information processing system. For example, an enterprise may replicate data from a production data center to a disaster recovery data center. In the event of a disaster at the production site, applications can be started at the disaster recovery site using the data that has been replicated to that site so that the enterprise can continue its business.

[0003] Data replication in these and other contexts can be implemented using asynchronous replication at certain times and synchronous replication at other times. For example, asynchronous replication may be configured to periodically transfer data in multiple cycles from a source site to a target site, while synchronous replication may be configured to mirror host writes from the source site to the target site as the writes are made at the source site. Storage systems participating in a replication process can therefore each be configured to support both asynchronous and synchronous replication modes.

[0004] Conventional approaches to data replication can be problematic under certain conditions. For example, it can be difficult to perform a data consistency check for one or more storage volumes of a consistency group subject to synchronous replication.

SUMMARY

[0005] Illustrative embodiments provide storage systems configured to perform data consistency checking in synchronous replication using an active snapshot set. The data consistency check is performed in a substantially simplified manner relative to conventional approaches. For example, some embodiments provide simplified data consistency checking that avoids the need to repeatedly create host access logical unit (LUN) mappings for all storage volumes in a snapshot set of a given replication consistency group each time a data consistency check is performed on synchronously replicated data. These and other embodiments provide a streamlined approach to data consistency checking for synchronously replicated data that advantageously avoids unduly complex management orchestration, host device setup, and manual user intervention in the data consistency checking process.

[0006] The source and target storage systems are illustratively implemented as respective content addressable storage systems, although other types of storage systems can be used in other embodiments.

[0007] In one embodiment, an apparatus comprises at least one processing device comprising a processor coupled to a memory. The at least one processing device is configured to define a consistency group comprising a plurality of storage volumes subject to replication from a source storage system to a target storage system. In conjunction with a transition from asynchronous replication of the consistency group to synchronous replication of the consistency group, the apparatus completes a current asynchronous replication cycle, utilizes a current asynchronous snapshot set on the target storage system to establish an active snapshot set on the target storage system, transfers differential data of a subsequent asynchronous replication cycle to the active snapshot set on the target storage system, specifies the active snapshot set on the target storage system as a destination of the synchronous replication, and performs a data consistency check on the target storage system utilizing an additional snapshot set generated from the active snapshot set.

[0008] In some embodiments, performing a data consistency check on the target storage system utilizing an additional snapshot set generated from the active snapshot set comprises generating the additional snapshot set from the active snapshot set, refreshing the consistency group on the target storage system to the additional snapshot set, and performing the data consistency check utilizing the refreshed consistency group. Refreshing the consistency group on the target storage system to the additional snapshot set illustratively comprises performing a snap-and-reassign operation for the consistency group utilizing the additional snapshot set.

[0009] Illustrative embodiments may be further configured to perform a replication failover operation for the synchronous replication. For example, performing the replication failover operation in some embodiments comprises terminating a current session of synchronous replication from the source storage system to the target storage system, generating a failover snapshot set from the active snapshot set, refreshing the consistency group on the target storage system to the failover snapshot set, and initiating a new session of synchronous replication from the target storage system to the source storage system.

[0010] The processing device in some embodiments is part of at least one of the source storage system and the target storage system, and more particularly implements at least a portion of a storage controller of at least one of the source and target storage systems, although numerous alternative implementations are possible. For example, in other embodiments the processing device is implemented at least in part in a host device configured to communicate over a network with the source and target storage systems. Again, these are only examples, and alternative implementations are possible.

[0011] A given one of the snapshot sets illustratively comprises a plurality of snapshot trees each associated with a different one of the storage volumes of the consistency group. A given one of the storage volumes illustratively comprises one or more logical storage volumes each comprising at least a portion of a physical storage space of one or more storage devices. The term "storage volume" as used herein is therefore intended to be broadly construed, so as to encompass a set of one or more logical storage volumes.

[0012] The source storage system in some embodiments comprises a clustered implementation of a content addressable storage system having a distributed storage controller. The content addressable storage system in arrangements of this type is illustratively configured to utilize non-volatile memory storage devices, such as flash-based storage devices. For example, the storage devices of the source storage system in such embodiments can be configured to collectively provide an all-flash storage array. The target storage system can similarly comprise an all-flash storage array, or another type of content addressable storage system. Numerous other storage system arrangements are possible in other embodiments. Content addressable storage is therefore not required.

[0013] These and other illustrative embodiments include, without limitation, apparatus, systems, methods and processor-readable storage media.

BRIEF DESCRIPTION OF THE DRAWINGS

[0014] FIG. **1** is a block diagram of an information processing system comprising source and target storage systems configured to perform data consistency checking in synchronous replication using an active snapshot set in an illustrative embodiment.

[0015] FIG. **2** is a flow diagram of a process for data consistency checking in synchronous replication using an active snapshot set in an illustrative embodiment.

[0016] FIG. **3** illustrates a snapshot tree data structure of a target storage system in an illustrative embodiment.

[0017] FIG. **4** shows a content addressable storage system having a distributed storage controller configured to perform data consistency checking in synchronous replication using an active snapshot set in an illustrative embodiment.

[0018] FIGS. **5** and **6** show examples of processing platforms that may be utilized to implement at least a portion of an information processing system in illustrative embodiments.

DETAILED DESCRIPTION

[0019] Illustrative embodiments will be described herein with reference to exemplary information processing systems and associated computers, servers, storage devices and other processing devices. It is to be appreciated, however, that these and other embodiments are not restricted to the particular illustrative system and device configurations shown. Accordingly, the term "information processing system" as used herein is intended to be broadly construed, so as to encompass, for example, processing systems comprising cloud computing and storage systems, as well as other types of processing systems comprising various combinations of physical and virtual processing resources. An information processing system may therefore comprise, for example, at least one data center or other cloud-based system that includes one or more clouds hosting multiple tenants that share cloud resources. Numerous different types of enterprise computing and storage systems are also encompassed by the term "information processing system" as that term is broadly used herein.

[0020] FIG. **1** shows an information processing system **100** configured in accordance with an illustrative embodiment. The information processing system **100** comprises a plurality of host devices **101**, a source storage system **102**S and a target storage system **102**T, all of which are configured to communicate with one another over a network **104**. The source and target storage systems **102** are more particularly configured in this embodiment to participate in replication process in which one or more storage volumes are replicated from the source storage system **102**S to the target storage system **102**T, possibly with involvement of at least one of the host devices **101**. The replication process illustratively comprises both asynchronous and synchronous replication modes. The one or more storage volumes that are replicated from the source storage system **102**S to the target storage system **102**T are illustratively part of a designated consistency group.

[0021] Each of the storage systems **102** is illustratively associated with a corresponding set of one or more of the host devices **101**. The host devices **101** illustratively comprise servers or other types of computers of an enterprise computer system, cloud-based computer system or other arrangement of multiple compute nodes associated with respective users.

[0022] The host devices **101** in some embodiments illustratively provide compute services such as execution of one or more applications on behalf of each of one or more users associated with respective ones of the host devices. Such applications illustratively generate input-output (IO) operations that are processed by a corresponding one of the storage systems **102**. The term "input-output" as used herein refers to at least one of input and output. For example, IO operations may comprise write requests and/or read requests directed to stored data of a given one of the storage systems **102**.

[0023] The storage systems **102** illustratively comprise respective processing devices of one or more processing platforms. For example, the storage systems **102** can each comprise one or more processing devices each having a processor and a memory, possibly implementing virtual machines and/or containers, although numerous other configurations are possible.

[0024] The storage systems **102** can additionally or alternatively be part of cloud infrastructure such as an Amazon Web Services (AWS) system. Other examples of cloud-based systems that can be used to provide at least portions of the storage systems **102** include Google Cloud Platform (GCP) and Microsoft Azure.

[0025] The storage systems **102** may be implemented on a common processing platform, or on separate processing platforms.

[0026] The host devices **101** are illustratively configured to write data to and read data from the storage systems **102** in accordance with applications executing on those host devices for system users.

[0027] The term "user" herein is intended to be broadly construed so as to encompass numerous arrangements of human, hardware, software or firmware entities, as well as combinations of such entities. Compute and/or storage services may be provided for users under a Platform-as-a-Service (PaaS) model, an Infrastructure-as-a-Service (IaaS) model and/or a Function-as-a-Service (FaaS) model, although it is to be appreciated that numerous other cloud infrastructure arrangements could be used. Also, illustrative embodiments can be implemented outside of the cloud infrastructure context, as in the case of a stand-alone computing and storage system implemented within a given enterprise.

[0028] The network **104** is assumed to comprise a portion of a global computer network such as the Internet, although other types of networks can be part of the network **104**, including a wide area network (WAN), a local area network (LAN), a satellite network, a telephone or cable network, a cellular network, a wireless network such as a WiFi or WiMAX network, or various portions or combinations of these and other types of networks. The network **104** in some embodiments therefore comprises combinations of multiple different types of networks each comprising processing devices configured to communicate using Internet Protocol (IP) or other communication protocols.

[0029] As a more particular example, some embodiments may utilize one or more high-speed local networks in which associated processing devices communicate with one another utilizing Peripheral Component Interconnect express (PCIe) cards of those devices, and networking protocols such as InfiniBand, Gigabit Ethernet or Fibre Channel. Numerous alternative networking arrangements are possible in a given embodiment, as will be appreciated by those skilled in the art.

[0030] The source storage system **102**S comprises a plurality of storage devices **106**S and an associated storage controller **108**S. The storage devices **106**S store storage volumes **110**S. The storage volumes **110**S illustratively comprise respective logical units (LUNs) or other types of logical storage volumes.

[0031] Similarly, the target storage system **102**T comprises a plurality of storage devices **106**T and an associated storage controller **108**T. The storage devices **106**T store storage volumes **110**T, at least a portion of which represent respective LUNs or other types of logical storage volumes that are replicated from the source storage system **102**S to the target storage system **102**T in accordance with a replication process.

[0032] The storage devices **106** of the storage systems **102** illustratively comprise solid state drives (SSDs). Such SSDs are implemented using non-volatile memory (NVM) devices such as flash memory. Other types of NVM devices that can be used to implement at least a portion of the storage devices **106** include non-volatile random access memory (NVRAM), phase-change RAM (PC-RAM) and magnetic RAM (MRAM). These and various combinations of multiple different types of NVM devices may also be used. For example, hard disk drives (HDDs) can be used in combination with or in place of SSDs or other types of NVM devices.

[0033] However, it is to be appreciated that other types of storage devices can be used in other embodiments. For example, a given storage system as the term is broadly used herein can include a combination of different types of storage devices, as in the case of a multi-tier storage system comprising a flash-based fast tier and a disk-based capacity tier. In such an embodiment, each of the fast tier and the capacity tier of the multi-tier storage system comprises a plurality of storage devices with different types of storage devices being used in different ones of the storage tiers. For example, the fast tier may comprise flash drives while the capacity tier comprises hard disk drives. The particular storage devices used in a given storage tier may be varied in other embodiments, and multiple distinct storage device types may be used within a single storage tier. The term "storage device" as used herein is intended to be broadly construed, so as to encompass, for example, SSDs, HDDs, flash drives, hybrid drives or other types of storage devices.

[0034] In some embodiments, at least one of the storage systems **102** illustratively comprises a scale-out all-flash content addressable storage array such as an XtremIO™ storage array from Dell EMC of Hopkinton, Mass. Other types of storage arrays, including by way of example VNX® and Symmetrix VMAX® storage arrays also from Dell EMC, can be used to implement storage systems **102** in other embodiments.

[0035] The term "storage system" as used herein is therefore intended to be broadly construed, and should not be viewed as being limited to content addressable storage systems or flash-based storage systems. A given storage system as the term is broadly used herein can comprise, for example, network-attached storage (NAS), storage area networks (SANs), direct-attached storage (DAS) and distributed DAS, as well as combinations of these and other storage types, including software-defined storage.

[0036] Other particular types of storage products that can be used in implementing storage systems **102** in illustrative embodiments include all-flash and hybrid flash storage arrays such as Unity™, software-defined storage products such as ScaleIO™ and ViPR®, cloud storage products such as Elastic Cloud Storage (ECS), object-based storage products such as Atmos®, and scale-out NAS clusters comprising Isilon® platform nodes and associated accelerators, all from Dell EMC. Combinations of multiple ones of these and other storage products can also be used in implementing a given storage system in an illustrative embodiment.

[0037] In some embodiments, communications between the host devices **101** and the storage systems **102** comprise Small Computer System Interface (SCSI) commands. Other types of SCSI or non-SCSI commands may be used in other embodiments, including commands that are part of a standard command set, or custom commands such as a "vendor unique command" or VU command that is not part of a standard command set. The term "command" as used herein is therefore intended to be broadly construed, so as to encompass, for example, a composite command that comprises a combination of multiple individual commands. Numerous other commands can be used in other embodiments.

[0038] The storage controller **108**S of source storage system **102**S in the FIG. **1** embodiment includes replication control logic **112**S and a snapshot generator **114**S.

[0039] Similarly, the storage controller **108**T of target storage system **102**T includes replication control logic **112**T and a snapshot generator **114**T.

[0040] Although not explicitly shown in the figure, additional components can be included in the storage controllers **108**, such as signature generators utilized in generating content-based signatures of data pages.

[0041] The instances of replication control logic **112**S and **112**T are collectively referred to herein as replication control logic **112**. Such replication control logic instances are also referred to herein as individually or collectively comprising at least a portion of a "replication engine" of the system **100**.

[0042] The replication control logic **112** of the storage systems **102** controls performance of the replication process carried out between those storage systems, which as noted above in some embodiments further involves at least one of the host devices **101**. The data replicated from the source storage system **102**S to the target storage system **102**T can include all of the data stored in the source storage system **102**S, or only certain designated subsets of the data stored in the source storage system **102**S, such as particular designated sets of LUNs or other logical storage volumes. Different replication processes of different types can be implemented for different parts of the stored data. Also, the storage systems **102** can be configured to operate in different replication modes of different types at different times. For example, as described in more detail below, the storage systems **102** can transition from an asynchronous replication mode to a synchronous replication mode and vice versa.

[0043] A given storage volume designated for replication from the source storage system **102**S to the target storage system **102**T illustratively comprises a set of one or more LUNs or other instances of the storage volumes **110**S of the source storage system **102**S. Each such LUN or other storage volume illustratively comprises at least a portion of a physical storage space of one or more of the storage devices **106**S. The corresponding replicated LUN or other storage volume of the storage volumes **110**T of the target storage system **102**T illustratively comprises at least a portion of a physical storage space of one or more of the storage devices **106**T.

[0044] The replication control logic **112** of the storage systems **102** in some embodiments is configured to control the performance of corresponding portions of a replication process of the type illustrated in the flow diagram of FIG. **2**. At least one of the host devices **101** in some embodiments can also include one or more instances of replication control logic and possibly also one or more snapshot generators, as well as additional or alternative components such as a signature generator.

[0045] The storage controllers **108** of the storage systems **102** should also be understood to include additional modules and other components typically found in conventional implementations of storage controllers and storage systems, although such additional modules and other components are omitted from the figure for clarity and simplicity of illustration.

[0046] It will be assumed for the following description of the FIG. **1** embodiment that there is an ongoing asynchronous replication process being carried out between the source storage system **102**S and the target storage system **102**T in the system **100**, utilizing their respective instances of replication control logic **112**S and **112**T.

[0047] The asynchronous replication process more particularly comprises a cycle-based asynchronous replication process in which a consistency group comprising one or more storage volumes is replicated from the source storage system **102**S to the target storage system **102**T over a plurality of asynchronous replication cycles. Such an arrangement is illustratively configured to guarantee data consistency between the storage volumes of the consistency group on the source and their corresponding replicated versions on the target. The asynchronous replication is performed periodically over the multiple cycles. The asynchronous replication is illustratively implemented at least in part by or otherwise under the control of the source and target instances of replication control logic **112**S and **112**T. Other types of replication arrangements can be used in other embodiments.

[0048] The source storage system **102**S illustratively initiates a transition from the above-described asynchronous replication of the consistency group to synchronous replication of the consistency group. In conjunction with the transition from asynchronous replication of the consistency group to the synchronous replication of the consistency group, the source and target storage systems **102** complete a current asynchronous replication cycle. A current snapshot set previously generated on the target storage system **102**T as part of the asynchronous replication is utilized to create or otherwise establish what is generally referred to herein as an "active snapshot set" on the target storage system **102**T. Like the current snapshot set from which it is generated, the active snapshot set comprises a plurality of snapshot trees for respective storage volumes of the consistency group, but is more particularly utilized to permit highly efficient consistency checking of data synchronously replicated from the source storage system **102**S to the target storage system **102**T.

[0049] The source storage system **102**S transfers differential data of a subsequent asynchronous replication cycle to the active snapshot set on the target storage system **102**T, and specifies the active snapshot set on the target storage system **102**T as a destination of the synchronous replication. At this point, the synchronous replication of the consistency group commences, and the asynchronous replication is terminated. The transition from the asynchronous replication the synchronous replication is therefore complete, and the replication process continues with the synchronous replication.

[0050] One or more data consistency checks are then performed on the synchronously replicated data of the consistency group. For example, such data consistency checks can be performed periodically as the synchronous replication progresses, possibly at predetermined points in time. Additionally or alternatively, one or more data consistency checks can be performed on demand, possibly responsive to a user command entered by a storage administrator or other user of the system **100**.

[0051] A given such data consistency check is illustratively performed on the target storage system **102**T utilizing an additional snapshot set generated from the active snapshot set. Again, each such snapshot set comprises a plurality of snapshot trees for respective storage volumes of the consistency group. A phrase such as "performing a data consistency check on the target storage system" or other similar language as used herein is intended to be broadly construed so as encompass a wide variety of different arrangements in which replicated data on the target storage system **102**T is checked for consistency with corresponding data in other parts of the system **100**, and a given such data consistency check illustratively involves not only target storage system **102**T but possibly also source storage system **102**S and/or one or more of the host devices **101** of the system **100**.

[0052] In some embodiments, transferring differential data of a subsequent asynchronous replication cycle to the active snapshot set on the target storage system **102**T comprises transferring differential data representing a differential between a previous asynchronous snapshot set of the consistency group on the source storage system **102**S and a current asynchronous snapshot set on the source storage system **102**S.

[0053] The current asynchronous snapshot set on the source storage system **102**S illustratively corresponds to the current asynchronous snapshot set on the target storage system **102**T that was utilized to establish the active snapshot set. The term "asynchronous snapshot set" as broadly used herein refers to a snapshot set that is generated for a consistency group as part of or otherwise in conjunction with asynchronous replication.

[0054] In some embodiments, the active snapshot set on the target storage system is specified as a destination of the synchronous replication responsive to the synchronous replication not being in an active-active configuration. Such an active-active configuration is sometimes referred to as a “metro” configuration, and implements synchronous replication with the source and target storage systems **102** having an active-active relationship. If the source and target storage systems **102** are in an active-active configuration for performance of the synchronous replication, the consistency group on the target storage system **102**T is illustratively specified as the destination of the synchronous replication. Accordingly, some embodiments disclosed herein support synchronous replication in an active-active configuration.

[0055] The data consistency check is illustratively performed on synchronously replicated data of the target storage system **102**T by generating the additional snapshot set from the active snapshot set, refreshing the consistency group on the target storage system **102**T to the additional snapshot set, and performing the data consistency check utilizing the refreshed consistency group. Refreshing the consistency group on the target storage system **102**T to the additional snapshot set illustratively comprises performing a snap-and-reassign operation for the consistency group utilizing the additional snapshot set.

[0056] For example, in a given snap-and-reassign operation, target consistency group storage volumes are refreshed to the snapshot set to be checked. Comparison data can then be read from the consistency group storage volumes. Such an arrangement allows a single set of host access LUN mappings to be used to perform multiple data consistency checks on synchronously replicated data at different times. More particularly, host access LUN mappings of a target consistency group can be set up only once at replication session creation time, and can be reused for multiple data consistency checks in that replication session no matter how many times such data consistency checks are performed.

[0057] Some embodiments are additionally or alternatively configured to perform a replication failover operation for the synchronous replication. For example, such a replication failover operation can be performed by terminating a current session of synchronous replication from the source storage system **102**S to the target storage system **102**T, generating a failover snapshot set from the active snapshot set, refreshing the consistency group on the target storage system to the failover snapshot set, and initiating a new session of synchronous replication from the target storage system **102**T to the source storage system **102**S.

[0058] As mentioned previously, the active snapshot set and other snapshot sets referred to herein each illustratively comprise a plurality of snapshot trees corresponding to respective ones of the storage volumes of the consistency group.

[0059] A given one of the snapshot trees corresponding to a particular one of the storage volumes more particularly comprises a root node, at least one branch node, and a plurality of leaf nodes, with a given one of the branch nodes representing a particular version of the storage volume from which a corresponding snapshot is taken. A first one of the leaf nodes which is a child of the given branch node represents a subsequent version of the storage volume, and a second one of the leaf nodes which is a child of the given branch node comprises the corresponding snapshot providing a point-in-time (PIT) copy of the particular version of the storage volume.

[0060] Accordingly, a snapshot tree illustratively provides a representation of a storage volume and its snapshots over time. Each leaf node represents a particular version of the storage volume or a snapshot of the storage volume, and each branch node represents a shared ancestor between a version of the storage volume, a snapshot of the storage volume, or a child branch node. When a given snapshot of the storage volume is created, two child leaf nodes are created, one representing new updates to the storage volume after creation of the snapshot, and the other representing the snapshot. The volume node from which the snapshot was created therefore becomes a branch node in the snapshot tree. When a given snap set of the consistency group is created for its member storage volumes, two new leaf nodes are created in each of the snapshot trees of the respective storage volumes.

[0061] An example of a particular snapshot tree data structure characterizing multiple snapshot trees is shown in FIG. **3**, and will be described in greater detail below in conjunction with the flow diagram of FIG. **2**.

[0062] In some embodiments, the snapshot trees comprise or are otherwise associated with additional information also arranged in the form of a tree structure. For example, a given one of the snapshot trees may be associated with one or more additional trees including at least one of a “dirty” tree that characterizes updates to logical addresses of the corresponding storage volume, and a hash tree comprising content-based signatures of respective ones of the logical addresses of the corresponding storage volume. A particular instance of a differential scan performed for the given snapshot tree in embodiments of this type can illustratively comprise aggregating information of at least one of the dirty tree and the hash tree between designated start and stop nodes of the given snapshot tree.

[0063] A wide variety of other types of snapshot trees and possibly one or more associated additional trees can be used in other embodiments. Also, the term “tree” as used herein is intended to be broadly construed so as to comprise any type of data structure characterizing a plurality of nodes and a plurality of edges interconnecting respective pairs of the nodes.

[0064] The content-based signatures of the above-noted hash tree associated with a given storage volume in some embodiments comprise hash digests of their respective pages, each generated by application of a hash function such as the well-known Secure Hashing Algorithm 1 (SHA1) to the content of its corresponding page. Other types of secure hashing algorithms, such as SHA2 or SHA256, or more generally other hash functions, can be used in generating content-based signatures herein.

[0065] A given hash digest in illustrative embodiments is unique to the particular content of the page from which it is generated, such that two pages with exactly the same content will have the same hash digest, while two pages with different content will have different hash digests. It is also possible that other types of content-based signatures may be used, such as hash handles of the type described elsewhere herein. A hash handle generally provides a shortened representation of its corresponding hash digest. More particularly, the hash handles are shorter in length than respective hash digests that are generated by applying a secure hashing algorithm to respective ones of the data pages. Hash handles are considered examples of “content-based signatures” as that term is broadly used herein.

[0066] In embodiments in which the storage systems **102** comprise content addressable storage systems, address metadata is illustratively utilized to provide content addressable storage functionality within those systems. The address metadata in some embodiments comprises at least a portion of one or more logical layer mapping tables that map logical addresses of respective ones of the data pages of the storage volume to corresponding content-based signatures of the respective data pages. Examples of logical layer mapping tables and other metadata structures maintained by at least the storage controller **108**T of target storage system **102**T will be described elsewhere herein.

[0067] As mentioned previously, the term “storage volume” as used herein is intended to be broadly construed, and should not be viewed as being limited to any particular format or configuration of LUNs or other logical volumes. The term “consistency group” as used herein is also intended to be broadly construed, and illustratively comprises a plurality of storage volumes subject to a replication process.

[0068] A more particular example of the data consistency checking functionality described above will now be presented. In this example, the replication control logic instances **112**S and **112**T are assumed to cooperate to facilitate data consistency checking using an active snapshot set in conjunction with ongoing synchronous replication of a consistency group from the source to the target. Like other embodiments herein, the process utilized in the present example advantageously allows consistency checking to be performed for one or more storage volumes in a particularly efficient manner as part of an ongoing synchronous replication process.

[0069] The process in the present example is carried out by the source storage system **102**S (“source”) in cooperation with the target storage system **102**T (“target”) for a particular consistency group (CG) and includes the following steps:

[0070] 1. Transition from asynchronous replication to synchronous replication:

- **[0071]** (a) Wait for completion of current asynchronous replication cycle n.
- **[0072]** (b) Create active snapshot set (“snap set”) based on existing asynchronous snap set Sn' on target.
- **[0073]** (c) Start transition to synchronous replication by transferring differential data (“delta”) between CG and snap set Sn from source to target active snap set Sa.

[0074] 2. Establish synchronous replication IO flow:

- **[0075]** (a) If the IO is read from source CG and the replication mode is synchronous, specify the IO destination as the target active snap set Sa.
- **[0076]** (b) If the IO is read from source CG and the replication mode is metro, specify IO destination as the target CG.

[0077] 3. Perform data consistency check:

- **[0078]** (a) Create a snap set S' based on the active snap set Sa. S' represents the latest content of Sa at the time of the data consistency check.
- **[0079]** (b) Refresh the target CG to the new snap set S'.
- **[0080]** (c) Check the target CG data content.

[0081] 4. Perform replication failover if needed:

- **[0082]** (a) Terminate replication session and stop synchronous replication IO.
- **[0083]** (b) Create a snap set S" based on the active snap set Sa.
- **[0084]** (c) Refresh the target CG to the new snap set S".
- **[0085]** (d) Start new replication session with the target CG as the new source CG.

[0086] An illustration of the snap sets of the present example is shown in FIG. **3** and will be described in more detail below. As mentioned previously, the process steps in the above example are assumed to be performed by cooperative interaction of the source storage system **102**S and the target storage system **102**T. Such cooperative interaction illustratively occurs via a replication engine comprising their respective replication control logic instances **112**S and **112**T. Other arrangements of process steps can be used in other embodiments. Also, the particular ordering of the steps shown above can be varied.

[0087] The above-described illustrative embodiments are examples of what are referred to herein as “data consistency checking” performed in synchronous replication. Such arrangements illustratively provide particularly efficient data consistency checking in synchronous replication through the use of an active snapshot set on the target storage system **102**T. For example, such arrangements avoid drawbacks such as excessive repetition of LUN mapping creation, unduly complex management orchestration, host device setup, and manual user intervention in the data consistency checking process. As a result, data consistency checking in illustrative embodiments considerably facilitates efficient usability of such functionality for storage administrators and other system users.

[0088] As indicated previously, these and other operations carried out in conjunction with a process for data consistency checking in synchronous replication using an active snapshot set are illustratively performed at least in part under the control of the replication control logic **112**.

[0089] The storage systems **102** in the FIG. **1** embodiment are assumed to be implemented using at least one processing platform each comprising one or more processing devices each having a processor coupled to a memory. Such processing devices can illustratively include particular arrangements of compute, storage and network resources.

[0090] The storage systems **102** may be implemented on respective distinct processing platforms, although numerous other arrangements are possible. At least portions of their associated host devices may be implemented on the same processing platforms as the storage systems **102** or on separate processing platforms.

[0091] The term “processing platform” as used herein is intended to be broadly construed so as to encompass, by way of illustration and without limitation, multiple sets of processing devices and associated storage systems that are configured to communicate over one or more networks. For example, distributed implementations of the system **100** are possible, in which certain components of the system reside in one data center in a first geographic location while other components of the system reside in one or more other data centers in one or more other geographic locations that are potentially remote from the first geographic location. Thus, it is possible in some implementations of the system **100** for the storage systems **102** to reside in different data centers. Numerous other distributed implementations of the storage systems **102** and their respective associated sets of host devices are possible.

[0092] Additional examples of processing platforms utilized to implement storage systems and possibly their associated host devices in illustrative embodiments will be described in more detail below in conjunction with FIGS. **5** and **6**.

[0093] It is to be appreciated that these and other features of illustrative embodiments are presented by way of example only, and should not be construed as limiting in any way.

[0094] Accordingly, different numbers, types and arrangements of system components such as host devices **101**, storage systems **102**, network **104**, storage devices **106**, storage controllers **108** and storage volumes **110** can be used in other embodiments.

[0095] It should be understood that the particular sets of modules and other components implemented in the system **100** as illustrated in FIG. **1** are presented by way of example only. In other embodiments, only subsets of these components, or additional or alternative sets of components, may be used, and such components may exhibit alternative functionality and configurations.

[0096] For example, in other embodiments, at least portions of the above-described functionality for data consistency checking in synchronous replication using an active snapshot set can be implemented in one or more host devices, or partially in a host device and partially in a storage system. Illustrative embodiments are not limited to arrangements in which all such functionality is implemented in source and target storage systems or a host device, and therefore encompass various hybrid arrangements in which the functionality is distributed over one or more storage systems and one or more associated host devices, each comprising one or more processing devices. References herein to "one or more processing devices" configured to implement particular operations or other functionality should be understood to encompass a wide variety of different arrangements involving one or more processing devices of at least one storage system and/or at least one host device.

[0097] As another example, it is possible in some embodiments that the source storage system and the target storage system can comprise different portions of the same storage system. In such an arrangement, a replication process is illustratively implemented to replicate data from one portion of the storage system to another portion of the storage system. The terms "source storage system" and "target storage system" as used herein are therefore intended to be broadly construed so as to encompass such possibilities.

[0098] The operation of the information processing system **100** will now be described in further detail with reference to the flow diagram of the illustrative embodiment of FIG. **2**, which implements a replication process involving a transition from asynchronous replication to synchronous replication. The steps of the process illustratively involve interactions between a source storage system and a target storage system, referred to as respective "source" and "target" in these figures, illustratively utilizing replication control logic instances and snapshot generators of storage controllers of the source and target. For example, replication control logic of the source interacts with replication control logic of the target in performing multiple cycles of asynchronous replication for a consistency group, transitioning from asynchronous to synchronous replication, and performing synchronous replication. It is possible in other embodiments that at least one of the storage systems does not include replication control logic and a snapshot generator, and in such embodiments these components are instead implemented in one or more host devices.

[0099] The replication process as illustrated in FIG. **2** includes steps **200** through **216**, and is suitable for use in system **100** but is more generally applicable to other types of information processing systems in which data is replicated from source to target. Also, the roles of source and target can be reversed, as in a situation in which a failover from source to target occurs.

[0100] In step **200**, a transition from asynchronous replication to synchronous replication is initiated for a consistency group (CG) comprising multiple storage volumes. This assumes that the consistency group is currently subject to cycle-based asynchronous replication from the source to the target. The asynchronous replication illustratively utilizes a differential scan of a current snap set relative to a previous snap set generated for the consistency group. The differential scan is illustratively utilized to generate differential data, also referred to as representing a "delta" between the two snap sets, for communication from source to target in a given replication cycle of the cycle-based asynchronous replication process.

[0101] In step **202**, a determination is made as to whether or not the current asynchronous replication cycle is complete. If the current asynchronous replication cycle is complete, the process advances to step **204**, and otherwise waits until the current asynchronous replication cycle is complete.

[0102] In step **204**, an active snap set is created on the target utilizing a current asynchronous snap set.

[0103] In step **206**, differential data of the next asynchronous replication cycle is transferred to the active snap set on the target.

[0104] In step **208**, the active snap set on the target is specified as a destination of the synchronous replication, and the process then begins synchronous replication of the consistency group from the source to the target using the active snap set as the destination.

[0105] In step **210**, a consistency check is performed on synchronously replicated data using an additional snap set created from the active snap set.

[0106] In step **212**, a determination is made as to whether or not one or more specified failover conditions have been satisfied. If the one or more failover conditions have been satisfied, the process moves to step **214**, and otherwise moves to step **216** as shown.

[0107] In step **214**, a snap set created from the active snap set is utilized to perform failover from the source to the target. At this point, replication may be terminated and possibly reinitiated to replicate from target to source.

[0108] In step **216**, the synchronous replication from source to target continues, and periodically and/or on demand returns to step **210** to perform an additional consistency check on the synchronously replicated data.

[0109] An example of one possible implementation of the FIG. **2** process will now be described with reference to FIG. **3**, which shows a snap set tree **300** on the target. The snap set tree **300** illustratively characterizes multiple snapshot trees, one for each storage volume of the consistency group denoted CG, with each such snapshot tree having the same general tree structure of the snap set tree as illustrated. The snap set tree **300** comprises a root node and a plurality of branch nodes denoted Sn−1, Sn and Sa, where Sa denotes the previously-described active snap set on the target. The root node represents a version of the consistency group from which an initial PIT copy is captured as snap set CG. The branch nodes Sn−1 and Sn represent subsequent versions of the consistency group from which respective PIT copies are captured as respective subsequent snap sets Sn−1' and Sn'. The branch nodes Sn−1 and Sn are associated with respective previous and current asynchronous replication cycles denoted as cycle n−1 and cycle n. The snap set tree **300** shows the point of the asynchronous to synchronous ("async to sync") transition between asynchronous snap set Sn and target active snap set Sa. Synchronous replication ("sync rep") writes directed to the active snap set Sa produce additional nodes corresponding to snap sets Sa' and Sa" as shown.

[0110] In the FIG. **3** example, the snap sets CG' and CG" denote respective PIT copies of the respective snap sets Sa and Sa'. The snap set Sa' is an example of what is more generally referred to herein as an "additional snapshot set" generated from an active snapshot set, in this case from the active snap set Sa. The snap sets CG' and CG" are illustratively utilized in performing respective data consistency checks involving corresponding snap sets denoted S' and S" in the figure.

[0111] Other types of tree structures and arrangements of multiple snap sets can be used in other embodiments.

[0112] The particular processing operations and other system functionality described in conjunction with the flow diagram of FIG. **2** are presented by way of illustrative example only, and should not be construed as limiting the scope of the disclosure in any way. Alternative embodiments can use other types of processing operations to provide data consistency checking in synchronous replication using an active snapshot set. For example, the ordering of the process steps may be varied in other embodiments, or certain steps may be performed at least in part concurrently with one another rather than serially. Also, one or more of the process steps may be repeated periodically, or multiple instances of the process can be performed in parallel with one another in order to implement a plurality of different data consistency checking arrangements in synchronous replication processes for respective different consistency groups comprising different sets of storage volumes or for different storage systems or portions thereof within a given information processing system.

[0113] Functionality such as that described in conjunction with the flow diagram of FIG. **2** can be implemented at least in part in the form of one or more software programs stored in memory and executed by a processor of a processing device such as a computer or server. As will be described below, a memory or other storage device having executable program code of one or more software programs embodied therein is an example of what is more generally referred to herein as a "processor-readable storage medium."

[0114] For example, storage controllers such as storage controllers **108** of storage systems **102** that are configured to control performance of one or more steps of the FIG. **2** process in their corresponding system **100** can be implemented as part of what is more generally referred to herein as a processing platform comprising one or more processing devices each comprising a processor coupled to a memory. A given such processing device may correspond to one or more virtual machines or other types of virtualization infrastructure such as Docker containers or Linux containers (LXCs). The storage controllers **108**, as well as other system components, may be implemented at least in part using processing devices of such processing platforms. For example, in a distributed implementation of a given one of the storage controllers **108**, respective distributed modules of such a storage controller can be implemented in respective containers running on respective ones of the processing devices of a processing platform.

[0115] In some implementations of the FIG. **2** process, the source and target storage systems comprise content addressable storage systems configured to maintain various metadata structures that are utilized in the data consistency checking. Examples of metadata structures maintained by the source and target storage systems in illustrative embodiments include the logical layer and physical layer mapping tables described below. It is to be appreciated that these particular tables are only examples, and other tables or metadata structures having different configurations of entries and fields can be used in other embodiments.

[0116] An address-to-hash ("A2H") utilized in some embodiments comprises a plurality of entries accessible utilizing logical addresses as respective keys, with each such entry of the A2H table comprising a corresponding one of the logical addresses, a corresponding hash handle, and possibly one or more additional fields.

[0117] A hash-to-data ("H2D") table utilized in some embodiments comprises a plurality of entries accessible utilizing hash handles as respective keys, with each such entry of the H2D table comprising a corresponding one of the hash handles, a physical offset of a corresponding one of the data pages, and possibly one or more additional fields.

[0118] A hash metadata ("HMD") table utilized in some embodiments comprises a plurality of entries accessible utilizing hash handles as respective keys. Each such entry of the HMD table comprises a corresponding one of the hash handles, a corresponding reference count and a corresponding physical offset of one of the data pages. A given one of the reference counts denotes the number of logical pages in the storage system that have the same content as the corresponding data page and therefore point to that same data page via their common hash digest. The HMD table may also include one or more additional fields.

[0119] A physical layer based ("PLB") table utilized in some embodiments illustratively comprises a plurality of entries accessible utilizing physical offsets as respective keys, with each such entry of the PLB table comprising a corresponding one of the physical offsets, a corresponding one of the hash digests, and possibly one or more additional fields.

[0120] As indicated above, the hash handles are generally shorter in length than the corresponding hash digests of the respective data pages, and each illustratively provides a short representation of the corresponding full hash digest. For example, in some embodiments, the full hash digests are 20 bytes in length, and their respective corresponding hash handles are illustratively only 4 or 6 bytes in length.

[0121] Also, it is to be appreciated that terms such as "table" and "entry" as used herein are intended to be broadly construed, and the particular example table and entry arrangements described above can be varied in other embodiments. For example, additional or alternative arrangements of entries can be used.

[0122] In some embodiments, the storage system comprises an XtremIO™ storage array or other type of content addressable storage system suitably modified to incorporate functionality for data consistency checking in synchronous replication using an active snapshot set as disclosed herein.

[0123] An illustrative embodiment of such a content addressable storage system will now be described with reference to FIG. **4**. In this embodiment, a content addressable storage system **405** comprises a plurality of storage devices **406** and an associated storage controller **408**. The content addressable storage system **405** may be viewed as a particular implementation of a given one of the storage systems **102**, and accordingly is assumed to be coupled to the other one of the storage systems **102** and to one or more host devices of a computer system within information processing system **100**.

[0124] Although it is assumed that both the source storage system **102**S and the target storage system **102**T are content addressable storage systems in some embodiments, other types of storage systems can be used for one or both of the source storage system **102**S and the target storage system **102**T in other embodiments. For example, it is possible that at least one of the storage systems **102** in an illustrative embodiment need not be a content addressable storage system and need not include an ability to generate content-based signatures. In such an embodiment, at least portions of the data consistency checking functionality of the one or more storage systems can be implemented in a host device.

[0125] The storage controller **408** in the present embodiment is configured to implement data consistency checking in synchronous replication using an active snapshot set of the type previously described in conjunction with FIGS. **1** through **3**. For example, the content addressable storage system **405** illustratively participates as a source storage system in a synchronous replication process with a target storage system that may be implemented as another instance of the content addressable storage system **405**.

[0126] The storage controller **408** includes distributed modules **412** and **414**, which are configured to operate in a manner similar to that described above for respective corresponding replication control logic **112** and snapshot generators **114** of the storage controllers **108** of system **100**. Module **412** is more particularly referred to as distributed replication control logic, and illustratively comprises multiple replication control logic instances on respective ones of a plurality of distinct nodes. Module **414** is more particularly referred to as a distributed snapshot generator, and illustratively comprises multiple snapshot generation instances on respective ones of the distinct nodes.

[0127] The content addressable storage system **405** in the FIG. **4** embodiment is implemented as at least a portion of a clustered storage system and includes a plurality of storage nodes **415** each comprising a corresponding subset of the storage devices **406**. Such storage nodes **415** are examples of the "distinct nodes" referred to above, and other clustered storage system arrangements comprising multiple storage nodes and possibly additional or alternative nodes can be used in other embodiments. A given clustered storage system may therefore include not only storage nodes **415** but also additional storage nodes, compute nodes or other types of nodes coupled to network **104**. Alternatively, such additional storage nodes may be part of another clustered storage system of the system **100**. Each of the storage nodes **415** of the storage system **405** is assumed to be implemented using at least one processing device comprising a processor coupled to a memory.

[0128] The storage controller **408** of the content addressable storage system **405** is implemented in a distributed manner so as to comprise a plurality of distributed storage controller components implemented on respective ones of the storage nodes **415**. The storage controller **408** is therefore an example of what is more generally referred to herein as a "distributed storage controller." In subsequent description herein, the storage controller **408** is referred to as distributed storage controller **408**.

[0129] Each of the storage nodes **415** in this embodiment further comprises a set of processing modules configured to communicate over one or more networks with corresponding sets of processing modules on other ones of the storage nodes **415**. The sets of processing modules of the storage nodes **415** collectively comprise at least a portion of the distributed storage controller **408** of the content addressable storage system **405**.

[0130] The modules of the distributed storage controller **408** in the present embodiment more particularly comprise different sets of processing modules implemented on each of the storage nodes **415**. The set of processing modules of each of the storage nodes **415** comprises at least a control module **408**C, a data module **408**D and a routing module **408**R. The distributed storage controller **408** further comprises one or more management ("MGMT") modules **408**M. For example, only a single one of the storage nodes **415** may include a management module **408**M. It is also possible that management modules **408**M may be implemented on each of at least a subset of the storage nodes **415**. A given set of processing modules implemented on a particular one of the storage nodes **415** therefore illustratively includes at least one control module **408**C, at least one data module **408**D and at least one routing module **408**R, and possibly a management module **408**M.

[0131] Communication links may be established between the various processing modules of the distributed storage controller **408** using well-known communication protocols such as IP, Transmission Control Protocol (TCP), and remote direct memory access (RDMA). For example, respective sets of IP links used in data transfer and corresponding messaging could be associated with respective different ones of the routing modules **408**R.

[0132] Although shown as separate modules of the distributed storage controller **408**, the modules **412** and **414** in the present embodiment are assumed to be distributed at least in part over at least a subset of the other modules **408**C, **408**D, **408**R and **408**M of the storage controller **408**. Accordingly, at least portions of the data consistency checking functionality of the modules **412** and **414** may be implemented in one or more of the other modules of the storage controller **408**. In other embodiments, the modules **412** and **414** may be implemented as stand-alone modules of the storage controller **408**.

[0133] The storage devices **406** are configured to store metadata pages **420** and user data pages **422**, and may also store additional information not explicitly shown such as checkpoints and write journals. The metadata pages **420** and the user data pages **422** are illustratively stored in respective designated metadata and user data areas of the storage devices **406**. Accordingly, metadata pages **420** and user data pages **422** may be viewed as corresponding to respective designated metadata and user data areas of the storage devices **406**.

[0134] A given "page" as the term is broadly used herein should not be viewed as being limited to any particular range of fixed sizes. In some embodiments, a page size of 8 kilobytes (KB) is used, but this is by way of example only and can be varied in other embodiments. For example, page sizes of 4 KB, 16 KB or other values can be used. Accordingly, illustrative embodiments can utilize any of a wide variety of alternative paging arrangements for organizing the metadata pages **420** and the user data pages **422**.

[0135] The user data pages **422** are part of a plurality of LUNs configured to store files, blocks, objects or other arrangements of data, each also generally referred to herein as a "data item," on behalf of users of the content addressable storage system **405**. Each such LUN may comprise particular ones of the above-noted pages of the user data area. The user data stored in the user data pages **422** can include any type of user data that may be utilized in the system **100**. The term "user data" herein is therefore also intended to be broadly construed.

[0136] A given storage volume for which content-based signatures are generated using modules **412** and **414** illustratively comprises a set of one or more LUNs, each including multiple ones of the user data pages **422** stored in storage devices **406**.

[0137] The content addressable storage system **405** in the embodiment of FIG. **4** is configured to generate hash metadata providing a mapping between content-based digests of respective ones of the user data pages **422** and corresponding physical locations of those pages in the user data area. Content-based digests generated using hash functions are also referred to herein as "hash digests." Such hash digests or other types of content-based digests are examples of what are more generally referred to herein as "content-based signatures" of the respective user data pages **422**. The hash metadata generated by the content addressable storage system **405** is illustratively stored as metadata pages **420** in the metadata area. The generation and storage of the hash metadata is assumed to be performed under the control of the storage controller **408**.

[0138] Each of the metadata pages **420** characterizes a plurality of the user data pages **422**. For example, a given set of user data pages representing a portion of the user data pages **422** illustratively comprises a plurality of user data pages denoted User Data Page **1**, User Data Page **2**, . . . User Data Page n. Each of the user data pages in this example is characterized by a LUN identifier, an offset and a content-based signature. The content-based signature is generated as a hash function of content of the corresponding user data page. Illustrative hash functions that may be used to generate the content-based signature include the above-noted SHA1 secure hashing algorithm, or other secure hashing algorithms known to those skilled in the art, including SHA2, SHA256 and many others. The content-based signature is utilized to determine the location of the corresponding user data page within the user data area of the storage devices **406**.

[0139] Each of the metadata pages **420** in the present embodiment is assumed to have a signature that is not content-based. For example, the metadata page signatures may be generated using hash functions or other signature generation algorithms that do not utilize content of the metadata pages as input to the signature generation algorithm. Also, each of the metadata pages is assumed to characterize a different set of the user data pages.

[0140] A given set of metadata pages representing a portion of the metadata pages **420** in an illustrative embodiment comprises metadata pages denoted Metadata Page **1**, Metadata Page **2**, . . . Metadata Page m, having respective signatures denoted Signature **1**, Signature **2**, . . . Signature m. Each such metadata page characterizes a different set of n user data pages. For example, the characterizing information in each metadata page can include the LUN identifiers, offsets and content-based signatures for each of the n user data pages that are characterized by that metadata page. It is to be appreciated, however, that the user data and metadata page configurations described above are examples only, and numerous alternative user data and metadata page configurations can be used in other embodiments.

[0141] Ownership of a user data logical address space within the content addressable storage system **405** is illustratively distributed among the control modules **408**C.

[0142] The data consistency checking functionality provided by modules **412** and **414** in this embodiment is assumed to be distributed across multiple distributed processing modules, including at least a subset of the processing modules **408**C, **408**D, **408**R and **408**M of the distributed storage controller **408**.

[0143] For example, the management module **408**M of the storage controller **408** may include a replication control logic instance that engages corresponding replication control logic instances in all of the control modules **408**C and routing modules **408**R in order to implement an asynchronous replication process.

[0144] In some embodiments, the content addressable storage system **405** comprises an XtremIO™ storage array suitably modified to incorporate data consistency checking functionality as disclosed herein.

[0145] In arrangements of this type, the control modules **408**C, data modules **408**D and routing modules **408**R of the distributed storage controller **408** illustratively comprise respective C-modules, D-modules and R-modules of the XtremIO™ storage array. The one or more management modules **408**M of the distributed storage controller **408** in such arrangements illustratively comprise a system-wide management module ("SYM module") of the XtremIO™ storage array, although other types and arrangements of system-wide management modules can be used in other embodiments. Accordingly, data consistency checking functionality in some embodiments is implemented under the control of at least one system-wide management module of the distributed storage controller **408**, utilizing the C-modules, D-modules and R-modules of the XtremIO™ storage array.

[0146] In the above-described XtremIO™ storage array example, each user data page has a fixed size such as 8 KB and its content-based signature is a 20-byte signature generated using the SHA1 secure hashing algorithm. Also, each page has a LUN identifier and an offset, and so is characterized by <lun_id, offset, signature>.

[0147] The content-based signature in the present example comprises a content-based digest of the corresponding data page. Such a content-based digest is more particularly referred to as a "hash digest" of the corresponding data page, as the content-based signature is illustratively generated by applying a hash function such as the SHA1 secure hashing algorithm to the content of that data page. The full hash digest of a given data page is given by the above-noted 20-byte signature. The hash digest may be represented by a corresponding "hash handle," which in some cases may comprise a particular portion of the hash digest. The hash handle illustratively maps on a one-to-one basis to the corresponding full hash digest within a designated cluster boundary or other specified storage resource boundary of a given storage system. In arrangements of this type, the hash handle provides a lightweight mechanism for uniquely identifying the corresponding full hash digest and its associated data page within the specified storage resource boundary. The hash digest and hash handle are both considered examples of "content-based signatures" as that term is broadly used herein.

[0148] Examples of techniques for generating and processing hash handles for respective hash digests of respective data pages are disclosed in U.S. Pat. No. 9,208,162, entitled "Generating a Short Hash Handle," and U.S. Pat. No. 9,286,003, entitled "Method and Apparatus for Creating a Short Hash Handle Highly Correlated with a Globally-Unique Hash Signature," both of which are incorporated by reference herein.

[0149] As mentioned previously, storage controller components in an XtremIO™ storage array illustratively include C-module, D-module and R-module components. For example, separate instances of such components can be associated with each of a plurality of storage nodes in a clustered storage system implementation.

[0150] The distributed storage controller in this example is configured to group consecutive pages into page groups, to arrange the page groups into slices, and to assign the slices to different ones of the C-modules. For example, if there are 1024 slices distributed evenly across the C-modules, and there are a total of 16 C-modules in a given implementation, each of the C-modules "owns" 1024/16=64 slices. In such arrangements, different ones of the slices are assigned to different ones of the control modules **408**C such that control of the slices within the storage controller **408** of the storage system **405** is substantially evenly distributed over the control modules **408**C of the storage controller **408**.

[0151] The D-module allows a user to locate a given user data page based on its signature. Each metadata page also has a size of 8 KB and includes multiple instances of the <lun_id, offset, signature> for respective ones of a plurality of the user data pages. Such metadata pages are illustratively generated by the C-module but are accessed using the D-module based on a metadata page signature.

[0152] The metadata page signature in this embodiment is a 20-byte signature but is not based on the content of the metadata page. Instead, the metadata page signature is generated based on an 8-byte metadata page identifier that is a function of the LUN identifier and offset information of that metadata page.

[0153] If a user wants to read a user data page having a particular LUN identifier and offset, the corresponding metadata page identifier is first determined, then the metadata page signature is computed for the identified metadata page, and then the metadata page is read using the computed signature. In this embodiment, the metadata page signature is more particularly computed using a signature generation algorithm that generates the signature to include a hash of the 8-byte metadata page identifier, one or more ASCII codes for particular predetermined characters, as well as possible additional fields. The last bit of the metadata page signature may always be set to a particular logic value so as to distinguish it from the user data page signature in which the last bit may always be set to the opposite logic value.

[0154] The metadata page signature is used to retrieve the metadata page via the D-module. This metadata page will include the <lun_id, offset, signature> for the user data page if the user page exists. The signature of the user data page is then used to retrieve that user data page, also via the D-module.

[0155] Write requests processed in the content addressable storage system **405** each illustratively comprise one or more IO operations directing that at least one data item of the storage system **405** be written to in a particular manner. A given write request is illustratively received in the storage system **405** from a host device over a network. In some embodiments, a write request is received in the distributed storage controller **408** of the storage system **405**, and directed from one processing module to another processing module of the distributed storage controller **408**. For example, a received write request may be directed from a routing module **408**R of the distributed storage controller **408** to a particular control module **408**C of the distributed storage controller **408**. Other arrangements for receiving and processing write requests from one or more host devices can be used.

[0156] The term "write request" as used herein is intended to be broadly construed, so as to encompass one or more IO operations directing that at least one data item of a storage system be written to in a particular manner. A given write request is illustratively received in a storage system from a host device.

[0157] In the XtremIO™ context, the C-modules, D-modules and R-modules of the storage nodes **415** communicate with one another over a high-speed internal network such as an InfiniBand network. The C-modules, D-modules and R-modules coordinate with one another to accomplish various IO processing tasks.

[0158] The write requests from the host devices identify particular data pages to be written in the storage system **405** by their corresponding logical addresses each comprising a LUN ID and an offset.

[0159] As noted above, a given one of the content-based signatures illustratively comprises a hash digest of the corresponding data page, with the hash digest being generated by applying a hash function to the content of that data page. The hash digest may be uniquely represented within a given storage resource boundary by a corresponding hash handle.

[0160] The content addressable storage system **405** utilizes a two-level mapping process to map logical block addresses to physical block addresses. The first level of mapping uses an address-to-hash ("A2H") table and the second level of mapping uses a hash metadata ("HMD") table, with the A2H and HMD tables corresponding to respective logical and physical layers of the content-based signature mapping within the content addressable storage system **405**. The HMD table or a given portion thereof in some embodiments disclosed herein is more particularly referred to as a hash-to-data ("H2D") table.

[0161] The first level of mapping using the A2H table associates logical addresses of respective data pages with respective content-based signatures of those data pages. This is also referred to as logical layer mapping.

[0162] The second level of mapping using the HMD table associates respective ones of the content-based signatures with respective physical storage locations in one or more of the storage devices **106**. This is also referred to as physical layer mapping.

[0163] Examples of these and other metadata structures utilized in illustrative embodiments were described above in conjunction with FIG. **2**. These particular examples include respective A2H, H2D, HMD and PLB tables. In some embodiments, the A2H and H2D tables are utilized primarily by the control modules **408**C, while the HMD and PLB tables are utilized primarily by the data modules **408**D.

[0164] For a given write request, hash metadata comprising at least a subset of the above-noted tables is updated in conjunction with the processing of that write request.

[0165] The A2H, H2D, HMD and PLB tables described above are examples of what are more generally referred to herein as "mapping tables" of respective distinct types. Other types and arrangements of mapping tables or other content-based signature mapping information may be used in other embodiments.

[0166] Such mapping tables are still more generally referred to herein as "metadata structures" of the content addressable storage system **405**. It should be noted that additional or alternative metadata structures can be used in other embodiments. References herein to particular tables of particular types, such as A2H, H2D, HMD and PLB tables, and their respective configurations, should be considered non-limiting and are presented by way of illustrative example only. Such metadata structures can be implemented in numerous alternative configurations with different arrangements of fields and entries in other embodiments.

[0167] The logical block addresses or LBAs of a logical layer of the storage system **405** correspond to respective physical blocks of a physical layer of the storage system **405**. The user data pages of the logical layer are organized by LBA and have reference via respective content-based signatures to particular physical blocks of the physical layer.

[0168] Each of the physical blocks has an associated reference count that is maintained within the storage system **405**. The reference count for a given physical block indicates the number of logical blocks that point to that same physical block.

[0169] In releasing logical address space in the storage system, a dereferencing operation is generally executed for each of the LBAs being released. More particularly, the reference count of the corresponding physical block is decremented. A reference count of zero indicates that there are no longer any logical blocks that reference the corresponding physical block, and so that physical block can be released.

[0170] It should also be understood that the particular arrangement of storage controller processing modules **408**C, **408**D, **408**R and **408**M as shown in the FIG. **4** embodiment is presented by way of example only. Numerous alternative arrangements of processing modules of a distributed storage controller may be used to implement data consistency checking functionality in a clustered storage system in other embodiments.

[0171] Additional examples of content addressable storage functionality implemented in some embodiments by control modules **408**C, data modules **408**D, routing modules **408**R and management module(s) **408**M of distributed storage controller **408** can be found in U.S. Pat. No. 9,104,326, entitled "Scalable Block Data Storage Using Content Addressing," which is incorporated by reference herein. Alternative arrangements of these and other storage node processing modules of a distributed storage controller in a content addressable storage system can be used in other embodiments.

[0172] Illustrative embodiments of a storage system with data consistency checking functionality as disclosed herein can provide a number of significant advantages relative to conventional arrangements.

[0173] For example, some embodiments provide particularly efficient arrangements for performing data consistency checks for a consistency group that is subject to an ongoing synchronous replication process.

[0174] Some embodiments perform data consistency checking in synchronous replication using an active snapshot set that allows the data consistency check to be performed in a substantially simplified manner relative to conventional approaches.

[0175] For example, some embodiments provide simplified data consistency checking that avoids the need to repeatedly create host access LUN mappings for all storage volumes in a snapshot set of a given replication consistency group each time a data consistency check is performed on synchronously replicated data.

[0176] These and other embodiments provide a streamlined approach to data consistency checking for synchronously replicated data that advantageously avoids unduly complex management orchestration, host device setup, and manual user intervention in the data consistency checking process.

[0177] In some embodiments, the source and target storage systems are illustratively implemented as respective content addressable storage systems, but in other embodiments one or more of the storage systems can instead be a traditional storage array, which does not support any type of content addressable storage functionality, with any missing functionality being provided by a host device.

[0178] Accordingly, functionality for data consistency checking in synchronous replication using an active snapshot set as disclosed herein can be implemented in a storage system, in a host device, or partially in a storage system and partially in a host device.

[0179] It is to be appreciated that the particular advantages described above and elsewhere herein are associated with particular illustrative embodiments and need not be present in other embodiments. Also, the particular types of information processing system features and functionality as illustrated in the drawings and described above are exemplary only, and numerous other arrangements may be used in other embodiments.

[0180] Illustrative embodiments of processing platforms utilized to implement host devices and storage systems with data consistency checking functionality will now be described in greater detail with reference to FIGS. **5** and **6**. Although described in the context of system **100**, these platforms may also be used to implement at least portions of other information processing systems in other embodiments.

[0181] FIG. **5** shows an example processing platform comprising cloud infrastructure **500**. The cloud infrastructure **500** comprises a combination of physical and virtual processing resources that may be utilized to implement at least a portion of the information processing system **100**. The cloud infrastructure **500** comprises multiple virtual machines (VMs) and/or container sets **502-1**, **502-2**, . . . **502**-L implemented using virtualization infrastructure **504**. The virtualization infrastructure **504** runs on physical infrastructure **505**, and illustratively comprises one or more hypervisors and/or operating system level virtualization infrastructure. The operating system level virtualization infrastructure illustratively comprises kernel control groups of a Linux operating system or other type of operating system.

[0182] The cloud infrastructure **500** further comprises sets of applications **510-1**, **510-2**, . . . **510**-L running on respective ones of the VMs/container sets **502-1**, **502-2**, . . . **502**-L under the control of the virtualization infrastructure **504**. The VMs/container sets **502** may comprise respective VMs, respective sets of one or more containers, or respective sets of one or more containers running in VMs.

[0183] In some implementations of the FIG. **5** embodiment, the VMs/container sets **502** comprise respective VMs implemented using virtualization infrastructure **504** that comprises at least one hypervisor. Such implementations can provide data consistency functionality of the type described above for one or more processes running on a given one of the VMs. For example, each of the VMs can implement replication control logic and/or snapshot generators for providing data consistency checking functionality in the system **100**.

[0184] An example of a hypervisor platform that may be used to implement a hypervisor within the virtualization infrastructure **504** is the VMware® vSphere® which may have an associated virtual infrastructure management system such as the VMware® vCenter™. The underlying physical machines may comprise one or more distributed processing platforms that include one or more storage systems.

[0185] In other implementations of the FIG. **5** embodiment, the VMs/container sets **502** comprise respective containers implemented using virtualization infrastructure **504** that provides operating system level virtualization functionality, such as support for Docker containers running on bare metal hosts, or Docker containers running on VMs. The containers are illustratively implemented using respective kernel control groups of the operating system. Such implementations can also provide data consistency checking functionality of the type described above. For example, a container host device supporting multiple containers of one or more container sets can implement one or more instances of replication control logic and/or snapshot generators for providing data consistency checking functionality in the system **100**.

[0186] As is apparent from the above, one or more of the processing modules or other components of system **100** may each run on a computer, server, storage device or other processing platform element. A given such element may be viewed as an example of what is more generally referred to herein as a "processing device." The cloud infrastructure **500** shown in FIG. **5** may represent at least a portion of one processing platform. Another example of such a processing platform is processing platform **600** shown in FIG. **6**.

[0187] The processing platform **600** in this embodiment comprises a portion of system **100** and includes a plurality of processing devices, denoted **602-1**, **602-2**, **602-3**, . . . **602**-K, which communicate with one another over a network **604**.

[0188] The network **604** may comprise any type of network, including by way of example a global computer network such as the Internet, a WAN, a LAN, a satellite network, a telephone or cable network, a cellular network, a wireless network such as a WiFi or WiMAX network, or various portions or combinations of these and other types of networks.

[0189] The processing device **602-1** in the processing platform **600** comprises a processor **610** coupled to a memory **612**.

[0190] The processor **610** may comprise a microprocessor, a microcontroller, an application-specific integrated circuit (ASIC), a field-programmable gate array (FPGA), graphics processing unit (GPU) or other type of processing circuitry, as well as portions or combinations of such circuitry elements.

[0191] The memory **612** may comprise random access memory (RAM), read-only memory (ROM), flash memory or other types of memory, in any combination. The memory **612** and other memories disclosed herein should be viewed as illustrative examples of what are more generally referred to as "processor-readable storage media" storing executable program code of one or more software programs.

[0192] Articles of manufacture comprising such processor-readable storage media are considered illustrative embodiments. A given such article of manufacture may comprise, for example, a storage array, a storage disk or an integrated circuit containing RAM, ROM, flash memory or other electronic memory, or any of a wide variety of other types of computer program products. The term "article of manufacture" as used herein should be understood to exclude transitory, propagating signals. Numerous other types of computer program products comprising processor-readable storage media can be used.

[0193] Also included in the processing device **602-1** is network interface circuitry **614**, which is used to interface the processing device with the network **604** and other system components, and may comprise conventional transceivers.

[0194] The other processing devices **602** of the processing platform **600** are assumed to be configured in a manner similar to that shown for processing device **602-1** in the figure.

[0195] Again, the particular processing platform **600** shown in the figure is presented by way of example only, and system **100** may include additional or alternative processing platforms, as well as numerous distinct processing platforms in any combination, with each such platform comprising one or more computers, servers, storage devices or other processing devices.

[0196] For example, other processing platforms used to implement illustrative embodiments can comprise converged infrastructure such as VxRail™, VxRack™, VxRack™ FLEX, VxBlock™, or Vblock® converged infrastructure from VCE, the Virtual Computing Environment Company, now the Converged Platform and Solutions Division of Dell EMC.

[0197] It should therefore be understood that in other embodiments different arrangements of additional or alternative elements may be used. At least a subset of these elements may be collectively implemented on a common processing platform, or each such element may be implemented on a separate processing platform.

[0198] As indicated previously, components of an information processing system as disclosed herein can be implemented at least in part in the form of one or more software programs stored in memory and executed by a processor of a processing device. For example, at least portions of the data consistency checking functionality of one or more components of a storage system as disclosed herein are illustratively implemented in the form of software running on one or more processing devices.

[0199] It should again be emphasized that the above-described embodiments are presented for purposes of illustration only. Many variations and other alternative embodiments may be used. For example, the disclosed techniques are applicable to a wide variety of other types of information processing systems, host devices, storage systems, storage nodes, storage devices, storage controllers, asynchronous replication processes, snapshot generators and associated control logic and metadata structures. Also, the particular configurations of system and device elements and associated processing operations illustratively shown in the drawings can be varied in other embodiments. Moreover, the various assumptions made above in the course of describing the illustrative embodiments should also be viewed as exemplary rather than as requirements or limitations of the disclosure. Numerous other alternative embodiments within the scope of the appended claims will be readily apparent to those skilled in the art.

What is claimed is:

1. An apparatus comprising:

at least one processing device comprising a processor coupled to a memory;

said at least one processing device being configured:

to define a consistency group comprising a plurality of storage volumes subject to replication from a source storage system to a target storage system;

in conjunction with a transition from asynchronous replication of the consistency group to synchronous replication of the consistency group:

to complete a current asynchronous replication cycle;

to utilize a current asynchronous snapshot set on the target storage system to establish an active snapshot set on the target storage system;

to transfer differential data of a subsequent asynchronous replication cycle to the active snapshot set on the target storage system;

to specify the active snapshot set on the target storage system as a destination of the synchronous replication; and

to perform a data consistency check on the target storage system utilizing an additional snapshot set generated from the active snapshot set.

2. The apparatus of claim **1** wherein said at least one processing device is implemented at least in part within at least one of the source storage system and the target storage system.

3. The apparatus of claim **2** wherein said at least one processing device comprises at least a portion of a storage controller of at least one of the source storage system and the target storage system.

4. The apparatus of claim **1** wherein the storage volumes comprise respective logical storage volumes each comprising at least a portion of a physical storage space of one or more storage devices.

5. The apparatus of claim **1** wherein a given one of the snapshot sets comprises a plurality of snapshot trees each associated with a different one of the storage volumes of the consistency group.

6. The apparatus of claim **5** wherein a given one of the snapshot trees corresponding to a particular one of the storage volumes comprises:

a root node;

at least one branch node; and

a plurality of leaf nodes;

wherein a given one of the leaf nodes comprises a snapshot providing a particular point-in-time copy of the storage volume.

7. The apparatus of claim **5** wherein a given one of the snapshot trees is associated with one or more additional trees including at least one of:

a dirty tree that characterizes updates to logical addresses of the corresponding storage volume; and

a hash tree comprising content-based signatures of respective ones of the logical addresses of the corresponding storage volume.

8. The apparatus of claim **1** wherein transferring differential data of a subsequent asynchronous replication cycle to the active snapshot set on the target storage system comprises transferring differential data representing a differential between a previous asynchronous snapshot set of the consistency group on the source storage system and a current asynchronous snapshot set on the source storage system, the current asynchronous snapshot set on the source storage system corresponding to the current asynchronous snapshot set on the target storage system that was utilized to establish the active snapshot set.

9. The apparatus of claim **1** wherein specifying the active snapshot set on the target storage system as a destination of the synchronous replication comprises:

specifying the active snapshot set on the target storage system as the destination of the synchronous replication responsive to the synchronous replication not being in an active-active configuration; and

specifying the consistency group on the target storage system as the destination of the synchronous replication responsive to the synchronous replication being in an active-active configuration.

10. The apparatus of claim **1** wherein performing a data consistency check on the target storage system utilizing an additional snapshot set generated from the active snapshot set comprises:

generating the additional snapshot set from the active snapshot set;

refreshing the consistency group on the target storage system to the additional snapshot set; and

performing the data consistency check utilizing the refreshed consistency group.

11. The apparatus of claim **10** wherein refreshing the consistency group on the target storage system to the additional snapshot set comprises performing a snap-and-reassign operation for the consistency group utilizing the additional snapshot set.

12. The apparatus of claim **1** wherein said at least one processing device is further configured to perform a replication failover operation for the synchronous replication.

13. The apparatus of claim **12** wherein performing the replication failover operation comprises:

terminating a current session of synchronous replication from the source storage system to the target storage system;

generating a failover snapshot set from the active snapshot set;

refreshing the consistency group on the target storage system to the failover snapshot set; and

initiating a new session of synchronous replication from the target storage system to the source storage system.

14. A method comprising:

defining a consistency group comprising a plurality of storage volumes subject to replication from a source storage system to a target storage system;

in conjunction with a transition from asynchronous replication of the consistency group to synchronous replication of the consistency group:

completing a current asynchronous replication cycle;

utilizing a current asynchronous snapshot set on the target storage system to establish an active snapshot set on the target storage system;

transferring differential data of a subsequent asynchronous replication cycle to the active snapshot set on the target storage system;

specifying the active snapshot set on the target storage system as a destination of the synchronous replication; and

performing a data consistency check on the target storage system utilizing an additional snapshot set generated from the active snapshot set;

wherein the method is implemented by at least one processing device comprising a processor coupled to a memory.

15. The method of claim **14** wherein a given one of the snapshot sets comprises a plurality of snapshot trees each associated with a different one of the storage volumes of the consistency group.

16. The method of claim **14** wherein performing a data consistency check on the target storage system utilizing an additional snapshot set generated from the active snapshot set comprises:

generating the additional snapshot set from the active snapshot set;

refreshing the consistency group on the target storage system to the additional snapshot set; and

performing the data consistency check utilizing the refreshed consistency group.

17. The method of claim **16** wherein refreshing the consistency group on the target storage system to the additional snapshot set comprises performing a snap-and-reassign operation for the consistency group utilizing the additional snapshot set.

18. A computer program product comprising a non-transitory processor-readable storage medium having stored therein program code of one or more software programs, wherein the program code when executed by at least one processing device causes said at least one processing device:

to define a consistency group comprising a plurality of storage volumes subject to replication from a source storage system to a target storage system;

in conjunction with a transition from asynchronous replication of the consistency group to synchronous replication of the consistency group:

to complete a current asynchronous replication cycle;

to utilize a current asynchronous snapshot set on the target storage system to establish an active snapshot set on the target storage system;

to transfer differential data of a subsequent asynchronous replication cycle to the active snapshot set on the target storage system;

to specify the active snapshot set on the target storage system as a destination of the synchronous replication; and

to perform a data consistency check on the target storage system utilizing an additional snapshot set generated from the active snapshot set.

19. The computer program product of claim **18** wherein performing a data consistency check on the target storage system utilizing an additional snapshot set generated from the active snapshot set comprises:

generating the additional snapshot set from the active snapshot set;

refreshing the consistency group on the target storage system to the additional snapshot set; and

performing the data consistency check utilizing the refreshed consistency group.

20. The computer program product of claim **19** wherein refreshing the consistency group on the target storage system to the additional snapshot set comprises performing a snap-and-reassign operation for the consistency group utilizing the additional snapshot set.

* * * * *