



- (51) **International Patent Classification:**
G06F 12/00 (2006.01) G06F 12/16 (2006.01)
- (21) **International Application Number:**
PCT/IB2015/059510
- (22) **International Filing Date:**
10 December 2015 (10.12.2015)
- (25) **Filing Language:** English
- (26) **Publication Language:** English
- (30) **Priority Data:**
62/097,077 28 December 2014 (28.12.2014) US
- (71) **Applicant:** STRATO SCALE LTD. [IL/IL]; 6 Galgalei Haplada Street, 46733 Herzliya (IL).
- (72) **Inventors:** HOROWITZ, Alon; 3 Liber Ze'ev Street, 4252484 Netanya (IL). BUCHBINDER, Rafael; 75/3 Tsa-hal Street, 5545149 Kiryat Ono (IL).
- (74) **Agent:** D. KLIGLER I.P. SERVICES LTD.; P.O. Box 57651, 61576 Tel Aviv (IL).
- (81) **Designated States** (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM,

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

- (84) **Designated States** (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

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

(54) **Title:** RECOVERY SYNCHRONIZATION IN A DISTRIBUTED STORAGE SYSTEM

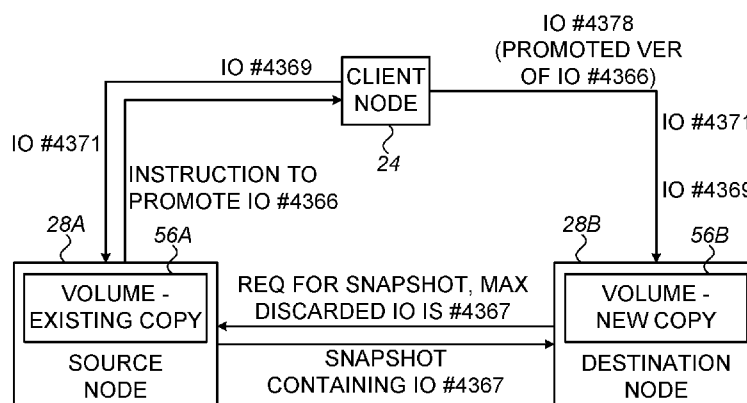


FIG. 2

(57) **Abstract:** A method for data storage includes sending write commands to a first storage node (28A), for storing data in a data volume (56A) on the first storage node. In response to a decision to create a copy (56B) of the data volume on a second storage node (28B), (i) the write commands are sent both to the first storage node and to the second storage node, and (ii) a snapshot of the data volume is created and sent from the first storage node to the second storage node. The copy of the data volume is created by the second storage node based on the snapshot and on one or more of the write commands received at the second storage node.

RECOVERY SYNCHRONIZATION IN A DISTRIBUTED STORAGE SYSTEM

CROSS-REFERENCE TO RELATED APPLICATIONS

This application claims the benefit of U.S. Provisional Patent Application 62/097,077, filed December 28, 2014, whose disclosure is incorporated herein by reference.

FIELD OF THE INVENTION

The present invention relates generally to data storage, and particularly to methods and systems for failure recovery in data storage systems.

BACKGROUND OF THE INVENTION

In many data storage schemes, the system maintains two or more copies of a data volume on different storage devices or compute nodes for resilience against failures. When a data volume is lost or corrupted due to failure, the system may create a new copy of the volume from an existing copy, to preserve the original level of redundancy.

SUMMARY OF THE INVENTION

An embodiment of the present invention that is described herein provides a method for data storage, including sending write commands to a first storage node, for storing data in a data volume on the first storage node. In response to a decision to create a copy of the data volume on a second storage node, (i) the write commands are sent both to the first storage node and to the second storage node, and (ii) a snapshot of the data volume is created and sent from the first storage node to the second storage node. The copy of the data volume is created by the second storage node based on the snapshot and on one or more of the write commands received at the second storage node.

In some embodiments, creating the snapshot includes including in the snapshot at least a write command having a specified serial number, and creating the copy of the data volume includes, in the second storage node, receiving the snapshot, and creating the copy of the data volume from the received snapshot and the received write commands whose serial numbers are larger than the specified serial number.

In an embodiment, the method includes sending from the second storage node to the first storage node a snapshot request that specifies the specified serial number, and discarding in the second storage node any received write commands whose serial numbers are smaller than or equal to the specified serial number. In another embodiment, the method includes detecting in the first storage node a write command that (i) is received after the snapshot has

been created, and (ii) has a serial number larger than the specified serial number, and requesting the client node to re-issue the write command with a new serial number.

Typically, creating the copy of the data volume includes initializing the copy in the second storage node from the snapshot, and then executing the one or more received write
5 commands in the initialized copy.

There is additionally provided, in accordance with an embodiment of the present invention, a computing system including a client node and at least a first storage node and a second storage node. The client node is configured to send write commands to the first storage node for storing data in a data volume on the first storage node. In response to a decision to
10 create a copy of the data volume on the second storage node, the client node is configured to send the write commands both to the first storage node and to the second storage node, the first storage node is configured to create and send to the second storage node a snapshot of the data volume, and the second storage node is configured to create the copy of the data volume by the second storage node, based on the snapshot and on one or more of the write commands
15 received at the second storage node.

There is also provided, in accordance with an embodiment of the present invention, a storage node including an interface and a processor. The interface is configured for communicating with a client node and with an additional storage node over a computer network. The processor is configured to receive from the client node write commands having
20 respective serial numbers for storing data in a data volume, to receive from the additional storage node a request to create a snapshot of the data volume that includes at least a write command having a specified serial number, and to create and send the snapshot to the additional storage node.

There is further provided, in accordance with an embodiment of the present invention,
25 a storage node including an interface and a processor. The interface is configured for communicating with a client node and with an additional storage node over a computer network. The processor is configured, in response to a decision to create on the storage node a copy of a data volume that is stored on the additional storage node, to start receiving write commands for storing data in the copy of the data volume, to request the additional storage
30 node to create and send a snapshot of the data volume, and to create the copy of the data volume based on the snapshot and on one or more of the received write commands.

The present invention will be more fully understood from the following detailed description of the embodiments thereof, taken together with the drawings in which:

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a block diagram that schematically illustrates a computing system, in accordance with an embodiment of the present invention;

Fig. 2 is a diagram that schematically illustrates synchronized recovery of a data volume, in accordance with an embodiment of the present invention; and

Fig. 3 is a flow chart that schematically illustrates a method for synchronized recovery of a data volume, in accordance with an embodiment of the present invention.

DETAILED DESCRIPTION OF EMBODIMENTS

OVERVIEW

Embodiments of the present invention that are described herein provide improved methods and systems for recovering data volumes. In some embodiments, a computing system comprises multiple compute nodes. At some point in time, a new copy of a data volume is to be created on a destination node, from an existing copy of the data volume that is stored on a source node. The source node continues to receive new write commands to the data volume from a client node during the recovery process.

In some disclosed embodiments, once a decision is made to create the new copy, the client node begins to send each write command both to the source node and to the destination node. The client node sends the write commands with respective serial numbers attached thereto. In addition, the destination node requests the source node to create and send a snapshot of the data volume. In the request, the destination node specifies a certain serial number X , and requests that the snapshot include the write commands up to and including this serial number.

Until receiving the snapshot from the source node, the destination node discards write commands arriving from the client node if their serial numbers are smaller than or equal to X , and accumulates the write commands having serial numbers larger than X . Upon receiving the snapshot from the source node, the destination node creates the new copy of the data volume from the snapshot (which spans the write commands up to serial number X) plus the locally-accumulated write commands (starting from serial number $X+1$).

The recovery process described above creates a new copy that is identical to the existing copy, provided that the write commands arrive in-order at the source node. If write commands possibly arrive out-of-order, it is possible that a write command with a serial number larger than X will arrive in the source node after the source node has already created the snapshot. Unless accounted for, such a write command will not be included in the new

copy of the data volume, because (i) it is not included in the snapshot created by the source node, and (ii) it is discarded by the destination node.

In some embodiments, the system uses an additional mechanism referred to as “promotion” to account for possible out-of-order arrival of write commands. In these
5 embodiments, after creating the snapshot, the source node checks each newly-arriving write command and verifies whether its serial number is indeed larger than X. Upon detecting a write command that arrived after the snapshot was created but has a serial number larger than X, the source node sends the write command back to the client node and requests the client node to re-issue the write command with a new serial number. The newly-issued (“promoted”)
10 write command is sent to the destination node and is added to the new copy.

In summary, the methods and systems described herein a new copy on-the-fly, i.e., without having to stop receiving write commands to the data volume. Moreover, the disclosed techniques do not assume that write commands arrive in-order.

SYSTEM DESCRIPTION

15 Fig. 1 is a block diagram that schematically illustrates a computing system 20, in accordance with an embodiment of the present invention. System 20 may comprise, for example, a data center, a High-Performance Computing (HPC) cluster, or any other suitable type of computing system.

System 20 comprises multiple compute nodes of various kinds, referred to simply as
20 “nodes” for brevity. In the example of Fig. 1, the system comprises one or more client nodes 24 and multiple disk nodes 28, also referred to as storage nodes. Fig. 1 shows a single client node and three disk nodes for the sake of clarity. Real-life implementations of system 20, however, may comprise large numbers of client nodes and disk nodes. In some embodiments, a given compute node (or even all compute nodes) may serve as both a client node and a disk node.
25 Client nodes 24 and disk nodes 28 communicate with one another over a computer network 32, e.g., a Local Area Network (LAN).

Each client node 24 typically runs client applications that, among other tasks, send data for persistent storage and retrieve data from persistent storage. In the present example, client node 24 comprises a network interface, such as a Network Interface Controller (NIC) 36, for
30 communicating over network 32, and a processor 40 for running the client applications and carrying out various other processing tasks of the client node.

Each disk node 28 comprises one or more storage devices 52, e.g., Solid State Drives (SSDs) or Hard Disk Drives (HDDs), for storing data generated by client nodes 24. In the

present example, disk node 28 comprises a network interface, such as a NIC 44, for communicating over network 32, and a processor 48 for managing the storage of data in storage devices 52 and carrying out various other processing tasks of the disk node.

Typically, the applications running on client nodes 24 store data in data volumes 56 that are stored on storage devices 52 of disk nodes 28. Each client node 24 typically stores data by sending write commands. Each write command specifies the data, and the data volume in which the data is to be stored.

In some embodiments, each data volume is replicated across multiple disk nodes 28 for redundancy. In other words, system 20 typically stores two or more copies of each data volume on different disk nodes. When a copy is lost or corrupted, e.g., due to failure of a disk node or a storage device, the system uses an existing copy to create another copy of the data volume on a functional disk node, so as to preserve the expected level of redundancy.

In the example of Fig. 1, system 20 comprises a cluster management unit 60, which manages the cluster-wide data storage functions. Unit 60 comprises an interface, e.g., a NIC 64, for communicating with the client nodes and disk nodes over network 32, and a processor 68 for carrying out the various management tasks. Among other tasks, processor 68 decides when to create a new copy of a data volume, and on which disk node and from which existing copy to create the new copy. Processor 68 instructs the various client nodes and disk nodes accordingly. In alternative embodiments, however, such management functions may be carried out by any other suitable entity. Distributed management, in which the functions of unit 60 are distributed among processors 40 of the client nodes and/or processors 48 of the disk nodes, is also possible.

The system, client-node, disk-node and cluster-management-unit configurations shown in Fig. 1 are example configurations that are chosen purely for the sake of conceptual clarity. In alternative embodiments, any other suitable configurations can be used. The various elements of system 20 may be implemented using hardware/firmware, such as in one or more Application-Specific Integrated Circuit (ASICs) or Field-Programmable Gate Array (FPGAs). Alternatively, some system elements may be implemented in software or using a combination of hardware/firmware and software elements.

Typically, processors 40, 48 and 68 comprise general-purpose processors, which are programmed in software to carry out the functions described herein. The software may be downloaded to the processors in electronic form, over a network, for example, or it may, alternatively or additionally, be provided and/or stored on non-transitory tangible media, such as magnetic, optical, or electronic memory.

SYNCHRONIZED ON-THE-FLY RECOVERY OF DATA VOLUMES

Fig. 2 is a diagram that schematically illustrates synchronized recovery of a data volume, in accordance with an embodiment of the present invention. The figure shows three nodes of system 20 – Client node 24, a source node 28A (one of disk nodes 28) on which an existing copy 56A of the data volume is stored, and a destination node 28B (another one of disk nodes 28) on which a new copy 56B of the data volume is to be created.

In some embodiments, once a decision is made (e.g., by management unit 60) to create new copy 56B, client node 24 is instructed to send each write command addressed to the data volume both to source node 28A and to destination node 28B. Client node 24 assigns respective serial numbers to the write commands, and marks each write command with the appropriate serial number.

The figure shows two example write commands (also referred to as Input-Output (IO) commands) having serial numbers #4369 and #4371 that are sent from client node 24 in parallel to source node 28A and to destination node 28B.

In addition, in response to the decision to create the new copy, destination node 28B requests source node 28A to create and send a snapshot of the data volume. In the request, destination node 28B specifies a certain serial number X, and requests that the snapshot include the write commands up to and including this serial number. In the present example, the specified serial number X is #4367.

Until receiving the snapshot from source node 28A, destination node 28B discards write commands arriving from client node 24 if their serial numbers are smaller than or equal to X, and accumulates the write commands having serial numbers larger than X. Upon receiving the snapshot from source node 28A, destination node 28B creates new copy 56B of the data volume from the snapshot, and from the locally-accumulated write commands.

As long as the write commands arrive in-order, the snapshot includes the write commands up to serial number X, and the write commands accumulated by destination node 28B start from serial number X+1. Thus, destination node 28B has the complete information for constructing new copy 56B that is identical to existing copy 56A, even though write commands continue to arrive and modify the data volume.

If, on the other hand, write commands arrive out-of-order, it is possible that a write command with a serial number larger than X will arrive in source node 28A after the source node has already created the snapshot. Such a write command may not be included in new copy 56B, because (i) it is not included in the snapshot, and (ii) it is discarded by destination node 28B.

In some embodiments, system 20 overcomes this issue using an additional mechanism referred to as “promotion” of write commands. In an embodiment, after creating the snapshot, source node 28A checks each newly-arriving write command and verifies whether its serial number is larger than X or not.

5 Upon detecting a write command that arrived after the snapshot was created but has a serial number larger than X, source node 28A does not acknowledge the write command. The source node sends the write command back to the client node 24, and requests client node 24 to re-issue the write command with a new serial number. In response, client node 24 issues the write command again with a new serial number. The newly-issued (“promoted”) write
10 command is sent to the destination node. Since the new serial number is larger than X, the newly-issued write command is added to the new copy.

In the example of Fig. 2, a write command with IO #4366 arrives at source node 28A after the snapshot has been created (recall that $X = \#4367$). The source node instructs client node 24A to promote this write command. In response, client 24A re-issues the write
15 command with serial number #4378. Since the new serial number is larger than X, destination node 28B adds the newly-issued write command to new copy 56B.

Thus, the promotion mechanism described above guarantees that new copy 56B is kept identical to existing copy 56A, notwithstanding the out-of-order arrival of write commands.

Fig. 3 is a flow chart that schematically illustrates a method for synchronized recovery
20 of a data volume, in accordance with an embodiment of the present invention. The recovery method begins with the client node sending IO commands to both the source node and the destination node, at a duplicate sending step 80.

At a snapshot requesting step 84, the destination node requests the source node to create and send a snapshot of the data volume that contains IO #X. The source node creates
25 the snapshot and begins to send it to the destination node. At a selective accumulation step 88, the destination node discards write commands whose serial number $\leq X$, and accumulates write commands whose serial number $> X$.

At a snapshot checking step 92, the destination node checks whether the entire snapshot has arrived from the source node. If not, the method loops back to step 88 above.
30 Once the entire snapshot arrives, the destination node creates the new copy of the data volume based on the snapshot and based on the accumulated write commands, at a copy creation step 96. The destination node typically initializes the new copy from the snapshot, and then executes the accumulated write commands in the initialized copy.

In parallel, at a promotion checking step 100, the source node checks whether a write command having a serial number $< X$ is received after the snapshot has been created. If not, the method loops back to step 88 above. Upon detecting a write command whose serial number is smaller than X , the source node requests the client node to “promote” the write command, i.e.,
5 re-issue the write command with a new serial number, at a promotion requesting step 104.

The process described in Figs. 2 and 3 is depicted by way of example. In alternative embodiments, other suitable recovery processes can be used. For example, if it can be guaranteed that write commands arrive in-order, the “promotion” mechanism can be omitted.

Although the embodiments described herein mainly address recovery of data volumes,
10 the methods and systems described herein can also be used in other suitable storage applications such as object stores and distributed file-systems.

It will thus be appreciated that the embodiments described above are cited by way of example, and that the present invention is not limited to what has been particularly shown and described hereinabove. Rather, the scope of the present invention includes both combinations
15 and sub-combinations of the various features described hereinabove, as well as variations and modifications thereof which would occur to persons skilled in the art upon reading the foregoing description and which are not disclosed in the prior art. Documents incorporated by reference in the present patent application are to be considered an integral part of the application except that to the extent any terms are defined in these incorporated documents in
20 a manner that conflicts with the definitions made explicitly or implicitly in the present specification, only the definitions in the present specification should be considered.

CLAIMS

1. A method for data storage, comprising:

sending write commands to a first storage node, for storing data in a data volume on the first storage node;

5 in response to a decision to create a copy of the data volume on a second storage node, (i) sending the write commands both to the first storage node and to the second storage node, and (ii) creating and sending from the first storage node to the second storage node a snapshot of the data volume; and

creating the copy of the data volume by the second storage node, based on the snapshot and on one or more of the write commands received at the second storage node.

2. The method according to claim 1, wherein creating the snapshot comprises including in the snapshot at least a write command having a specified serial number, and wherein creating the copy of the data volume comprises, in the second storage node, receiving the snapshot, and creating the copy of the data volume from the received snapshot and the received write commands whose serial numbers are larger than the specified serial number.

3. The method according to claim 2, and comprising sending from the second storage node to the first storage node a snapshot request that specifies the specified serial number, and discarding in the second storage node any received write commands whose serial numbers are smaller than or equal to the specified serial number.

4. The method according to claim 2, and comprising detecting in the first storage node a write command that (i) is received after the snapshot has been created, and (ii) has a serial number larger than the specified serial number, and requesting the client node to re-issue the write command with a new serial number.

5. The method according to claim 1, wherein creating the copy of the data volume comprises initializing the copy in the second storage node from the snapshot, and then executing the one or more received write commands in the initialized copy.

6. A computing system, comprising:

a client node; and

at least a first storage node and a second storage node,

wherein the client node is configured to send write commands to the first storage node for storing data in a data volume on the first storage node, and wherein, in response to a decision to create a copy of the data volume on the second storage node, the client node is

configured to send the write commands both to the first storage node and to the second storage node, the first storage node is configured to create and send to the second storage node a snapshot of the data volume, and the second storage node is configured to create the copy of the data volume by the second storage node, based on the snapshot and on one or more of the write commands received at the second storage node.

7. The system according to claim 6, wherein the first storage node is configured to include in the snapshot at least a write command having a specified serial number, and wherein the second storage node is configured to receive the snapshot and create the copy of the data volume from the received snapshot and the received write commands whose serial numbers are larger than the specified serial number.

8. The system according to claim 7, wherein the second storage node is configured to send to the first storage node a snapshot request that specifies the specified serial number, and to discard any received write commands whose serial numbers are smaller than or equal to the specified serial number.

9. The system according to claim 7, wherein the first storage node is configured to detect a write command that (i) is received after the snapshot has been created, and (ii) has a serial number larger than the specified serial number, and to request the client node to re-issue the write command with a new serial number.

10. The system according to claim 6, wherein the second storage node is configured to create the copy of the data volume by initializing the copy from the snapshot, and then executing the one or more received write commands in the initialized copy.

11. A storage node, comprising:

an interface for communicating with a client node and with an additional storage node over a computer network; and

a processor, which is configured to receive from the client node write commands having respective serial numbers for storing data in a data volume, to receive from the additional storage node a request to create a snapshot of the data volume that includes at least a write command having a specified serial number, and to create and send the snapshot to the additional storage node.

12. The storage node according to claim 11, wherein the processor is configured to detect a write command that (i) is received after the snapshot has been created, and (ii) has a serial

number larger than the specified serial number, and to request the client node to re-issue the write command with a new serial number.

13. A storage node, comprising:

an interface for communicating with a client node and with an additional storage node
5 over a computer network; and

a processor, which is configured, in response to a decision to create on the storage node a copy of a data volume that is stored on the additional storage node, to start receiving write commands for storing data in the copy of the data volume, to request the additional storage node to create and send a snapshot of the data volume, and to create the copy of the data
10 volume based on the snapshot and on one or more of the received write commands.

14. The storage node according to claim 13, wherein the processor is configured to request the additional storage node to include in the snapshot at least a write command having a specified serial number, and to create the copy of the data volume from the received snapshot and the received write commands whose serial numbers are larger than the specified serial
15 number.

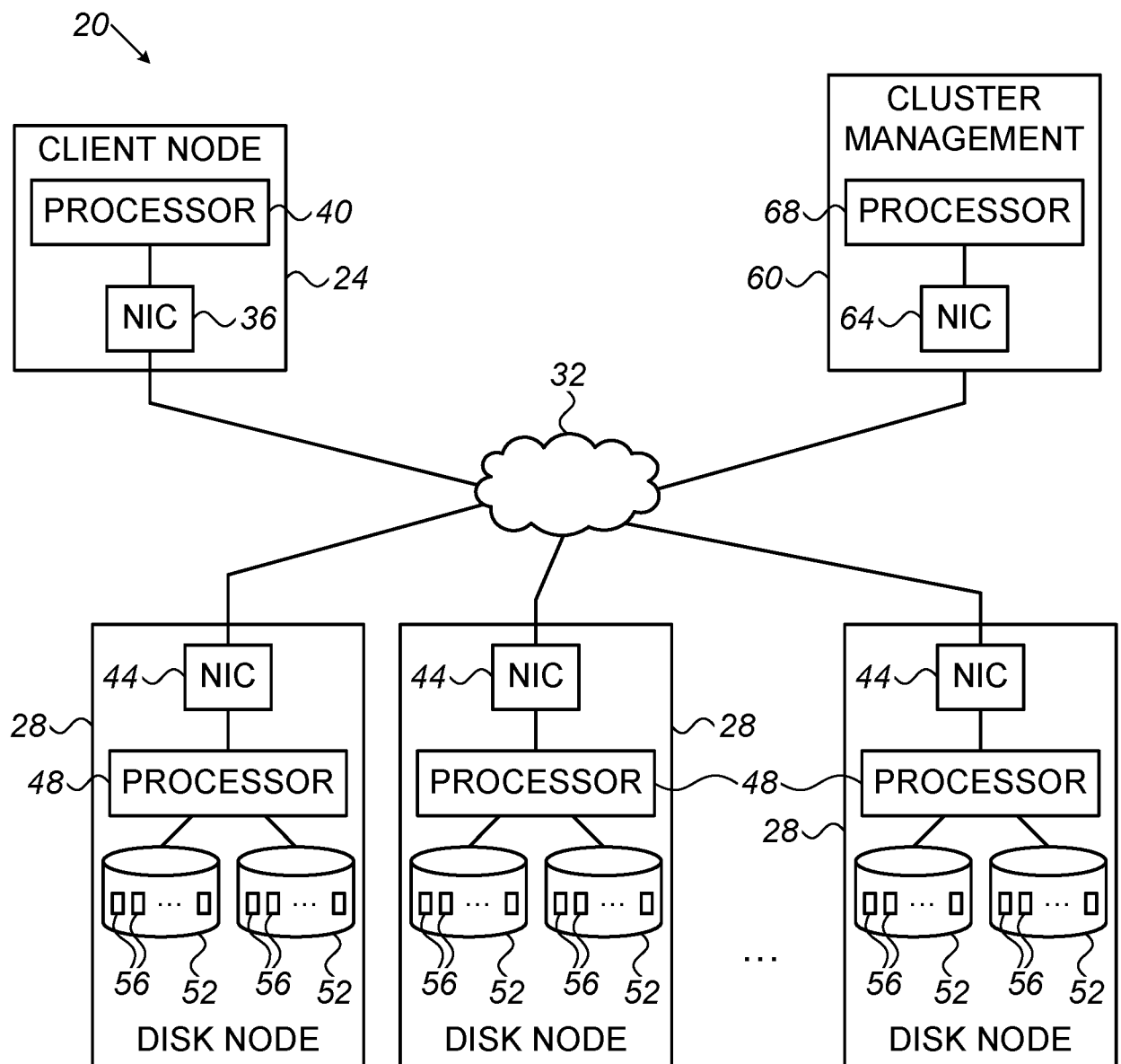


FIG. 1

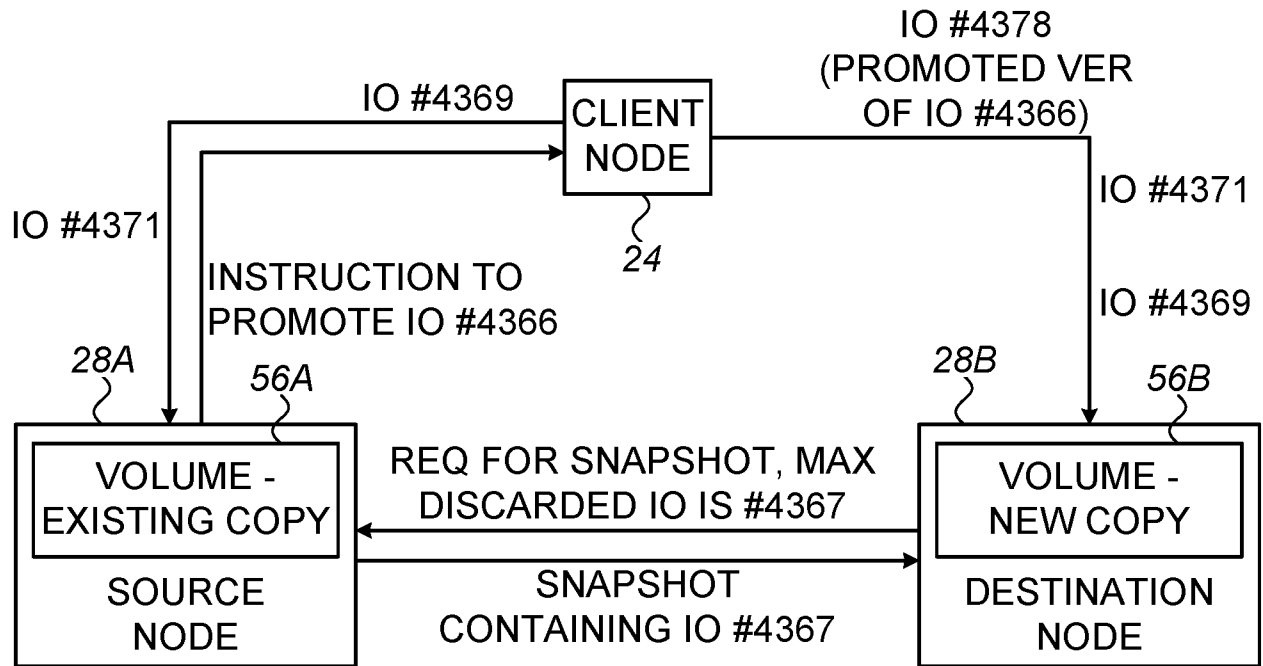
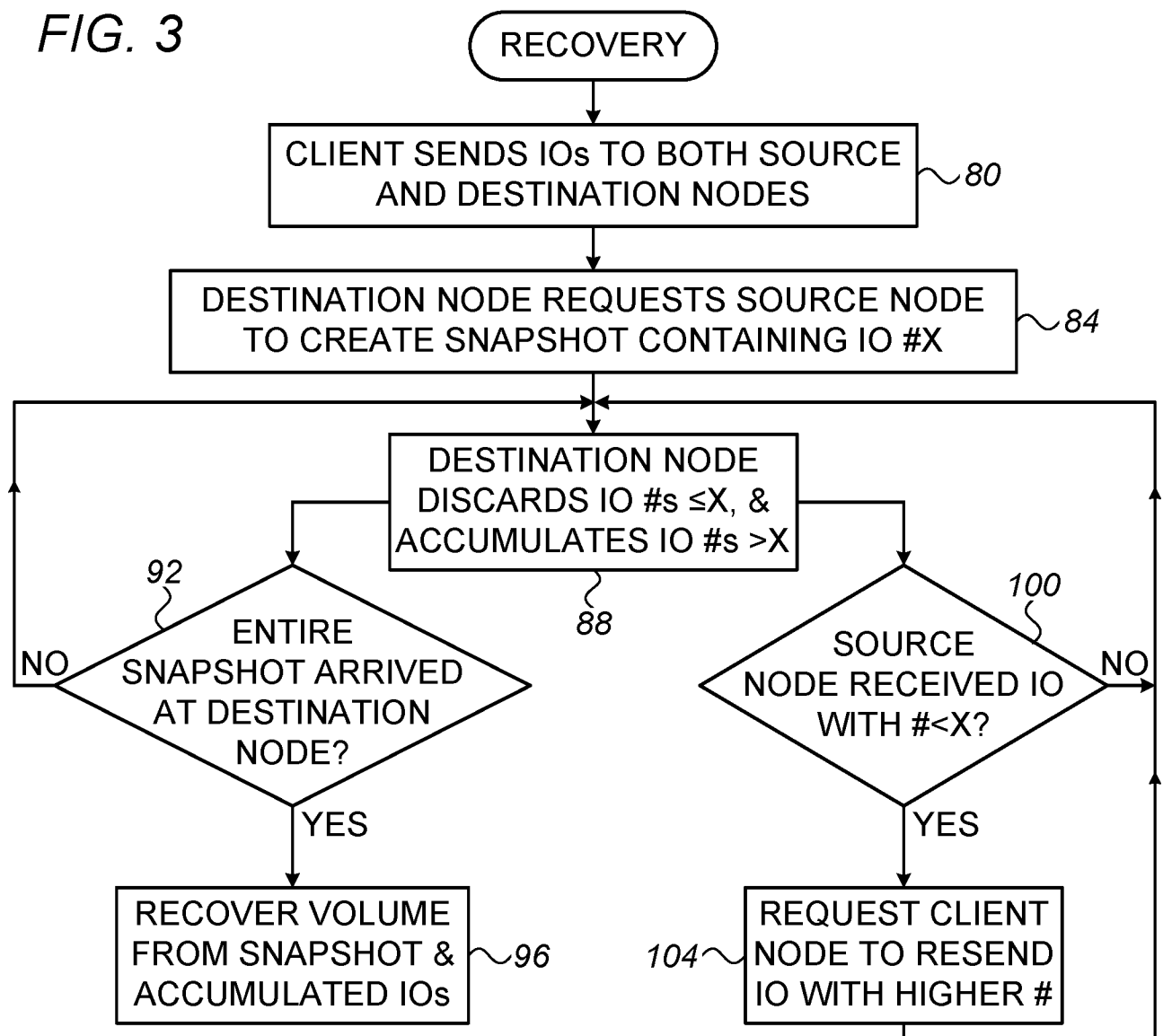


FIG. 2

FIG. 3



INTERNATIONAL SEARCH REPORT

International application No.

PCT/IB2015/059510

A. CLASSIFICATION OF SUBJECT MATTER

IPC (2016.01) G06F 12/00, G06F 12/16

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC (2016.01) G06F 12/00, G06F 12/16

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

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

Databases consulted: THOMSON INNOVATION, Esp@cenet, Google Patents

Search terms used: write command, storage, snapshot, data volume, copy, recovery, discarding, resending, reissue, number, identification, serial

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	CA 2632935 A1 COMMVault SYSTEMS, INC. 05 Jul 2007 (2007/07/05) The whole document	1-14
A	US 2004010487 A1 Prahlad et al. 15 Jan 2004 (2004/01/15) The whole document	1-14
A	US 8370302 B2 Otani 05 Feb 2013 (2013/02/05) The whole document	1-14

☐ Further documents are listed in the continuation of Box C.

☒ See patent family annex.

* Special categories of cited documents:

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

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

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

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

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

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

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

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

“&” document member of the same patent family

Date of the actual completion of the international search

29 Mar 2016

Date of mailing of the international search report

31 Mar 2016

Name and mailing address of the ISA:

Israel Patent Office

Technology Park, Bldg.5, Malcha, Jerusalem, 9695101, Israel

Facsimile No. 972-2-5651616

Authorized officer

GORBUNOVA Yelena

Telephone No. 972-2-5651669

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/IB2015/059510

Patent document cited search report			Publication date	Patent family member(s)			Publication Date
CA	2632935	A1	05 Jul 2007	CA	2632935	A1	05 Jul 2007
				CA	2632935	C	04 Feb 2014
				AU	2006331932	A1	05 Jul 2007
				AU	2006331932	A2	25 Sep 2008
				AU	2006331932	B2	06 Sep 2012
				EP	1974296	A1	01 Oct 2008
				US	2007185937	A1	09 Aug 2007
				US	7617253	B2	10 Nov 2009
				US	2007185939	A1	09 Aug 2007
				US	7617262	B2	10 Nov 2009
				US	2007185852	A1	09 Aug 2007
				US	7636743	B2	22 Dec 2009
				US	2007185938	A1	09 Aug 2007
				US	7651593	B2	26 Jan 2010
				US	2007226438	A1	27 Sep 2007
				US	7661028	B2	09 Feb 2010
				US	2007183224	A1	09 Aug 2007
				US	7870355	B2	11 Jan 2011
				US	2010094808	A1	15 Apr 2010
				US	7962455	B2	14 Jun 2011
				US	2007186068	A1	09 Aug 2007
				US	7962709	B2	14 Jun 2011
				US	2010049753	A1	25 Feb 2010
				US	8121983	B2	21 Feb 2012
				US	2010100529	A1	22 Apr 2010
				US	8271830	B2	18 Sep 2012
				US	2010122053	A1	13 May 2010
				US	8285684	B2	09 Oct 2012
				US	2013006926	A1	03 Jan 2013
				US	8656218	B2	18 Feb 2014
				US	2013006938	A1	03 Jan 2013

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/IB2015/059510

Patent document cited search report	Publication date	Patent family member(s)	Publication Date
		US 8793221 B2	29 Jul 2014
		US 2014324772 A1	30 Oct 2014
		US 9020898 B2	28 Apr 2015
		US 2014181029 A1	26 Jun 2014
		US 9208210 B2	08 Dec 2015
		US 2015199375 A1	16 Jul 2015
		WO 2007075587 A2	05 Jul 2007
<hr/>			
US 2004010487 A1	15 Jan 2004	US 2004010487 A1	15 Jan 2004
		US 7346623 B2	18 Mar 2008
		DE 60232165 D1	10 Jun 2009
		EP 1436873 A1	14 Jul 2004
		EP 1436873 A4	18 Apr 2007
		EP 1436873 B1	29 Apr 2009
		HK 1062502 A1	24 Dec 2009
		JP 2005505045 A	17 Feb 2005
		US 2008183775 A1	31 Jul 2008
		US 8055625 B2	08 Nov 2011
		US 2012089798 A1	12 Apr 2012
		US 8442944 B2	14 May 2013
		US 2013246361 A1	19 Sep 2013
		US 8655846 B2	18 Feb 2014
		US 2014114922 A1	24 Apr 2014
		WO 03028183 A1	03 Apr 2003
<hr/>			
US 8370302 B2	05 Feb 2013	US 2010306174 A1	02 Dec 2010
		US 8370302 B2	05 Feb 2013
<hr/>			