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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
16 December 2010 (16.12.2010)

PCT

(10) International Publication Number
WO 2010/142634 A1

(51) International Patent Classification:
G06F 11/10 (2006.01)

(21) International Application Number:
PCT/EP2010/057904

(22) International Filing Date:
7 June 2010 (07.06.2010)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
12/481,953 10 June 2009 (10.06.2009) US

(71) Applicant *(for all designated States except US)*: **INTERNATIONAL BUSINESS MACHINES CORPORATION** [US/US]; New Orchard Road, Armonk, New York 10504 (US).

(71) Applicant *(for MG only)*: **IBM UNITED KINGDOM LIMITED** [GB/GB]; PO Box 41, North Harbour, Portsmouth Hampshire PO6 3AU (GB).

(72) Inventor; and

(75) Inventor/Applicant *(for US only)*: **ZVIBEL, Liran** [IL/IL]; IBM Israel, Azrieli Centre 1, Round Building, Floor 15, Tel Aviv, 67021 Israel (IL).

(74) Agent: **STRETTON, Peter, John**; IBM United Kingdom Limited, Intellectual Property Law, Hursley Park, Winchester Hampshire SO21 2JN (GB).

(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, 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, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, RO, RS, RU, 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,

[Continued on next page]

(54) Title: DATA VERIFICATION USING CHECKSUM SIDEFILE

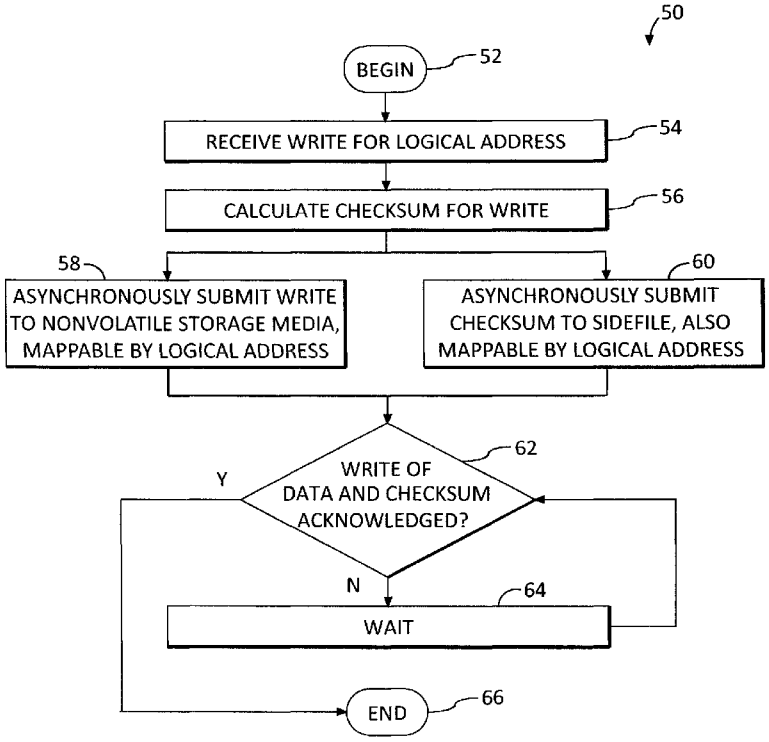

FIG. 2A

(57) Abstract: Exemplary method, system, and computer program product embodiments for data verification in a storage system are provided. A read of data is asynchronously submitted to nonvolatile storage media. A read of a first checksum signature is submitted to a solid state, sidefile memory location of a storage controller in the storage subsystem. The first checksum signature is representative of the data previously written to the nonvolatile storage media. A second checksum signature is calculated from the read of the data. The first and second checksum signatures are compared. If a match is not determined, a critical event is reported.

WO 2010/142634 A1

GM, KE, LR, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, 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, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

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

DATA VERIFICATION USING CHECKSUM SIDEFILE

BACKGROUND OF THE INVENTION

Field of the Invention

The present invention relates in general to computers, and more particularly to a method, system, and computer program product for facilitating data verification using a checksum in conjunction with a sidefile in data processing computer storage subsystems.

Description of the Related Art

A checksum or has sum is a fixed-size datum computed from an arbitrary block of digital data for the purpose of detecting accidental errors that may have been introduced during its transmission or storage. The integrity of the data may be checked at any time by recomputing the checksum and comparing it with the stored checksum. If the checksums do not match, the data is determined to be altered.

In data processing computer storage subsystems, a storage controller often computes a checksum signature such as a Cyclic Redundancy Check (CRC). A CRC is an error-detecting code. Its computation resembles a long division operation in which the quotient is discarded and the remainder becomes the result, with the important distinction that the arithmetic used is the carry-less arithmetic of a finite field. The length of the remainder is always less than or equal to the length of the divisor, which therefore determines how long the result can be. The definition of a particular CRC specifies the divisor to be used, among other things.

SUMMARY OF THE INVENTION

In a first aspect, the present invention provides method for data verification in a storage subsystem by at least one processor device in communication with at least one memory device, the method comprising: asynchronously submitting a read of data to nonvolatile

storage media; asynchronously submitting a read of a first checksum signature to a solid state, sidefile memory location of a storage controller in the storage subsystem, wherein the first checksum signature is representative of the data previously written to the nonvolatile storage media; calculating a second checksum signature from the read of the data; and comparing the first and second checksum signatures, wherein if a match is not determined, reporting a critical event.

Preferably, asynchronously submitting the read of the first checksum signature to the solid state, sidefile memory location includes submitting the read to at least a portion of a solid state disk (SSD) device in communication with the storage controller. Preferably, asynchronously submitting the read of the first checksum signature to the solid state, sidefile memory location includes submitting the read to at least a portion of a flash memory device in communication with the storage controller. Preferably, asynchronously submitting the read of the first checksum signature to the solid state, sidefile memory location includes submitting the read to a storage module of the storage controller responsible for the nonvolatile storage media, wherein the solid state sidefile memory location is one of an available plurality of solid state sidefile memory locations associated with each of an available plurality of storage modules in the storage controller.

The method may further include, previous to asynchronously submitting the read of the data to the nonvolatile storage media: calculating the first checksum signature representative of the data, asynchronously submitting a write of the data to the nonvolatile storage media, and asynchronously submitting the first checksum signature to the solid state, sidefile memory location. The method may further include, previous to asynchronously submitting the read of the data to the nonvolatile storage media, receiving one of a read and a verify instruction for a logical address.

In a second aspect, there is provided a system for data verification in a storage subsystem having nonvolatile storage media, comprising: a solid state, sidefile device of a storage controller operable in the storage subsystem, the sidefile device providing a memory location in addition to the nonvolatile storage media, wherein the sidefile device is adapted for storing a checksum signature calculated for data written to the nonvolatile storage media.

The system may further include a storage module operable in the storage controller, wherein: the sidefile device is further adapted for providing the checksum signature to the storage module for data verification, the storage module is adapted for calculating an additional checksum signature for a subsequent read of the data from the nonvolatile storage media, and the storage module is further adapted for comparing the checksum signature to the additional checksum signature.

Preferably, the storage module is further adapted for, pursuant to comparing the checksum signature to the additional checksum signature, determining if the checksum signature and the additional checksum signature match, wherein if a match is not determined, reporting a critical event. Preferably, the storage module is further adapted for receiving one of a read and a verify instruction for a logical address. Preferably, the data written to the nonvolatile storage media and the checksum signature calculated for the data are mappable by logical address. Preferably, the sidefile device is one of an available plurality of sidefile devices, each associated with one storage module of an available plurality of storage modules of the storage controller. Preferably, the sidefile device comprises a solid state disk (SSD) device.

Preferably, the sidefile device comprises a flash memory device. Preferably, the checksum signature comprises a cyclic redundancy check (CRC).

In a third aspect, there is provided a computer program comprising computer program code to, when loaded into a computer and executed thereon, cause said computer to perform all the steps of the method of the first aspect.

The computer program of the third aspect may be embodied in the form of a computer program product for data verification in a storage subsystem by at least one processor device in communication with at least one memory device, the computer program product comprising a computer-readable storage medium having computer-readable program code portions stored therein, the computer-readable program code portions comprising: a first executable portion for asynchronously submitting a read of data to nonvolatile storage media; a second executable portion for asynchronously submitting a read of a first checksum signature to a solid state, sidefile memory location of a storage controller in the storage

subsystem, wherein the first checksum signature is representative of the data previously written to the nonvolatile storage media; a third executable portion for calculating a second checksum signature from the read of the data; and a fourth executable portion for comparing the first and second checksum signatures, wherein if a match is not determined, reporting a critical event.

Preferably, the second executable portion for asynchronously submitting the read of the first checksum signature to the solid state, sidefile memory location is further adapted for submitting the read to at least a portion of a solid state disk (SSD) device in communication with the storage controller. Preferably, the second executable portion for asynchronously submitting the read of the first checksum signature to the solid state, sidefile memory location is further adapted for submitting the read to at least a portion of a flash memory device in communication with the storage controller. Preferably, the second executable portion for asynchronously submitting the read of the first checksum signature to the solid state, sidefile memory location is further adapted for submitting the read to a storage module of the storage controller responsible for the nonvolatile storage media, wherein the solid state sidefile memory location is one of an available plurality of solid state sidefile memory locations associated with each of an available plurality of storage modules in the storage controller.

The computer program product may further include a fifth executable portion for, previous to asynchronously submitting the read of the data to the nonvolatile storage media: calculating the first checksum signature representative of the data, asynchronously submitting a write of the data to the nonvolatile storage media, and asynchronously submitting the first checksum signature to the solid state, sidefile memory location. The computer program product may further include a fifth executable portion for, previous to asynchronously submitting the read of the data to the nonvolatile storage media, receiving one of a read and a verify instruction for a logical address.

In a further embodiment there may be provided a preferred method of manufacturing a system for data verification in a storage subsystem having nonvolatile storage media, comprising: providing a solid state, sidefile device of a storage controller in the storage

subsystem, the sidefile device providing a memory location in addition to the nonvolatile storage media, wherein the sidefile device is adapted for storing a checksum signature calculated for data written to the nonvolatile storage media.

The method of manufacture may further include providing a storage module operable in the storage controller, wherein: the sidefile device is further adapted for providing the checksum signature to the storage module for data verification, the storage module is adapted for calculating an additional checksum signature for a subsequent read of the data from the nonvolatile storage media, and the storage module is further adapted for comparing the checksum signature to the additional checksum signature. Preferably, the storage module is further adapted for, pursuant to comparing the checksum signature to the additional checksum signature, determining if the checksum signature and the additional checksum signature match, wherein if a match is not determined, reporting a critical event. Preferably, the sidefile device comprises one of a solid state disk (SSD) device and a flash memory device.

Storage controllers typically place checksum signatures such as CRCs near the stored data, for example on the same nonvolatile storage media as the stored data. As a result, if the data becomes corrupted, the error may not be detectable by the CRC. For example, situations may arise where a write of data and/or the corresponding CRC does not occur (the data is not written). In such a case, the next read of this data will read the old data and/or the old CRC, and the system may determine that the old data and/or CRC is valid. In view of the foregoing, a need exists for a mechanism whereby checksum signatures such as CRCs may be retained, yet in a storage location apart from the stored data.

Accordingly, various exemplary method, system, and computer program product embodiments for data verification in a storage subsystem are provided. In one such exemplary embodiment, by way of example only, a read of data is asynchronously submitted to nonvolatile storage media. A read of a first checksum signature is submitted to a solid state, sidefile memory location of a storage controller in the storage subsystem. The first checksum signature is representative of the data previously written to the nonvolatile storage media. A second checksum signature is calculated from the read of the

data. The first and second checksum signatures are compared. If a match is not determined, a critical event is reported.

Related system and computer program product embodiments are also disclosed and provide additional advantages.

Brief Description of the Drawings

A preferred embodiment of the present invention will now be described, by way of example only, with reference to the accompanying drawings, in which:

Fig. 1 is a block diagram of an exemplary computing environment including a data processing storage subsystem in which various aspects of the following description and claimed subject matter may be implemented;

Fig. 2A is a flow chart diagram of an exemplary method for submitting a checksum signature to a sidefile;

Fig. 2B is a flow chart diagram of an exemplary method of writing data wherein a data error results;

Fig. 2C is a flow chart diagram of an exemplary method of writing data wherein an additional data error results; and

Fig. 3 is a flow chart diagram of an exemplary method for verifying data using a checksum signature submitted to a solid-state sidefile memory location according to the present invention.

Detailed Description of the Drawings

The illustrated embodiments below provide mechanisms for data verification using a sidefile storage device as a memory location to store checksum signature information in lieu of

placing the checksum signatures in close proximity to the stored data. Such a sidefile may be implemented using a small, high-performance and zero-seek time storage device, such as a solid-state drive (SSD) or flash memory device. The sidefile establishes a memory location in addition to the nonvolatile memory location in which the stored data is located. Accordingly, each checksum is recorded in the sidefile instead of being recorded on the storage controller disks. As a result, if data is never written to disk, the corresponding checksum is still written to the sidefile location. The associated error established by the stored checksum is thus ensured to be discovered.

Turning now to the drawings, reference is initially made to Fig. 1, which is a block diagram of an exemplary data processing storage subsystem 10, in accordance with a disclosed embodiment of the invention. The particular subsystem shown in Fig. 1 is presented to facilitate an explanation of the invention. However, as the skilled artisan will appreciate, the invention can be practiced using other storage subsystems with diverse architectures and capabilities.

The storage subsystem 10 receives, from one or more host computers 12, I/O requests, which are commands to read or write data at logical addresses on logical volumes. Any number of host computers 12 are coupled to the storage subsystem 10 by any means known in the art, for example, using a network. Herein, by way of example, the host computers 12 and the storage subsystem 10 are assumed to be coupled by a storage area network (SAN) 16 incorporating data connections 14 and host bus adapters (HBAs) 18. The logical addresses specify a range of data blocks within a logical volume, each block herein being assumed by way of example to contain 512 bytes. For example, a 10 KB data record used in a data processing application on a host computer would require 20 blocks, which the host computer might specify as being stored at a logical address comprising blocks 1000 through 1019 of a logical volume. The storage subsystem 10 typically operates in, or as, a network attached storage (NAS) or a SAN system.

The storage subsystem 10 comprises a clustered storage controller 24 coupled between the SAN 16 and a private network 36 using data connections 20 and 34, respectively, and incorporating adapters 22 and 32, again respectively. Clustered storage controller 24

implements clusters of storage modules 26, each of whom includes an interface 28 (in communication between adapters 22 and 32), and a cache 30. Each storage module 26 is responsible for a number of disks 40 by way of data connection 38 as shown.

As described previously, each storage module 26 further comprises a cache 30. However, it will be appreciated that the number of caches used in the storage subsystem 10 and in conjunction with clustered storage controller 24 may be any convenient number. While all caches 30 in the storage subsystem 10 may operate in substantially the same manner and to comprise substantially similar elements, this is not a requirement. Each of the caches 30 is typically, but not necessarily approximately equal in size and is assumed to be coupled, by way of example, in a one-to-one correspondence with a set of physical storage units, which are typically disks. In one embodiment, the disks 40 may comprise such disks. Those skilled in the art will be able to adapt the description herein to caches of different sizes, and to caches and storage devices in other correspondences, such as the many-to-many correspondence described in U.S. Patent Application Publication No. 2005/0015566, entitled "Data Allocation in a Distributed Storage System."

Each set of physical storage comprises multiple slow and/or fast access time mass storage devices, herein below assumed to be multiple hard disks. Fig. 1 shows the caches 30 coupled to respective sets of physical storage. Typically, the sets of physical storage comprise one or more disks 40, which can have different performance characteristics. In response to an I/O command, the cache 30, by way of example, may read or write data at addressable physical locations of physical storage. In the embodiment of Fig. 1, the caches 30 are shown to exercise certain control functions over the physical storage. These control functions may alternatively be realized by hardware devices such as disk controllers, which are linked to the caches 30.

In an embodiment of the present invention, the routing of logical addresses is implemented according to methods described in the above-referenced U.S. Patent Application Publication No. 2005/0015566. Routing records, indicating the association of logical addresses of logical volumes with partitions and the association of the partitions with caches, are distributed by the SAN 16 to one or more generally similar network interfaces 28 of the

storage modules 26. It will be understood that the storage subsystem 10, and thereby, the clustered storage controller 24, may comprise any convenient number of network interfaces 28. Subsequent to the formation of the disks 40, the network interfaces 28 receive I/O commands from the host computers 12 specifying logical addresses of the disks 40. The network interfaces use the routing records to break the commands into I/O instructions, or command subsets, that are then distributed among the caches 30.

Each storage module 26 is operative to monitor its state, including the states of associated caches 30, and to transmit configuration information to other components of the storage subsystem 10 for example, configuration changes that result in blocking intervals, or limit the rate at which I/O requests for the sets of physical storage are accepted, as explained in further detail herein below.

Routing of commands and data from the HBAs 18 to the clustered storage controller 24 to each cache 30 is typically performed over a network and/or a switch. Herein, by way of example, the HBAs 18 may be coupled to the storage modules 26 by at least one switch (not shown) of the SAN 16 which can be of any known type having a digital cross-connect function. In additional implementations, the HBAs 18 may be directly connected to the storage modules 26.

Data having contiguous logical addresses are generally distributed among the disks 40. This can be accomplished using the techniques disclosed in the above-referenced U.S. Patent Application Publication No. 2005/0015566. Alternatively, the data can be distributed using other algorithms, e.g., byte or block interleaving. In general, this increases bandwidth, for instance, by allowing a volume in a SAN or a file in network attached storage to be read from or written to more than one disk at a time. However, this technique requires coordination among the various disks, and in practice may require complex provisions for disk failure, and a strategy for dealing with error checking information, e.g., a technique for storing parity information relating to distributed data. Indeed, when logical unit partitions are distributed in sufficiently small granularity, data associated with a single logical unit may span all of the disks 40.

While not explicitly shown for purposes of illustrative simplicity, the skilled artisan will appreciate that in some embodiments, the clustered storage controller 24 may be adapted for implementation in conjunction with certain hardware, such as a rack mount system, a midplane, and/or a backplane. Indeed, the private network 36 in one embodiment may be implemented using a backplane. Additional hardware such as the aforementioned switches, processors, controllers, memory devices, and the like may also be incorporated into the clustered storage controller 24 and elsewhere within the storage subsystem 10, again as the skilled artisan will appreciate. Further, a variety of software components, operating systems, firmware, and the like may be integrated.

Each storage module 26 in the clustered storage controller 24 includes a sidefile 42. As previously mentioned, the sidefile 42 may include such storage devices as a solid-state drive (SSD) storage device or a flash memory device such as a Compact Flash (CF) storage device. The storage capacity of the sidefile 42 is not required to be nearly as large as that of the nonvolatile storage disks 40 as the checksums associated with stored write data are correspondingly small in comparison (typically a few bytes). As a result, a small, high-performance device may be implemented in conjunction with each storage module in a cost effective manner.

While the illustrated embodiment shows sidefiles 42 associated with each storage module 26, the skilled artisan will appreciate that other embodiments may implement sidefiles 42 in differing ways. For example, a single sidefile device may store checksum information for data stored in a number of volumes associated with a number of storage modules.

Figs. 2A-2C, and 3, following depict exemplary methods for implementing sidefile mechanisms to improve data verification. As one skilled in the art will appreciate, various steps in the methods may be implemented in differing ways to suit a particular application. In addition, the described methods may be implemented by various means, such as hardware, software, firmware, or a combination thereof operational on or otherwise associated with the computing environment. For example, the methods may be implemented at least partially as a computer program product including a computer-readable storage medium having computer-readable program code portions stored therein. The computer-

readable storage medium may include disk drives, flash memory, digital versatile disks (DVDs), compact disks (CDs), and other types of storage mediums.

Turning to Fig. 2A, reference is made to an exemplary method 50 for performing data writes implementing the sidefile device illustrated in Fig. 1, previously. Method 50 begins (step 52) with a write being received for a certain logical address (step 54). A checksum signature is calculated by the system for the write (step 56). The write is asynchronously submitted to nonvolatile storage media, mappable by the logical address (step 58). In addition, the checksum is asynchronously submitted to the sidefile to be recorded (step 60). Both steps 58 and 60 occur asynchronously with each other so as to allow for increased system performance. The checksum signature is also mappable by logical address. Method 50 queries whether the write of the data and the checksum signature have been acknowledged (step 62). If no, the system waits (step 64) until this is the case. The method 50 then ends (step 66).

Turning now to Figs. 2B and 2C, reference is made to an exemplary methods for writing data in which a data error occurs. In both exemplary methods, the data errors may be not discoverable but for the mechanisms of the present invention implementing the sidefile device previously described. Turning first to Fig. 2B, method 100 begins (step 102) with a write received for a logical address (step 104). The applicable checksum is calculated for the write (step 106), and the write is asynchronously submitted to the nonvolatile storage media, mappable by the logical address (step 108). In the present example however, the data is not written, yet an acknowledgement is received (step 110). Asynchronous with the write of the data, the checksum signature is submitted to the sidefile for storage (again mappable by the logical address) (step 112). The system waits until the writes of both data and checksum are acknowledged (steps 114, 116), and then the method 100 ends.

Turning now to Fig. 2C, method 150 begins (step 152) with a write received for a logical address (step 154). The applicable checksum is calculated for the write (step 156), and the write is asynchronously submitted to the nonvolatile storage media, mappable by the logical address (step 158). Asynchronous with the write of the data, the checksum signature is submitted to the sidefile for storage (again mappable by the logical address) (step 160).

Subsequent to the submittal of the checksum signature, and in similar fashion to the previous example, the data is not written, yet an acknowledgement is received that the data was written (step 162). The system waits until the writes of both data and checksum are acknowledged (steps 164, 166), and then the method 150 ends.

Turning now to Fig. 3, an exemplary method 200 for data verification using the sidefile mechanisms previously described is now shown. Method 200 allows for the detection of data errors resulting from either of the previous examples set forth in Figs. 2B and 2C, where write errors prevent data from being written, yet an acknowledgement is received that the data was written. Method 200 begins (step 202) with the receipt of a read/verify instruction for a logical address (step 204). The read is asynchronously submitted to the nonvolatile storage media, mappable by the logical address (step 206) while the read of the stored checksum, also mappable by the logical address, is asynchronously submitted to the sidefile location (step 208).

As a next step, the method 200 queries whether the read of data and checksum signature information is acknowledged (step 210), and waits (step 212) until this is the case. Next, an additional checksum signature is calculated for the read of data (step 214). The stored checksum is then compared against the checksum calculated for the read (step 216). If the two checksums match, then the data is verified and the method 200 ends (step 218). If not, then a critical event that system data has been corrupted is reported (step 220). The corrupted data is locked (step 222), and a read/verify error is reported (step 224). The method 200 ends (again, step 218).

Use of a sidefile location in conjunction with a storage controller to store checksum signature information provides an efficient and effective mechanism to improve data reliability in storage subsystems.

Some of the functional units described in this specification have been labeled as modules in order to more particularly emphasize their implementation independence. For example, a module may be implemented as a hardware circuit comprising custom VLSI circuits or gate arrays, off-the-shelf semiconductors such as logic chips, transistors, or other discrete

components. A module may also be implemented in programmable hardware devices such as field programmable gate arrays, programmable array logic, programmable logic devices, or the like.

Modules may also be implemented in software for storage for execution by various types of processors. An identified module of executable code may, for instance, comprise one or more physical or logical blocks of computer instructions, which may, for instance, be organized as an object, procedure, or function. Nevertheless, the executables of an identified module need not be physically located together, but may comprise disparate instructions stored in different locations which, when joined logically together, comprise the module and achieve the stated purpose for the module.

Indeed, a module of executable code may be a single instruction, or many instructions, and may even be distributed over several different code segments, among different programs, and across several memory devices. Similarly, operational data may be identified and illustrated herein within modules, and may be embodied in any suitable form and organized within any suitable type of data structure. The operational data may be collected as a single data set, or may be distributed over different locations including over different storage devices, and may exist, at least partially, as electronic signals on a system or network.

While one or more embodiments of the present invention have been illustrated in detail, the skilled artisan will appreciate that modifications and adaptations to those embodiments may be made without departing from the scope of the present invention as set forth in the following claims.

CLAIMS

1. A method for data verification in a storage subsystem by at least one processor device in communication with at least one memory device, the method comprising:

asynchronously submitting a read of data to nonvolatile storage media;

asynchronously submitting a read of a first checksum signature to a solid state, sidefile memory location of a storage controller in the storage subsystem, wherein the first checksum signature is representative of the data previously written to the nonvolatile storage media;

calculating a second checksum signature from the read of the data; and

comparing the first and second checksum signatures, wherein if a match is not determined, reporting a critical event.

2. The method of claim 1, wherein asynchronously submitting the read of the first checksum signature to the solid state, sidefile memory location includes submitting the read to at least a portion of a solid state disk (SSD) device in communication with the storage controller.

3. The method of claim 1, wherein asynchronously submitting the read of the first checksum signature to the solid state, sidefile memory location includes submitting the read to at least a portion of a flash memory device in communication with the storage controller.

4. The method of claim 1, wherein asynchronously submitting the read of the first checksum signature to the solid state, sidefile memory location includes submitting the read to a storage module of the storage controller responsible for the nonvolatile storage media, wherein the solid state sidefile memory location is one of an available plurality of solid state sidefile memory locations associated with each of an available plurality of storage modules in the storage controller.

5. The method of any preceding claim, further including, previous to asynchronously submitting the read of the data to the nonvolatile storage media:

calculating the first checksum signature representative of the data,

asynchronously submitting a write of the data to the nonvolatile storage media, and

asynchronously submitting the first checksum signature to the solid state, sidefile memory location.

6. The method of any preceding claim, further including, previous to asynchronously submitting the read of the data to the nonvolatile storage media, receiving one of a read and a verify instruction for a logical address.

7. A system for data verification in a storage subsystem having nonvolatile storage media, comprising:

a solid state, sidefile device of a storage controller operable in the storage subsystem, the sidefile device providing a memory location in addition to the nonvolatile storage media, wherein the sidefile device is adapted for storing a checksum signature calculated for data written to the nonvolatile storage media.

8. The system of claim 7, further including a storage module operable in the storage controller, wherein:

the sidefile device is further adapted for providing the checksum signature to the storage module for data verification,

the storage module is adapted for calculating an additional checksum signature for a subsequent read of the data from the nonvolatile storage media, and

the storage module is further adapted for comparing the checksum signature to the additional checksum signature.

9. The system of claim 8, wherein the storage module is further adapted for, pursuant to comparing the checksum signature to the additional checksum signature, determining if the checksum signature and the additional checksum signature match, wherein if a match is not determined, reporting a critical event.

10. The system of claim 8 or claim 9, wherein the storage module is further adapted for receiving one of a read and a verify instruction for a logical address.

11. The system of any of claims 7 to 10, wherein the data written to the nonvolatile storage media and the checksum signature calculated for the data are mappable by logical address.

12. The system of any of claims 7 to 11, wherein the sidefile device is one of an available plurality of sidefile devices, each associated with one storage module of an available plurality of storage modules of the storage controller.

13. The system of any of claims 7 to 12, wherein the sidefile device comprises one of a solid state disk (SSD) device or a flash memory device.

14. The system of any of claims 7 to 13, wherein the checksum signature comprises a cyclic redundancy check (CRC).

15. A computer program comprising computer program code to, when loaded into a computer system and executed thereon, cause said computer system to perform all the steps of the method of any of claims 1 to 6.

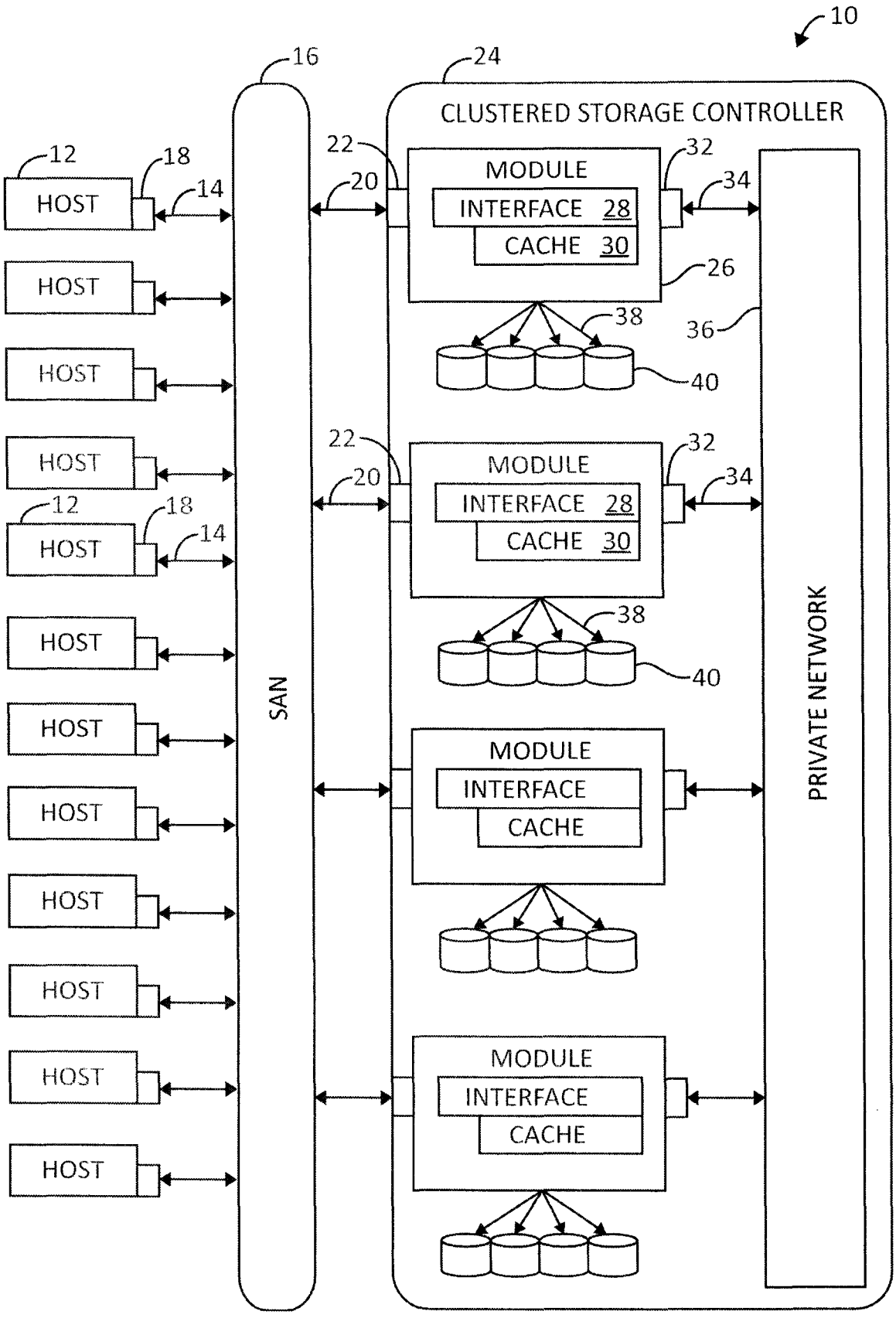
10
16
24
CLUSTERED STORAGE CONTROLLER
12
18
14
HOST
22
20
MODULE
INTERFACE 28
CACHE 30
32
34
26
38
36
40
SAN
PRIVATE NETWORK

FIG. 1

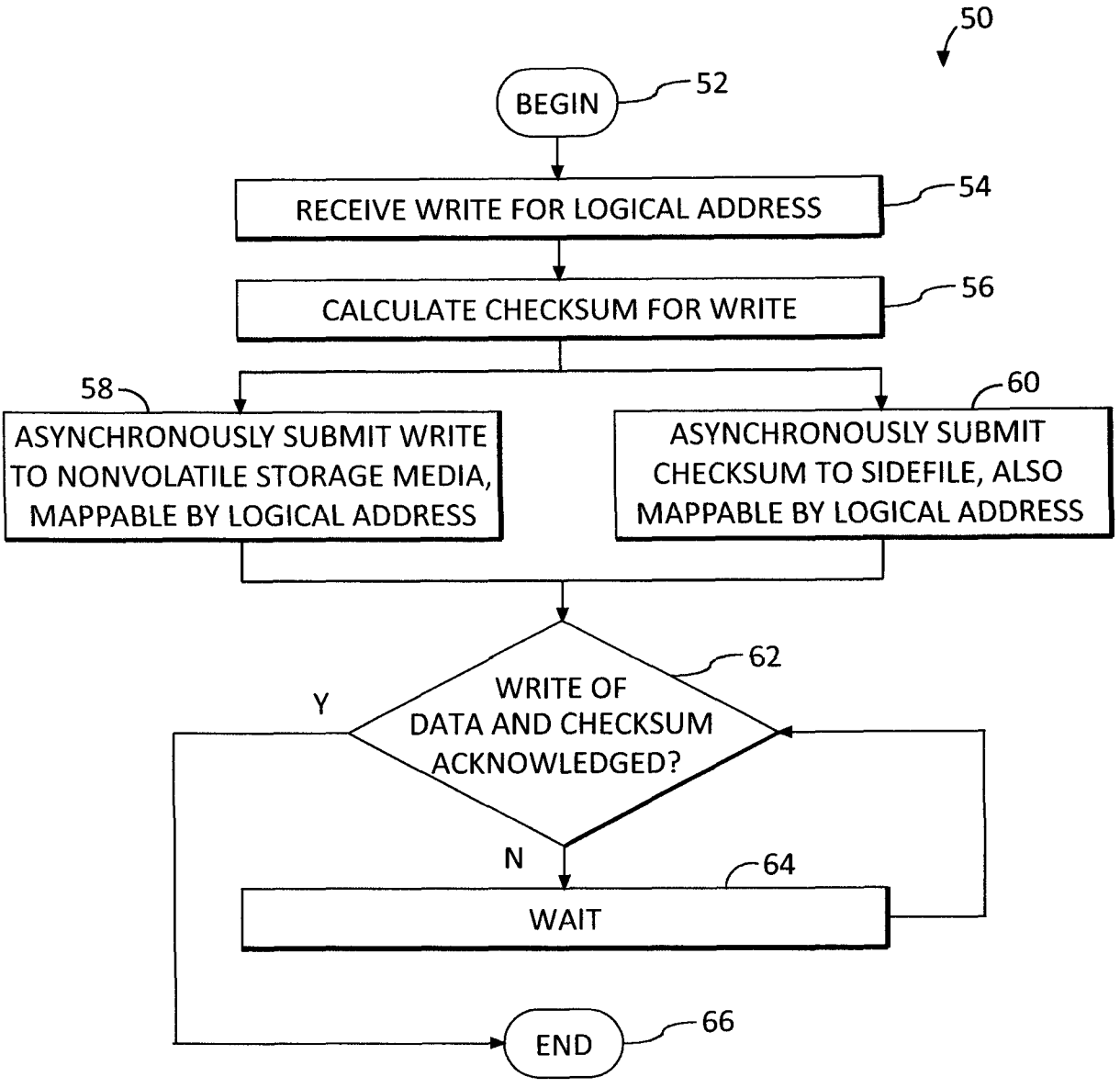
50
BEGIN
52
RECEIVE WRITE FOR LOGICAL ADDRESS
54
CALCULATE CHECKSUM FOR WRITE
56
58
ASYNCHRONOUSLY SUBMIT WRITE TO NONVOLATILE STORAGE MEDIA, MAPPABLE BY LOGICAL ADDRESS
60
ASYNCHRONOUSLY SUBMIT CHECKSUM TO SIDEFILE, ALSO MAPPABLE BY LOGICAL ADDRESS
62
WRITE OF DATA AND CHECKSUM ACKNOWLEDGED?
Y
N
64
WAIT
END
66

FIG. 2A

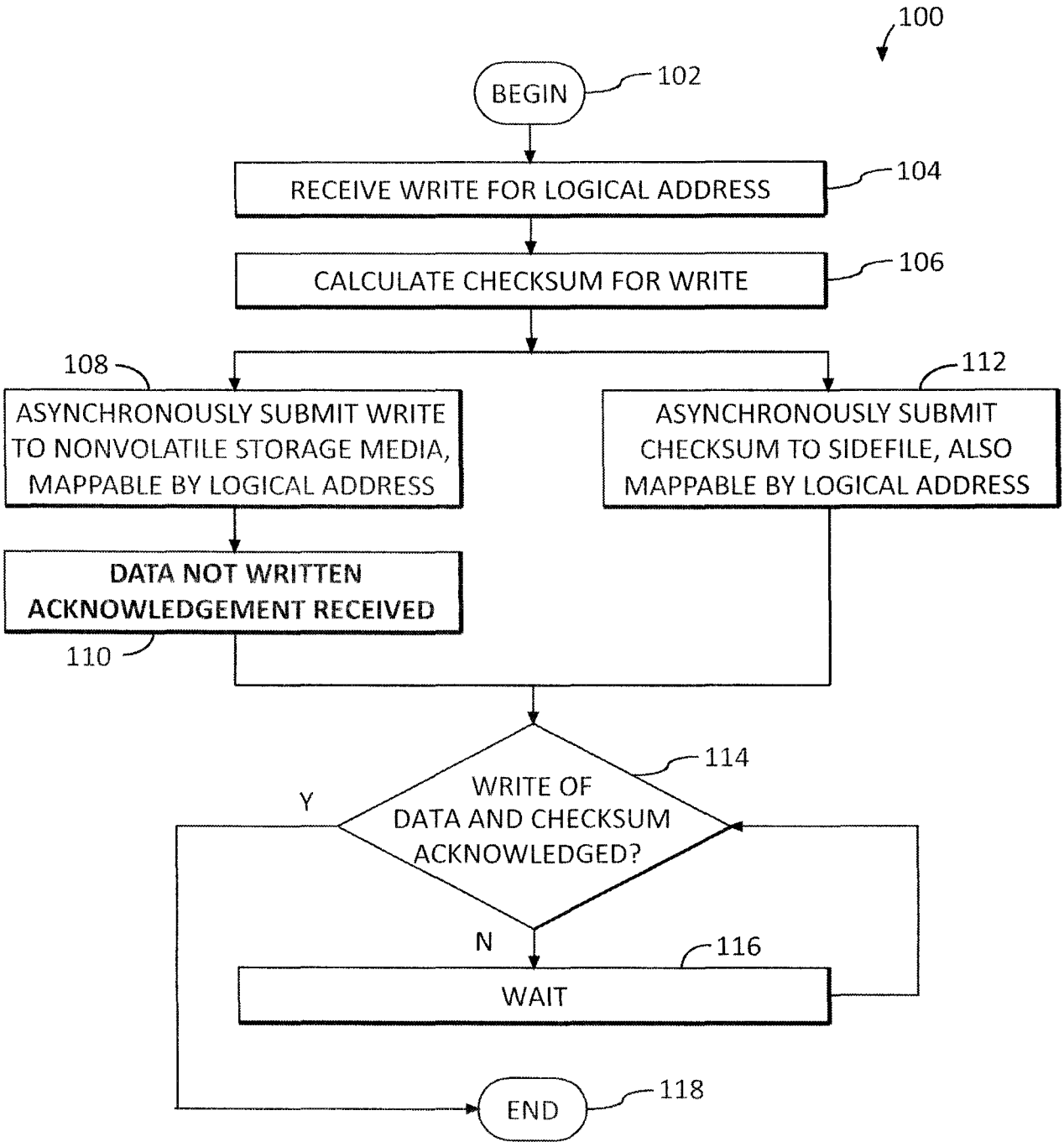
100
BEGIN
102
RECEIVE WRITE FOR LOGICAL ADDRESS
104
CALCULATE CHECKSUM FOR WRITE
106
108
ASYNCHRONOUSLY SUBMIT WRITE TO NONVOLATILE STORAGE MEDIA, MAPPABLE BY LOGICAL ADDRESS
112
ASYNCHRONOUSLY SUBMIT CHECKSUM TO SIDEFILE, ALSO MAPPABLE BY LOGICAL ADDRESS
DATA NOT WRITTEN ACKNOWLEDGEMENT RECEIVED
110
114
WRITE OF DATA AND CHECKSUM ACKNOWLEDGED?
Y
N
116
WAIT
END
118

FIG. 2B

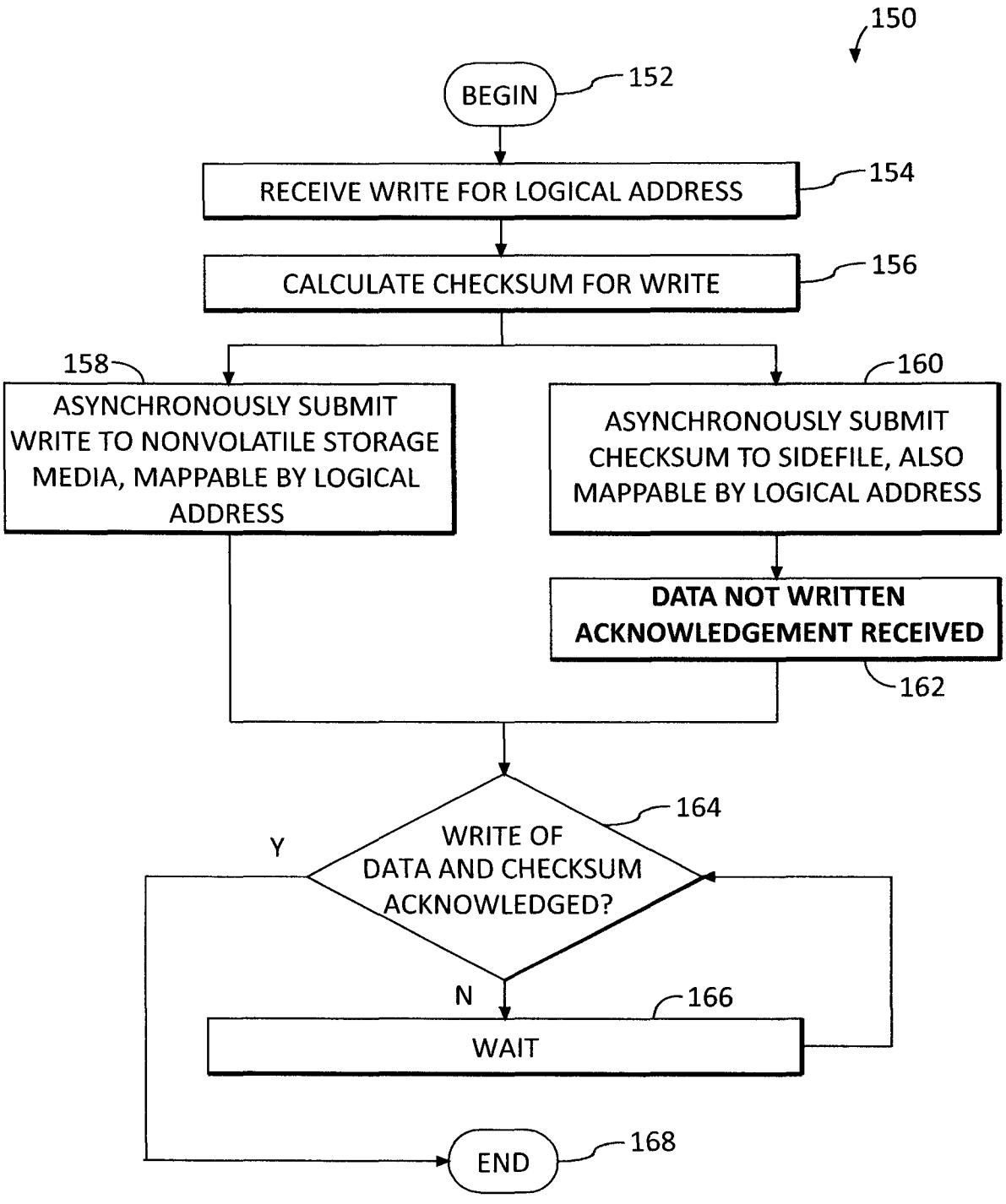
150
BEGIN
152
RECEIVE WRITE FOR LOGICAL ADDRESS
154
CALCULATE CHECKSUM FOR WRITE
156
158
ASYNCHRONOUSLY SUBMIT WRITE TO NONVOLATILE STORAGE MEDIA, MAPPABLE BY LOGICAL ADDRESS
160
ASYNCHRONOUSLY SUBMIT CHECKSUM TO SIDEFILE, ALSO MAPPABLE BY LOGICAL ADDRESS
DATA NOT WRITTEN ACKNOWLEDGEMENT RECEIVED
162
164
Y
WRITE OF DATA AND CHECKSUM ACKNOWLEDGED?
N
166
WAIT
END
168

FIG. 2C

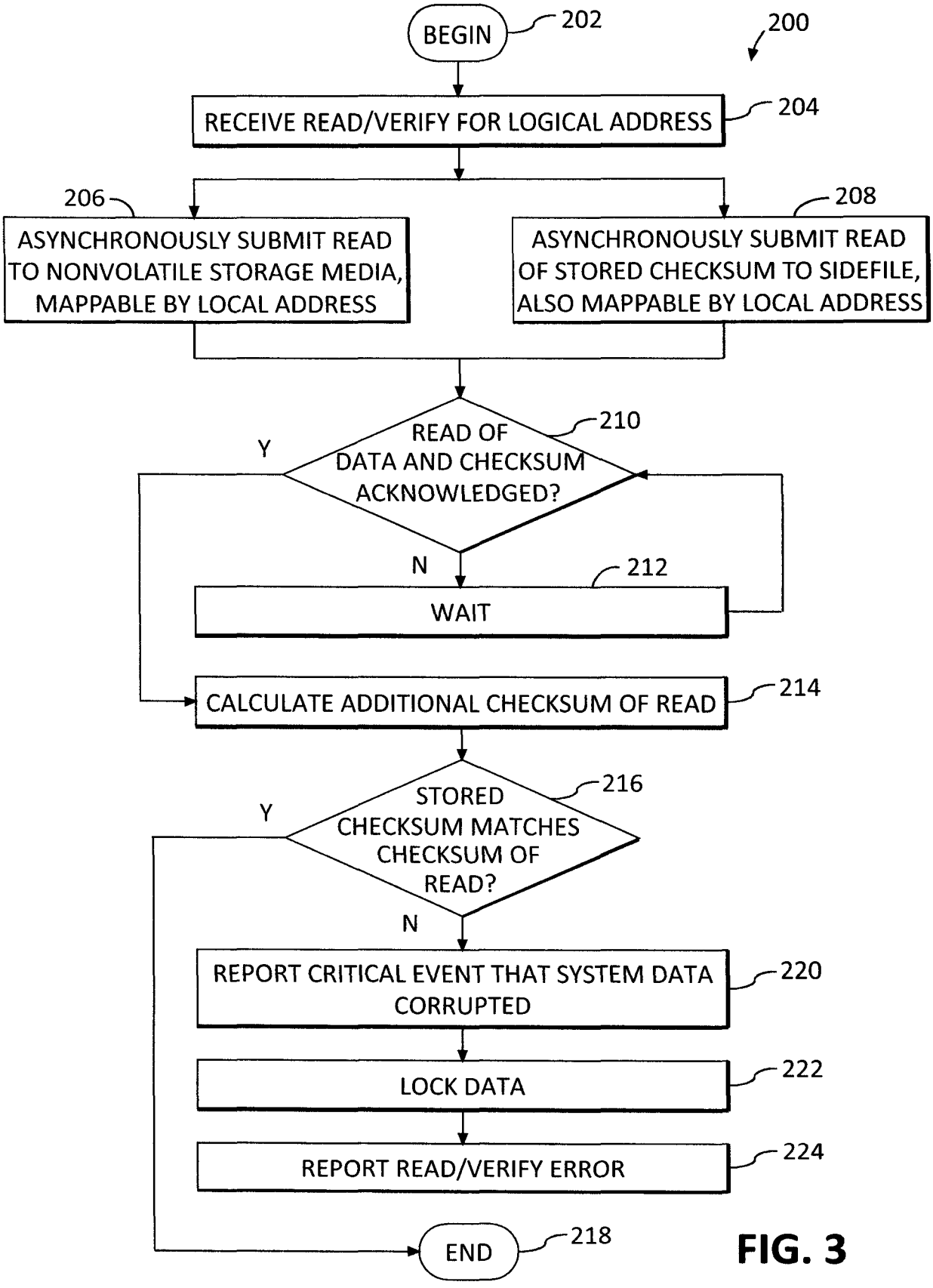
BEGIN
202
200
RECEIVE READ/VERIFY FOR LOGICAL ADDRESS
204
206
ASYNCHRONOUSLY SUBMIT READ TO NONVOLATILE STORAGE MEDIA, MAPPABLE BY LOCAL ADDRESS
208
ASYNCHRONOUSLY SUBMIT READ OF STORED CHECKSUM TO SIDEFILE, ALSO MAPPABLE BY LOCAL ADDRESS
READ OF DATA AND CHECKSUM ACKNOWLEDGED?
210
Y
N
212
WAIT
CALCULATE ADDITIONAL CHECKSUM OF READ
214
STORED CHECKSUM MATCHES CHECKSUM OF READ?
216
Y
N
REPORT CRITICAL EVENT THAT SYSTEM DATA CORRUPTED
220
LOCK DATA
222
REPORT READ/VERIFY ERROR
224
END
218

FIG. 3

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2010/057904

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F11/10
ADD.

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

B. FIELDS SEARCHED

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

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

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)
EPO-Internal, WPI Data, IBM-TDB, INSPEC, COMPENDEX

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 2007/226588 A1 (LEE SHEA-YUN [KR] ET AL) 27 September 2007 (2007-09-27) paragraphs [0008] - [0014], [0031] - [0034], [0038] - [0044], [0055], [0058] - [0066], [0071] - [0073] claims 1-3,6,7 figures 1,2,6,7,8-10	1-15
Y	US 2003/028723 A1 (SEGEV DANIT [IL] ET AL) 6 February 2003 (2003-02-06) paragraphs [0004], [0008] - [0013], [0039] - [0044] claims 1,13 figures 1,2	1-15

☐ 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 document but published on or after the international filing date

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

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

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

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

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

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

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
24 July 2010	30/07/2010

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

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/EP2010/057904

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2007226588 A1	27-09-2007	US 2008016428 A1	17-01-2008
US 2003028723 A1	06-02-2003	NONE	