

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
9 March 2006 (09.03.2006)

PCT

(10) International Publication Number
WO 2006/024583 A2

(51) International Patent Classification: Not classified

(21) International Application Number:
PCT/EP2005/053657

(22) International Filing Date: 27 July 2005 (27.07.2005)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
10/932,774 2 September 2004 (02.09.2004) 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) Inventors; and

(75) Inventors/Applicants (for US only): **BENHASE, Michael, Thomas** [US/US]; 4801 North Placita Lirio,

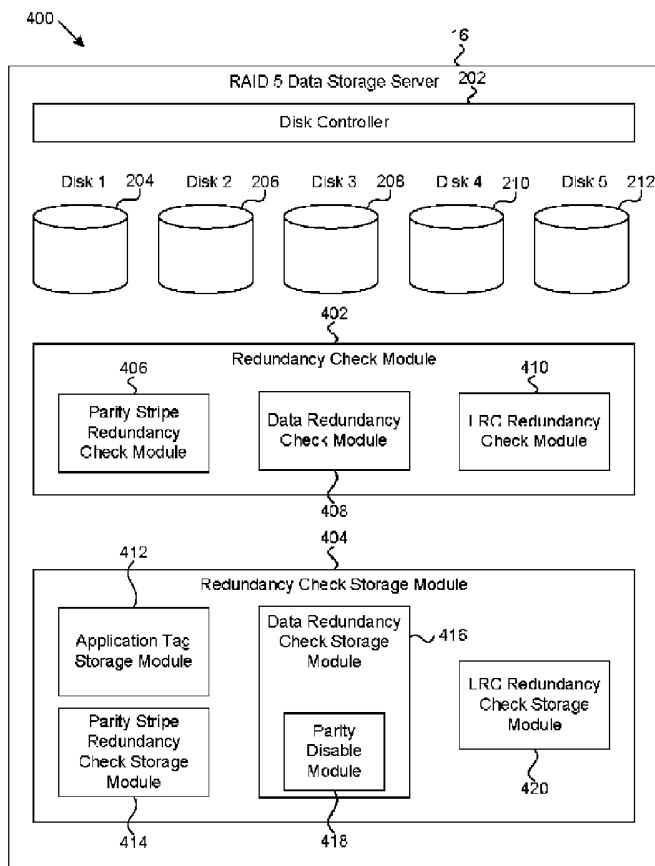
Tucson, Arizona 85749-9442 (US). **PALMER, Michael, John** [GB/GB]; 37 Launcelyn Close, North Baddesley, Southampton Hampshire SO52 9NP (GB). **VERDOORN, JR., William, Garrett** [US/US]; 704 29th Street, N.W., Rochester, Minnesota 55901 (US). **WALLS, Andrew, Dale** [US/US]; 7219 Via Bella, San Jose, California 95139 (US).

(74) Agent: **LING, Christopher, 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, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SM, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

[Continued on next page]

(54) Title: ERROR CHECKING AND RECOVERY OF TRANSMITTED DATA



(57) Abstract: An apparatus, system, and method are disclosed for data error checking and recovery in a data storage device. A redundancy check module creates a redundancy check for data on a data storage device in a SCSI End-to-End Checking Standard environment and a redundancy check storage module stores the redundancy check in a guard associated with the data.



(84) **Designated States** (*unless otherwise indicated, for every kind of regional protection available*): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— without international search report and to be republished upon receipt of that report

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

Description

ERROR CHECKING AND RECOVERY OF TRANSMITTED DATA

Technical Field

[0001] This invention relates to data error checking and recovering and more particularly relates to data error checking and recovery in a data storage device in a Small Computer System Interface ("SCSI") End-to-End Checking System.

Background Art

[0002] For current data systems, data is typically transmitted in a SCSI environment without any redundancy checking information such as parity. Currently, individual devices have internal error checking. For example, a SCSI bus may have a parity bit added to each byte. The parity is tracked from where data is sent to the target along the bus, but once the data reaches the target device, the parity information is not retained. In recognition of this problem, a working-copy SCSI standard is emerging called the SCSI T10 End-to-End Checking Standard which addresses the problem of errors introduced somewhere in the data stream between a source and a target that may not be detected by individual devices between the source, target, and storage device.

[0003] One element of the standard is to include information along with the data in the form of a 2-byte guard, a 2-byte application tag, and a 4-byte reference tag. This allows a redundancy check to be created and stored in the guard. The application tag is used by an application and the reference tag is a logical block address. Since the redundancy check can now be done at the source and transmitted to the target and stored on disk, it can be used to endure data consistency from end to end.

[0004] A problem is created, however, in that the standard calls for the redundancy check to be a cyclic redundancy check ("CRC") using a particular polynomial. This creates a problem for a data storage device, such as a RAID 5 configured device, in that the CRC is not transitive. (RAID is an acronym for Redundant Array of Independent (or Inexpensive) Disks.) The problem is manifest when the RAID 5 data storage device performs a redundancy check on the parity drive, which is done by performing an exclusive OR ("XOR") operation on a row of data blocks. In this case, even though the data may be valid, the value returned will not match the XOR of the guard and a parity error will be flagged. The same problem occurs if the redundancy check is created using Reed Solomon Codes. The problem exists because the redundancy check chosen for the standard is not a type that is transitive such as a Longitudinal Redundancy Check ("LRC"). A transitive redundancy check type would allow a redundancy check of the parity drive to equal an XOR of the guard.

[0005] From the foregoing discussion, it should be apparent that a need exists for data error checking and recovery that overcomes the redundancy check problem associated with the CRC required by the SCSI End-to-End Checking Standard when storing data. Beneficially, such data error checking and recovery would be performed in a SCSI environment and especially an End-to-End Checking Standard Environment.

Disclosure of Invention

[0006] The present invention provides an apparatus as claimed in claim 1 and corresponding signal-bearing medium and method.

Brief Description of the Drawings

[0007] In order that the advantages of the invention will be readily understood, a more particular description of the invention briefly described above will be rendered by reference to specific embodiments that are illustrated in the appended drawings. Understanding that these drawings depict only typical embodiments of the invention and are not therefore to be considered to be limiting of its scope, the invention will be described and explained with additional specificity and detail through the use of the accompanying drawings, in which:

[0008] Figure 1 is a schematic block diagram illustrating one embodiment of a system for data error checking and recovery in accordance with the present invention;

[0009] Figure 2 is a schematic block diagram illustrating a typical RAID 5 configured data storage device in accordance with the present invention;

[0010] Figure 3 is a schematic block diagram illustrating a typical RAID 6 configured data storage device in accordance with the present invention;

[0011] Figure 4 is a schematic block diagram illustrating an apparatus for data error checking and recovery in accordance with the present invention;

[0012] Figure 5 is a schematic block diagram illustrating an alternate embodiment of an apparatus of for data error checking and recovery in accordance with the present invention;

[0013] Figure 6 is a schematic block diagram illustrating another embodiment of an apparatus of for data error checking and recovery in accordance with the present invention;

[0014] Figure 7 is a schematic block diagram illustrating yet another embodiment of an apparatus of for data error checking and recovery in accordance with the present invention;

[0015] Figure 8 is a schematic flow chart diagram illustrating one embodiment of a method for data error checking and recovery in accordance with the present invention;

[0016] Figure 9 is a schematic flow chart diagram illustrating a more detailed embodiment of a method for data error checking and recovery in accordance with the present

invention;

[0017] Figure 10 is a schematic diagram of an example of the data error checking and recovery from Figure 9 in accordance with the present invention;

[0018] Figure 11 is a schematic diagram of an alternate embodiment of an example of the data error checking and recovery from Figure 9 in accordance with the present invention;

[0019] Figure 12 is a schematic flow chart diagram illustrating a more detailed alternate embodiment of a method for data error checking and recovery in accordance with the present invention;

[0020] Figure 13 is a schematic flow chart diagram illustrating another alternate embodiment of a method for data error checking and recovery in accordance with the present invention;

[0021] Figure 14 is a schematic diagram of an example of the data error checking and recovery from Figure 12 in accordance with the present invention; and

[0022] Figure 15 is a schematic diagram of an example of the data error checking and recovery from Figure 13 in accordance with the present invention.

Best Mode for Carrying Out the Invention

[0023] Many of the functional units described in this specification have been labelled 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.

[0024] Modules may also be implemented in software 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.

[0025] 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 data storage devices, and may exist, at least partially, merely as electronic signals on a system or network.

[0026] Reference throughout this specification to “one embodiment,” “an embodiment,” or similar language means that a particular feature, structure, or characteristic described in connection with the embodiment is included in at least one embodiment of the present invention. Thus, appearances of the phrases “in one embodiment,” “in an embodiment,” and similar language throughout this specification may, but do not necessarily, all refer to the same embodiment.

[0027] Furthermore, the described features, structures, or characteristics of the invention may be combined in any suitable manner in one or more embodiments. In the following description, numerous specific details are provided, such as examples of programming, software modules, user selections, network transactions, database queries, database structures, hardware modules, hardware circuits, hardware chips, etc., to provide a thorough understanding of embodiments of the invention. One skilled in the relevant art will recognize, however, that the invention can be practiced without one or more of the specific details, or with other methods, components, materials, and so forth. In other instances, well-known structures, materials, or operations are not shown or described in detail to avoid obscuring aspects of the invention.

[0028] Figure 1 depicts a system 10 for data error checking and recovery in a SCSI End-to-End Checking Standard environment that overcomes the problems and limitations of the prior art. The system 10 includes one or more host servers 12 connected to a storage area network 14. A RAID 5 data storage server 16 is also connected to the storage area network 14. The RAID 5 data storage server 16 may alternatively be a RAID 6 data storage server, or any other data storage device where the error checking standard causes a redundancy check problem similar to the problem described above in the Background of the Invention section. The system 10 may also include one or more other data storage servers 18.

[0029] The system 10 also may include a computer network 20. One or more workstations 22, personal computers 24, printers 26, or other devices may be connected to the computer network 20. In addition, a host server 12 may have an Internet connection 28 with other computers 30 connected to the Internet connection 28. One skilled in the art will recognize other storage area network, computer network, and Internet configurations including other data storage devices, computers, workstations, mainframe computers, personal computers, printers, and other peripherals that relevant to the present system 10.

[0030] Figure 2 is a more detailed block diagram 200 of the RAID 5 data storage server 16. Typically, a Raid 5 data storage server includes a disk controller 202 and an array of disks. The array of disks illustrated is a 5 disk array labelled disks 1 to disk 5 204, 206,

208, 210, 212. Other RAID 5 configurations may have a different number of disks. A RAID 5 device as described herein by way of example distributes data on the five drives by dividing it up into a convenient amount such as a 512 data block with 512 bytes of data per block. The disk controller 202 stores the first block on disk 1 204 in data stripe (1,1) 214. The next block is stored on disk 2 206 in data stripe (1,2) 216, the third data block on disk 3 208 in data stripe (1,3) 218, and the fourth data block on disk 4 210 in data stripe (1,4) 220. On disk 5, parity information is stored in a parity stripe (1) 222 for a first row of data blocks. Typically, the parity information is created by XORing each of the 512 bytes for each of the data blocks of that row. Another embodiment may include a more sophisticated redundancy check to create the parity information.

[0031] Once the first four blocks of data are stored in the first row, the disk controller 202 stores the fifth block on a second row on disk 1 204 in data stripe (2,1) 224. The sixth block is stored on the second row on disk 2 206 in data stripe (2,2) 226. The seventh data block is stored on disk 3 208 in data stripe (2,3) 228. The eighth data block, however, is stored on disk 5 212 in data stripe (2,4) 230. The parity information for the second row is stored on disk 4 210 in parity stripe (2) 232. The third row repeats this pattern storing the parity information on disk 3. This pattern continues for N rows. Here the Nth row is depicted with the parity information for the Nth row stored in parity stripe (n) 234 on disk 1 204, and the data for the Nth row is stored on the other four disks.

[0032] The RAID 5 configuration allows data to be reconstructed from a failed drive using the parity information. A RAID 5 configured device is one fault tolerant so if a second drive fails or there is a media error with one drive failed, the data cannot be reconstructed.

[0033] Figure 3 is a block diagram 300 of a RAID 6 configured data storage server 302. Unlike a RAID 5 configured device, a RAID 6 configured data storage server 302 is two-fault tolerant because it stores two sets of parity data. The RAID 6 data storage server 302 is shown with a disk controller 304 and five disks 1-5 306, 308, 310, 312, 314. In one embodiment, when data is stored, it is first stored in the first row. Data block one is stored on disk 1 306 in data stripe (1,1) 316, data block two is stored in on disk 2 308 in data stripe (1,2) 318, and data block three is stored in on disk 3 310 in data stripe (1,3) 320. The first parity information is stored on disk 4 312 in parity stripe P(1) 322 and the second parity information is stored on disk 5 314 in parity stripe Q(1) 324. The fourth data block is then stored by the disk controller 304 on a second row on disk 1 306 in data stripe (2,1) 326, the fifth data block is stored on disk 2 308 in data stripe (2,2) 328, and the sixth data block is stored on disk 5 in data stripe (2,3) 330. The first parity information for the second row is stored on disk 3 310 in parity stripe

P(2) 332 and the second parity information is stored on disk 4 312 in parity stripe Q(2) 334. This continues to the N^{th} row where here it is shown that the first parity information for the row is stored on disk 1 306 in parity stripe P(n) 336 and the second parity information for the row is stored on disk 2 308 in parity stripe Q(n). The redundancy check problem with the SCSI End-to-End Checking Standard is similar to RAID 5.

[0034] Figure 4 is a depiction of an apparatus 400 for data error checking and recovery and may be in a SCSI End-to-End Checking Standard environment. In this embodiment, the RAID 5 data storage server 16 is again configured with a disk controller 202 and five disks 1-5 204, 206, 208, 210, 212. In addition, the apparatus 400 includes a redundancy check module 402 configured to create a redundancy check for a row of data stored on the disks. In a SCSI End-to-End Checking Standard environment, the data may be sent with a guard. The apparatus 400 also includes a redundancy check storage module 404 configured to store the redundancy check in a guard that is associated with the parity stripe of the row of data.

[0035] In one embodiment, the redundancy check module 402 includes a parity stripe redundancy check module 406 configured to create a first redundancy check of the parity stripe associated with the data. The first redundancy check of the parity stripe may be derived by creating a CRC of the parity stripe. A data redundancy check module 408 is included and configured to create a second redundancy check of the guards of the data. In one embodiment, the second redundancy check of the guards of the data is created by XORing the guards of each block of data in a row. An LRC redundancy check module 410 creates a longitudinal redundancy check of a second data block where the second data block includes each data block of the row, the associated guard, an application tag associated with the data block, and a reference tag associated with the data block, and creates a longitudinal redundancy check of a second parity data block, where the second parity data block includes the parity stripe, the guard of the parity stripe, an application tag of the parity stripe, and a reference tag of the parity stripe.

[0036] The redundancy check storage module 404 may include an application tag storage module 412 configured to store the first redundancy check of the parity stripe in the guard of the parity stripe and to store the second redundancy check of the guards of the data in an application tag of the parity stripe. The redundancy check storage module 404 may also include a parity stripe redundancy check module 414 configured to store the first redundancy check of the parity stripe in the guard of the parity stripe. The redundancy check storage module 404 may also include a data redundancy check storage module 416 configured to store the second redundancy check of the guards of the data in the guard of the parity stripe. The data redundancy check storage module

416 may also include a parity disable module 418 configured to disable parity checking of the data. The redundancy check storage module 404 may also include an LRC redundancy check storage module 420 that stores the longitudinal redundancy check of each second data block in a second guard associated with each second data block and stores the longitudinal redundancy check of the second parity data block in a second guard associated with the second parity data block.

[0037] Figure 5 is a block diagram of an apparatus 500 of an alternate embodiment of the RAID 5 data storage server 16 which again includes a disk controller 202 and five disks 204, 206, 208, 210, 212. In this embodiment, the RAID 5 data storage server 16 includes a redundancy check parity stripe module 502 configured to create a first redundancy check of a parity stripe associated with data, where the data is sent with a guard and comprises a row of data on the RAID 5 data storage server 16 along with the related parity information. The data may be sent and stored in a SCSI End-to-End Checking Standard environment. The RAID 5 data storage server 16 includes a redundancy check data module 504 configured to create a second redundancy check of the guards of the data. The RAID 5 data storage server 16 also includes a guard and application tag storage module 506 configured to store the first redundancy check of the parity stripe in a guard associated with the parity stripe and to store the second redundancy check of the guards of the data in an application tag associated with the parity stripe.

[0038] Figure 6 is a block diagram of another embodiment of the RAID 5 data storage server 16. An apparatus 600 includes a disk controller 202 and five disks 204, 206, 208, 210, 212. In this embodiment, the RAID 5 data storage server 16 includes a parity stripe check module 602 configured to create a redundancy check of a parity stripe associated with data, where the data is sent with a guard and comprises a row of data on the RAID 5 data storage server 16 along with the related parity information. The data may be sent and stored in a SCSI End-to-End Checking Standard environment. In addition, the RAID 5 data storage server 16 includes a parity stripe check storage module 604 configured to store the redundancy check of the parity stripe in a guard associated with the parity stripe.

[0039] Figure 7 is a block diagram of yet another embodiment of the RAID 5 data storage server 16. An apparatus 700 includes a disk controller 202 and five disks 204, 206, 208, 210, 212. In this embodiment, the RAID 5 data storage server 16 includes a redundancy check data module 702 configured to create a redundancy check of data, where the data is sent with a guard and comprises a row of data on the RAID 5 data storage server 16 along with the related parity information. The data may be sent and stored in a SCSI End-to-End Checking Standard environment. The RAID 5 data storage server 16 also includes a data check storage module 704 configured to store the

redundancy check of the guards of the data in a guard associated with the parity stripe. The RAID 5 data storage server 16 may also include, in one embodiment, a disable parity module 706 configured to disable parity checking of the data.

[0040] Figure 8 is a block diagram of an alternate embodiment of the RAID 5 data storage server 16. An apparatus 800 includes a disk controller 202 and five disks 204, 206, 208, 210, 212. In this embodiment, the RAID 5 data storage server 16 includes an LRC guard module 802 configured to create a second guard LRC by performing a longitudinal redundancy check for each second data block. A second data block is configured to include a data block of a row of data on a data storage device, an associated guard sent with the data block, an application tag sent with the data block, and a reference tag sent with the data block. An LRC guard storage module 804 is also included and is configured to store the second guard LRC in a second guard associated with each second data block. An LRC parity stripe module 806 is included and configured to create a parity stripe by performing a checksum on each data block in the row, to create a parity guard by performing a checksum of the guards, to create a parity application tag by performing a checksum of the application tags, and to create a parity reference tag by performing a checksum of the reference tags. An LRC parity stripe check module is included and is configured to create a second parity guard by performing a longitudinal redundancy check on the parity stripe, the parity guard, the parity application tag, and the parity reference tag, and an LRC parity guard storage module is included and configured to store the second parity guard in a second guard associated with the parity stripe.

[0041] The schematic flow chart diagrams that follow are generally set forth as logical flow chart diagrams. As such, the depicted order and labelled steps are indicative of one embodiment of the presented method. Other steps and methods may be conceived that are equivalent in function, logic, or effect to one or more steps, or portions thereof, of the illustrated method. Additionally, the format and symbols employed are provided to explain the logical steps of the method and are understood not to limit the scope of the method. Although various arrow types and line types may be employed in the flow chart diagrams, they are understood not to limit the scope of the corresponding method. Indeed, some arrows or other connectors may be used to indicate only the logical flow of the method. For instance, an arrow may indicate a waiting or monitoring period of unspecified duration between enumerated steps of the depicted method. Additionally, the order in which a particular method occurs may or may not strictly adhere to the order of the corresponding steps shown.

[0042] Figure 9 is a schematic flow chart of a method 900 of data error checking and recovery in a SCSI End-to-End Checking Standard environment. The method 900 begins 902 with the redundancy check module 402 creating 904 a redundancy check

for data, where the data is sent with a guard and is a row of data on a storage device along with a related parity stripe. The redundancy check storage module 404 then stores 906 the redundancy check in a guard associated with the parity stripe and the method 900 ends 908.

[0043] Figure 10 is a schematic flow chart of an alternate method 1000 of data error checking and recovery where the data may be sent or stored in a SCSI End-to-End Checking Standard environment. The method 1000 begins 1002 when the parity stripe redundancy check module 406 creates 1004 a first redundancy check of a parity stripe associated with the data. The data redundancy check module 408 then creates 1006 a second redundancy check of the guards of the data. The application tag storage module 412 then stores 1008 the first redundancy check of the parity stripe in the guard of the parity stripe. The application tag storage module 412 also stores 1010 the second redundancy check of the guards of the data in an application tag associated with the parity stripe and the method 1000 ends 1012. A detailed example is provided in Figure 11 along with the benefits of method 1000.

[0044] Figure 11 is a data chart 1100 illustrating an example of the method 1000 of figure 10. The chart 1100 depicts data as stored on the disks of a RAID 5 data storage server 16 which can have any number of disks. N disks are depicted in the chart 1100. The data associated with each disk is shown in columns where disk 1 data is in column 1 1102, disk 2 data is in column 2 1104, disk 3 data is in column 3 1106. This continues to disk N in column N 1108. For this example, a data block is configured to be a 512 data block 1110 with 512 bytes. In a SCSI End-to-End Checking Standard environment, each data block 1110 includes a 2-byte guard 1112, a 2-byte application tag 1114, and a 4-byte reference tag 1116. For each row of data this pattern of data is repeated. The Mth row is shown here as a 512 data block 1118, guard 1120, application tag 1122, and reference tag 1124. For simplicity, the parity information is always shown in column N, whereas an actual RAID 5 configured storage server 16 will have the parity information distributed as shown in Figure 2.

[0045] The guard 1112 is a CRC of the 512 data block 1110 based on a polynomial specified in the SCSI End-to-End Checking Standard. The application tag 1114 may contain application data and may either be set at a source or at a target depending upon whether or not an application tag protection bit is set that is part of a SCSI command under the SCSI End-to-End Checking Standard. If the application tag 1114 is set by the source and therefore protected, the application tag 1114 may not be overwritten by the target. In this example, the application tag may be overwritten at the target and used for redundancy check data.

[0046] In the chart 1100, column N 1108 is associated with parity information. A parity stripe, depicted here as parity block 1126, is created by XORing each of the 512 data

blocks 1110 together in the row and storing the result in the parity block 1126. The guard 1128 of the parity disk in column N 1108 in this example is configured to create a CRC of the parity block 1126. In the chart 1100, since the application tag 1114 can be set by the target, the application tag 1130 of column N 1108 is created by XORing the guards 1112 of each column of data. The logical block address of the column N 1108 data is stored in the reference tag 1132. The pattern is repeated for M rows so that in this example 1100, the Mth row of column N 1108 includes parity data 1134, a CRC of the parity data stored in the guard 1136, an XOR of the data guard CRCs stored in the application tag 1138, and a reference tag 1140.

[0047] Since the CRC data stored in each guard 1112, 1128 is specified in depicted embodiments by the SCSI End-to-End Checking Standard and is created using a polynomial that is not transitive, such as is the case with an LRC, the CRC of the parity block 1126 stored in the column N 1108 guard 1128 will be different than the value stored in the application tag 1130. Both the parity block CRC 1128 in the parity column N 1108 and the XOR of the data guards stored in the parity column N 1108 application tag 1130 are useful information. The CRC of the parity block stored in the guard 1128 is useful in detecting parity errors, but this information cannot be used alone to regenerate a guard CRC 1112 if a drive fails. With only the guard CRC of the parity 1128, the guard CRC 1112 of the failed drive must be recreated after the 512 data block 1110 of the failed drive is recreated. This is risky because the recreated CRC may be based on incorrect data.

[0048] If only an XOR of the guard CRCs 1112 is created and stored in the guard 1128 of column N 1108, but the CRC of the parity block 1126 is not created, parity checking of the target data storage device must be disabled since the guard 1128 of column N 1108 with the XOR of the guard CRCs 1112 will generate a parity error. The ability to store both the CRC of the parity block 1126 and the XOR of the data block CRCs 1112 allows both parity checking and recreation of a CRC from other CRCs of the row. This allows the newly recreated CRC to be used to check the newly recreated data to ensure the recreated data does not have an error.

[0049] Figure 12 is data chart 1200 illustrating a further example of the method 1000 of figure 10. In this example, the application data bit is such that the application data cannot be overwritten by a target storage device. In an alternate embodiment, the application data may be overwritten, but the method 1000 in Figure 10 is chosen. In the chart 1200, the RAID 5 data storage server 16 is configured to add an additional 8 bytes of data to each stored data block. As in the example of Figure 11, there are N columns where column 1 1202 is associated with disk 1, column 2 1204 is associated with disk 2, column 3 1206 is associated with disk 3, and so forth to column N 1208 which is associated with parity information. Again, in a RAID 5 environment, for each

row, the parity information shifts to a new column.

- [0050] As with the previous example, each column includes a 512 data block 1210, a guard 1212 with a CRC of the 512 data block 1210, an application tag 1214, and a reference tag 1216. In this example, an additional 2-byte guard 1218, a 2-byte application tag 1220, and a 4-byte reference tag 1222 are added to the 512 data block 1210 and tags 1212, 1214, 1216. The original guard 1212, application tag 1214, and reference tag 1216 are treated as data by the RAID 5 data storage server 16. The new guard 1218 is used to store a new CRC of all of data above 1210, 1212, 1214, 1216. Since the original application tag 1214 may not be changed, the RAID 5 data storage server 16 may store information in the new application tag 1220 as well as in the new reference tag 1222.
- [0051] For the parity drive in column N 1208, the parity data block 1224 is again created by XORing the 512 data blocks 1210. In the chart 1200, since all data above the new tags 1218, 1220, 1222 is treated like a 512 data block 1210, the guard 1226 of column N 1208 is also created by XORing the other original guards 1212. Similarly, the application tag 1228 and reference tag 1230 of column N 1208 are created by XORing the data in the other columns. In the chart 1200, a CRC of the parity information is created of all data above 1224, 1226, 1228 and stored in a new guard 1232. An XOR of the new guards 1218 is stored in the new application tag 1234 of column N 1208. A new reference tag 1236 of column N 1208 is also created. As with the previous example, in this example there are M rows of data and each row is similar to the row of data described above. The Mth row is shown 1238. With this example, the benefits of storing a CRC of the parity data and an XOR of the guards of the data are realized along with the cost of using 8 additional bytes per row and creating a CRC for the data.
- [0052] In an alternate embodiment of the method 900 of Figure 9, when the new guards 1218 are created, instead of using a CRC, an LRC is used. The same is true for the guard 1232 of the parity information in column N 1208. In this case, the LRC of the parity information stored in the guard 1232 for column N 1208 will match a value that is computed by a checksum of the LRCs stored in the new guards 1218. This makes it unnecessary for the new application tag 1234 to be used to store the checksum.
- [0053] Figure 13 is a flow chart diagram of another alternate embodiment of method 900 for data error checking and recovery. The method 1300 begins 1302 when the parity stripe redundancy check module 406 creates 1304 a first redundancy check of a parity stripe associated with the data. The parity stripe redundancy check storage module 414 then stores 1306 the first redundancy check of the parity stripe in the guard of the parity stripe and the method 1300 ends 1308. A more detailed example of this method 1300 is provided with relation to in Figure 15.
- [0054] Figure 14 is a schematic flow chart of yet another alternate embodiment of method

800 for data error checking and recovery. The method 1400 begins 1402 when the data redundancy check module 408 creates 1404 a second redundancy check of the guards of the data. The data redundancy check storage module 416 then stores 1406 the second redundancy check of the data in the guard of the parity stripe and the method 1400 ends 1408. In an alternate embodiment of the method 1400, the parity disable module 418 disables parity checking on the RAID 5 data storage server 16. A more detailed example of this method 1400 is provided with relation to Figure 16.

[0055] Figure 15 is a data chart 1500 provided as an example of the method 1300 of figure 13. In this example, the application tag is not available for the RAID 5 data storage server 16 to overwrite. Additionally, extra bytes are not stored as illustrated in the example 1200 of Figure 12. In this example, there are again N disk drives where the data of disk drive 1 is shown in column 1 1502, the data of disk drive 2 is shown in column 2 1504, the data of disk drive 3 is shown in column 3 1506, and so forth to column N 1508, where parity information is stored. In the SCSI End-to-End Checking Standard environment, a 512 data block 1510 is stored along with a CRC of the 512 data block 1510 in a guard 1512, an application tag 1514 and a reference tag 1516.

[0056] In column N 1508, a parity block 1518 is created by XORing the 512 data blocks 1510 of a row of data. In this example, the guard 1520 of column N 1508 contains an XOR of the guards 1512 of the 512 data blocks 1510. One advantage of the method 1300 of this arrangement is that if a disk drive is lost, in addition to being able to regenerate the missing data from the other 512 data blocks 1510 and the parity information 1518, the missing CRC stored in the guard 1512 of the missing drive can be recreated from the other CRCs in guards 1512 and the data in the guard 1520 on column N 1508. This arrangement ensures continuity of the data since the recreated CRC can be used to verify that the recreated data is accurate. As discussed in the example of Figure 10, a disadvantage of method 1200 and this example is that parity checking must be disabled since the data stored in the guard 1520 of column N 1508 will cause a parity error.

[0057] Since the parity information of column N 1508 consists of data created by XORing the other data on the same row 1518, 1520, the data stored in the application tag 1522 of column N 1508 is created by XORing the other application tags 1514 of the row. Similarly, the reference tag 1524 of column N 1508 is created by XORing the other reference tags 1516 of the row. Again, the pattern repeats for M rows of data and the Mth row is shown 1526.

[0058] Figure 16 is a data chart 1600 illustrating an example of the method 1400 of Figure 14. In the chart 1600, the application tag is not available for the RAID 5 data storage server 16 to overwrite. Again, extra bytes are not stored as illustrated in the example 1200 of Figure 12. In this arrangement, there are once again N disk drives where the

data of disk drive 1 is shown in column 1 1602, the data of disk drive 2 is shown in column 2 1604, the data of disk drive 3 is shown in column 3 1606, and in column N 1608 the parity information is stored. In the SCSI End-to-End Checking Standard environment, a 512 data block 1610 is stored along with a CRC of the 512 data block 1610 in a guard 1612, an application tag 1614 and a reference tag 1616.

[0059] In column N 1608, parity information 1618 is created by XORing the 512 data blocks 1610. In this arrangement, the guard 1620 of column N 1608 contains a CRC calculated from the parity information 1618. One advantage of the method 1300 of this example 1600 is that the parity information in the guard 1620 of column N 1608 is accurate so parity checking can be enabled. Another advantage of this method 1300 over the method 900 of the example 1100 shown in Figure 11 is that extra bytes beyond what is required by the SCSI End-to-End Checking Standard are not used. As discussed previously in the example of Figure 10, the disadvantage of this method 1300 is that if a disk drive is lost, the CRC of the regenerated data cannot be generated from the other existing CRCs stored in the guards 1612 and the information stored in the guard 1620 of column N 1608.

[0060] As in the previous example, the data stored in the application tag 1622 of column N 1608 is created by XORing the other application tags 1614 of the row. Similarly, the reference tag 1624 of column N 1608 is created by XORing the other reference tags 1616 of the row. Again, the pattern repeats for M rows of data and the Mth row is shown 1626.

Claims

- [0001] 1. An apparatus for data error checking and recovery in a data storage device, the apparatus comprising: a redundancy check module (402) configured to create a redundancy check for data, wherein the data is sent with a guard and comprises a row of data on a storage device along with a related parity stripe; and a redundancy check storage module (404) configured to store the redundancy check in a guard associated with the parity stripe.
- [0002] 2. The apparatus of claim 1, wherein the data sent with a guard is sent in a SCSI environment.
- [0003] 3. The apparatus of claim 2, wherein the SCSI environment is a SCSI End-to-End Checking Standard environment.
- [0004] 4. The apparatus of claim 1, further comprising a parity stripe redundancy check module configured to create a first redundancy check of the parity stripe associated with the data, a data redundancy check module configured to create a second redundancy check of the guards of the data, and an application tag storage module configured to store the first redundancy check of the parity stripe in the guard associated with the parity stripe and to store the second redundancy check of the guards of the data in an application tag associated with the parity stripe.
- [0005] 5. The apparatus of claim 4, wherein the first redundancy check is a cyclic redundancy check of the parity stripe.
- [0006] 6. The apparatus of claim 4, wherein the first redundancy check is created using Reed Solomon codes for the parity stripe.
- [0007] 7. The apparatus of claim 4, wherein the first redundancy check is a longitudinal redundancy check of the parity stripe.
- [0008] 8. The apparatus of claim 4, wherein the second redundancy check is a check sum of the guards associated with each block of data in the row.
- [0009] 9. The apparatus of claim 4, wherein the second redundancy check is created using an exclusive OR operation on a cyclic redundancy check stored in the guards associated with each block of data in the row.
- [0010] 10. The apparatus of claim 1, wherein the row of data comprises the data blocks stored together in a row on multiple disks of a data storage device.
- [0011] 11. The apparatus of claim 1, wherein the redundancy check module comprises a parity stripe redundancy check module configured to create a first redundancy check of the parity stripe associated with the data, and the redundancy check storage module comprises a parity stripe redundancy check storage module configured to store the first redundancy check of the parity stripe in the guard

associated with the parity stripe.

- [0012] 12. The apparatus of claim 1, wherein the redundancy check module comprises a data redundancy check module configured to create a second redundancy check of the guards of the data and the redundancy check storage module comprises a data redundancy check storage module configured to store the second redundancy check of the data in the guard associated with the parity stripe.
- [0013] 13. The apparatus of claim 12 further comprising a parity disable module configured to disable parity checking of the data.
- [0014] 14. The apparatus of claim 1, wherein the redundancy check module comprises an LRC redundancy check module configured to create a longitudinal redundancy check of a second data block wherein the second data block comprises each data block of the row, the associated guard, an application tag associated with the data block, and a reference tag associated with the data block, and wherein the redundancy check storage module comprises an LRC redundancy check storage module configured to store the longitudinal redundancy check of each second data block in a second guard associated with each second data block.
- [0015] 15. The apparatus of claim 14, wherein the LRC redundancy check module is further configured to create a longitudinal redundancy check of a second parity data block, wherein the second parity data block comprises the parity stripe, the guard of the parity stripe, an application tag of the parity stripe, and a reference tag of the parity stripe, and wherein the LRC redundancy check storage module is further configured to store the longitudinal redundancy check of the second parity data block in a second guard associated with the second parity data block.
- [0016] 16. The apparatus of claim 15, wherein the parity stripe, the guard of the parity stripe, the application tag of the parity stripe and the reference tag of the parity stripe are created with a checksum of the data blocks, guards, application tags, and reference tags respectively of the row of data.
- [0017] 17. The apparatus of claim 1, wherein the data storage device further comprises a RAID configured data storage device.
- [0018] 18. The apparatus of claim 17 wherein the RAID configured data storage device comprises a RAID 5 configured device.
- [0019] 19. The apparatus of claim 17 wherein the RAID configured data storage device comprises a RAID 6 configured device.
- [0020] 20. A signal bearing medium tangibly embodying a program of machine-readable instructions executable by a digital processing apparatus to perform operations of data error checking and recovery in a data storage device, the operations comprising: creating a redundancy check for data on a data storage

device, wherein the data is sent with a guard and is a row of data on a storage device along with a related parity stripe; and storing the redundancy check in a guard associated with the parity stripe.

- [0021] 21. A method for data error checking and recovery in a data storage device, the method comprising: means for creating a redundancy check for data, wherein the data is sent with a guard and is a row of data on a storage device along with a related parity stripe; and means for storing the redundancy check in a guard associated with the parity stripe.

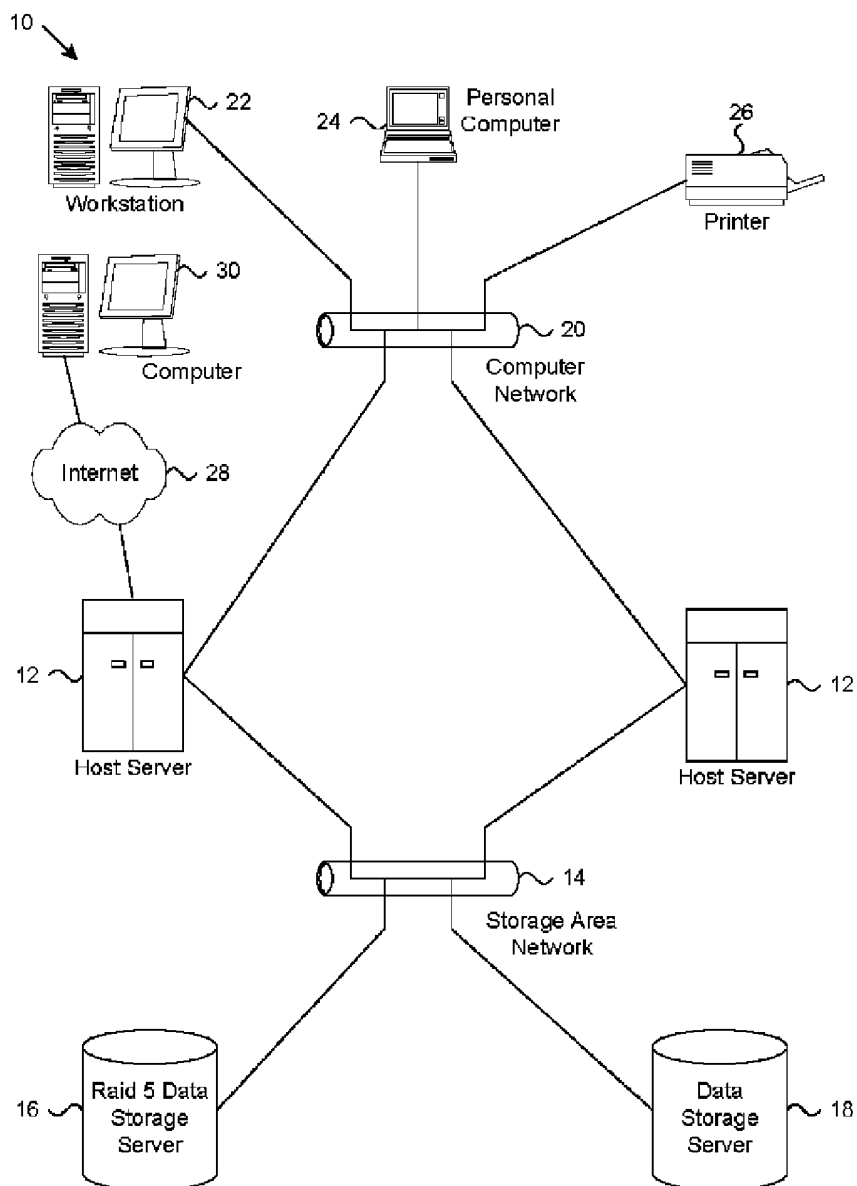


FIG. 1

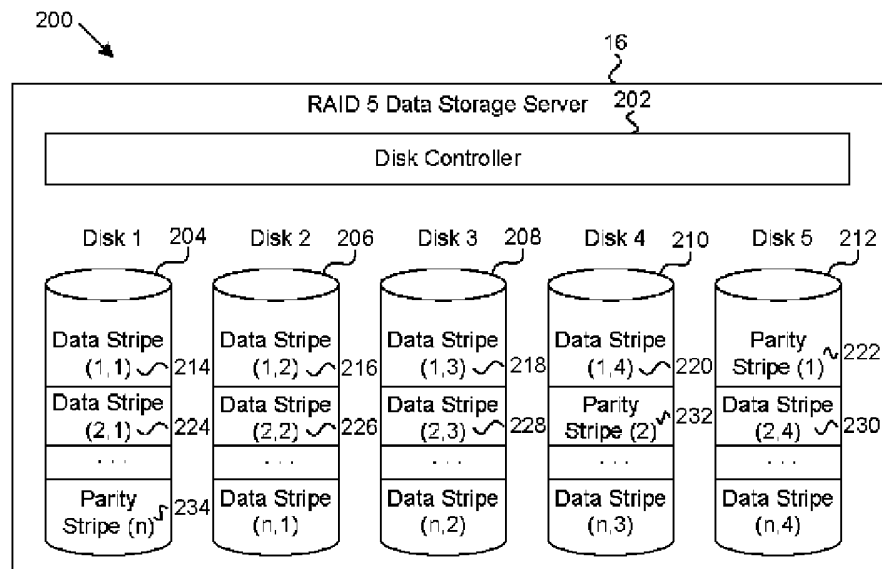


FIG. 2

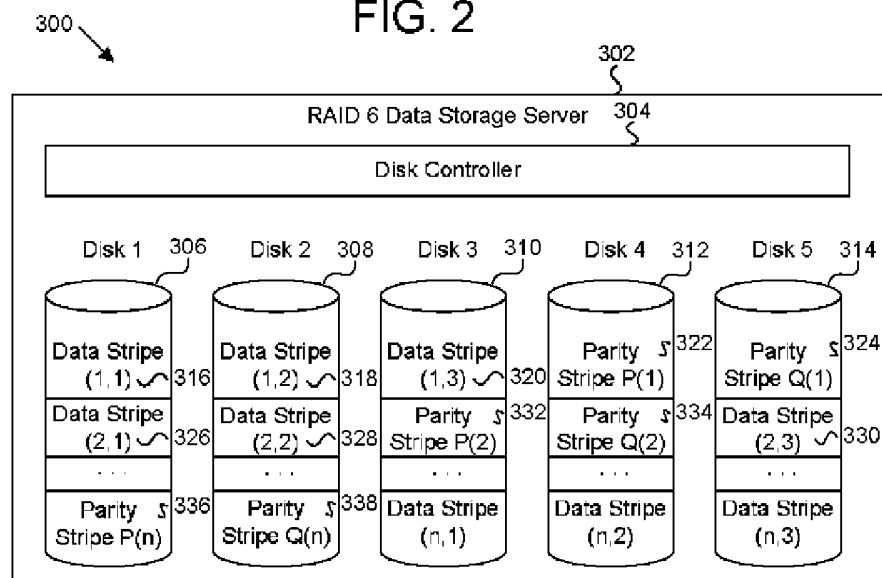


FIG. 3

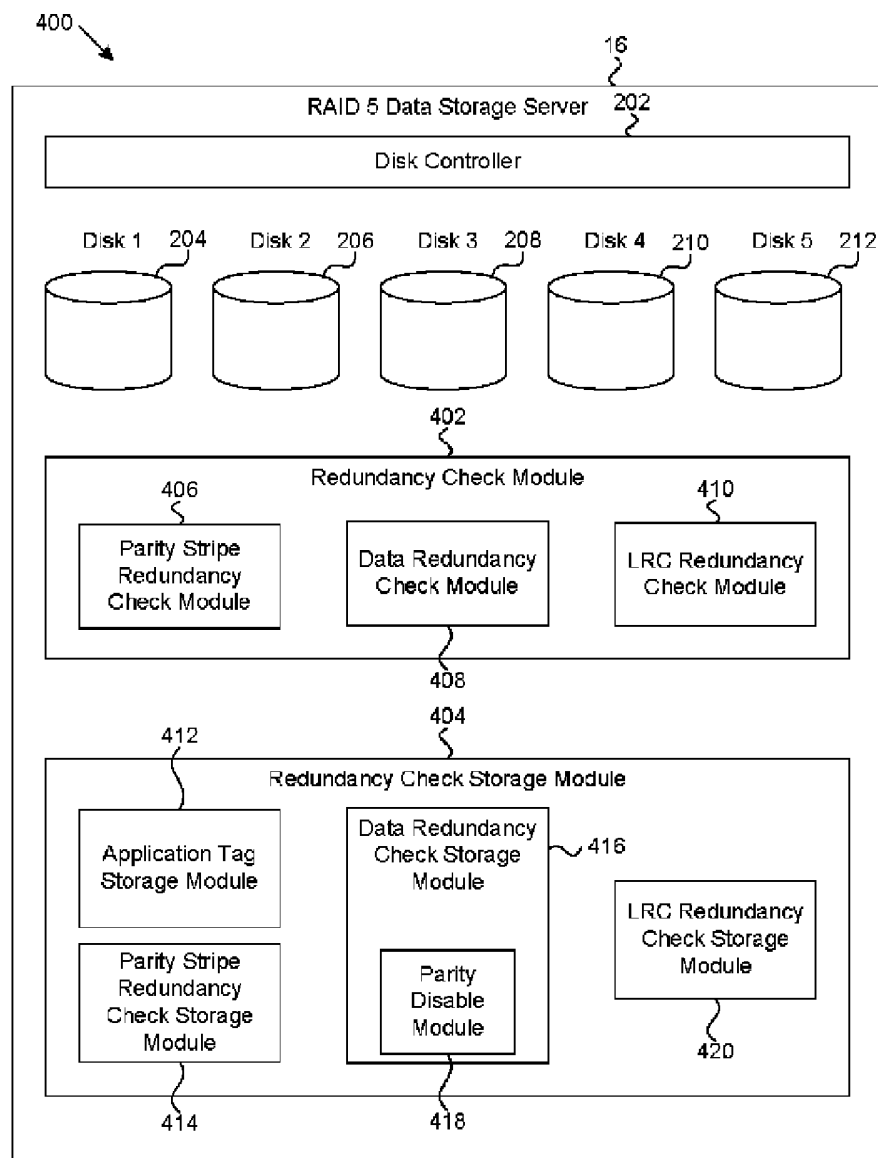


FIG. 4

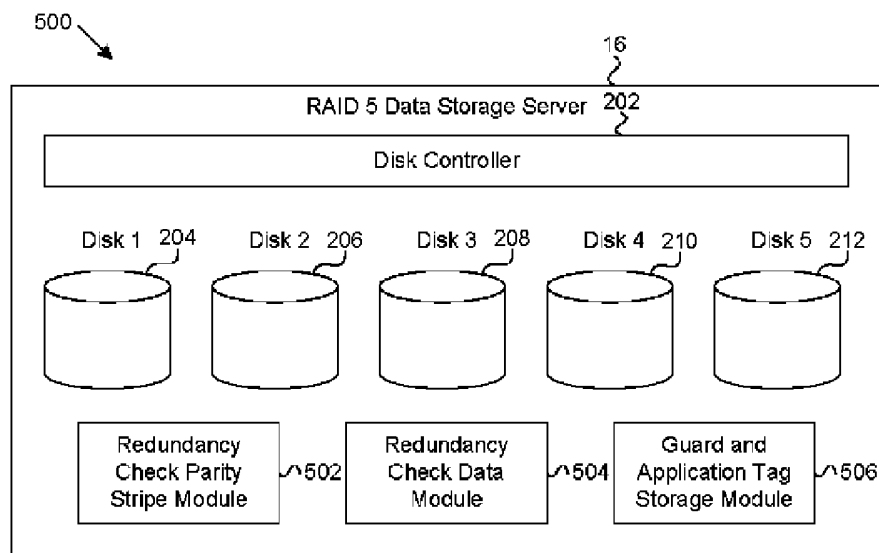


FIG. 5

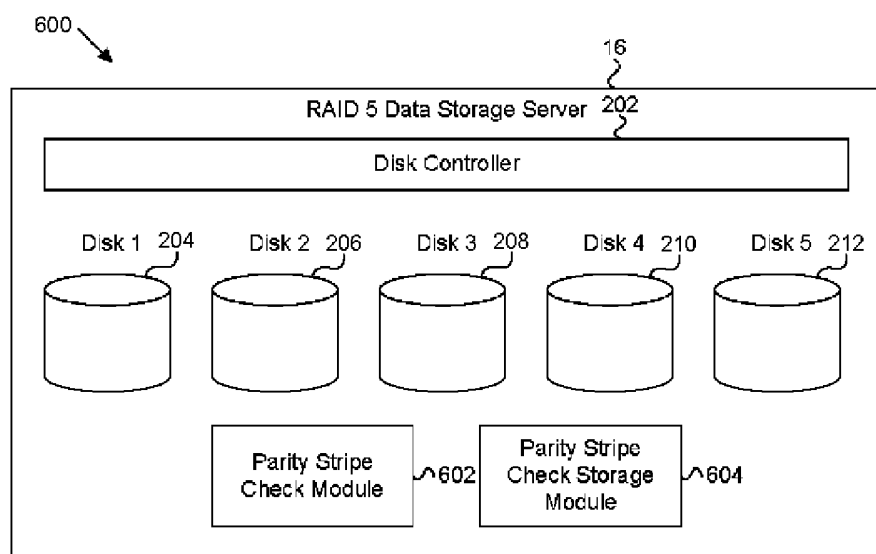


FIG. 6

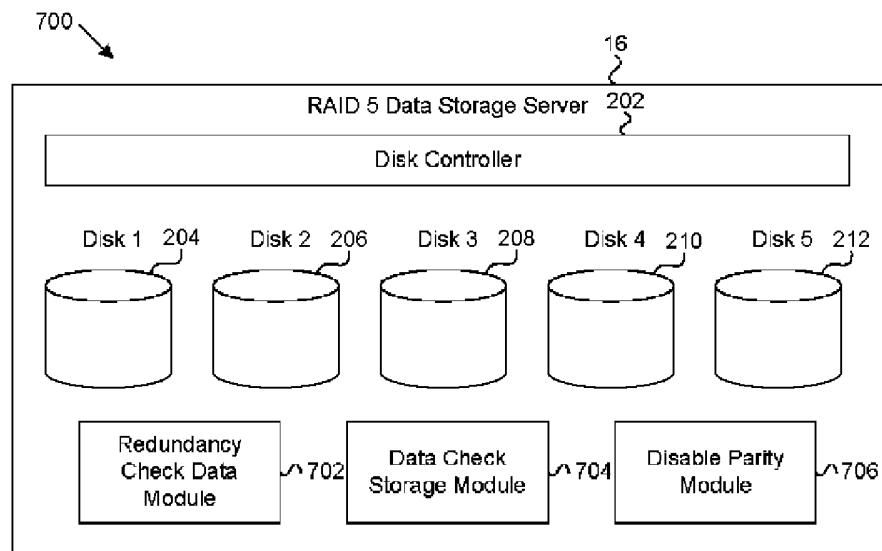


FIG. 7

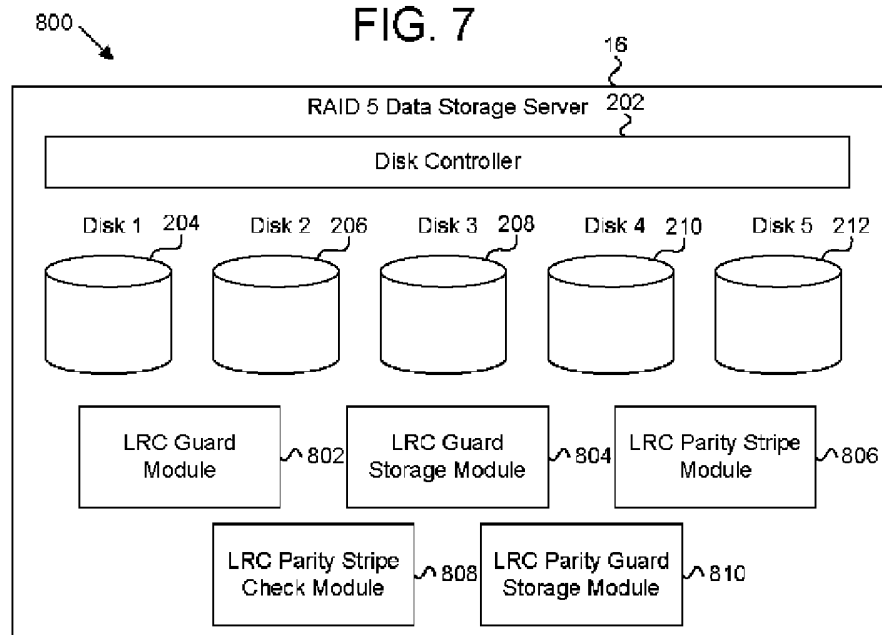


FIG. 8

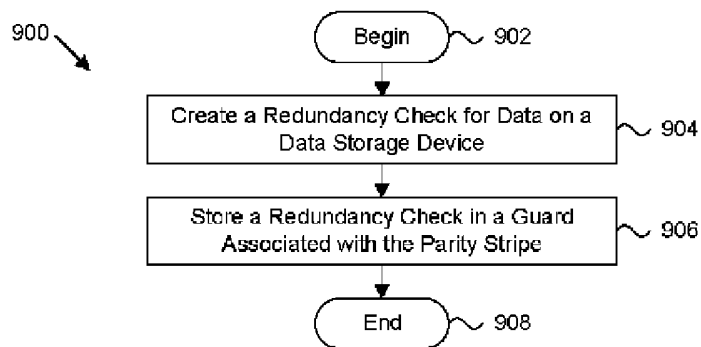


FIG. 9

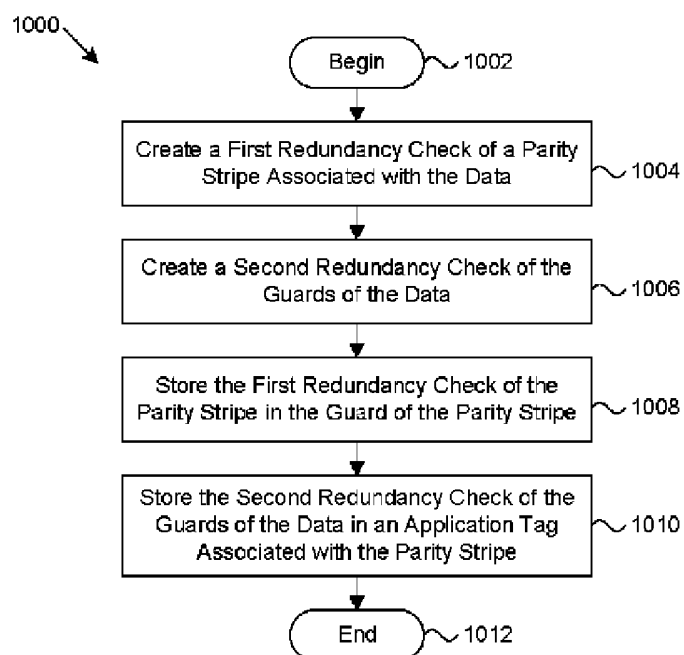


FIG. 10

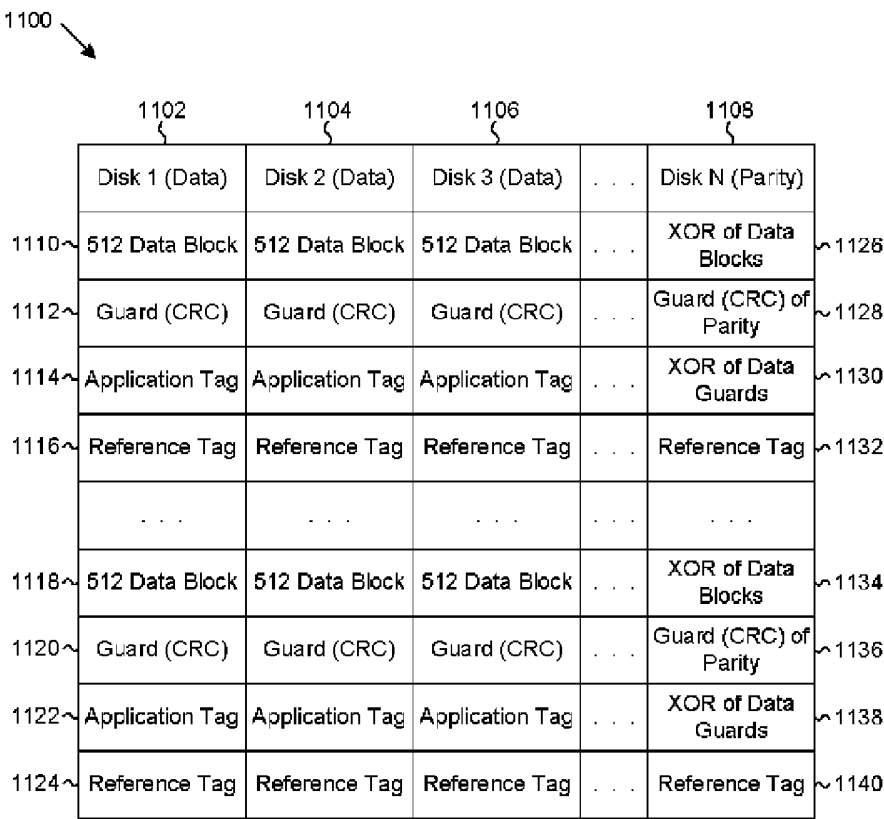


FIG. 11

1200

	1202	1204	1206	1208	
	Disk 1 (Data)	Disk 2 (Data)	Disk 3 (Data)	...	Disk N (Parity)
1210~	512 Data Block	512 Data Block	512 Data Block	...	XOR of Data Blocks
1212~	Guard (CRC)	Guard (CRC)	Guard (CRC)	...	XOR of Guards
1214~	Application Tag	Application Tag	Application Tag	...	XOR of App Tags
1216~	Reference Tag	Reference Tag	Reference Tag	...	XOR of Reference Tags
1218~	Guard 2 (CRC)	Guard 2 (CRC)	Guard 2 (CRC)	...	Guard 2 (CRC) of Parity
1220~	Application Tag 2	Application Tag 2	Application Tag 2	...	XOR of Data Guards 2s
1222~	Reference Tag 2	Reference Tag 2	Reference Tag 2	...	Reference Tag 2
	...	...	...	...	...
Row M 1238	512 Data Block	512 Data Block	512 Data Block	...	XOR of Data Blocks
	Guard (CRC)	Guard (CRC)	Guard (CRC)	...	XOR of Guards
	Application Tag	Application Tag	Application Tag	...	XOR of App Tags
	Reference Tag	Reference Tag	Reference Tag	...	XOR of Reference Tags
	Guard 2 (CRC)	Guard 2 (CRC)	Guard 2 (CRC)	...	Guard 2 (CRC) of Parity
	Application Tag 2	Application Tag 2	Application Tag 2	...	XOR of Data Guards 2s
	Reference Tag 2	Reference Tag 2	Reference Tag 2	...	Reference Tag 2

1224

1226

1228

1230

1232

1234

1236

FIG. 12

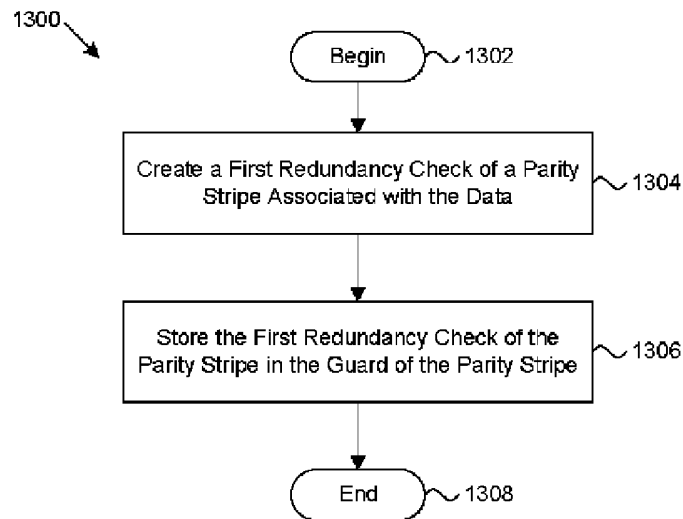


FIG. 13

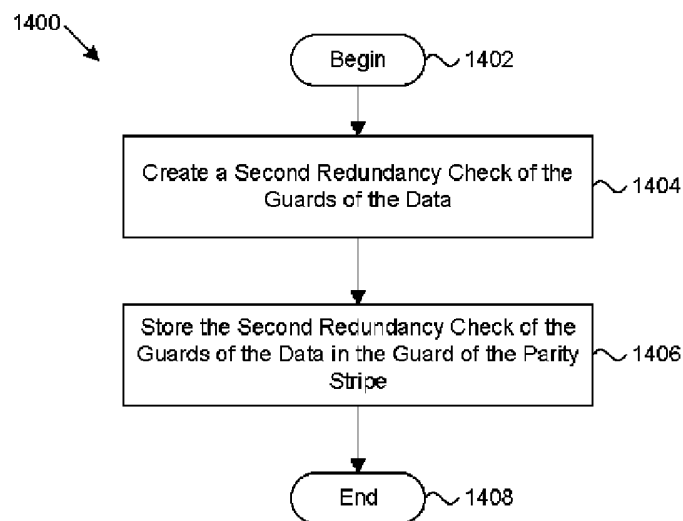


FIG. 14

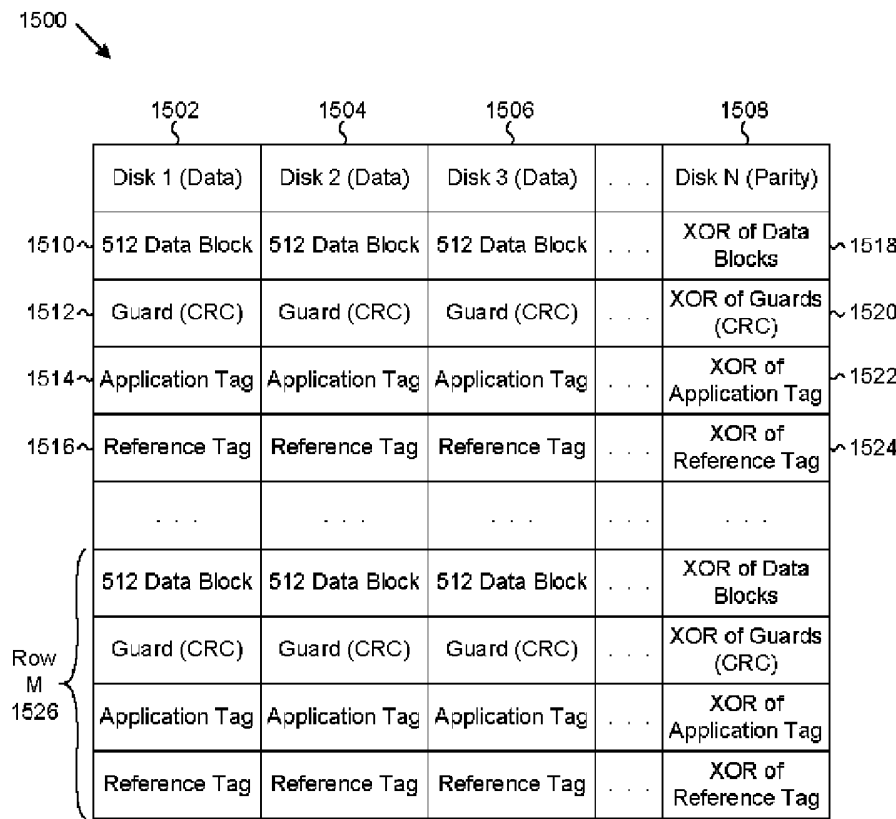


FIG. 15

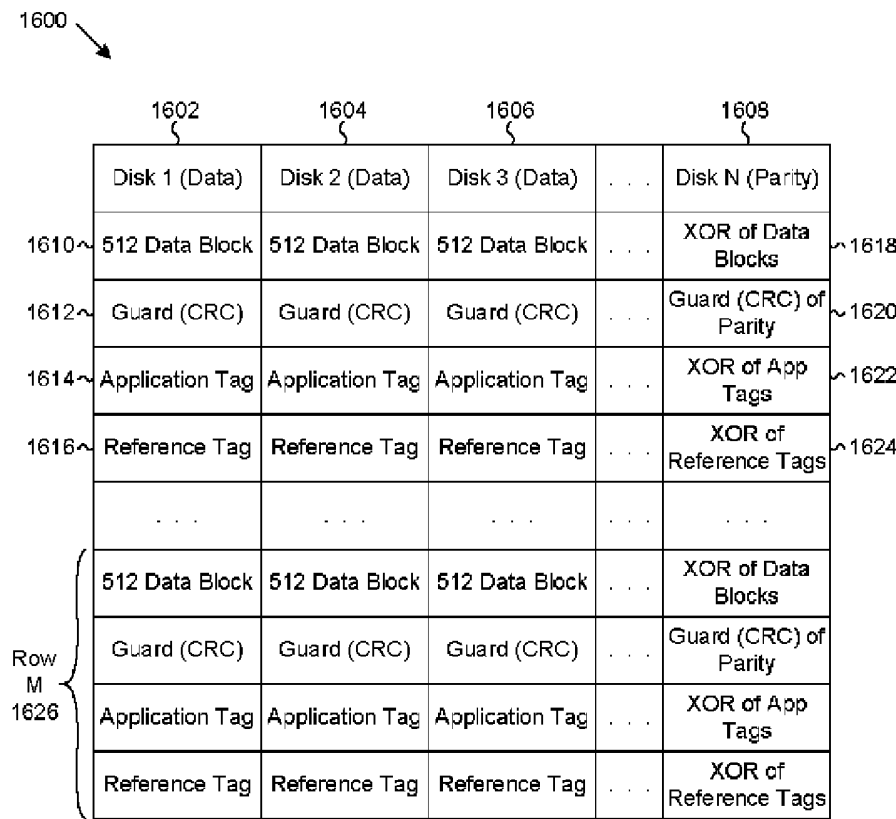


FIG. 16