(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 3 163 446 B1

(12) **EUROPEAN PATENT SPECIFICATION**

(45) Date of publication and mention of the grant of the patent:
14.08.2019 Bulletin 2019/33

(21) Application number: **14895667.5**

(22) Date of filing: **04.11.2014**

(51) Int Cl.:
G06F 11/14 *(2006.01)* ***G06F 11/20*** *(2006.01)*

(86) International application number:
PCT/CN2014/090212

(87) International publication number:
WO 2015/196686 (30.12.2015 Gazette 2015/52)

(54) **DATA STORAGE METHOD AND DATA STORAGE MANAGEMENT SERVER**

DATENSPEICHERUNGSVERFAHREN UND DATENSPEICHERVERWALTUNGSSERVER

PROCÉDÉ DE MÉMORISATION DE DONNÉES ET SERVEUR DE GESTION DE LA MÉMORISATION DE DONNÉES

(84) Designated Contracting States:
AL AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HR HU IE IS IT LI LT LU LV MC MK MT NL NO PL PT RO RS SE SI SK SM TR

(30) Priority: **26.06.2014 CN 201410292889**

(43) Date of publication of application:
03.05.2017 Bulletin 2017/18

(73) Proprietor: **Hangzhou Hikvision System Technology Co., Ltd.**
Hangzhou, Zhejiang 310052 (CN)

(72) Inventors:
- **ZENG, Fanli**
  Hangzhou
  Zhejiang 310052 (CN)
- **LIN, Peng**
  Hangzhou
  Zhejiang 310052 (CN)
- **LIN, Qiqian**
  Hangzhou
  Zhejiang 310052 (CN)
- **WANG, Weichun**
  Hangzhou
  Zhejiang 310052 (CN)

(74) Representative: **AWA Sweden AB**
P.O. Box 45086
104 30 Stockholm (SE)

(56) References cited:
CN-A- 101 187 948
CN-A- 101 394 347
CN-A- 102 096 698
CN-A- 102 622 184
US-A1- 2012 079 424
US-A1- 2013 046 735
US-A1- 2014 095 457

EP 3 163 446 B1

Description

Technical Field

[0001] The present disclosure relates to data processing technology, and particularly, to a method of storing data and a data storage managing server.

Background

[0002] When data is stored according to a conventional mechanism, original data stored is generally copied for backup

[0003] Data backup refers to a technology in which a copy or a duplicate of original data is stored separately from the original data, and is used to recover the original data in case the original data is lost or damaged due to factors such as hardware failure, virus attack, human error, or the like.

[0004] Ordinarily, more backup copies are required to increase data reliability.

[0005] CN 102622184 A discloses a data storage system and a method for storing data by using the system. The system comprises a plurality of disk storage units, a redundant strategy database and a storage control unit. The data storage method and the system select redundant strategies respectively according to different data units, save disk spaces on the basis of system safety guarantee, further improve system performance, and reduce operation cost successfully.

[0006] CN 101394347 A discloses a service data management method and a system thereof, which are mainly used in the field of data storage. The service data management method comprises the following steps: acquiring service data; determining the service levels of the services corresponding to the service data; determining the priority level of the service data; and storing the service data into corresponding storage subsystems in accordance with the service levels of the services and the priority level of the service data. The service data management method and the system thereof select different storage subsystems to store the service data in accordance with the key factors of different services and the priority level of the same service, so as to improve the access speed of the service data; and periodically detect the service levels of different services and the priority level of the service data to re-determine the value of the service data and dynamically transfer the service data according to the value of the service data, and store all the service data for backup at the same time so as to improve the safety of the service data.

[0007] CN 102096698 A discloses a video data storage format, a video data storage method and a video data search method. The storage format is that: a logical volume in a memory is divided into two types, namely an equipment head and a plurality of pages, wherein the equipment head stores related information of the pages; the related information comprises a current page field and a current offset field into which video data is written; each page is divided into two parts, namely a page head and at least one data block, wherein the page head comprises a start time field when the video data is stored into the page; and each data block is divided into two parts, namely a block head and a real video data storage area, wherein the block head comprises a data block size field and a time offset field opposite to the page head. Received video data is directly stored into a raw device based on the storage method disclosed by the storage format, disk fragmentation can be reduced, and search efficiency can be improved. In addition, the video data takes data blocks as units, is stored in the pages of each logical volume, is convenient to manage and has high data access performance.

[0008] US 2013/0046735 A discloses that a mail saving-and-backup system including a mail server and backup servers performs a backup method. The mail server determines the degree of importance of a received email by reference to a mail-importance determination table. Subsequently, the mail server determines the number of one or more backup servers for use in saving the email, according to the degree of importance of the email by reference to a backup-server management table, in which information on the status of each of the backup servers is stored. Then, the mail server selects one or more of the backup servers for use in saving the email. The mail server sends the email to each of the selected one or more of the backup servers. Then, each of the selected one or more of the backup servers stores the email in a memory.

[0009] US 2014/0095457 A discloses technology for regulating data storage based on a popularity of data files ("the technology"). Various embodiment of the technology includes maintaining a fixed durability level of data files stored in a storage system by regulating a number of copies of the data files stored in the storage system. One embodiment includes regulating the number of copies of a particular data file based on popularity of the particular data file among various users using the storage system. The number of copies stored in the storage system is increased or decreased, including from/to zero, based on the popularity of the particular data file. The popularity is determined based on at least one of: a number of computing devices of various users having the particular data file, a latency, network bandwidth and/or availability with the computing devices for reading the particular data file, or access pattern of the particular data file.

[0010] US 2012/0079424 A discloses that a redundant array of independent nodes are networked together. Each node executes an instance of an application that provides object-based storage. The nodes are grouped into a plurality of systems each having multiple nodes. A data protection level method comprises setting a numerical value of a system dynamic data protection level for each system of nodes; and in response to user input, setting a value of a data protection level for one or more

data objects in a system, the value being a fixed numerical value or a dynamic value. If the one or more data objects have a dynamic value of the data protection level in a system, the one or more data objects inherit the system dynamic data protection level of the system in which the one or more data objects reside. The value of the data protection level represents a number of copies of data of the one or more data objects to be kept where the one or more data objects reside.

Summary

[0011] In view of the foregoing, the present disclosure provides a method of storing data and a data storage managing server to enable rational use of storage resources.

[0012] To achieve the above objectives, it is proposed, in accordance with the invention, a method of storing data as defined in the appended independent claim 1 and a data storage managing server as defined in the appended independent claim 8.

[0013] A method of storing data may include:

determining the number of copies to be initially stored for a data block x after receiving a request for storing the data block x, and storing the data block x according to the number of copies to be initially stored; and

performing the following actions periodically: determining a current importance level of the data block x, determining whether the number of copies corresponding to the importance level is identical to the number of currently stored copies of the data block x, and adjusting the number of currently stored copies of the data block x to be consistent with the number of copies corresponding to the importance level when the number of copies corresponding to the importance level is not identical to the number of currently stored copies of the data block x; wherein a higher importance level is corresponding to more copies than a lower importance level;

wherein determining a current importance level of the data block x comprises: determining the current importance level of the data block x, according to the obtained data type and the obtained importance degree and a total number of visits to the data block x within a latest period.

[0014] A data storage managing server may include:

a first processing module, configured to determine the number of copies to be initially stored for a data block x after receiving a request for storing the data block x sent by a user device, instruct the user device to store the data block x into data nodes according to the number of copies to be initially stored, and instruct a second processing module to start functioning; and

the second processing module, configured to perform the following actions periodically: determining a current importance level of the data block x, determining whether the number of copies corresponding to the importance level is identical to the number of currently stored copies of the data block x, and adjusting the number of the currently stored copies of the data block x to be consistent with the number of copies corresponding to the importance level when the number of copies corresponding to the importance level is not identical to the number of currently stored copies of the data block x; wherein a higher importance level is corresponding to more copies than a lower importance level;

wherein the second processing module is configured to determine the current importance level of the data block x according to the obtained data type and the obtained importance degree of the data block x and the total number of visits to the data block x within the latest period.

[0015] A server may include:

a storage device;

at least one processor; and

one or a plurality of instructions stored in the storage device, executable by the at least one processor to:

determine the number of copies to be initially stored for a data block x after receiving a request for storing the data block x, and store the data block x according to the number of copies; and

perform the following actions periodically: determining a current importance level of the data block x, determining whether the number of copies corresponding to the importance level is identical to the number of currently stored copies of the data block x, and adjusting the number of currently stored copies of the data block x to be consistent with the number of copies corresponding to the importance level if the number of copies corresponding to the importance level is not identical to the number of currently stored copies of the data block x; wherein a higher importance level is corresponding to more copies than a lower importance level.

Brief Description of the Drawings

[0016]

FIG. 1 is a flowchart illustrating a method of storing data in accordance with an embodiment of the present disclosure.

FIG. 2 is a schematic diagram illustrating connections between a user device, a data storage managing server and a data node in accordance with the present disclosure.

FIG. 3 is a flowchart illustrating a method of storing data in accordance with another embodiment of the present disclosure.

FIG. 4 is a flowchart illustrating a method of increasing stored copies in accordance with an embodiment of the present disclosure.

FIG. 5 is a flowchart illustrating a method of decreasing stored copies in accordance with an embodiment of the present disclosure.

FIG. 6 is a schematic diagram illustrating index information of data block x in accordance with the present disclosure.

FIG. 7 is a flowchart illustrating a method of reading data in accordance with an embodiment of the present disclosure.

FIG. 8 is a schematic diagram illustrating the structure of a data storage managing server in accordance with an embodiment of the present disclosure.

FIG. 9 is a schematic diagram illustrating the structure of a data storage managing server in accordance with another embodiment of the present disclosure.

Detailed Descriptions

[0017] When multiple copies are backup for original data, if there are a large number of original data items to be backup, it is not advisable to store a fixed number of backup copies for each original data item without taking different importance levels of the original data items into consideration. For example, when the fixed number of copies is 2, two backup copies are generated for each of an original data item having a higher importance level and an original data item having a lower importance level. Two backup copies may be not enough for the original data item having a higher importance level and may be too many for the original data item having a lower importance level, thus the storage resources are not used in a rational way.

[0018] In view of the disadvantages of the conventional mechanism, some embodiments of the present disclosure provide a data storage mechanism to use storage resources rationally.

[0019] FIG. 1 is a flowchart illustrating a method of storing data in accordance with an embodiment of the present disclosure. As shown in FIG. 1, the method may include the following procedures 11 to 12.

[0020] At block 11: the number of copies to be initially stored for a data block x is determined after receiving a request for storing the data block x and the data block x may be stored according to the number of copies to be initially stored.

[0021] For simplicity, the data block x represents an arbitrary data block to be stored.

[0022] At block 12: the following actions is performed periodically: determining a current importance level of the data block x, determining whether the number of copies corresponding to the importance level is identical to the number of currently stored copies of the data block x, and adjusting the number of currently stored copies of the data block x to be consistent with the number of copies corresponding to the importance level, if the number of copies corresponding to the importance level is not identical to the number of currently stored copies of the data block x. A higher importance level corresponds to more copies to be stored than a lower importance level.

[0023] In various examples, there is no distinction between original data and backup data. If the data block x have 3 copies, the 3 copies have equal status.

[0024] The above procedures may generally be carried out by a data storage managing server. The data storage managing server may receive a storing request from a user device, and the data block x is stored into data nodes. In practice, each storage server in a system may be referred to as a data node.

[0025] The period may be a preset time period, and the specific length of the time period may be determined according to the application scenario. For example, if the frequency of system data being visited is estimated to reach stability within two hours, the period may be defined as 1 hour.

[0026] FIG. 2 is a schematic diagram illustrating connections between a user device 201, a data storage managing server 202 and data nodes 203. As shown in FIG. 2, there may generally be a plurality of data nodes 203 and a plurality of user device 201. For simplicity, only one user device is shown in FIG. 2.

[0027] Some examples of the present disclosure are hereinafter described with reference to the above devices. FIG. 3 is a flowchart illustrating a method of storing data in accordance with another embodiment of the present disclosure. The method of storing data may include the following procedures.

[0028] At block 301, after receiving a request for storing a data block x sent by a user device, the number of copies to be initially stored for the data block x is determined by the data storage managing server, according to the data type and importance degree of the data block x, and create index information for the data block x.

[0029] The data type and the importance degree of the data block x may be sent to the data storage managing server together with the request by the user device. The importance degree may be defined by a user.

[0030] The manner of determining the number of copies to be initially stored for data block x according to the

data type and the importance degree of data block x is not limited. For example, different data types and different importance degrees may be associated to different values. The weighted average of the two values may be determined to be the number of copies to be initially stored. The number of copies may be one or larger than one.

[0031] The index information may include: the data type of the data block x, the importance degree of the data block x, the total number of visits to the data block x within the latest period, the latest number of the stored copies of the data block x, the data node in which each copy of the data block x is stored and the specific location of the copy of the data block x in the data node.

[0032] At block 302, the data storage managing server obtains current load information of each data node in the system respectively, sorts the data nodes according to an ascending order of load, selects P data nodes which are ahead of the other data nodes in the sorted data nodes, and stores a copy of data block x in each of the P data nodes. The value of P is a positive integer, and is equal to the number of copies to be initially stored.

[0033] In an example, the data storage managing server may inform the user device of the number of copies to be initially stored and the P data nodes to enable the user device to store a copy of data block x in each of the P data nodes.

[0034] The load information may include: the number of service channels borne by a data node, network traffic (bit stream), CPU (Central processing unit) utilization, I/O (Input/Output) information, storage device information, or the like. The data storage managing server may obtain the load information of each data node according to a conventional mechanism.

[0035] The above procedures in blocks 301 and 302 implement the initial storage of data. Then, the data storage managing server may execute the procedures in blocks 303 to 309 as shown in FIG. 3 periodically.

[0036] At block 303, the current importance level of the data block x is determined, and the number of copies corresponding to the current importance level of the data block x may be determined. A higher importance level corresponds to a larger number of copies to be stored, and a lower importance level corresponds to a smaller number of copies to be stored.

[0037] The specific number of copies to be stored corresponding to each importance level may be pre-determined according to the needs so long as a higher importance level corresponds to a larger number of copies to be stored than a lower importance level, and is not limited herein.

[0038] The data storage managing server may determine the current importance degree of the data block x, according to the data type and the importance level of the data block x and the total number of visits to the data block x within the latest period.

[0039] The manner of determining the current importance level of the data block x is not limited in the present disclosure. For example, the importance level may be obtained from calculations using a pre-determined formula. In addition, supposing the data block x has 3 copies, the total number of visits to the data block x in the latest period may refer to the total number of visits to the 3 copies.

[0040] At block 304, it is judged whether the number of copies corresponding to the current importance level of the data block x is identical to the number of currently stored copies of the data block x: the procedure in block 305 is performed, if the number corresponding to the current importance level is identical to the number of currently stored copies of the data block x; otherwise the number of currently stored copies of the data block x is adjusted to be identical to the number of copies corresponding to the current importance level of the data block x, and the procedure in block 306 is further performed.

[0041] At block 305, if the number of currently stored copies of the data block x is equal to the number of copies corresponding to the current importance level of the data block x, the number of copies of the data block x does not need to be adjusted, and the process is terminated.

[0042] The adjusting of the number of copies may include two types of operations, i.e., increasing the number of copies and decreasing the number of copies. The increasing and decreasing are described in the following with reference to examples.

[0043] At block 306: it is judged whether the number of currently stored copies of the data block x is smaller than the number corresponding to the current importance level of the data block x: the number of copies of the data block x is to be increased and the procedure in block 307 is performed, if the number of currently stored copies of the data block x is smaller than the number corresponding to the current importance level; otherwise, the number of copies of the data block x is to be decreased and the procedure in block 308 is performed.

[0044] At block 307, when the number of currently stored copies of the data block x is smaller than the number corresponding to the current importance level of the data block x, the number of copies of the data block x is increased.

[0045] The process for increasing the copies may be as shown in FIG. 4. FIG. 4 is a flowchart illustrating the process of increasing the number of copies in accordance with an embodiment of the present disclosure.

[0046] At block 401, the difference N1 between the number of copies corresponding to the current importance level of data block x and the number of currently stored copies of the data block x is determined. N1 refers to the number of copies to be added.

[0047] At block 402, the number of data nodes that do not store a copy of the data block x is determined and denoted as M1.

[0048] M1 and N1 are both positive integers.

[0049] At block 403, the value of M1 is compared with the value of N1.

[0050] When M1 is larger than N1, the procedure in

block 404 is performed;
when M1 equals N1, the procedure in block 405 is performed;
when M1 is smaller than N1, the procedure in block 406 is performed;
At block 404: current load information of each of the M1 data nodes which do not store a copy of the data block x is obtained respectively, the M1 data nodes are sorted in an ascending order of load, N1 data nodes which are ahead of the other data nodes in the sorted data nodes are selected, and a copy of the data block x is stored in each of the N1 data nodes respectively.

[0051] For example, when M1 is 6 and N1 is 3, current load information of 6 data nodes which do not store a copy of the data block x is obtained, and the 6 data node are sorted in an ascending order of load. 3 data nodes which are ahead of the other data nodes in the sorted data nodes are selected, and a copy of data block x is stored in each of the 3 selected data nodes respectively.

[0052] At block 405, when M1 equals N1, a copy of the data block x is stored in each of the M1 data nodes that do not store a copy of the data block x respectively.

[0053] For example, when M1 is 3 and N1 is also 3, a copy of the data block x is stored in each of the 3 data nodes that do not store a copy of the data block x respectively.

[0054] At block 406, when M1 is smaller than N1, N1 copies of the data block x are stored in the M1 data nodes that do not store a copy of the data block x, with each of the data nodes storing at least one copy of the data bock x, i.e., to distribute the additional data storage task as evenly as possible among the M1 data nodes.

[0055] For example, when M1 is 2 and N1 is 3, 3 copies of data block x are stored in the 2 data nodes that do not store a copy of the data block x. Supposing the 2 data nodes that do not store a copy of the data block x are data node a and data node b, and data node a has a lighter load than the data node b, 2 copies of the data block x may be stored in the data node a, and 1 copy of the data block x may be stored in the data node b.

[0056] It should be noted that the above are mere examples. Any method of increasing the stored copies may be used according to the actual demand.

[0057] According to the above mechanism, copies of data block x may be stored to disperse as widely as possible among different data nodes to enhance data safety.

[0058] In addition, if M1 is 0, i.e., there is no data node that does not store the data block x, the additional copies may be stored in data nodes that have stored copies of the data block x, and data nodes with relatively lighter loads may be selected for storing the additional copies.

[0059] In an example, in the above block 404, supposing N1 is 3, the data storage managing server may store a copy of data block x in each of selected 3 data nodes. Supposing the selected 3 data nodes are data node a, data node b and data node c respectively, the data storage managing server may obtain current load information of each data node which has stored copies of the data block x, select the data node having the lightest load, read the data block x from the data node having the lightest load, send the data block x to the data node a, data node b and data node c respectively, and inform the data node a, the data node b and the data node c of the number of copies to be stored in each of the data nodes, so that the data node a, data node b and data node c to store the copies accordingly.

[0060] At block 308, when the number of currently stored copies of the data block x is larger than the number corresponding to the current importance level of the data block x, the number of copies of the data block x is decreased.

[0061] The process for decreasing the number is as shown in FIG. 5. FIG. 5 is a flowchart illustrating the process of decreasing the number of copies in accordance with an embodiment of the present disclosure.

[0062] At block 501, the difference N2 between the number of copies corresponding to the current importance level of data block x and the number of currently stored copies of the data block x is determined. N2 refers to the number of copies to be removed.

[0063] At block 502, the number of data nodes that store copies of the data block x is determined and denoted as M2.

[0064] M2 and N2 are both positive integers.

[0065] At block 503, the value of M2 is compared with the value of N2.

[0066] When M2 is larger than N2, the procedure in block 504 is performed;
when M2 is equal to N2, the procedure in block 505 is performed;
when M2 is smaller than N2, the procedure in block 506 is performed.

[0067] At block 504, if M2 is larger than N2, current load information of each of the M2 data nodes which store copies of the data block x may be obtained, the M2 data nodes are sorted in a descending order of load, N2 data nodes which are ahead of the other data nodes in the sorted data nodes are selected, and a copy of data block x is removed from each of the N2 data nodes.

[0068] For example, when M2 is 6 and N2 is 3, 6 data nodes which store copies of the data block x are obtained and sorted in a descending order of load. 3 data nodes which are ahead of the other data nodes in the sorted data nodes are selected, and a copy of data block x is removed from each of the 3 selected data nodes.

[0069] At block 505, when M2 equals N2, a copy of the data block x is removed from each of the M2 data nodes that store copies of the data block x respectively.

[0070] For example, when M2 is 3 and N2 is also 3, a copy of the data block x may be removed from each of 3 data nodes that have stored copies of the data block x.

[0071] At block 506, when M2 is smaller than N2, N2 copies of the data block x are removed from the M2 data nodes that have stored copies of the data block x, with each of the data nodes removing at least one copy of the data bock x, i.e., to distribute the removing task as evenly

as possible among the M2 data nodes.

[0072] For example, when M2 is 2 and N2 is 3, supposing data nodes that have stored copies of the data block x are data node a and data node b which respectively store 2 copies of the data block x, and supposing the data node a has a heavier load than the data node b, two copies of the data block x may be removed from the data node a and one copy of the data block x may be removed from the data node b.

[0073] It should be noted that the above are mere examples. Any method of decreasing the stored copies may be used according to the actual demand.

[0074] At block 309, when contents in the index information is changed, the index information is updated.

[0075] In practice, data bocks as mentioned herein may be video data blocks, and the index information of the data block x may also include: the start time and end time of the data block x, i.e., the start time and end time of the video.

[0076] FIG. 6 is a schematic diagram illustrating index information of data block x in accordance with the present disclosure. When the number of copies of the data block x or the total number of visits to the data block x is changed, the index information as shown in FIG. 6 may be updated accordingly. In addition, the data storage managing server may obtain information required by looking up the index information as shown in FIG. 6, e.g., the data nodes that have stored copies of the data block x, the data type of the data block x, the importance degree of the data block x, and the total number of visits to the data block x, or the like.

[0077] As in the above, the data blocks may generally be video data blocks, the data storage managing server may identify the data block to be searched for and the corresponding data node from which the data block is to be read, and read the data block from each of the corresponding data node, after receiving a lookup request for video data within a specified time range sent by a user device. FIG. 7 is a flowchart illustrating a method of reading data in accordance with an embodiment of the present disclosure. The method may include the following procedures.

[0078] At block 701, the data storage managing server receives a lookup request for video data within a specified time range.

[0079] At block 702, it is judged whether the start time and end time of each data block defines a time range having an intersection with the specified time range, according to stored index information of each data block: the procedure in block 703 is performed if they do not have an intersection; if the time range has an intersection with the specified time range, data blocks whose time range having intersection with the specified time range are identified from among the data blocks, and the procedure in block 704 is performed.

[0080] For example, when the specified time range is 13:00~15:00, the start time and end time of data block x1 are 12:30 and 13:30, the start time and end time of data block x2 are 13:30 and 14:30, the start time and end time of data block x3 are 14:30 and 15:30, data block x1, data block x2 and data block x3 are determined as the data blocks whose the time range defined by respective start time and end time has an intersection with the specified time range.

[0081] At block 703, it is determined that there is no video data required by the lookup request, and the process is terminated.

[0082] At block 704, the following operations is performed for each data block identified: obtaining current load information of each of data nodes that has stored copies of the data block, and selecting the data node with the lightest load as the data node from which the data block is to be read.

[0083] For example, data node a, which is the data node having the lightest load in all of data nodes that have stored copies of the data block x1, may be selected as the data node from which the data block x1 is to be read; data node b, which is the data node having the lightest load in all of data nodes that have stored copies of the data block x2, may be selected as the data node from which the data block x2 is to be read; data node c, which is the data node having the lightest load in all of data nodes that have stored copies of the data block x3, may be selected as the data node from which the data block x3 is to be read.

[0084] At block 705, the data blocks are read from the data nodes selected respectively.

[0085] In an example, the data storage managing server may send read commands concurrently to the data node a, the data node b and the data node c, which are selected for reading the data blocks so as to read the data blocks x1, x2 and x3. The reading procedure may follow a conventional procedure.

[0086] Based on the above, FIG. 8 is a schematic diagram illustrating the structure of a data storage managing server in accordance with an embodiment of the present disclosure. As shown in FIG. 8, the server may include: a first processing module 801 and a second processing module 802.

[0087] The first processing module 801 is configured to determine the number of copies to be initially stored for a data block x after receiving a request for storing the data block x sent by a user device, instruct the user device to store the data block x into data nodes according to the number of copies, and instruct a second processing module to start functioning.

[0088] The second processing module 802 is configured to perform the following actions periodically: determining a current importance level of the data block x, determining whether the number of copies corresponding to the importance level is identical to the number of currently stored copies of the data block x, and adjusting the number of the currently stored copies of the data block x to be consistent with the number of copies corresponding to the importance level when the number of copies corresponding to the importance level is not identical to

the number of currently stored copies. A higher importance level is corresponding to more copies than a lower importance level.

[0089] In an example, the first processing module may determine the number of copies to be initially stored for the data block x according to the data type and the importance degree of the data block x. The importance degree may be defined by a user.

[0090] The second processing module may determine the current importance degree of the data block x according to the data type and the importance level of the data block x and the total number of visits to the data block x within the latest period.

[0091] In an example, the first processing module may also obtain current load information of the data nodes, sort the data nodes in an ascending order of load, select P data nodes which are ahead of the other data nodes in the sorted data nodes, and instruct the user device to store a copy of the data block x in each of the P data nodes. The value of P is the same with the number of copies.

[0092] In an example, the second processing module may also determine the difference N1 between the number of copies corresponding to the importance level and the number of currently stored copies of the data block x when the number of currently stored copies is smaller than the number corresponding to the importance level, and determine the number M1 of data nodes that do not store a copy of the data block x.

[0093] When M1 is larger than N1, the second processing module may obtain current load information of each of the M1 data nodes which do not store a copy of the data block x, sort the M1 data nodes in an ascending order of load, select N1 data nodes which are ahead of the other data nodes in the sorted data nodes, and store a copy of the data block x in each of the N1 data nodes.

[0094] When M1 is equal to N1, the second processing module may store a copy of the data block x in each of the M1 data nodes that do not store a copy of the data block x.

[0095] When M1 is smaller than N1, the second processing module may store N1 copies of the data block x in the M1 data nodes that have not stored a copy of the data block x with each of the data nodes storing at least one copy.

[0096] The second processing module may also obtain current load information of data nodes that have stored copies of the data block x respectively, select the data node that has the lightest load, and read the data block x from the selected data node for storing the N1 copies.

[0097] In an example, the second processing module may also determine the difference N2 between the number of currently stored copies of the data block x and the number of copies corresponding to the importance level when the number of currently stored copies is larger than the number corresponding to the importance level, and determine the number M2 of data nodes that have stored copies of the data block x.

[0098] When M2 is larger than N2, the second processing module may obtain current load information of each of the M2 data nodes which store copies of the data block x, sort the M2 data nodes in a descending order of load, select N2 data nodes which are ahead of the other data nodes in the sorted data nodes, and remove a copy of data block x from each of the N2 data nodes.

[0099] When M2 equals N2, the second processing module may remove a copy of the data block x from each of the M2 data nodes that store copies of the data block x.

[0100] When M2 is smaller than N2, the second processing module may remove N2 copies of the data block x from the M2 data nodes that store copies of the data block x with at least one copy removed from each of the data nodes.

[0101] In an example, the first processing module may also establish index information for the data block x, and update the index information when contents in the index information is changed.

[0102] The index information may include: the data type of the data block x, the importance degree of the data block x, the total number of visits to the data block x within the latest period, a latest number of stored copies of the data block x, the data node in which each copy of the data block x is stored and the specific location of the copy of the data block x in the data node.

[0103] In an example, when the data block x is a video data block, the index information may also include: start time and end time of the data block x.

[0104] Correspondingly, the second processing module may also identify data nodes meeting the following conditions from all of data blocks according to index information of the data nodes after receiving a lookup request for video data within a specified time range: a time range defined by the start time and end time of the data block has an intersection with the specified time range. The following procedures may be performed for each of the data blocks identified: obtaining current load information of each data node that stores a copy of the data block, selecting the data node having the lightest load as the data node from which the data block is to be read; sending a read command to the data node to read the data block.

[0105] In an example, the second processing module may send read commands concurrently to the data nodes storing the data blocks.

[0106] FIG. 9 is a schematic diagram illustrating the structure of a data storage managing server in accordance with another embodiment of the present disclosure. As shown in FIG. 9, a target device may include at least one processor 901, storage device 902 and at least one instruction 903 which is stored in the storage device 902 and is executable by the at least one processor 901. The storage device 902 may include a first processing module and a second processing module. The virtual modules may include instructions 903 for fulfilling respective functions. When the processor 901 communicates with the storage device 902, reads and executes the instructions,

the device may implement corresponding functions.

[0107] Generally, the mechanism of the present disclosure, any data block x may be stored according to a determined number of initially stored copies which are backup for each other. The number of copies may be adjusted according to changes in the importance level of the data block x, i.e., when the importance level becomes higher, the number of copies may be increased; when the importance level becomes lower, the number of copies may be decreased. As such, at least a problem in the conventional mechanism is solved, and storage resources can be used rationally.

[0108] In addition, the backup operation is generally implemented manually in the conventional mechanism. Each procedure of the present disclosure may be implemented automatically, thus user's operation is simplified.

[0109] In addition, the data storage and copy number adjustment of the present disclosure all take load balancing into consideration. Read commands are sent to data nodes concurrently for reading data, thus the system performances and stability can be improved.

[0110] Further, the backup data is used only when the original data is lost or damaged in conventional mechanism, and is left idle in other time. But according to the present disclosure, original data is not distinguished from backup data, and the data block x is read from any data node (having the lightest load) that stores the data block x, i.e., storage resources occupied by each data block x are not left completely idle, thus utility ratio of the storage resource can be improved.

[0111] The foregoing descriptions are only preferred embodiments of this invention and are not for use in limiting the protection scope thereof.

Claims

1. A method of storing data, the method comprising determining a number of copies to be initially stored for a data block x, after receiving a request for storing the data block x, storing the data block x according to the number of copies to be initially stored (11); and performing the following actions periodically: determining a current importance level of the data block x, determining whether a number of copies corresponding to the importance level is identical to a number of currently stored copies of the data block x, and adjusting the number of currently stored copies of the data block x to be consistent with the number of copies corresponding to the importance level if the number of copies corresponding to the importance level is not identical to the number of currently stored copies of the data block x; wherein a higher importance level is corresponding to more copies than a lower importance level (12);
   wherein determining a current importance level of the data block x comprises: determining the current importance level of the data block x, according to the obtained data type and the obtained importance degree and a total number of visits to the data block x within a latest period.

2. The method of claim 1, wherein
   determining the number of copies to be initially stored for the data block x comprises: determining the number of copies to be initially stored for the data block x, according to an obtained data type and an obtained importance degree of the data block x, wherein the importance degree is defined by a user.

3. The method of claim 2, wherein
   storing the data block x according to the number of copies to be initially stored comprises:

   obtaining current load information of each of data nodes respectively, and sorting the data nodes in an ascending order of load;
   selecting P data nodes which are ahead of other data nodes in the sorted data nodes, and storing a copy of the data block x in each of the P selected data nodes; wherein the value of P is a positive integer and equal to the number of copies to be initially stored.

4. The method of claim 3, wherein
   adjusting the number of currently stored copies of the data block x to be consistent with the number of copies corresponding to the importance level comprises:

   determining a difference N1 between the number of copies corresponding to the importance level and the number of currently stored copies of the data block x, when the number of currently stored copies is smaller than the number corresponding to the importance level, and determining the number M1 of data nodes that do not store a copy of the data block x; wherein M1 and N1 are positive integers;
   when M1 is larger than N1, obtaining current load information of each of the M1 data nodes which do not store a copy of the data block x, sorting the M1 data nodes in an ascending order of load, selecting N1 data nodes which are ahead of other data nodes in the sorted data nodes, and storing a copy of the data block x in each of the N1 data nodes;
   when M1 is equal to N1, storing a copy of the data block x in each of the M1 data nodes that do not store a copy of the data block x; and
   when M1 is smaller than N1, storing N1 copies of the data block x in the M1 data nodes that have not stored a copy of the data block x with each of the M1 data nodes storing at least one copy.

5. The method of claim 3, wherein
adjusting the number of currently stored copies of the data block x to be consistent with the number of copies corresponding to the importance level comprises:

determining a difference N2 between the number of copies corresponding to the importance level and the number of currently stored copies of the data block x, when the number of currently stored copies is larger than the number corresponding to the importance level, and determining a number M2 of data nodes that store a copy of the data block x; wherein M2 and N2 are positive integers;
when M2 is larger than N2, obtaining current load information of each of the M2 data nodes which store copies of the data block x, sorting the M2 data nodes in a descending order of load, selecting N2 data nodes which are ahead of other data nodes in the sorted data nodes, and removing a copy of data block x from each of the N2 data nodes;
when M2 equals N2, removing a copy of the data block x from each of the M2 data nodes that store a copy of the data block x; and
when M2 is smaller than N2, removing N2 copies of the data block x from the M2 data nodes that store at least one copy of the data block x with at least one copy removed from each of the M2 data nodes.

6. The method of claim 4 or 5, further comprising:

establishing index information for the data block x, and updating the index information when contents in the index information are changed;
wherein the index information comprises: the data type of the data block x, the importance degree of the data block x, the total number of visits to the data block x within the latest period, a latest number of stored copies of the data block x, a data node in which each copy of the data block x is stored and a specific location of the copy of the data block x in the data node.

7. The method of claim 6, wherein
when the data block x is a video data block, the index information further comprises: start time and end time of the data block x;
the method further comprises:

after receiving a lookup request for video data within a specified time range, determining data blocks which meet the following condition from all of data blocks, according to index information of the data blocks: a time range defined by start time and end time of the data block has an intersection with the specified time range;
performing the following actions for each of the data blocks determined: obtaining current load information of each of data nodes that stores a copy of the data block, and selecting a data node having the lightest load as a data node from which the data block is to be read;
reading the data blocks from the data node selected.

8. A data storage managing server, comprising, a first processing module (801), configured to determine a number of copies to be initially stored for a data block x, after receiving a request for storing the data block x sent by a user device, instruct the user device to store the data block x into data nodes according to the number of copies to be initially stored, and instruct a second processing module (802) to start functioning; and
the second processing module (802), configured to perform the following actions periodically: determining a current importance level of the data block x, determining whether a number of copies corresponding to the importance level is identical to a number of currently stored copies of the data block x, and adjusting the number of the currently stored copies of the data block x to be consistent with the number of copies corresponding to the importance level when the number of copies corresponding to the importance level is not identical to the number of currently stored copies of the data block x; wherein a higher importance level is corresponding to more copies than a lower importance level;
wherein the second processing module (802) is configured to determine the current importance level of the data block x according to the obtained data type and the obtained importance degree of the data block x and the total number of visits to the data block x within the latest period.

9. The data storage managing server of claim 8, wherein
the first processing module is configured to determine the number of copies to be initially stored for the data block x according to an obtained data type and an obtained importance degree of the data block x, wherein the importance degree is defined by a user.

10. The data storage managing server of claim 9, wherein
the first processing module is further configured to obtain current load information of respective data nodes, sort the data nodes in an ascending order of load, select P data nodes which are ahead of other data nodes in the sorted data nodes, and instruct the user device to store a copy of the data block x in each of the P data nodes; wherein the value of P is

the same with the number of copies to be initially stored, wherein P is a positive integer.

11. The data storage managing server of claim 10, wherein
the second processing module is further configured to determine a difference N1 between the number of copies corresponding to the importance level and the number of currently stored copies of the data block x when the number of currently stored copies is smaller than the number corresponding to the importance level, and determining the number M1 of data nodes that do not store a copy of the data block x; wherein M1 and N1 are positive integers;
when M1 is larger than N1, obtain current load information of each of the M1 data nodes which do not store a copy of the data block x, sort the M1 data nodes in an ascending order of load, selecting N1 data nodes which are ahead of other data nodes in the sorted data nodes, and store a copy of the data block x in each of the N1 data nodes;
when M1 is equal to N1, store a copy of the data block x in each of the M1 data nodes that do not store a copy of the data block x; and
when M1 is smaller than N1, store N1 copies of the data block x in the M1 data nodes that have not stored a copy of the data block x with each of the M1 data nodes storing at least one copy.

12. The data storage managing server of claim 11, wherein
the second processing module is further configured to obtain current load information of each of data nodes that stores a copy of the data block x, select a data node that has the lightest load, and read the data block x from the selected data node for storing the N1 copies.

13. The data storage managing server of claim 10, wherein
the second processing module is further configured to determine a difference N2 between the number of currently stored copies of the data block x and the number of copies corresponding to the importance level when the number of currently stored copies is larger than the number corresponding to the importance level, and determining a number M2 of data nodes that store a copy of the data block x; wherein M2 and N2 are positive integers;
when M2 is larger than N2, obtain current load information of each of the M2 data nodes which store copies of the data block x, sort the M2 data nodes in a descending order of load, select N2 data nodes which are ahead of other data nodes in the sorted data nodes, and remove a copy of data block x from each of the N2 data nodes;
when M2 equals N2, remove a copy of the data block x from each of the M2 data nodes that store a copy of the data block x;
when M2 is smaller than N2, remove N2 copies of the data block x from the M2 data nodes that store at least one copy of the data block x with at least one copy removed from each of the M2 data nodes.

14. The data storage managing server of claim 11, 12 or 13, wherein
the first processing module is further configured to establish index information for the data block x, and update the index information when contents in the index information are changed;
wherein the index information comprises: the data type of the data block x, the importance degree of the data block x, the total number of visits to the data block x within the latest period, a latest number of stored copies of the data block x, a data node in which each copy of the data block x is stored and a specific location of the copy of the data block x in the data node.

15. The data storage managing server of claim 14, wherein
when the data block x is a video data block, the index information further comprises: start time and end time of the data block x;
the second processing module is further configured to identify data blocks meeting the following condition from all of data blocks, according to index information of the data nodes after receiving a lookup request for video data within a specified time range: a time range defined by start time and end time of the data block has an intersection with the specified time range; perform the following procedures for each of the data blocks identified: obtaining current load information of each of data nodes that store a copy of the data block, selecting data nodes having the lightest load as the data node from which the data block is to be read; sending a read command to each of the data nodes selected to read the data block.

16. The data storage managing server of claim 15, wherein the second processing module is configured to send a read command to each of the data nodes selected concurrently.

Patentansprüche

1. Verfahren zur Datenspeicherung, wobei das Verfahren Folgendes umfasst:

Bestimmen einer Anzahl an Kopien, die für einen Datenblock x anfänglich zu speichern sind, nach dem Empfangen einer Anfrage zum Speichern des Datenblocks x, Speichern des Datenblocks x gemäß der Anzahl an Kopien, die an-

fänglich zu speichern sind, (11) und
periodisches Ausführen der folgenden Aktionen: Bestimmen eines aktuellen Wichtigkeitsniveaus des Datenblocks x, Bestimmen, ob eine Anzahl an Kopien, die dem Wichtigkeitsniveau entspricht, identisch mit einer Anzahl an aktuell gespeicherten Kopien des Datenblocks x ist, und Einstellen der Anzahl der aktuell gespeicherten Kopien des Datenblocks x derart, dass sie mit der Anzahl an Kopien, die dem Wichtigkeitsniveau entspricht, übereinstimmt, wenn die Anzahl an Kopien, die dem Wichtigkeitsniveau entspricht, nicht identisch mit der Anzahl an aktuell gespeicherten Kopien des Datenblocks x ist, wobei ein höheres Wichtigkeitsniveau mehr Kopien entspricht als ein niedrigeres Wichtigkeitsniveau (12),
wobei das Bestimmen eines aktuellen Wichtigkeitsniveaus des Datenblocks x Folgendes umfasst: Bestimmen des aktuellen Wichtigkeitsniveaus des Datenblocks x gemäß der erhaltenen Datenart und des erhaltenen Wichtigkeitsgrades und einer Gesamtanzahl an Besuchen des Datenblocks x innerhalb einer jüngsten Periode.

2. Verfahren nach Anspruch 1, wobei
das Bestimmen der Anzahl an Kopien, die für den Datenblock x anfänglich zu speichern sind, Folgendes umfasst:
Bestimmen der Anzahl an Kopien, die für den Datenblock x anfänglich zu speichern sind, gemäß einer erhaltenen Datenart und eines erhaltenen Wichtigkeitsgrades des Datenblocks x, wobei der Wichtigkeitsgrad durch einen Anwender definiert wird.

3. Verfahren nach Anspruch 2, wobei
das Speichern des Datenblocks x gemäß der Anzahl an Kopien, die anfänglich zu speichern sind, Folgendes umfasst:

Erhalten aktueller Lastinformationen von jeweils jedem einzelnen von Datenknoten und Sortieren der Datenknoten in einer aufsteigenden Reihenfolge der Last,
Auswählen von P Datenknoten, die in den sortierten Datenknoten vor anderen Datenknoten liegen, und Speichern einer Kopie des Datenblocks x in jedem der P ausgewählten Datenknoten, wobei der Wert von P eine positive Ganzzahl und gleich der Anzahl an Kopien, die anfänglich zu speichern sind, ist.

4. Verfahren nach Anspruch 3, wobei
das Einstellen der Anzahl an aktuell gespeicherten Kopien des Datenblocks x derart, dass sie mit der Anzahl an Kopien übereinstimmt, die dem Wichtigkeitsniveau entspricht, Folgendes umfasst:

Bestimmen einer Differenz N1 zwischen der Anzahl an Kopien, die dem Wichtigkeitsniveau entspricht, und der Anzahl an aktuell gespeicherten Kopien des Datenblocks x, wenn die Anzahl an aktuell gespeicherten Kopien kleiner als die Anzahl an Kopien ist, die dem Wichtigkeitsniveau entspricht, und Bestimmen der Anzahl M1 an Datenknoten, die keine Kopie des Datenblocks x speichern, wobei M1 und N1 positive Ganzzahlen sind,
wenn M1 größer als N1 ist, Erhalten aktueller Lastinformationen von jedem der M1 Datenknoten, die keine Kopie des Datenblocks x speichern, Sortieren der M1 Datenknoten in einer aufsteigenden Reihenfolge der Last, Auswählen von N1 Datenknoten, die in den sortierten Datenknoten vor anderen Datenknoten liegen, und Speichern einer Kopie des Datenblocks x in jeden der N1 Daten knoten,
wenn M1 gleich N1 ist, Speichern einer Kopie des Datenblocks x in jedem der M1 Datenknoten, die keine Kopie des Datenblocks x speichern, und
wenn M1 kleiner als N1 ist, Speichern von N1 Kopien des Datenblocks x in den M1 Datenknoten, die keine Kopie des Datenblocks x speichern, wobei jeder der M1 Datenknoten mindestens eine Kopie speichert.

5. Verfahren nach Anspruch 3, wobei
das Einstellen der Anzahl an aktuell gespeicherten Kopien des Datenblocks x derart, dass sie mit der Anzahl an Kopien übereinstimmt, die dem Wichtigkeitsniveau entspricht, Folgendes umfasst:

Bestimmen einer Differenz N2 zwischen der Anzahl an Kopien, die dem Wichtigkeitsniveau entspricht, und der Anzahl an aktuell gespeicherten Kopien des Datenblocks x, wenn die Anzahl an aktuell gespeicherten Kopien größer als die Anzahl an Kopien ist, die dem Wichtigkeitsniveau entspricht, und Bestimmen einer Anzahl M2 an Datenknoten, die eine Kopie des Datenblocks x speichern, wobei M2 und N2 positive Ganzzahlen sind,
wenn M2 größer als N2 ist, Erzielen aktueller Lastinformationen von jedem der M2 Datenknoten, die Kopien des Datenblocks x speichern, Sortieren der M2 Datenknoten in einer absteigenden Reihenfolge der Last, Auswählen von N2 Datenknoten, die in den sortierten Datenknoten vor anderen Datenknoten liegen, und Entfernen einer Kopie des Datenblocks x aus jedem der N2 Datenknoten,
wenn M2 gleich N2 ist, Entfernen einer Kopie des Datenblocks x aus jedem der M2 Datenknoten, die eine Kopie des Datenblocks x speichern, und

wenn M2 kleiner als N2 ist, Entfernen von N2 Kopien des Datenblocks x aus den M2 Datenknoten, die mindestens eine Kopie des Datenblocks x speichern, wobei aus jedem der M2 Datenknoten mindestens eine Kopie entfernt wird.

6. Verfahren nach Anspruch 4 oder 5, ferner Folgendes umfassend:

Ermitteln von Indexinformationen für den Datenblock x und Aktualisieren der Indexinformationen, wenn Inhalte in den Indexinformationen verändert werden,
wobei die Indexinformationen Folgendes umfassen: die Datenart des Datenblocks x, den Wichtigkeitsgrad des Datenblocks x, die Gesamtanzahl an Besuchen des Datenblocks x innerhalb der jüngsten Periode, eine jüngste Anzahl an gespeicherten Kopien des Datenblocks x, einen Datenknoten, in dem jede Kopie des Datenblocks x gespeichert ist, und ein spezifischer Standort der Kopie des Datenblocks x in dem Datenknoten.

7. Verfahren nach Anspruch 6, wobei
die Indexinformationen, wenn der Datenblock x ein Videodatenblock ist, ferner Startzeit und Endzeit des Datenblocks x umfassen,
wobei das Verfahren ferner Folgendes umfasst:

nach dem Empfangen einer Nachschlageanfrage für Videodaten innerhalb eines spezifizierten Zeitbereichs Bestimmen von Datenblöcken aus allen Datenblöcken, gemäß Indexinformationen der Datenblöcke, welche die folgende Bedingung erfüllen: ein Zeitbereich, der durch Startzeit und Endzeit des Datenblocks definiert ist, weist eine Überschneidung mit dem spezifizierten Zeitbereich auf,
Ausführen der folgenden Aktionen für jeden der bestimmten Datenblöcke: Erhalten von aktuellen Lastinformationen von jedem der Datenknoten, der eine Kopie des Datenblocks speichert, und Auswählen eines Datenknotens, der die geringste Last aufweist, als einen Datenknoten, aus dem der Datenblock zu lesen ist,
Lesen der Datenblöcke aus dem ausgewählten Datenknoten.

8. Datenspeicher-Verwaltungsserver, Folgendes umfassend:

ein erstes Verarbeitungsmodul (801), das dafür konfiguriert ist, eine Anzahl an Kopien zu bestimmen, die anfänglich für einen Datenblock x zu speichern sind, nach dem Empfangen einer Anfrage zum Speichern des Datenblocks x, die von einem Anwendergerät gesendet wurde, Anweisen des Anwendergerätes, den Datenblock x gemäß der Anzahl an Kopien, die anfänglich zu speichern sind, in Datenknoten zu speichern, und Anweisen eines zweiten Verarbeitungsmoduls (802), den Betrieb zu starten, und
wobei das zweite Verarbeitungsmodul (802) dafür konfiguriert ist, die folgenden Aktionen periodisch auszuführen: Bestimmen eines aktuellen Wichtigkeitsniveaus des Datenblocks x, Bestimmen, ob eine Anzahl an Kopien, die dem Wichtigkeitsniveau entspricht, identisch mit einer Anzahl an aktuell gespeicherten Kopien des Datenblocks x ist, und Einstellen der Anzahl der aktuell gespeicherten Kopien des Datenblocks x derart, dass sie mit der Anzahl an Kopien, die dem Wichtigkeitsniveau entspricht, übereinstimmt, wenn die Anzahl an Kopien, die dem Wichtigkeitsniveau entspricht, nicht identisch mit der Anzahl an aktuell gespeicherten Kopien des Datenblocks x ist, wobei ein höheres Wichtigkeitsniveau mehr Kopien entspricht als ein niedrigeres Wichtigkeitsniveau,
wobei das zweite Verarbeitungsmodul (802) dafür konfiguriert ist, das aktuelle Wichtigkeitsniveau des Datenblocks x gemäß der erhaltenen Datenart und des erhaltenen Wichtigkeitsgrades des Datenblocks x und der Gesamtanzahl an Besuchen des Datenblocks x innerhalb der jüngsten Periode zu bestimmen.

9. Datenspeicher-Verwaltungsserver nach Anspruch 8, wobei
das erste Verarbeitungsmodul dafür konfiguriert ist, die Anzahl an Kopien, die anfänglich für den Datenblock x zu speichern sind, gemäß einer erhaltenen Datenart und einem erhaltenen Wichtigkeitsgrad des Datenblocks x zu bestimmen, wobei der Wichtigkeitsgrad durch einen Anwender definiert ist.

10. Datenspeicher-Verwaltungsserver nach Anspruch 9, wobei
das erste Verarbeitungsmodul ferner dafür konfiguriert ist, aktuelle Lastinformationen von entsprechenden Datenknoten zu erhalten, die Datenknoten in einer aufsteigenden Reihenfolge der Last zu sortieren, P Datenknoten auszuwählen, die in den sortierten Datenknoten vor anderen Datenknoten liegen, und das Anwendergerät anzuweisen, eine Kopie des Datenblocks x in jedem der P ausgewählten Datenknoten zu speichern, wobei der Wert von P gleich der Anzahl an Kopien ist, die anfänglich zu speichern sind, wobei P eine positive Ganzzahl ist.

11. Datenspeicher-Verwaltungsserver nach Anspruch 10, wobei
das zweite Verarbeitungsmodul ferner dafür konfiguriert ist, eine Differenz N1 zwischen der Anzahl an Kopien, die dem Wichtigkeitsniveau entspricht, und

der Anzahl an aktuell gespeicherten Kopien des Datenblocks x zu bestimmen, wenn die Anzahl an aktuell gespeicherten Kopien kleiner als die Anzahl an Kopien ist, die dem Wichtigkeitsniveau entspricht, und die Anzahl M1 an Datenknoten zu bestimmen, die keine Kopie des Datenblocks x speichern, wobei M1 und N1 positive Ganzzahlen sind,
wenn M1 größer als N1 ist, aktuelle Lastinformationen von jedem der M1 Datenknoten zu erhalten, die keine Kopie des Datenblocks x speichern, die M1 Datenknoten in einer aufsteigenden Reihenfolge der Last zu sortieren, N1 Datenknoten auszuwählen, die in den sortierten Datenknoten vor anderen Datenknoten liegen, und eine Kopie des Datenblocks x in jedem der N1 Datenknoten zu speichern,
wenn M1 gleich N1 ist, eine Kopie des Datenblocks x in jeden der M1 Datenknoten zu speichern, die keine Kopie des Datenblocks x speichern, und
wenn M1 kleiner als N1 ist, N1 Kopien des Datenblocks x in den M1 Datenknoten zu speichern, die keine Kopie des Datenblocks x speichern, wobei jeder der M1 Datenknoten mindestens eine Kopie speichert.

12. Datenspeicher-Verwaltungsserver nach Anspruch 11, wobei
das zweite Verarbeitungsmodul ferner dafür konfiguriert ist, aktuelle Lastinformationen von jedem der Datenknoten zu erzielen, die eine Kopie des Datenblocks x speichern, einen Datenknoten auszuwählen, der die geringste Last aufweist und den Datenblock x aus dem ausgewählten Datenkonen zu lesen, um die N1 Kopien zu speichern.

13. Datenspeicher-Verwaltungsserver nach Anspruch 10, wobei
das zweite Verarbeitungsmodul ferner dafür konfiguriert ist, eine Differenz N2 zwischen der Anzahl an aktuell gespeicherten Kopien des Datenblocks x und der Anzahl an Kopien, die dem Wichtigkeitsniveau entspricht, zu bestimmen, wenn die Anzahl an aktuell gespeicherten Kopien größer als die Anzahl an Kopien ist, die dem Wichtigkeitsniveau entspricht, und Bestimmen einer Anzahl M2 an Datenknoten, die eine Kopie des Datenblocks x speichern, wobei M2 und N2 positive Ganzzahlen sind,
wenn M2 größer als N2 ist, aktuelle Lastinformationen von jedem der M2 Datenknoten zu erhalten, die Kopien des Datenblocks x speichern, die M2 Datenknoten in einer absteigenden Reihenfolge der Last zu sortieren, N2 Datenknoten auszuwählen, die in den sortierten Datenknoten vor anderen Datenknoten liegen, und eine Kopie des Datenblocks x aus jedem der N2 Datenknoten zu entfernen,
wenn M2 gleich N2 ist, eine Kopie des Datenblocks x aus jedem der M2 Datenknoten zu entfernen, die eine Kopie des Datenblocks x speichern,
wenn M2 kleiner als N2 ist, N2 Kopien des Datenblocks x aus den M2 Datenknoten zu entfernen, die mindestens eine Kopie des Datenblocks x speichern, wobei aus jedem der M2 Datenknoten mindestens eine Kopie entfernt wird.

14. Datenspeicher-Verwaltungsserver nach Anspruch 11, 12 oder 13, wobei
das erste Verarbeitungsmodul ferner dafür konfiguriert ist, Indexinformationen für den Datenblock x zu ermitteln und die Indexinformationen zu aktualisieren, wenn Inhalte in den Indexinformationen verändert werden,
wobei die Indexinformationen Folgendes umfassen: die Datenart des Datenblocks x, den Wichtigkeitsgrad des Datenblocks x, die Gesamtanzahl an Besuchen des Datenblocks x innerhalb der jüngsten Periode, eine jüngste Anzahl an gespeicherten Kopien des Datenblocks x, einen Datenknoten, in dem jede Kopie des Datenblocks x gespeichert ist, und einen spezifischen Standort der Kopie des Datenblocks x in dem Datenknoten.

15. Datenspeicher-Verwaltungsserver nach Anspruch 14, wobei
die Indexinformationen, wenn der Datenblock x ein Videodatenblock ist, ferner Startzeit und Endzeit des Datenblocks x umfassen,
wobei das zweite Verarbeitungsmodul ferner dafür konfiguriert ist, nach dem Empfangen einer Nachschlageanfrage für Videodaten innerhalb eines spezifizierten Zeitbereichs gemäß Indexinformationen der Datenblöcke aus allen Datenblöcken Datenblöcke zu identifizieren, welche die folgenden Bedingung erfüllen: ein Zeitbereich, der durch Startzeit und Endzeit des Datenblocks definiert ist, weist eine Überschneidung mit dem spezifizierten Zeitbereich auf, die folgenden Vorgänge für jeden der bestimmten Datenblöcke auszuführen: Erhalten von aktuellen Lastinformationen von jedem der Datenknoten, die eine Kopie des Datenblocks speichern, Auswählen eines Datenknotens, der die geringste Last aufweist, als den Datenknoten, aus dem der Datenblock zu lesen ist, Senden eines Lesebefehls an jeden der Datenkonten, die ausgewählt sind, den Datenblock zu lesen.

16. Datenspeicher-Verwaltungsserver nach Anspruch 15, wobei das zweite Verarbeitungsmodul ferner dafür konfiguriert ist, einen Lesebefehl an jeden der gleichzeitig ausgewählten Datenknoten zu senden.

Revendications

1. Procédé de mémorisation de données, ce procédé comprenant
la détermination d'un nombre de copies à mémoriser initialement pour un bloc de données x, après la ré-

ception d'une requête de mémorisation du bloc de données x, la mémorisation du bloc de données x en fonction du nombre de copies à mémoriser initialement (11) ; et

la réalisation des actions suivantes périodiquement : la détermination d'un niveau d'importance actuel du bloc de données x, la détermination si un nombre de copies correspondant au niveau d'importance est identique à un nombre de copies actuellement mémorisé du bloc de données x, et l'ajustement du nombre de copies actuellement mémorisées du bloc de données x pour qu'il soit cohérent avec le nombre de copies correspondant au niveau d'importance si le nombre de copies correspondant au niveau d'importance n'est pas identique au nombre de copies actuellement mémorisé du bloc de données x ; un niveau d'importance plus élevé correspondant à plus de copies qu'un niveau d'importance plus bas (12) ; la détermination d'un niveau d'importance actuel du bloc de données x comprenant : la détermination du niveau d'importance actuel du bloc de données x en fonction du type de données obtenu et du degré d'importance obtenu et d'un nombre total de visites sur le bloc x pendant une période la plus récente.

2. Procédé selon la revendication 1, dans lequel la détermination du nombre de copies à mémoriser initialement pour le bloc de données x comprend : la détermination du nombre de copies à mémoriser initialement pour le bloc de données x en fonction d'un type de données obtenu et d'un degré d'importance obtenu du bloc de données x, le degré d'importance étant défini par un utilisateur.

3. Procédé selon la revendication 2, dans lequel la sauvegarde du bloc de données x en fonction du nombre de copies à mémoriser initialement comprend :

l'obtention d'une information de charge actuelle respectivement de chacun des nodules de données, et le tri des nodules de données dans un ordre de charge ascendant ;

la sélection de P nodules de données qui sont en tête des autres nodules de données dans les nodules de données triés, et la mémorisation d'une copie du bloc de données x dans chacun des P nodules de données sélectionnés ; la valeur de P étant un nombre entier positif et étant égale au nombre de copies à mémoriser initialement.

4. Procédé selon la revendication 3, dans lequel l'ajustement du nombre de copies actuellement mémorisé du bloc de données x devant être cohérent avec le nombre de copies correspondant au niveau d'importance comprend :

la détermination d'une différence N1 entre le nombre de copies correspondant au niveau d'importance et le nombre de copies actuellement mémorisé du bloc de données x si le nombre de copies actuellement mémorisé est inférieur au nombre correspondant au niveau d'importance, et la détermination du nombre M1 de nodules de données qui ne mémorisent pas une copie du bloc de données x ; M1 et M2 étant des nombres entiers positifs ;

si M1 est supérieur à N1, obtention d'une information de charge actuelle de chacun des M1 nodules de données qui ne mémorisent pas une copie du bloc de données x, tri des M1 nodules de données dans un ordre de charge ascendant, sélection des N1 nodules de données qui sont en tête des autres nodules de données dans les nodules de données triés, et mémorisation d'une copie du bloc de données x dans chacun des N1 nodules de données ;

si M1 est égal à M1, mémorisation d'une copie du bloc de données x dans chacun des M1 nodules de données qui ne mémorisent pas une copie de blocs de données x ; et

si M1 est inférieur à N1, mémorisation de N1 copies du bloc de données x dans les M1 nodules de données qui n'ont pas mémorisé une copie du bloc de données x, chacun des M1 nodules de données mémorisant au moins une copie.

5. Procédé selon la revendication 3, dans lequel l'ajustement du nombre de copies actuellement mémorisé du bloc de données x devant être cohérent avec le nombre de copies correspondant au niveau d'importance comprend :

la détermination d'une différence N2 entre le nombre de copies correspondant au niveau d'importance et le nombre de copies actuellement mémorisé du bloc de données x si le nombre de copies actuellement mémorisé est supérieur au nombre correspondant au niveau d'importance, et la détermination d'un nombre M2 de nodules de données qui mémorisent une copie du bloc de données x ; M2 et N2 étant des nombres entiers positifs ;

si M2 est supérieur à N2, obtention d'une information de charge actuelle de chacun des M2 nodules de données qui mémorisent des copies du bloc de données x, tri des M2 nodules de données dans un ordre de charge descendant, sélection de N2 nodules de données qui sont en tête d'autres nodules de données dans les nodules de données triés, et retrait d'une copie de bloc de données x de chacun des N2 nodules de données ;

si M2 est égal à N2, retrait d'une copie du bloc

de données x de chacun des M2 nodules de données qui mémorisent une copie du bloc de données x ; et
si M2 est inférieur à N2, retrait de N2 copies du bloc de données x des M2 nodules de données qui mémorisent au moins une copie du bloc de données, au moins une copie étant retirée de chacun des M2 nodules de données.

6. Procédé selon la revendication 4 ou 5, comprenant en outre :

l'établissement d'une information d'indexation pour le bloc de données x, et la mise à jour de l'information d'indexation lorsque les contenus de l'information d'indexation sont modifiés ;
l'information d'indexation comprenant : le type de données du bloc de données x, le degré d'importance du bloc de données x, le nombre total de visites au bloc de données x pendant la période la plus récente, un nombre le plus récent de copies mémorisé du bloc de données, un nodule de données x dans lequel chaque copie du bloc de données x et un emplacement spécifique de la copie du bloc de données x dans le nodule de données.

7. Procédé selon la revendication 6, dans lequel si le bloc de données x est un bloc de données vidéo, l'information d'indexation comprend en outre : l'heure de début et l'heure de fin du bloc de données x ; ce procédé comprenant en outre :

après la réception d'une demande de recherche de données vidéo dans une plage temporelle spécifique, la détermination de bloc de données qui répondent à la condition suivante parmi tous les blocs de données en fonction de l'information d'indexation des blocs de données ; une plage temporelle définie par la durée de début et la durée de fin du bloc de données s de syndrome se recoupant avec la plage temporelle spécifiée ;
la réalisation des actions suivantes pour chacun des blocs de données déterminés : l'obtention d'une information de charge actuelle de chacun des nodules de données qui mémorise une copie du bloc de données, et la sélection d'un nodule de données ayant la charge la plus légère soue forme d'un nodule de données dans lequel le bloc de données doit être lu ;
la lecture des blocs de données dans le nodule de données sélectionné.

8. Serveur de gestion de stockage de données, comprenant
un premier module de traitement (801) conçu pour déterminer un nombre de copies à mémoriser initialement pour un bloc de données x, après la réception d'une demande de mémorisation du bloc de données x envoyé par un dispositif utilisateur, l'instruction donnée au dispositif utilisateur de mémoriser le bloc de données x dans des nodules de données en fonction du nombre de copies à mémoriser initialement, et l'instruction donnée à un second module de traitement (802) de commencer à fonctionner ; et
le second module de traitement (802) étant conçu pour réaliser les actions suivantes périodiquement : la détermination d'un niveau d'importance actuelle du bloc de données x, la détermination si un nombre de copies correspondant au niveau d'importance est identique à un nombre de copies actuellement mémorisé du bloc de données x, et l'ajustement du nombre de copies actuellement mémorisé du bloc de données x pour qu'il soit cohérent avec le nombre de copies correspondant au niveau d'importance si le nombre de copies correspondant au niveau d'importance n'est pas identique au nombre de copies actuellement mémorisé du bloc de données x ; un niveau d'importance plus élevé correspondant à plus de copies qu'un niveau d'importance plus bas ;
le second module de traitement (802) étant conçu pour déterminer le niveau d'importance actuel du bloc de données x en fonction du type de données obtenu et du degré d'importance obtenu du bloc de données x et du nombre total de visites sur le bloc de données x pendant la période la plus récente.

9. Serveur de gestion de stockage de données selon la revendication 8, dans lequel
le premier module de traitement est conçu pour déterminer le nombre de copies à mémoriser initialement pour le bloc de données x en fonction d'un type de données obtenu et d'un degré d'importance obtenu oui du bloc de données x, le degré d'importance étant défini par un utilisateur.

10. Serveur de gestion de stockage de données selon la revendication 9, dans lequel
le premier module de traitement est en outre conçu pour déterminer une information de charge actuelle de nodules de données respectifs, trier les nodules de données dans un ordre de charge ascendant, sélectionner P modules de données qui sont en tête d'autres nodules de données dans les nodules de données triés, et donner au dispositif utilisateur l'instruction de mémoriser une copie du bloc de données x dans chacun des P modules de données ; la valeur de P étant la même que le nombre de copies à mémoriser initialement, P étant un nombre entier positif.

11. Serveur de gestion de stockage de données selon la revendication 10, dans lequel
le second module de traitement est en outre conçu pour déterminer une différence N1 entre le nombre de copies correspondant au niveau d'importance et

le nombre de copies actuellement mémorisé du bloc de données x si le nombre de copies actuellement mémorisé est inférieur au nombre correspondant au niveau d'importance, et déterminer le nombre M1 de nodules de données qui ne mémorisent pas une copie du bloc de données x ; M1 et N1 étant des nombres entiers positifs ;
si M1 est supérieur à N1, obtention d'une information de charge actuelle de chacun des M1 nodules de données qui ne mémorisent pas une copie du bloc de données x, tri des M1 nodules de données dans un ordre de charge ascendant, sélection de N1 nodules de données qui sont en tête des autres nodules de données dans les nodules de données triés, et mémorisation d'une copie du bloc de données x dans chacun des N1 nodules de données ;
si M1 est égal à N1, mémorisation d'une copie des blocs de données x dans chacun des M1 nodules de données qui ne mémorisent pas une copie du bloc de données x ; et
si M1 est inférieur à N1, mémorisation de N1 copies du bloc de données x dans les M1 blocs de données qui n'ont pas mémorisé une copie du bloc de données x, chacun des M1 blocs de données mémorisant au moins une copie.

12. Serveur de gestion de stockage de données selon la revendication 11, dans lequel
le second module de traitement est en outre conçu pour obtenir une information de charge actuelle de chacun des nodules de données qui mémorisent une copie du bloc de données x, sélectionner un nodule de données qui a la charge la plus légère, et lire le bloc de données x dans le nodule de données sélectionné pour mémoriser les N1 copies.

13. Serveur de gestion de stockage de données selon la revendication 10, dans lequel
le second module de traitement est en outre conçu pour déterminer une différence N2 entre le nombre de copies actuellement mémorisé du bloc de données x et le nombre de copies correspondant au niveau d'importance si le nombre de copies actuellement mémorisé est supérieur au nombre correspondant au niveau d'importance, et déterminer un nombre M2 de nodules de données qui mémorisent une copie du bloc de données x ; M2 et N2 étant des nombres entiers positifs ;
si M2 est supérieur à N2, obtention d'une information de charge actuelle de chacun des M2 nodules de données qui mémorisent des copies du bloc de données x, tri des M2 nodules de données dans un ordre de charge descendant, sélection de N2 nodules de données qui sont en tête des autres nodules de données dans les nodules de données triés, et retrait d'une copie de bloc de données x de chacun des N2 nodules de données ;
si M2 est égal à N2, retrait d'une copie du bloc de données x de chacun des M2 nodules de données qui mémorisent une copie du bloc de données x ;
si M2 est inférieur à N2, retrait de N2 copies du bloc de données x parmi les M2 nodules de données qui mémorisent au moins une copie du bloc de données, au moins une copie étant retirée chacun des M2 nodules de données.

14. Serveur de gestion de stockage de données selon la revendication 11, 12 ou 13, dans lequel
le premier module de traitement est en outre conçu pour établir une information d'indexation pour le bloc de données x, et pour mettre à jour l'information d'indexation lorsque des contenus de l'information d'indexation sont modifiés ;
l'information d'indexation comprenant : le type de données du bloc de données x, le degré d'importance du bloc de données x, le nombre total de visites sur le bloc de données x pendant la période la plus récente, le nombre le plus récent de copies mémorisé dans le bloc de données x, un nodule de données dans lequel chaque copie du bloc de données x est mémorisée et un emplacement spécifique de la copie du bloc de données x dans le nodule de données.

15. Serveur de gestion de stockage de données selon la revendication 14, dans lequel
si le bloc de données x est un bloc de données vidéo, l'information d'indexation comprend en outre : l'heure de début et l'heure de fin du bloc de données x ;
le second module de traitement est en outre conçu pour identifier des blocs de données répondant à la condition suivante parmi tous les blocs de données en fonction d'une information d'indexation des nodules de données après la réception d'une demande de recherche de données vidéo dans une plage temporelle spécifiée ; une plage temporelle définie par l'heure de début et lors de fin du bloc de données se recoupant avec la plage temporelle spécifiée ; la réalisation des procédures suivantes pour chacun des blocs de données identifiés ; l'obtention d'une information de charge actuelle de chacun des nodules de données qui mémorisent une copie du bloc de données, la sélection de nodules de données ayant la charge la plus légère sous forme du nodule de données dans lequel le bloc de données doit être lu ; l'envoi d'une commande de lecture à chacun des nodules de données sélectionnés pour lire le bloc de données.

16. Serveur de gestion de stockage de données selon la revendication 15, dans lequel le second module de traitement est conçu pour envoyer une commande de lecture à chacun des nodules de données sélectionnés concurremment.

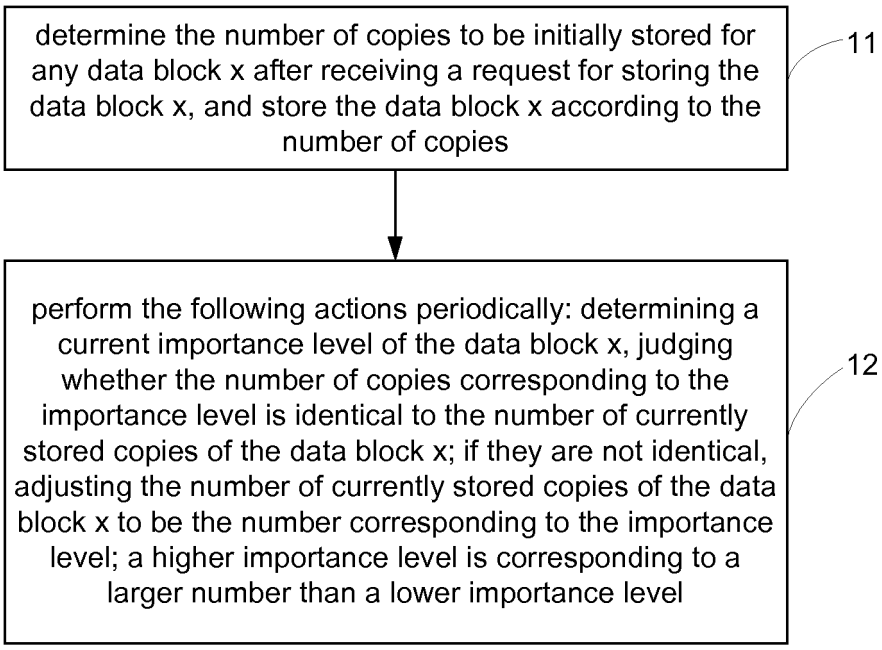
determine the number of copies to be initially stored for any data block x after receiving a request for storing the data block x, and store the data block x according to the number of copies
11
perform the following actions periodically: determining a current importance level of the data block x, judging whether the number of copies corresponding to the importance level is identical to the number of currently stored copies of the data block x; if they are not identical, adjusting the number of currently stored copies of the data block x to be the number corresponding to the importance level; a higher importance level is corresponding to a larger number than a lower importance level
12

FIG.1

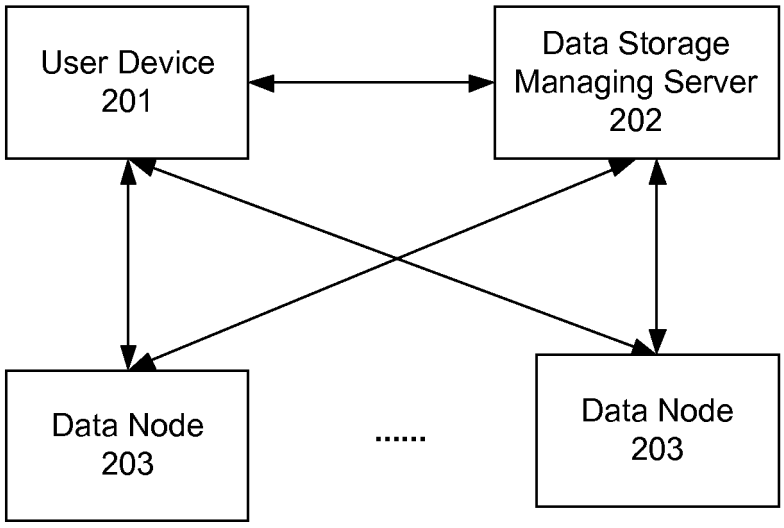
User Device 201
Data Storage Managing Server 202
Data Node 203
.....
Data Node 203

FIG.2

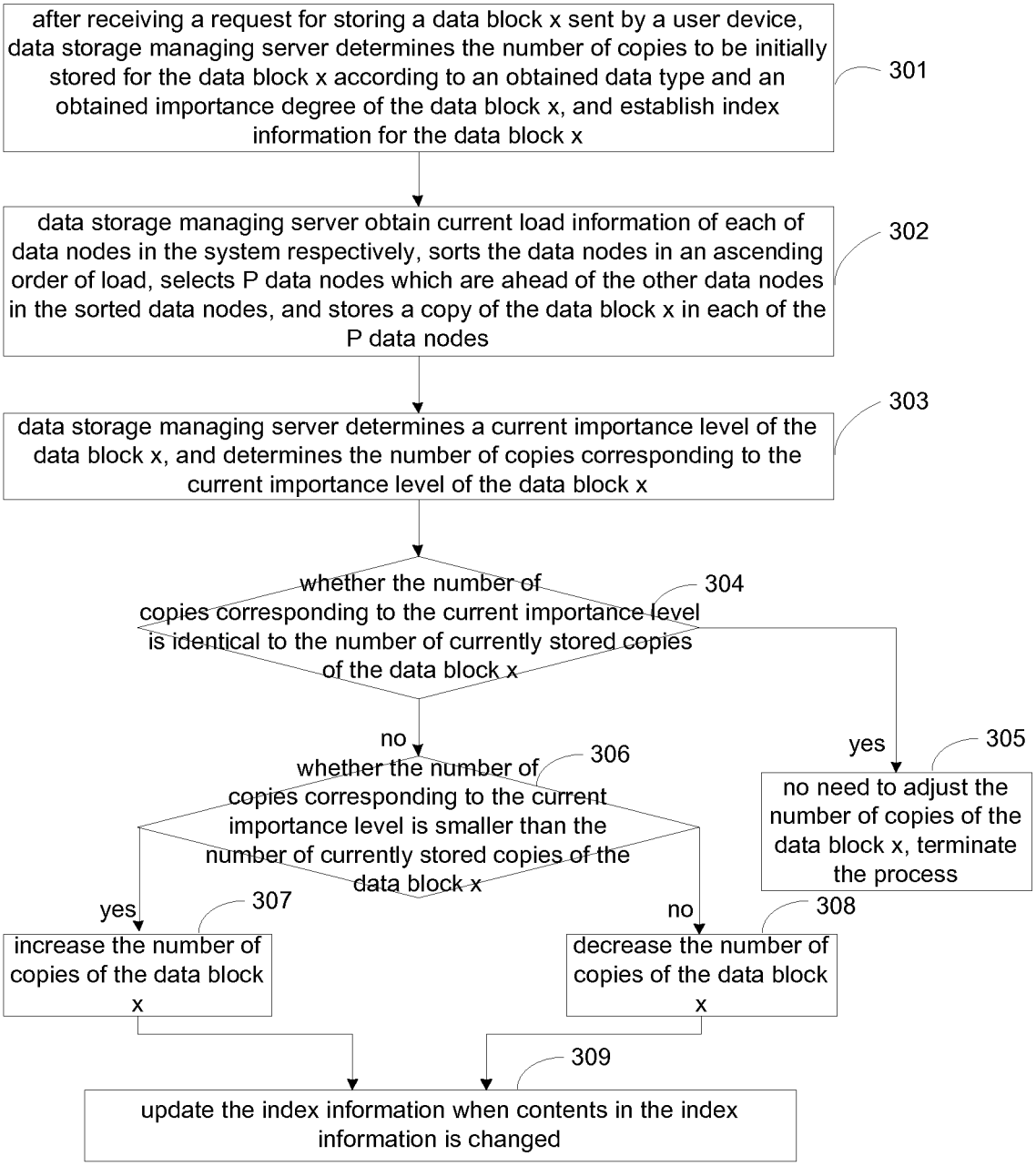
after receiving a request for storing a data block x sent by a user device, data storage managing server determines the number of copies to be initially stored for the data block x according to an obtained data type and an obtained importance degree of the data block x, and establish index information for the data block x
301
data storage managing server obtain current load information of each of data nodes in the system respectively, sorts the data nodes in an ascending order of load, selects P data nodes which are ahead of the other data nodes in the sorted data nodes, and stores a copy of the data block x in each of the P data nodes
302
data storage managing server determines a current importance level of the data block x, and determines the number of copies corresponding to the current importance level of the data block x
303
whether the number of copies corresponding to the current importance level is identical to the number of currently stored copies of the data block x
304
yes
no need to adjust the number of copies of the data block x, terminate the process
305
no
whether the number of copies corresponding to the current importance level is smaller than the number of currently stored copies of the data block x
306
yes
increase the number of copies of the data block x
307
no
decrease the number of copies of the data block x
308
update the index information when contents in the index information is changed
309

FIG.3

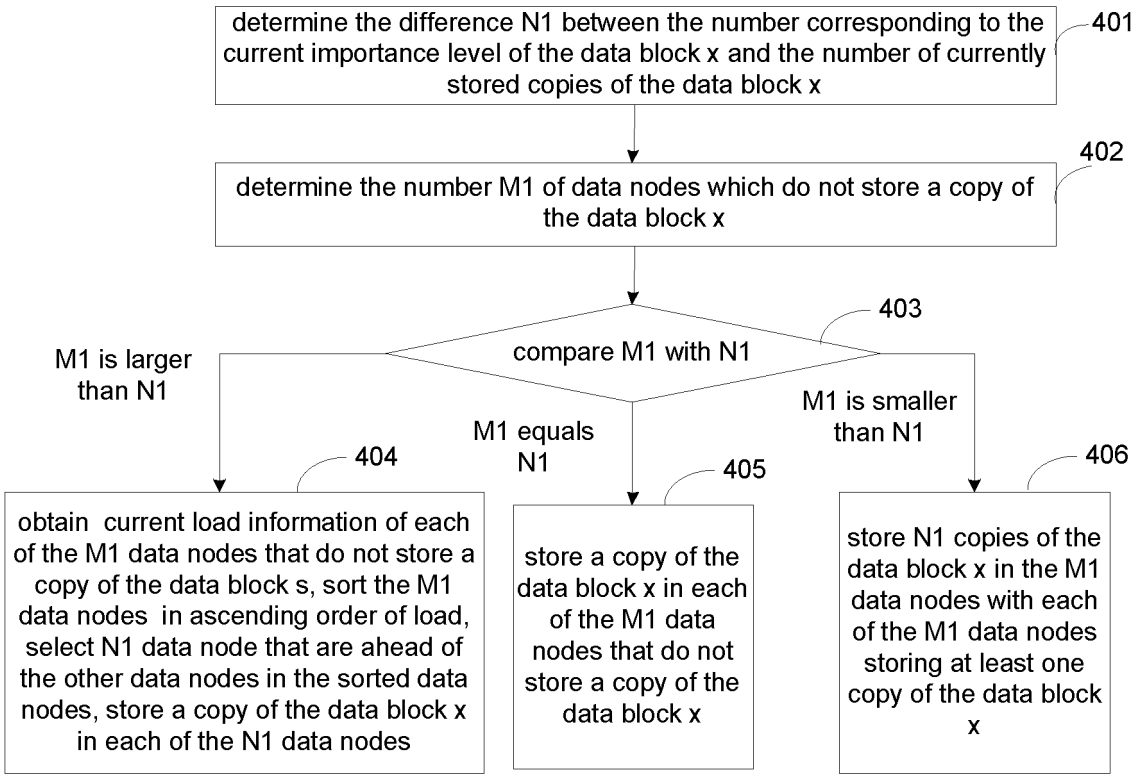
determine the difference N1 between the number corresponding to the current importance level of the data block x and the number of currently stored copies of the data block x
401
determine the number M1 of data nodes which do not store a copy of the data block x
402
compare M1 with N1
403
M1 is larger than N1
M1 equals N1
M1 is smaller than N1
404
405
406
obtain current load information of each of the M1 data nodes that do not store a copy of the data block s, sort the M1 data nodes in ascending order of load, select N1 data node that are ahead of the other data nodes in the sorted data nodes, store a copy of the data block x in each of the N1 data nodes
store a copy of the data block x in each of the M1 data nodes that do not store a copy of the data block x
store N1 copies of the data block x in the M1 data nodes with each of the M1 data nodes storing at least one copy of the data block x

FIG.4

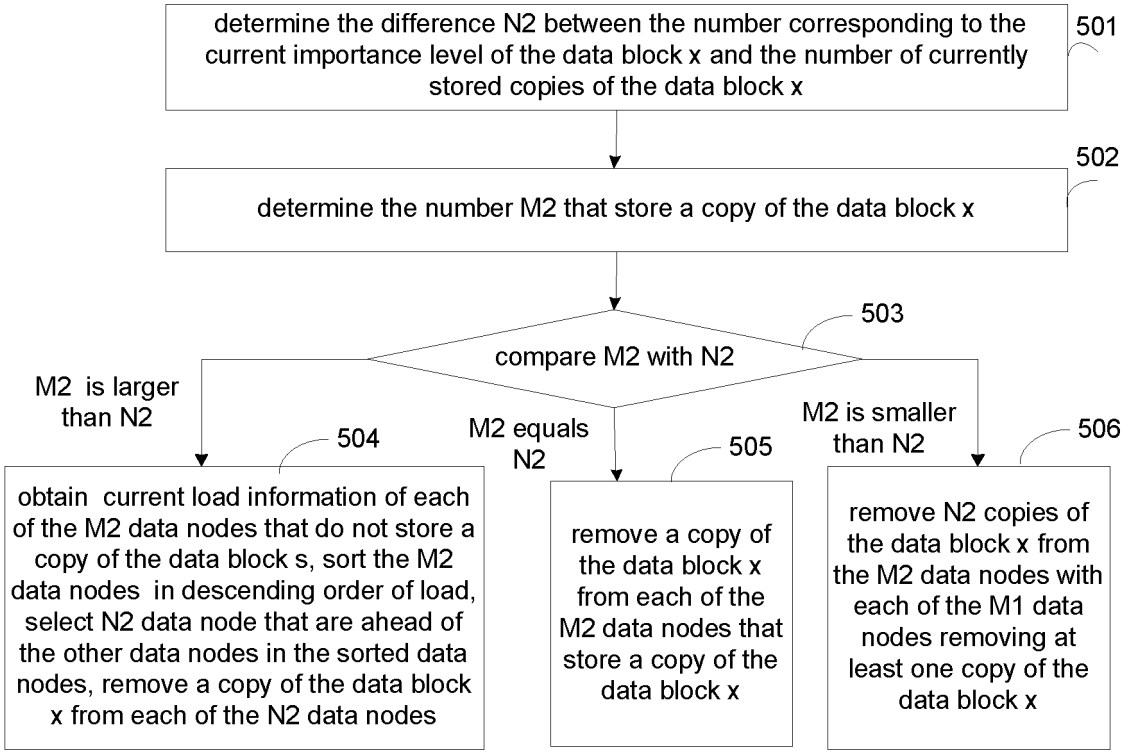
determine the difference N2 between the number corresponding to the current importance level of the data block x and the number of currently stored copies of the data block x
501
determine the number M2 that store a copy of the data block x
502
compare M2 with N2
503
M2 is larger than N2
M2 equals N2
M2 is smaller than N2
504
505
506
obtain current load information of each of the M2 data nodes that do not store a copy of the data block s, sort the M2 data nodes in descending order of load, select N2 data node that are ahead of the other data nodes in the sorted data nodes, remove a copy of the data block x from each of the N2 data nodes
remove a copy of the data block x from each of the M2 data nodes that store a copy of the data block x
remove N2 copies of the data block x from the M2 data nodes with each of the M1 data nodes removing at least one copy of the data block x

FIG.5

Data Type	Importance Level	Start Time	End Time	Total visits	Number of copies	Copy 1		...	Copy n	
						Data Node	Location	...	Data Node	Location

FIG.6

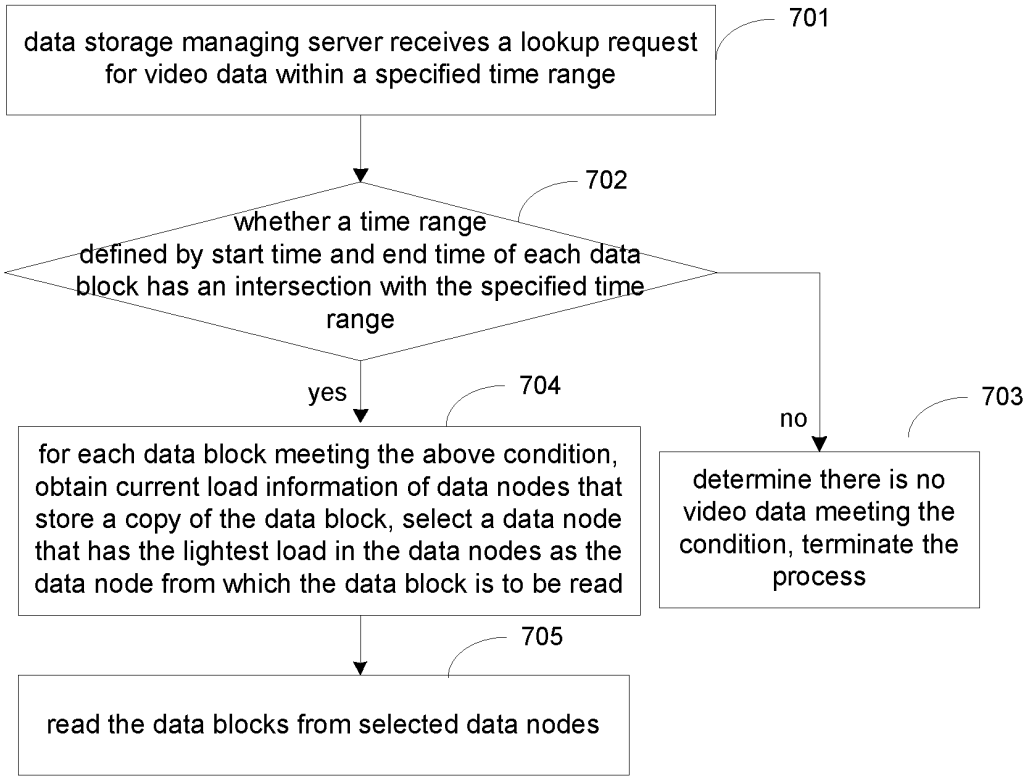

FIG.7

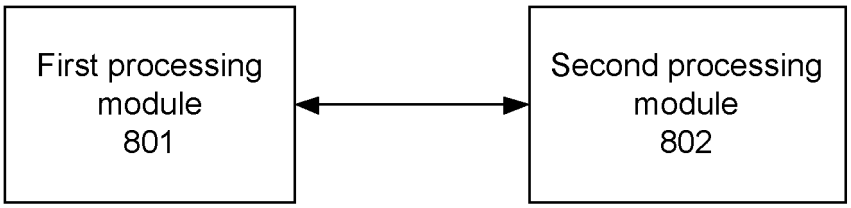
First processing module 801
Second processing module 802

FIG.8

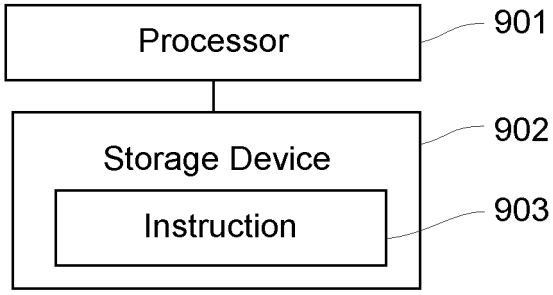
Processor
901
Storage Device
902
Instruction
903

FIG.9

REFERENCES CITED IN THE DESCRIPTION

This list of references cited by the applicant is for the reader's convenience only. It does not form part of the European patent document. Even though great care has been taken in compiling the references, errors or omissions cannot be excluded and the EPO disclaims all liability in this regard.

Patent documents cited in the description

- CN 102622184 A **[0005]**
- CN 101394347 A **[0006]**
- CN 102096698 A **[0007]**
- US 20130046735 A **[0008]**
- US 20140095457 A **[0009]**
- US 20120079424 A **[0010]**