

BLOCKCHAIN CLUSTER PROCESSING SYSTEM AND METHOD, COMPUTER DEVICE AND STORAGE MEDIUM

CROSS REFERENCE TO RELATED APPLICATIONS

[0001] The present application claims priority to Chinese application No. 2017101392761, filed on March 9, 2017, titled “Regional Chain Cluster Processing System and Method.” Afore-mentioned reference(s) are herein incorporated by reference in their entireties.

TECHNICAL FIELD

[0002] The present application relates to the technical field of the Internet, in particular to a blockchain cluster processing system, a method, a computer device and a storage medium.

BACKGROUND

[0003] Blockchain is generally understood as a distributed book, which is also essentially a distributed computing and storage system. The blockchain has a plurality of nodes, different nodes reach consensus through certain calculation rules, and store the transaction execution results after consensus in the database in the form of blocks. the plurality of blockchains constitute blockchains, Blocks may be backed up with each other between multiple nodes. Because each node in the blockchain is independent, failure of the node is inevitable. When there are many failed nodes, the blockchain may not work properly, resulting in low availability of the blockchain. How to effectively improve the availability of blockchain becomes a technical problem to be solved at present.

SUMMARY

[0004] According to various embodiments of the present application, there is provided a blockchain cluster processing system, a method, a computer device, and a storage medium.

[0005] A blockchain cluster processing system includes:

[0006] a computing cluster, the computing cluster including a plurality of computing nodes, only one computing node being operating at the same time; and

[0007] a storage cluster, the storage cluster including a memory database cluster and a disk database cluster, the memory database cluster including a plurality of memory databases, the disk database cluster including a plurality of disk databases; only one memory database and one disk database being operating at the same time;

[0008] the currently operating computing node obtains transaction information, calculates the transaction information to obtain corresponding blockchain data, and writes the blockchain data into the currently operating memory database in the memory database cluster, and writes the blockchain data into the currently operating disk database in the disk database cluster;

[0009] when the computing node malfunctions, the computing cluster switches the computing node to other computing nodes in the computing cluster to continue to work; when the memory database malfunctions, the memory database cluster switches the memory database to other memory databases in the memory database cluster to continue to work; when the disk database malfunctions, the disk database cluster switches the disk database to other disk databases in the disk database cluster to continue to work.

[0010] A block chain cluster processing method includes:

[0011] obtaining transaction information by using a computing node in a computing cluster, and calculating the transaction information to obtain corresponding block chain data; the computing cluster including a plurality of computing nodes, and only one computing node being operating at the same time;

[0012] writing the blockchain data into a memory database cluster so that a memory database currently operating in the memory database cluster receives the blockchain data and stores the blockchain data;

[0013] writing the blockchain data to a disk database cluster such that a disk database in which the disk database cluster currently operating receives the blockchain data and stores the blockchain data; And

[0014] switching to other compute nodes in the compute cluster to continue operating when the compute node fails.

[0015] A computer device includes a memory and a processor, the memory storing computer-executable instructions, the computer-executable instructions, when executed by the processor, causes the processor to perform the following steps:

[0016] obtaining transaction information, calculating the transaction information, and obtaining corresponding blockchain data; the computer device belonging to one computing node in a computing cluster, and only one computing node in the computing cluster is in a working state at the same time;

[0017] writing the blockchain data into a memory database cluster so that a memory database currently working in the memory database cluster receives the blockchain data and stores the blockchain data;

[0018] writing the blockchain data to a disk database cluster such that a disk database in which the disk database cluster currently operates receives the blockchain data and stores the blockchain data; and

[0019] when a failure occurs, switching to other compute nodes in the compute cluster continues to work.

[0020] One or more non-transitory readable storage media storing computer-executable instructions that, when executed by one or more processors, cause the one or more processors to perform the following steps:

[0021] obtaining transaction information, calculating the transaction information, and obtaining corresponding blockchain data; the computer device belonging to one computing node in a computing cluster, and only one computing node in the computing cluster is in a working state at the same time;

[0022] writing the blockchain data into a memory database cluster so that a memory database currently working in the memory database cluster receives the blockchain data and stores the blockchain data;

[0023] writing the blockchain data to a disk database cluster such that a disk database in which the disk database cluster currently operates receives the blockchain data and stores the blockchain data; and

[0024] when a failure occurs, switching to other compute nodes in the compute cluster continues to work.

[0025] One or more embodiments of the present application will be described below with reference to the drawings. It is considered obvious for an ordinary person skilled in the art to obtain the drawings of other embodiments without creative efforts based on these drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

[0026] In order to more clearly illustrate the embodiments of the present

application or the technical solution in the prior art, brief description of the drawings to be used in the description of the embodiments or the prior art is given. The drawings in the description are merely some embodiments of the present application, and for those of ordinary skill in the art, other embodiments of the drawings may also be obtained from these drawings without the effort of creative labors.

[0027] FIG. 1 is a block diagram of a blockchain cluster processing system in one embodiment;

[0028] FIG. 2 is a block diagram of multiple computing clusters in one embodiment;

[0029] FIG. 3 is a flowchart of a blockchain cluster processing method in one embodiment;

[0030] FIG. 4 is a block diagram of a server in one embodiment;

[0031] FIG. 5 is a block diagram of a blockchain cluster processing device in one embodiment.

DETAILED DESCRIPTION OF THE EMBODIMENTS

[0032] The present disclosure will be clearly and completely described in combination with the accompanying drawings of the embodiments such that the purposes, the technical solutions and advantages of the embodiments will be more apparent. Apparently, the embodiments described are part, rather than all, of the embodiments of the invention. It should be understood that any and all embodiments derived by a person skilled in the art based on the embodiments of the invention and without creative effort shall be considered within the scope of the invention.

[0033] In one embodiment, as shown in FIG. 1, a blockchain cluster processing system 100 includes a computing cluster 102 including a plurality of computing nodes and a storage cluster 104. The storage cluster includes a memory database cluster 106 including a plurality of memory databases and a disk database cluster 108 including a plurality of disk databases. The computing cluster 102 has only one computing node operating at the same time, the memory database cluster 106 has only one memory database operating at the same time, and the disk database has only one disk database operating at the same time. The currently operating computing node obtains transaction information, calculates the transaction information to obtain corresponding blockchain data, and writes the blockchain data into the currently operating memory database in the memory database cluster 106, writes the blockchain data into the currently working disk database in the disk database cluster 108, switches to other compute nodes in the compute cluster 102 to continue to operate when the compute node fails. When the memory database fails, switching to other memory databases in the memory database cluster 106 continues to work. When the disk database fails, switching to other disk databases in the disk database cluster 108 continues to work.

[0034] The computing cluster includes a plurality of computing nodes. The computing cluster may be divided according to a preset rule, for example, the preset rule may constitute a computing cluster with a plurality of computing nodes belonging to an institution identifier. The computing cluster may include two or more server nodes. The computing cluster may serve the blockchain through a virtual address. The compute cluster has only one compute node working at the same time, and the other compute nodes remain in standby state. The computing node in operation obtains the transaction information, calculates the transaction information, and calculates the corresponding block chain data. The blockchain data

includes transaction execution results and transaction logs.

[0035] Storage clusters include a memory database cluster and a disk database cluster. Storage clusters can also be divided according to preset rules. The memory database cluster includes multiple memory databases, and the disk database cluster includes multiple disk databases. The memory database and the disk database may also be collectively referred to as the storage nodes of the blockchain. The memory database cluster has only one memory database working at the same time, and other memory databases remain on standby. The disk database cluster operates with only one disk database at the same time, and the other disk databases remain on standby. Multiple memory databases in the memory database cluster maintain data consistency through data replication. Multiple disk databases in a disk database cluster maintain data consistency through data replication.

[0036] The currently working computing node can store the computed blockchain data to the memory database cluster and the disk database cluster, respectively. Specifically, the currently working computing node writes the blockchain data to the unique memory database currently working in the memory database cluster through the memory database interface layer for storage. The currently operating computing node writes the blockchain data to the unique disk database currently operating in the disk database cluster for storage through the disk database interface, data routing, sub-database and sub-table middleware. Since the write speed of the memory database is much higher than the write speed of the disk database, storing the blockchain data into the memory database can effectively improve the storage efficiency. Blockchain data is written to the memory database at a high frequency without affecting blockchain performance. However, the capacity of the memory database is relatively limited, and the disk database is also required to provide permanent storage of blockchain data.

[0037] The computing cluster performs fault monitoring on the currently working computing node. Specifically, the computing cluster may perform fault monitoring on the currently working computing node by means of load balancing, or may perform fault monitoring on the currently working computing node by means of a message-passing-based consistency algorithm, or through the distributed application coordination service Zookeeper (a distributed, open source distributed application coordination service) to perform fault monitoring on the currently working computing node. If the currently operating compute node fails, switching to the other compute nodes in the compute cluster to continue to operate by way of load balancing and / or virtual address drift. By performing load balancing on each computing node in the computing cluster and managing the consistency of the blockchain data, the availability of the computing nodes in the blockchain can be effectively improved.

[0038] The memory database cluster monitors the failure of the currently working memory database. In particular, the fault monitoring may be performed on the currently working memory database by means of load balancing, or the fault monitoring may be performed on the currently working memory database by means of a consistency algorithm based on message passing, or fault monitoring is performed on the currently working memory database through the distributed application coordination service Zookeeper. The blockchain data in each memory database of the memory database cluster is consistent. If the current working memory database fails, switch to other memory databases in the memory database cluster through load balancing to continue to work. Thus, the availability of the memory database in the blockchain can be effectively improved.

[0039] The disk database cluster monitors the failure of the currently working disk database. In particular, fault monitoring may be performed on the currently working disk database by means of load balancing, or fault monitoring may be performed on

the currently working disk database by means of a consistency algorithm based on message passing, or fault monitoring of the currently working disk database through the distributed application coordination service Zookeeper. The blockchain data in each disk database of the disk database cluster is consistent. If the disk database currently working fails, switch to other disk databases in the disk database cluster through load balancing to continue working. Thus, the availability of the disk database in the blockchain can be effectively improved.

[0040] In that embodiment, only one compute node of the computing cluster works at the same time, and the working node obtains transaction information, calculate the transaction information to obtain corresponding block chain data. In addition, that blockchain data is written into the memory database of the current unique work of the memory database cluster for storage, and the blockchain data is written into the disk database of the current unique work of the disk database cluster for storage. In that event of a failure of the compute node, switching to other compute nodes in the computing cluster continue to work. Thus, the availability of the computing nodes in the blockchain can be effectively improved by performing load balancing processing on each computing node and consistency management on the blockchain data. When the memory database fails, switching to other memory databases in the memory database cluster continues to work, which can effectively improve the availability of the memory database in the blockchain. When the disk database fails, switching to other disk databases in the disk database cluster continues to work, which can effectively improve the availability of disk databases in the blockchain. Since the availability of both computing nodes and storage nodes in the blockchain is effectively improved, the availability of the whole blockchain is effectively improved.

[0041] Further, when the computing node in the computing cluster reads the blockchain data, it first reads the memory database that is currently the only working

memory database in the memory database cluster. Since the read speed of the memory database is much faster than the read speed of the disk database, reading data from the memory database cluster can effectively improve the read speed of data. If that read fail, the compute node reads the only currently work disk database in the disk database cluster. If the reading fails again, the local data cache is read, and if the reading fails again, the prompt message of the data reading failure is returned.

[0042] In one embodiment, that compute cluster include a plurality, the computing cluster includes a corresponding institutional identifier, the computing node of one institution identifier corresponding to the computing cluster sends the transaction information to the computing node in the operation of the computing cluster corresponding to the other institution identifier for consensus, obtain the transaction information after the consensus returned by the computing cluster corresponding to the identifier of other institutions, and the computing node calculates the transaction information after the consensus to obtain the corresponding blockchain data.

[0043] The computing cluster may include a corresponding institution identifier. It is assumed that the institution identifiers are A and B, respectively. The computing cluster corresponding to A includes three computing nodes, and the computing cluster corresponding to B includes five computing nodes. Before the computing node calculates the transaction information, a plurality of computing nodes need to agree on the transaction information. In the conventional method, if one of the computing nodes in A initiates a consensus request, the transaction information to be agreed needs to be sent to the other two computing nodes corresponding to A and the five computing nodes corresponding to B for consensus. If the computing node corresponding to A and / or the computing node corresponding to B fail, the consensus result may be inaccurate, which may affect the performance of the blockchain and reduce the availability of the blockchain.

[0044] In this embodiment, each computing cluster has only one computing node operating at the same time. Multiple computing clusters may provide servers to the blockchain through virtual addresses, respectively. A block diagram of multiple computing clusters is shown in FIG. 2. When the computing node in one of the computing clusters needs to agree on the transaction information, the transaction information may be sent to the computing node corresponding to the identifier of the other institution that currently works uniquely in the computing cluster for consensus. If the computing node currently working in the computing cluster, it may be switched to the other computing nodes to continue working in the manner provided in the above embodiment. Therefore, it is possible to avoid the situation where the consensus result is inaccurate due to the computing node failure, and to effectively improve the availability of the blockchain.

[0045] In one embodiment, when that compute node fails to write the blockchain data to the memory database cluster, the computing node establishes a local data cache for the blockchain data. After the memory database cluster switches the memory database that failed to write to other memory databases, the computing node writes the local cached blockchain data to the switched memory database.

[0046] In this embodiment, when the current working memory database in the memory database cluster fails, the memory database cluster switches from the failed memory database to other memory databases. In this switching process, if the computing node writes the blockchain data to the memory database cluster, the write failure will occur. In order to ensure that the blockchain data is not lost and can be stored in the memory database cluster, the computing node establishes a local cache for the blockchain data. After the memory database cluster switches from the failed memory database to the other memory databases, the computing node writes the locally cached blockchain data into the memory database in which the memory

database cluster currently works. In order to ensure that that blockchain data can be completely stored to the memory database cluster.

[0047] In one embodiment, when that compute node fails to write the blockchain data to the disk database cluster, the computing node establishes a local data cache for the blockchain data. After the disk database cluster switches the failed disk database to another disk database, the computing node writes the locally cached blockchain data to the switched disk database.

[0048] In this embodiment, when the disk database currently working in the disk database cluster fails, the disk database cluster switches from the failed disk database to other disk databases. During this switch, if the compute node writes blockchain data to the disk database cluster, a write failure occurs. In order to ensure that the blockchain data is not lost and can be stored in the disk database cluster, the computing node establishes a local cache for the blockchain data. After the disk database cluster switches from the failed disk database to the other disk databases, the compute node writes locally cached blockchain data to the disk database where the disk database cluster is currently working. In order to ensure that that blockchain data can be stored completely to the cluster of disk databases.

[0049] In one embodiment, the disk database cluster includes a write database cluster and a read database cluster, the write database cluster receiving and storing the blockchain data written by the computing node, synchronizing the read database cluster and the write database cluster to save the blockchain data. When the computing node reads the blockchain data, the data is read in the degree database cluster.

[0050] In this embodiment, the disk database may include a write database and a read database. A plurality of write databases constitute a write database cluster, and

a plurality of read database clusters constitute a read database cluster. The write database cluster has only one write database to keep working at the same time, and the read database cluster has only one read database to keep working at the same time. The compute node writes to the write database cluster through the disk database interface layer, the data routing, the partition table middleware and the write database connection pool. The computing node performs a read operation on the read database cluster through the disk database interface layer, the data routing, the partition table middleware, and the read database connection pool.

[0051] When the computing node writes the blockchain data to the disk database cluster, the computing node writes the blockchain data to the write database in which the write database cluster currently works uniquely for storage. If the write database in the write database cluster fails, the switch to the other write databases in the write database cluster continues through the load balancing and/or consistency service. The write database cluster and the read database cluster maintain data consistency through data replication. When the computing node needs to read the blockchain data, the computing node accesses the read database, which is currently the only working one of the read database clusters, to read the data. If the read database in the read database cluster fails, switching to other read databases in the read database cluster through the load balancing and / or consistency service continues to work. By separating the read and write functions of the disk database cluster, the storage and read efficiency of the blockchain data can be effectively improved, and the availability of the blockchain can be effectively improved.

[0052] In one embodiment, the system further includes a security cluster including a plurality of security nodes, the security cluster running only one security node at the same time. The security node receives the verification information uploaded by the computing node, and verifies the validity of the blockchain user

according to the verification information. When the security node fails, the switch to other security nodes in the security cluster continues to run.

[0053] When a blockchain user logs in through a computing node and performs a transaction, the system verifies the validity of the blockchain user through a security cluster. In the conventional manner, a plurality of security nodes are independent, and when the security authentication needs to be performed, the computing node uploads the authentication information to the corresponding security node. If the security node fails, the security verification may not be completed, or the illegal intruder may be verified and passed, thereby adversely affecting the security of the blockchain and reducing the availability of the blockchain.

[0054] In this embodiment, the security cluster includes a plurality of security nodes, and the security node may also be referred to as a CA (Certificate Authority) or a security center. The security cluster serves the blockchain through a virtual address. The security cluster operates with only one security node at the same time, and the other security nodes remain in a standby state. The currently operating security node receives the verification information uploaded by the computing node, and verifies the validity of the blockchain user according to the verification information, thereby ensuring the security of the blockchain data.

[0055] The security cluster may perform fault monitoring on the currently operating security node in the manner provided in the above embodiments. When the currently operating security node fails, switching to other security nodes in the security cluster by way of load balancing and / or virtual address drift continues to work. As a result, the validity verification of the blockchain user can be carried out smoothly, thereby effectively improving the usability of the blockchain.

[0056] Further, in order to facilitate the management of the authentication

information of the blockchain user, the computing cluster corresponding to the plurality of institution identifiers may send the authentication information to the same security cluster. The security cluster receives the verification information sent by the computing node in the computing cluster corresponding to the plurality of institution identifiers, and verifies the validity of the blockchain user corresponding to the plurality of institution identifiers according to the verification information.

[0057] Further, in order to effectively improve the verification efficiency, the security cluster may have a plurality of, and the security cluster may have a corresponding institution identifier. The computing cluster corresponding to the same institution identifier sends verification information to the security cluster of the same institution identifier, and multiple security clusters can simultaneously verify the legality of the use of different blockchains.

[0058] In one embodiment, as shown in FIG. 3, a block chain cluster processing method is provided, and it should be understood that although the steps in the flowchart of FIG. 3 are sequentially displayed as indicated by arrows. However, these steps are not necessarily performed sequentially in the order indicated by the arrows. Unless expressly stated herein, the execution of these steps is not subject to strict sequence limitations, which may be performed in other sequences. Moreover, at least a portion of the steps in FIG. 3 may include a plurality of sub-steps or stages, which are not necessarily performed at the same time, but may be performed at different times. The order of execution is also not necessarily sequential, but may be performed in turn or alternately with at least a portion of the sub-steps or stages of other steps or other steps. The method is applied to a computing cluster, and only one computing node works in the computing cluster at the same time, specifically including the following steps:

[0059] Step 302: obtain transaction information, calculate the transaction

information, and obtain corresponding blockchain data;

[0060] Step 304: write the blockchain data into the memory database cluster, so that a memory database currently working in the memory database cluster receives the blockchain data and stores the blockchain data;

[0061] Step 306: write the blockchain data into the disk database cluster, so that a disk database currently working in the disk database cluster receives the blockchain data and stores the blockchain data;

[0062] Step 308: when the computing node fails, switch to other computing nodes in the computing cluster to continue to work.

[0063] In this embodiment, the computing cluster includes a plurality of computing nodes. The computing cluster may serve the blockchain through a virtual address. The compute cluster has only one compute node working at the same time, and the other compute nodes remain in standby state. The computing node in operation obtains the transaction information, calculates the transaction information, and calculates the corresponding block chain data. The blockchain data includes transaction execution results and transaction logs. If the currently operating compute node fails, switching to the other compute nodes in the compute cluster to continue to operate by way of load balancing and/or virtual address drift. By performing load balancing on each computing node in the computing cluster and managing the consistency of the blockchain data, the availability of the computing nodes in the blockchain can be effectively improved.

[0064] Storage clusters include a memory database cluster and a disk database cluster. Storage clusters can also be divided according to preset rules. The memory database cluster includes multiple memory databases, and the disk database cluster includes multiple disk databases. The memory database and the disk

database may also be collectively referred to as the storage nodes of the blockchain. The memory database cluster has only one memory database working at the same time, and other memory databases remain on standby. The disk database cluster operates with only one disk database at the same time, and the other disk databases remain on standby. Multiple memory databases in the memory database cluster maintain data consistency through data replication. Multiple disk databases in a disk database cluster maintain data consistency through data replication.

[0065] The currently operating computing node writes the blockchain data to the unique memory database currently operating in the memory database cluster through the memory database interface layer for storage. If the current working memory database fails, switch to other memory databases in the memory database cluster by load balancing to continue working. In one embodiment, that method further include establishing a local data cache for the blockchain data when the blockchain data fail to be written to the memory database cluster. After the memory database cluster switches the memory database that failed to write to other memory databases, the local cached blockchain data is written to the switched memory database.

[0066] The currently operating computing node writes the blockchain data to the unique disk database currently operating in the disk database cluster for storage through the disk database interface, data routing, sub-database and sub-table middleware. If the disk database currently working fails, switch to other disk databases in the disk database cluster by load balancing to continue working. In one embodiment, that method further include establishing a local data cache for the blockchain data when the blockchain data fail to be written to the disk database cluster. After the disk database cluster switches the failed disk database to another disk database, the local cached blockchain data is written to the switched disk database.

[0067] In that embodiment, only one compute node of the computing cluster works at the same time, and the working node obtains transaction information, calculate the transaction information to obtain corresponding block chain data. In addition, that blockchain data is written into the memory database of the current unique work of the memory database cluster for storage, and the blockchain data is written into the disk database of the current unique work of the disk database cluster for storage. In that event of a failure of the compute node, switching to other compute nodes in the computing cluster continue to work. Thus, the availability of the computing nodes in the blockchain can be effectively improved by performing load balancing processing on each computing node and consistency management on the blockchain data. When the memory database fails, switching to other memory databases in the memory database cluster continues to work, which can effectively improve the availability of the memory database in the blockchain. When the disk database fails, switching to other disk databases in the disk database cluster continues to work, which can effectively improve the availability of disk databases in the blockchain. Since the availability of both computing nodes and storage nodes in the blockchain is effectively improved, the availability of the whole blockchain is effectively improved.

[0068] In one embodiment, the computing cluster includes a plurality, and the computing cluster includes a corresponding institutional identity. After obtaining the transaction information, the method further includes sending the transaction information to a computing node corresponding to the other institution identifier in the operation of the computing cluster, so that the other computing clusters agree on the transaction information; receiving the transaction information after the consensus returned by the computing cluster corresponding to the identifier of the other institution; the transaction information after the consensus is used for calculation to

obtain corresponding blockchain data.

[0069] In this embodiment, each computing cluster has only one computing node operating at the same time. Multiple computing clusters may provide servers to the blockchain through virtual addresses, respectively. When the computing node in one of the computing clusters needs to agree on the transaction information, the transaction information may be sent to the computing node corresponding to the identifier of the other institution that currently works uniquely in the computing cluster for consensus. If the computing node currently working in the computing cluster, it may be switched to the other computing nodes to continue working in the manner provided in the above embodiment. Therefore, it is possible to avoid the situation where the consensus result is inaccurate due to the computing node failure, and to effectively improve the availability of the blockchain.

[0070] In one embodiment, the method further includes uploading authentication information to the security cluster so that the security node uniquely working at the same time in the security cluster verifies the legitimacy of the blockchain user according to the authentication information. The security cluster includes a plurality of security nodes, and if the currently working security node fails, switching to other security nodes in the security cluster to continue working.

[0071] In this embodiment, the security cluster provides services to the blockchain through a virtual address. The security cluster operates with only one security node at the same time, and the other security nodes remain in a standby state. The currently operating security node receives the verification information uploaded by the computing node, and verifies the validity of the blockchain user according to the verification information, thereby ensuring the security of the blockchain data.

[0072] The security cluster may perform fault monitoring on the currently

operating security node in the manner provided in the above embodiments. When the currently operating security node fails, switching to other security nodes in the security cluster by way of load balancing and / or virtual address drift continues to work. As a result, the validity verification of the blockchain user can be carried out smoothly, thereby effectively improving the usability of the blockchain.

[0073] In one embodiment, the computing node in the computing cluster may be a computer device, such as a server. As shown in FIG. 4, the server 400 includes a processor 401, an internal memory 402, a non-transitory storage medium 403, and a network interface 404 connected through a system bus. The processor 401 is configured to provide computing and control capabilities. The non-transitory storage medium 403 stores therein an operating system 4031 and computer-executable instructions 4032, and the internal memory 402 provides an environment for the execution of the computer-executable instructions 4032. When the computer-executable instructions 4032 are executed by the processor 401, the processor 401 may be caused to execute the aforementioned blockchain cluster processing method. Network interface 404 is used to communicate over a network connection with other computing or storage clusters. It will be understood by those skilled in the art that the structure shown in FIG. 4 are merely a block diagram of a portion of the structure associated with the present application and does not constitute a limitation on the server to which the present application applies. In particular, that serv may include more or fewer component than shown in the drawings, or combine certain components, or have different arrangement of components.

[0074] In one embodiment, as shown in FIG. 5, a block chain cluster processing device is provided, including an obtaining module 502, a calculating module 504, a storing module 506, and a switching module 508:

[0075] An obtaining module 502, configured to obtain transaction information.

[0076] The calculation module 504 is configured to calculate the transaction information to obtain corresponding blockchain data.

[0077] A storage module 506, configured to write the blockchain data into the memory database cluster, so that a memory database currently working in the memory database cluster receives the blockchain data and stores the blockchain data; writing the blockchain data to the disk database cluster so that a disk database in which the disk database cluster currently works receives the blockchain data and stores it.

[0078] The switching module 508 is configured to switch to other computing nodes in the computing cluster to continue to work when the computing node fails.

[0079] In one embodiment, the computing cluster includes a plurality, and the computing cluster includes a corresponding institutional identity. The device further includes a sending module 510 and a receiving module 512:

[0080] The sending module 510 is configured to send the transaction information to the computing nodes in the operation of the computing cluster corresponding to the identifier of other institutions, so as to make the other computing clusters agree on the transaction information.

[0081] The receiving module 512 is configured to receive the transaction information after the consensus returned by the computing cluster corresponding to the identifier of the other institution.

[0082] The calculation module 504 is further configured to perform calculation using the transaction information after the consensus to obtain corresponding blockchain data.

[0083] In one embodiment, the storage module 506 is further configured to establish a local data cache for the blockchain data when the blockchain data fails to

be written into the memory database cluster. After the memory database cluster switches the memory database that failed to write to other memory databases, the local cached blockchain data is written to the switched memory database.

[0084] In one embodiment, that storage module 506 establish a local data cache for the blockchain data when the blockchain data fails to be written to the disk database cluster, and after the disk database cluster switches the disk database fail to other disk databases, writes locally cached blockchain data to the switched disk database.

[0085] All or part of each module in the blockchain cluster processing device may be implemented by software, hardware, and combinations. For example, in the hardware implementation, the receiving module 512 receives the agreed transaction information returned by the computing cluster corresponding to the identifier of another institution through the network interface. The sending module 510 sends the transaction information through the network interface to the computing nodes in operation of the corresponding computing clusters identified by the other institutions so that the other computing clusters agree on the transaction information. The network interface may be an Ethernet card or a wireless network card. The above-mentioned modules may be embedded in or independent of the processor of the base station in hardware form, or may be stored in the memory of the base station in software form so that the processor calls to perform the operations corresponding to the above-mentioned modules. The processor may be a central processing unit (CPU), a microprocessor, or the like.

[0086] One of ordinary skill in the art will appreciate that all or part of the process of implementing the embodiments described above may be accomplished by computer programs that instruct associated hardware, that may be stored in a non-transitory computer readable storage medium. The program, when executed,

may include the flow of embodiments of the methods described above. The storage medium may be a disk, an optical disk, a read-only memory (ROM), or the like.

[0087] The various technical features of the above embodiments may be combined arbitrarily, and in order to simplify the description, all possible combinations of the various technical features in the above embodiments are not described, however, as long as there is no contradiction in the combination of these technical features, Should be considered as the scope of this specification.

[0088] The above embodiments only express several embodiments of the present application, and the description thereof is more specific and detailed, but is not to be construed as limiting the scope of the application. It should be noted that several variations and modifications may be made by those of ordinary skill in the art without departing from the spirit of the present application, which are within the scope of the present application. Therefore, the scope of protection of this patent application shall be subject to the appended claims.

[0089] The foregoing implementations are merely specific embodiments of the present disclosure, and are not intended to limit the protection scope of the present disclosure. It should be noted that any variation or replacement readily figured out by persons skilled in the art within the technical scope disclosed in the present disclosure shall all fall into the protection scope of the present disclosure. Therefore, the protection scope of the present disclosure shall be subject to the protection scope of the claims.

What is claimed is:

1. A blockchain cluster processing system comprises:

a computing cluster, the computing cluster comprising a plurality of computing nodes, wherein only one computing node being operating at the same time; and

a storage cluster, the storage cluster comprising a memory database cluster and a disk database cluster, the memory database cluster comprising a plurality of memory databases, the disk database cluster comprising a plurality of disk databases; only one memory database and one disk database being operating at the same time;

wherein the currently operating computing node obtains transaction information, calculates the transaction information to obtain corresponding blockchain data, and writes the blockchain data into the currently operating memory database in the memory database cluster, and writes the blockchain data into the currently operating disk database in the disk database cluster;

wherein when the computing node malfunctions, the computing cluster switches the computing node to other computing nodes in the computing cluster to continue to work; when the memory database malfunctions, the memory database cluster switches the memory database to other memory databases in the memory database cluster to continue to work; when the disk database malfunctions, the disk database cluster switches the disk database to other disk databases in the disk database cluster to continue to work.

2. The system according to claim 1, wherein the system comprises a plurality of computing clusters, the plurality of computing clusters comprising a corresponding plurality of institution identifiers, the calculation node of one institution identifier corresponding to the calculation cluster sends the transaction information to the

operating calculation node of the calculation cluster corresponding to the other institution identifiers for consensus, and obtains the transaction information after the consensus returned by the calculation cluster corresponding to the identifier of the other institution, the calculation node calculating the transaction information after the consensus to obtain the corresponding blockchain data.

3. The system according to claim 1, wherein when the computing node fails to write the blockchain data into the memory database cluster, the computing node establishes a local data cache for the blockchain data, after the memory database cluster switches the memory database which fail to write into other memory databases, the computing node writes the local cached blockchain data into the switched memory database.

4. The system according to claim 1, wherein when the computing node fails to write the blockchain data to the disk database cluster, the computing node establishes a local data cache for the blockchain data, after the disk database cluster switches the disk database that fails to write into other disk databases, the computing node writes the locally cached blockchain data to the switched disk database.

5. The system according to claim 1, wherein the disk database cluster comprises a write database cluster and a read database cluster, the write database cluster receiving and storing blockchain data written by a computing node, the read database cluster being synchronized with the write database cluster to store blockchain data; when the computing node reads the blockchain data, reading is performed in the database cluster.

6. The system according to claim 1, wherein the system further comprises a security cluster, the security cluster comprising a plurality of security nodes, the security cluster running only one security node at the same time; the security node being configured to receive the verification information uploaded by the computing node and verify the validity of the blockchain user according to the verification information; when the security node fails, the security cluster switches the security node to other security nodes in the security cluster to continue to operate.

7. A block chain cluster processing method comprises:

obtaining transaction information by using a computing node in a computing cluster, and calculating the transaction information to obtain corresponding block chain data; the computing cluster comprising a plurality of computing nodes, and only one computing node being operating at the same time;

writing the blockchain data into a memory database cluster so that a memory database currently operating in the memory database cluster receives the blockchain data and stores the blockchain data;

writing the blockchain data to a disk database cluster such that a disk database in which the disk database cluster currently operating receives the blockchain data and stores the blockchain data; And

switching to other compute nodes in the compute cluster to continue operating when the compute node fails.

8 The method according to claim 7, wherein after obtaining transaction information by using one of the computing nodes in the computing cluster, the method further

comprises:

sending the transaction information to an operating computing node of the computing cluster corresponding to identifiers of other institutions to make the other computing clusters reach on a consensus of the transaction information;

receiving the transaction information after the consensus returned by the computing cluster corresponding to the identifier of the other institution; and

using the transaction information after the consensus for calculation to obtain corresponding blockchain data.

9 The method according to claim 8 further comprising:

establishing a local data cache for the blockchain data when the blockchain data fails to be written into the memory database cluster; and

after the memory database cluster switches the memory database that failed to write to other memory databases, the local cached blockchain data is written to the switched memory database.

10. A method according to claim 8 further comprising:

establishing a local data cache for the blockchain data when the blockchain data fails to be written into the disk database cluster; and

after the disk database cluster switches the failed disk database to another disk database, the local cached blockchain data is written to the switched disk database.

11. The method according to claim 7 further comprising:

uploading verification information to the security nodes in the security cluster, so that the security nodes in the security cluster verify the legitimacy of the blockchain user according to the verification information; and the security cluster comprises a plurality of security nodes, and only one security node works at the same time.

12. A computer device comprises a memory and a processor, the memory storing computer-executable instructions, the computer-executable instructions, when executed by the processor, causes the processor to perform the following steps:

obtaining transaction information, calculating the transaction information, and obtaining corresponding blockchain data; the computer device belonging to one computing node in a computing cluster, and only one computing node in the computing cluster is in a working state at the same time;

writing the blockchain data into a memory database cluster so that a memory database currently working in the memory database cluster receives the blockchain data and stores the blockchain data;

writing the blockchain data to a disk database cluster such that a disk database in which the disk database cluster currently operates receives the blockchain data and stores the blockchain data; and

when a failure occurs, switching to other compute nodes in the compute cluster continues to work.

13. The computer device according to claim 12, wherein after obtaining transaction information by using one of the computing nodes in the computing cluster, the processor is also caused to perform the following steps:

sending the transaction information to the operating computing nodes of the computing cluster corresponding to the identifier of other institutions so that the computing nodes in the other computing clusters agree on the transaction information; The computer cluster includes a corresponding institution identifier;

receiving the transaction information after the consensus returned by the computing cluster corresponding to the identifier of the other institution; and

using the transaction information after the consensus for calculation to obtain corresponding blockchain data.

14. The computer device according to claim 13, wherein when the computer-executable instructions are executed by the processor, the processor is further caused to perform the following steps:

establishing a local data cache for the blockchain data when the blockchain data fails to be written into the memory database cluster; and

after the memory database cluster switches the memory database that failed to write to other memory databases, the local cached blockchain data is written to the switched memory database.

15. The computer device according to claim 13, wherein when the computer-executable instructions are executed by the processor, the processor is further caused to perform the following steps:

establishing a local data cache for the blockchain data when the blockchain data fails to be written into the disk database cluster; And

after the disk database cluster switches the failed disk database to another disk

database, the local cached blockchain data is written to the switched disk database.

16. The computer device according to claim 12, wherein when the computer-executable instructions are executed by the processor, the processor is also caused to perform the following steps:

uploading verification information to the security nodes in the security cluster, so that the security nodes in the security cluster verify the legitimacy of the blockchain user according to the verification information; and the security cluster includes a plurality of security nodes, and only one security node works at the same time.

17. one or more non-transitory readable storage media storing computer-executable instructions that, when executed by one or more processors, cause the one or more processors to perform the following steps:

obtaining transaction information, calculating the transaction information, and obtaining corresponding blockchain data; the computer device belonging to one computing node in a computing cluster, and only one computing node in the computing cluster is in a working state at the same time;

writing the blockchain data into a memory database cluster so that a memory database currently working in the memory database cluster receives the blockchain data and stores the blockchain data;

writing the blockchain data to a disk database cluster such that a disk database in which the disk database cluster currently operates receives the blockchain data and stores the blockchain data; and

when a failure occurs, switching to other compute nodes in the compute cluster

continues to work.

18. A non-transitory readable storage medium according to claim 17, wherein after obtaining transaction information using one of the computing nodes in the computing cluster, causing the one or more processors to perform the following steps:

sending the transaction information to the computing nodes in the operation of the computing cluster corresponding to the identifier of other institutions so that the computing nodes in the other computing clusters agree on the transaction information; the computer cluster comprising a corresponding institution identifier;

receiving the transaction information after the consensus returned by the computing cluster corresponding to the identifier of the other institution; and

using the transaction information after the consensus for calculation to obtain corresponding blockchain data.

19. A non-transitory readable storage medium according to claim 18, wherein when the computer-executable instructions are executed by one or more processors, the one or more processors are caused to perform the following steps:

establishing a local data cache for the blockchain data when the blockchain data fails to be written into the memory database cluster; and

after the memory database cluster switches the memory database that failed to write to other memory databases, the local cached blockchain data is written to the switched memory database.

20. A non-transitory readable storage medium according to claim 18, wherein when

the computer-executable instructions are executed by one or more processors, the one or more processors are caused to perform the following steps:

establishing a local data cache for the blockchain data when the blockchain data fails to be written into the disk database cluster; and

after the disk database cluster switches the failed disk database to another disk database, the local cached blockchain data is written to the switched disk database.

ABSTRACT

The disclosure relates to a blockchain cluster processing system, including: a compute cluster, in which only one compute node is in a working state at any one time; a memory database cluster, in which only one memory database is in the working state at any one time; and a disk database cluster, in which only one disk database is in the working state at any one time; the current working compute node obtains corresponding blockchain data by means of computation, writes the blockchain data into the current working memory database in the memory database cluster, and writes the blockchain data into the current working disk database in the disk database cluster; when the compute node fails, switching to another compute node in the compute cluster is performed to continue operation; when the memory database fails, switching to another memory database in the memory database cluster is performed to continue operation; and when the disk database fails, switching to another disk database in the disk database cluster is performed to continue operation.

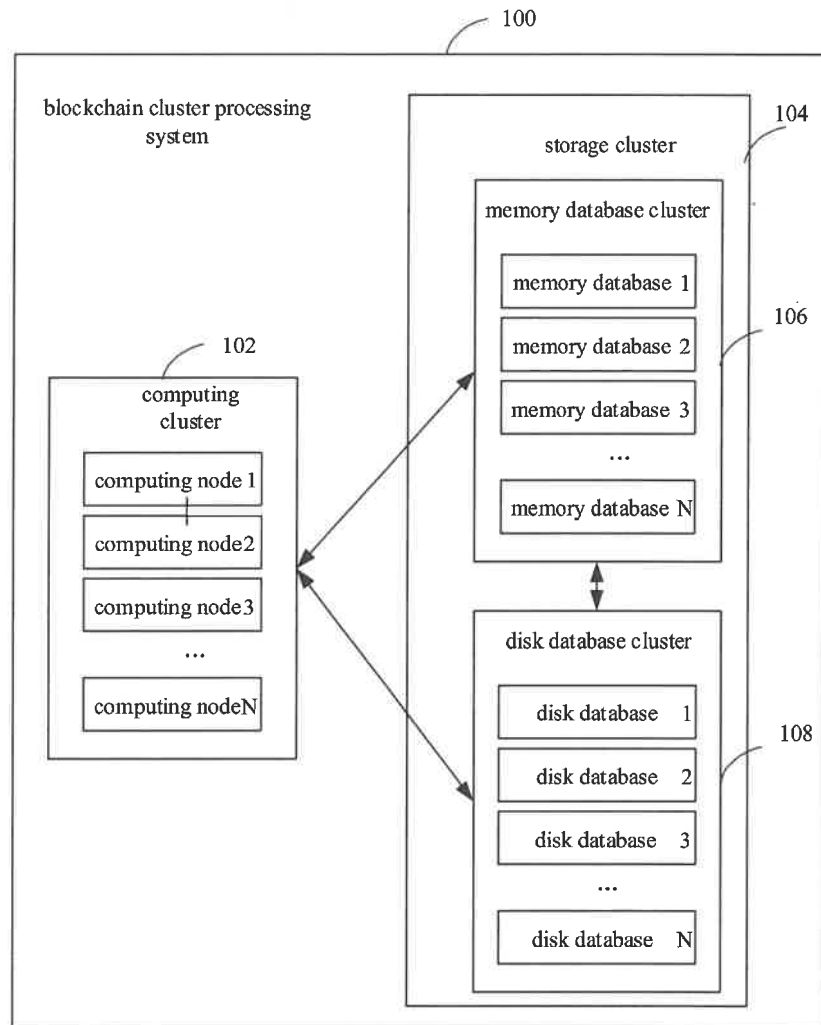


FIG. 1

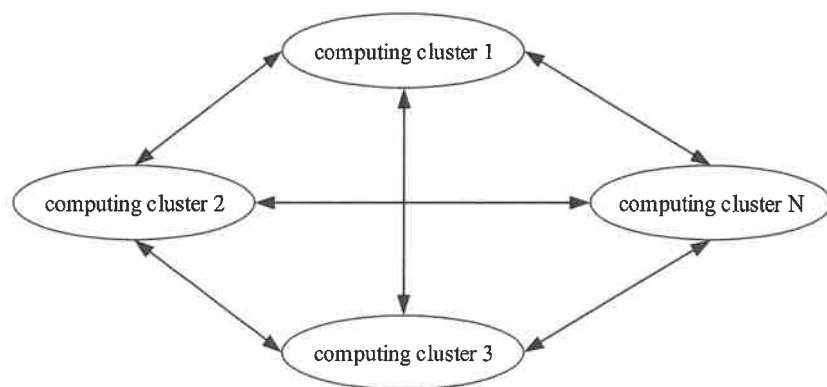


FIG. 2

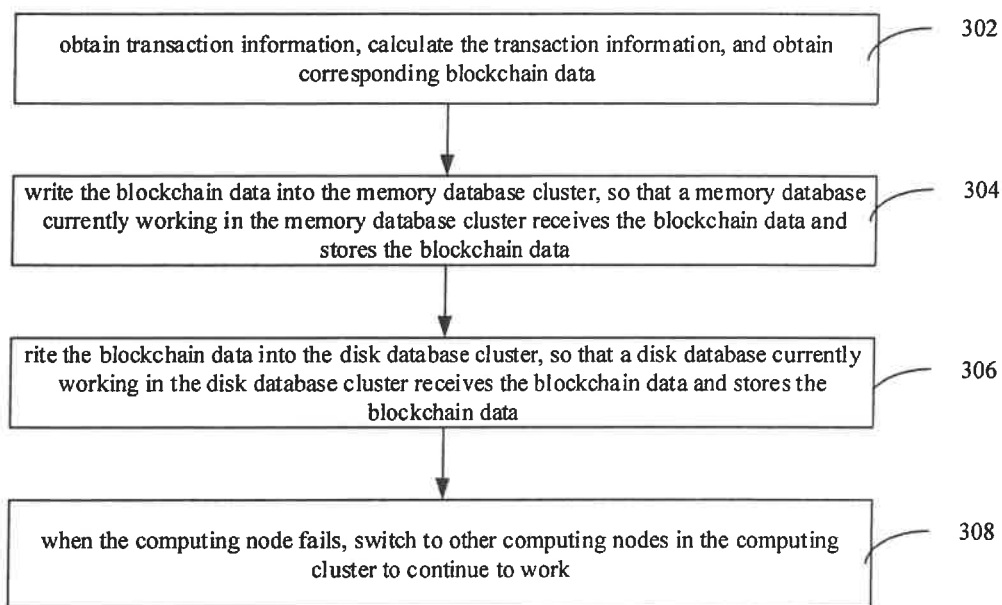


FIG. 3

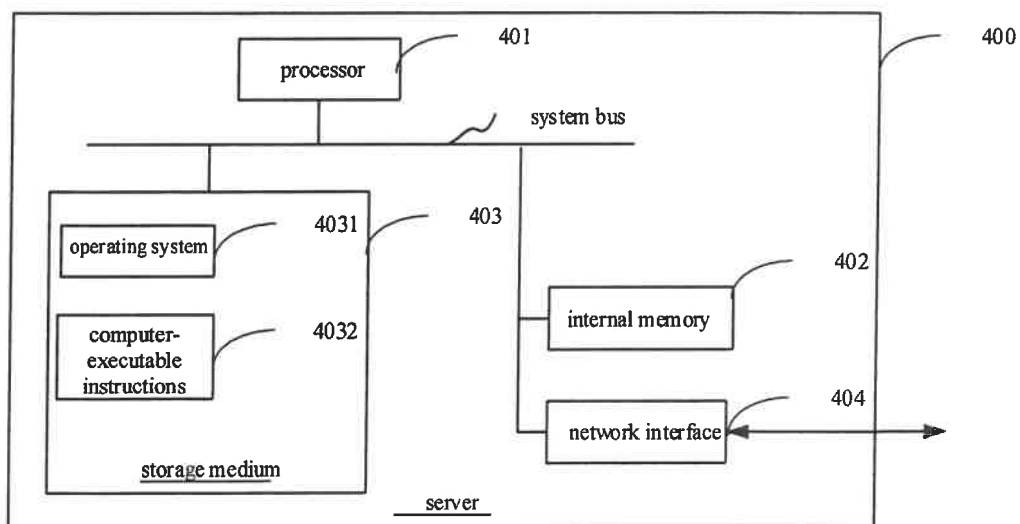


FIG. 4

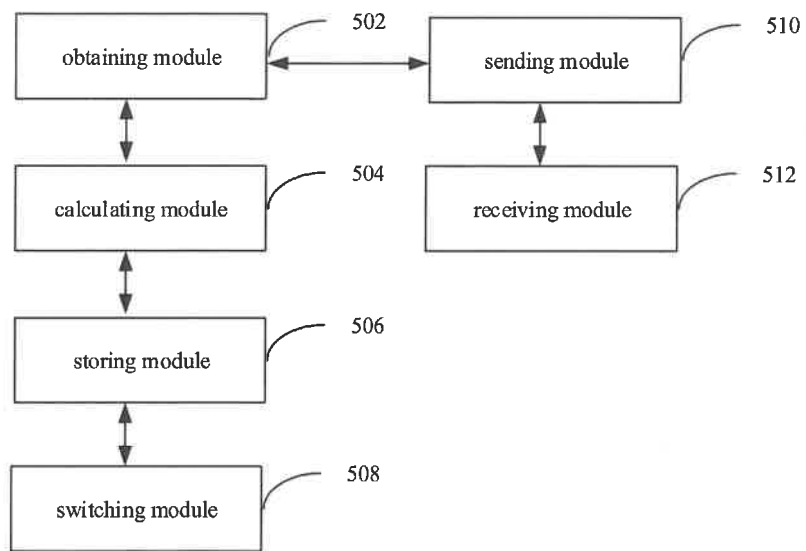


FIG. 5