

TRANSACTION DATA PROCESSING METHOD AND DEVICE, COMPUTER DEVICE, AND STORAGE MEDIUM

CROSS-REFERENCE TO RELATED APPLICATIONS

This application claims priority to the Chinese Patent Application No. 2017111807450, filed
5 with the National Intellectual Property Administration, PRC on November 23, 2017, and entitled
"TRANSACTION DATA PROCESSING METHOD AND DEVICE, COMPUTER DEVICE, AND
STORAGE MEDIUM", which is incorporated herein by reference in its entirety.

TECHNICAL FIELD

This application relates to a transaction data processing method and device, a computer device,
10 and a storage medium.

BACKGROUND

A blockchain is generally understood to be a distributed account book, which is essentially a
distributed computing and storage system. In a blockchain or a distributed transaction system
implemented based on a PaxOS algorithm, a plurality of transaction nodes are required to participate
15 in a transaction, and a final confirmation of a calculation result is achieved based on a certain
processing policy.

The inventor realizes that in a conventional technology, after receiving a transaction request, a
client directly allocates the transaction request to different transaction nodes for calculation, and the
different transaction nodes obtain a transaction result according to a processing policy and feed back
20 the transaction result to the client. Since the transaction result is obtained by intercommunication
between the transaction nodes, when a transaction volume is large, it is prone to cause transaction
data confusion, and as a result, a transaction result is wrong.

SUMMARY

According to various embodiments disclosed by this application, a transaction data processing

method and device, a computer device, and a storage medium are provided.

A transaction data processing method includes:

receiving, by a client, a transaction data processing request, the transaction data processing request carrying a transaction type and transaction data; transmitting, by the client, according to the transaction type, the transaction data to a corresponding channel;

transmitting, by the channel, the transaction data to a transaction node therein;

performing, by the transaction node, processing on the transaction data to obtain a preliminary transaction result, and feeding back the preliminary transaction result to the channel;

performing, by the channel, according to the preliminary transaction result, calculation to obtain a target transaction result, and returning the target transaction result to the client; and

issuing, by the client, the target transaction result.

A blockchain transaction data processing system includes a client, channels, and transaction nodes;

the client is configured to receive a transaction data processing request, the transaction data processing request carrying a transaction type and transaction data; the client transmits, according to the transaction type, the transaction data to a corresponding channel;

the channel is configured to transmit the transaction data to a transaction node therein;

the transaction node is configured to perform processing on the transaction data to obtain a preliminary transaction result, and feed back the preliminary transaction result to the channel;

the channel is further configured to perform, according to the preliminary transaction result, calculation to obtain a target transaction result, and return the target transaction result to the client; and

the client is further configured to issue the target transaction result.

A computer system includes a client, channels, and transaction nodes;

the client includes a first memory and one or more first processors; the memory stores a computer readable instruction therein; when the computer readable instruction is executed by the first processor, the one or more first processors execute the following steps: receiving a transaction data processing request, the transaction data processing request carrying a transaction type and transaction data; transmitting, according to the transaction type, the transaction data to a corresponding channel; and issuing a target transaction result;

the channel includes a second memory and one or more second processors; the memory stores therein a computer readable instruction; when the computer readable instruction is executed by the second processor, the one or more second processors execute the following steps: transmitting the transaction data to a transaction node in the channel; performing, according to a preliminary transaction result, calculation to obtain a target transaction result; and returning the target transaction result to the client;

the transaction node includes a third memory and one or more third processors; the memory stores therein a computer readable instruction; when the computer readable instruction is executed by the third processor, the one or more third processors execute the following steps: performing processing on the transaction data to obtain a preliminary transaction result; and feeding back the preliminary transaction result to the channel.

One or more non-volatile computer readable storage media storing a computer readable instruction are provided; when the computer readable instruction is executed by one or more first processors, the one or more first processors execute the following steps: receiving a transaction data processing request, the transaction data processing request carrying a transaction type and transaction data; transmitting, according to the transaction type, the transaction data to a corresponding channel; and issuing a target transaction result;

when the computer readable instruction is executed by one or more second processors, the one or more second processors execute the following steps: transmitting the transaction data to a transaction

node in the channel; performing, according to a preliminary transaction result, calculation to obtain a target transaction result; and returning the target transaction result to the client;

when the computer readable instruction is executed by one or more third processors, the one or more third processors execute the following steps: performing processing on the transaction data to obtain a preliminary transaction result; and feeding back the preliminary transaction result to the channel.

Details of one or more embodiments of this application are set forth in accompanying drawings and description below. Other features and advantages of this application will be apparent through the description, the accompanying drawings and the claims.

BRIEF DESCRIPTION OF DRAWINGS

In order to explain technical solutions in embodiments of this application more clearly, the accompanying drawings used in the embodiments will be briefly introduced below. Apparently, the accompanying drawings in the following description are only some embodiments of this application, and a person of ordinary skill in the art can further obtain other accompanying drawings according to these accompanying drawings without involving any creative effort.

FIG. 1 is an application environment diagram of a transaction data processing method according to one or more embodiments;

FIG. 2 is a flowchart of a transaction data processing method according to one or more embodiments; and

FIG. 3 is a block diagram of a computer system according to one or more embodiments.

DESCRIPTION OF EMBODIMENTS

To make the technical solutions and advantages of this application clearer and more comprehensible, the following further describes this application in detail with reference to the accompanying drawings and embodiments. It should be understood that the specific embodiments described herein are merely illustrative of this application and are not intended to limit this

application.

Referring to FIG. 1, illustrated is an application environment diagram of a transaction data processing method according to an embodiment. A computer system includes a client, channels, transaction nodes, an organization, and a CA server, where the client is independent of the transaction node, can be deployed in the form of a cluster, and provides a transaction request for the outside in a load-balancing mode. The external transaction request generally includes a transaction type and transaction data. The client agrees with the outside on an encryption and decryption algorithm and a key length thereof (namely a security level), a countersign and verification algorithm and a key length thereof, and a certificate format. The client generates a transaction identifier according to the transaction type, selects a matching channel according to the transaction type, and transmits the transaction identifier and transaction data after security processing to the matching channel. The transaction data processing method can be applied to a blockchain or a distributed transaction system implemented based on a PaxOS algorithm.

The CA server provides the download of a root CA certificate and provides the user key and certificate generation for each organization, and can provide the generation of security elements required for Transport Layer Security (TLS). The CA servers provide services for the outside in a cluster mode. The organization first downloads a root CA certificate from the CA server, then receives a user registration, and provides each user with a key and certificate according to the root CA certificate. One or more users can be registered in the organization. The organization allocates appropriate transaction nodes for each user, allocates the transaction nodes to appropriate channels according to the transaction type, and can set user permissions. The permissions may include a transaction execution permission and a transaction data read permission, where a user with the transaction execution permission can have the transaction data read permission at the same time, while a user with the transaction data read permission does not have the transaction execution permission.

The transaction node can support a cluster mode and a load-balancing mode in deployment. In addition to basic attributes, attributes of the transaction node further include an IP address of the

transaction node, a port of the transaction node, an IP address of an event, a port of an event, an organization identifier of the organization, a user of the organization on the channel where the transaction node is located, a user encryption key and certificate, a certificate format, an encryption and decryption algorithm agreed by the organization and the channel, a countersign and verification algorithm, a key length of each algorithm, and the like. After receiving a transaction request of the channel, the transaction node first verifies a field of a signature. After the successful verification, it is determined whether an organization identifier in the received field is consistent with an organization identifier stored in the transaction node; if the organization identifiers are inconsistent, the transaction is directly rejected; and if the organization identifiers are consistent, continue to determine whether user permissions are correct. If the user permissions are not correct, the transaction is directly rejected; and if the user permissions are correct, the transaction node executes the transaction, and returns a preliminary transaction result to the channel.

When initialized, the channel acquires a key and certificate of a user of an organization to which each transaction node belongs, and the key and certificate are stored only after the successful verification by the CA server, so that the user key and certificate can be prevented from being tampered. After receiving a message request sent by the client, the channel traverses the transaction nodes therein, and reads the user key and certificate of the transaction node stored in the channel, and transmits an obtained transaction identifier, transaction data, and an organization identifier to the corresponding transaction node, and before the transmission, the obtained transaction identifier, the transaction data, and the organization identifier can be encrypted in an agreed encryption mode and countersigned in an agreed countersign mode.

In one embodiment, referring to FIG. 2, a transaction data processing method is provided. This embodiment is exemplified by the application of the method to the computer system in FIG. 1 described above. A blockchain transaction data processing readable instruction is run on the computer system, and the transaction data processing method is implemented through the blockchain transaction data processing readable instruction. The method specifically includes the following steps.

In step S202: A client receives a transaction data processing request, where the transaction data processing request carries a transaction type and transaction data; and the client transmits, according to the transaction type, the transaction data to a corresponding channel.

Specifically, the client may receive a transmitted transaction data processing request transmitted by the outside, where the transaction data processing request carries a transaction type and transaction data. The client can pre-agree with the outside on an encryption and decryption algorithm and an algorithm length thereof, a countersign and verification algorithm and an algorithm length thereof, a certificate format, a definition of a transaction type, transaction data that needs to be encrypted and decrypted, data that needs to be countersigned and verified, and the like. During the transaction, the outside transmits the transaction type and the transaction data and the like to the client, and when receiving the transaction data processing request, the client first verifies the legality of the transaction data processing request. For example, countersigned data is verified according to the verification algorithm; after the successful verification, the encrypted data is decrypted to obtain a transaction type and transaction data; and a transaction identifier is generated according to the transaction type, and the transaction identifier and the transaction data are transmitted to the channel corresponding to the transaction type.

The transaction identifier is an identifier that can uniquely determine this transaction, which can be determined according to the transaction type and a serial number. For example, if the transaction type is the same, the serial number of the transaction is incremented and unique. The client uses the transaction identifier as a header field of a message and uses the transaction data as a body field of the message to generate a first message, and transmits the first message to the channel corresponding to the transaction type.

In step S204: The channel transmits the transaction data to a transaction node therein.

Specifically, the channel that receives the first message traverses transaction nodes in the channel, and reads a user key and certificate of the transaction node stored in the channel, uses the transaction identifier and the organization identifier of the organization to which the transaction node

belongs as a header field, uses the transaction data as a body field to generate a second message, encrypts a pre-determined field according to a pre-agreed encryption algorithm, for example, encrypts the body field, and then performs countersign according to a pre-determined signed field, for example, both the header field and the body field are countersigned, and then the signature and all the
5 fields are transmitted to each transaction node in the channel.

It should be noted that each channel and the transaction node of each organization can agree on a secure communication mode in two ways: the first way is that the channel acquires a user key and certificate of a user of the corresponding transaction type of the organization to which each transaction node added to the channel belongs or acquires the user key and certificate of the user from
10 the organization offline, verifies the acquired user key and certificate through the CA server, saves the successfully verified user key and certificate, and agrees with the organization offline on an encryption and decryption algorithm and an algorithm length thereof (namely, security level), a countersign and verification algorithm and an algorithm length thereof, and a certificate format. The second way is that the channel agrees with the transaction node of each organization on a TLS
15 protocol version and secure communication series options through Transport Layer Security (TLS). In addition, the communication between the client and channel is intranet communication, which further ensures communication security.

[In step](#) S206: The transaction node performs processing on the transaction data to obtain a preliminary transaction result, and feeds back the preliminary transaction result to the channel.

Specifically, the transaction node has two response modes to the channel request: one is a synchronous API call response mode, and the other is an asynchronous event message response mode. After receiving the transaction request of the channel, the transaction node first verifies the signed field. If the verification fails, the transaction node directly returns a preliminary transaction result of transaction failure, and if the verification is successful, the transaction node continues to decrypt the
25 encrypted field. If the decryption fails, a preliminary transaction result of transaction failure is directly returned, and if the decryption is successful, the transaction data obtained by the decryption is processed according to preset transaction logic. If the transaction is successful, a preliminary

transaction result of transaction success is obtained, and if the transaction fails, a preliminary transaction result of transaction failure is obtained and fed back to the channel, so that the channel obtains a target transaction result.

No matter whether the preliminary transaction result is a transaction success, the preliminary transaction result is returned to the channel. The second message can be transmitted to the channel in a response mode. The second message may also include a header field and a body field, and the body field is signed, where the header field may store a transaction identifier and an organization identifier, and the body field stores the preliminary transaction result.

[In step S208](#): The channel performs, according to the preliminary transaction result, calculation to obtain a target transaction result, and returns the target transaction result to the client.

Specifically, the channel performs, according to the preliminary transaction result returned by the transaction node participating in the transaction in the channel, calculation to obtain a target transaction result. When receiving the second message returned by the transaction node, the channel may first find a public key in the certificate of the user corresponding to the locally stored organization identifier according to the organization identifier to verify the second message. After the verification is successful, a preliminary transaction result is recorded through the transaction identifier, and then a target transaction result is obtained according to a preset transaction policy and all received preliminary transaction results and fed back to the client.

[In step S210](#): The client issues the target transaction result.

Specifically, after receiving the target transaction result, the client encrypts and countersigns the target transaction result according to the externally agreed encryption and decryption algorithm and the countersign and verification algorithm, and then issues the encrypted and countersigned target transaction result.

In the foregoing transaction data processing method, a transaction data processing request is first received through a client, and a specific channel is allocated for the transaction data processing request according to a transaction type; the channel allocates transaction data to a transaction node

therein, and the transaction node processes the transaction data to obtain a preliminary transaction result, so that the channel can obtain a target transaction result according to the preliminary transaction result of the transaction node participating in the transaction and feed back the target transaction result to the client, without interaction between the transaction nodes, which reduces data interaction, so that the occurrence of confusion of data between different transaction nodes can be avoided, thereby further ensuring the reliability of the transaction result. In one embodiment, the transaction data processing method may further include: allocating, by an organization, one or more transaction nodes to users registered in the organization, each transaction node corresponding to one transaction type; and allocating, by the organization, the transaction nodes to different channels according to the transaction types.

Specifically, when the computer system is initialized, the organization applies for a user name and a related user key and certificate of a user registered in the organization through the CA server and establishes channels, and the organization allocates one or more transaction nodes to the user registered in the organization. Each transaction node corresponds to one transaction type. The organization allocates the transaction nodes to different channels according to the transaction types, and the channel acquires attributes of each transaction node, thereby ensuring that the channel and the transaction type have a one-to-one correspondence. Each organization allocates users of the corresponding transaction type, and each organization allocates one or more transaction nodes to the corresponding channel. Attributes of the transaction node may include basic attributes of the transaction node, an IP address of the transaction node, a port number of the transaction node, an IP address of an event, a port number of an event, an organization to which the transaction node belongs, a user of the organization on the channel, user permissions, a user encryption key and certificate, a certificate form, an encryption and decryption algorithm agreed by the organization and the channel, a countersign and verification algorithm, a key length of each algorithm, and the like. If the response mode of the transaction node is an event mode, the channel establishes an event connection with each transaction node added into the channel. In addition, during initialization, the client establishes a mapping relationship between the transaction type and the channel, and agrees with the outside on an

encryption and decryption algorithm, a countersign and verification algorithm, a key length of each algorithm, and the like.

Specifically, in conjunction with FIG. 1, two users, a first user and a second user respectively, are registered in an organization 1, where the first user corresponds to a fifth transaction node, and the second user corresponds to a second transaction node; the fifth transaction node belongs to a first channel, and the second transaction node belongs to a second channel, so the transaction types processed by the fifth transaction node and the second transaction node are different. In this way, when the client receives the transaction data processing request, the transaction data is first allocated to the corresponding channel, for example, the first channel, according to the transaction type. The first channel allocates the transaction data to all the transaction nodes in the first channel, namely the first transaction node and the fifth transaction node. The first transaction node and the fifth transaction node perform calculation respectively to obtain preliminary transaction results and feed back same to the first channel. After obtaining a target transaction result according to the preliminary transaction results, the first channel feeds back the target transaction result to the client, and then the target transaction result is issued by the client.

Specifically, the organization may also update user information of the transaction type or an expired user certificate, and transmit the updated user information and the updated user certificate to the corresponding channel. For example, the first organization may update user information of a user corresponding to the fifth transaction node, and then notify the first channel. For example, when the changed user information is an encryption and decryption algorithm, the first channel needs to be notified in time, otherwise the transaction may fail.

In the foregoing embodiment, an organization attribute of the organization is introduced in the process of initialization, which improves the visibility of the organization participating in the transaction, thereby improving the scalability of the system. In an organization, generally only one transaction node is involved in calculation, which improves the utilization of resources. A plurality of transaction nodes in an organization participate in calculation of different types of transactions, and the isolation between different computational nodes of the same organization is designed during the

calculation of different types of transactions, which avoids data confusion and improves the visibility of transaction calculation.

In one embodiment, the step of transmitting, by the channel, the transaction data to a transaction node therein may include: acquiring, by the channel, an organization identifier corresponding to the transaction node in the channel, and transmitting the transaction data and the organization identifier to the corresponding transaction node in the channel. Before the step of performing, by the transaction node, processing on the transaction data to obtain a preliminary transaction result, the method may further include: verifying, by the transaction node, the organization identifier, and after the verification of the organization identifier is passed, verifying, by the transaction node, user permissions corresponding to the transaction nodes; and after the verification of the user permissions is passed, continuing to execute the step of the transaction node performing processing on the transaction data to obtain a preliminary transaction result.

Specifically, the step of transmitting, by the channel, the transaction data to a transaction node therein may include: acquiring an organization identifier corresponding to the transaction node in the channel, using the organization identifier and the transaction identifier as a header field, and transmitting the transaction data as a body field to the corresponding transaction node. For example, in FIG. 1, the first channel acquires an organization identifier of the first organization and an organization identifier of the second organization, transmits the organization identifier of the first organization as the header field and the transaction data as the body field to the fifth transaction node, and transmits the organization identifier of the second organization as the header field and the transaction data as the body field to the first transaction node, and thus the first transaction node and the fifth transaction node process the corresponding transaction data to obtain a preliminary transaction result.

Specifically, the first transaction node is taken as an example. When the first transaction node receives the transaction data and the organization identifier transmitted by the first channel, it is determined whether the organization identifier transmitted by the first channel is consistent with the organization identifier stored in the first transaction node, namely whether the organization

identifiers are both organization identifier of the second organization; if the organization identifiers are inconsistent, the transaction is directly rejected, and a preliminary transaction result of transaction failure is returned; and if the organization identifiers are consistent, it is continued to determine whether a current user has transaction execution permissions. If the current user does not have transaction execution permissions, the transaction is directly rejected, and a preliminary transaction result of transaction failure is returned; and if the current user has transaction execution permissions, the first transaction node processes the received transaction data to obtain a preliminary transaction result.

In the foregoing embodiment, attributes of the organization are introduced in the transaction process, which improves the visibility of the organization participation, thereby improving the scalability of the system. For example, a new organization can be added according to the attributes of the organization. In addition, user permissions are set. Therefore, it can be ensured that a user who have no permission cannot access the corresponding transaction data, that is, in one transaction, one user performs a login transaction, and if multiple transaction nodes belong to different organizations, the user can only access one of the transaction nodes. If the user wants to access other transaction nodes, the user needs to pass the corresponding signature verification and the like, thereby ensuring the security of the transaction data. In one embodiment, the transaction nodes are a transaction node cluster, and the transaction nodes in the transaction node cluster share a storage space. The step of performing, by the transaction node, processing on the transaction data to obtain a preliminary transaction result may include: detecting, by the transaction node cluster, whether a transaction node currently participating in a transaction has a fault; if the transaction node currently participating in the transaction has a fault, selecting, by the transaction node cluster, a transaction node that has no fault; and continuing to process, by the selected transaction node that has no fault, according to information stored in the shared storage space by the transaction node that has a fault, the transaction data to obtain a preliminary transaction result.

Specifically, the transaction node may exist in the form of a cluster, and the cluster shares a common storage space. After receiving transaction data sent by the channel, the transaction node

processes the transaction data. If the transaction node currently processing the transaction in this process has a fault, the cluster allocates another transaction node that has no fault for the current transaction to continue to participate in the transaction. For example, an allocation can be made according to a serial number. If the serial number of the transaction node currently participating in the transaction is 5, whether the transaction node with the serial number of 6 has a fault is queried. If the transaction node with the serial number of 6 has no fault, the transaction node with the serial number of 6 is directly allocated for continuing to participate in the transaction. If the transaction node with the serial number of 6 has a fault, it is necessary to continue to traverse a next transaction node until a transaction node that has no fault is acquired to process the current transaction. Since all transaction nodes in the cluster share the same storage space, the selected transaction node that has no fault can continue to process this transaction at a breakpoint without reprocessing this transaction.

In the foregoing embodiment, if the transaction node currently processing the transaction has a fault, a new transaction node is reselected to continue the processing of the transaction. Moreover, since all the transaction nodes in the cluster share the storage space, the new transaction node can continue to process this transaction without repeating the processing process of the last faulted transaction node.

In one embodiment, the step of performing, by the channel, according to the preliminary transaction result, calculation to obtain a target transaction result may include: determining, by the channel, whether the number of preliminary transaction results indicating a transaction success that are fed back by the transaction nodes participating in the transaction is greater than or equal to a preset value; when the number of the preliminary transaction results indicating a transaction success is greater than or equal to the preset value, a target transaction result indicates a transaction success; and when the number of the preliminary transaction results indicating a transaction success is less than the preset value, the target transaction result indicates a transaction failure.

Specifically, after receiving the preliminary transaction results fed back by the transaction nodes, the channel queries the number of transaction results indicating a success, and obtains a target transaction result according to a transaction policy.

For example, a transaction policy is that preliminary transaction results of all the transaction nodes in the channel need to indicate a transaction success and then the target transaction result is a transaction success, and if the preliminary transaction result of one transaction node indicates a failure, the target transaction result is a transaction failure. Under this transaction policy, the channel queries the preliminary transaction results returned by the transaction nodes. When all the preliminary transaction results indicate a transaction success, the target transaction result indicates a transaction success is output. When one preliminary transaction result indicates a transaction failure, the target transaction result indicates a transaction failure is output.

A transaction policy is that preliminary transaction results of at least $n/2+1$ transaction nodes in the channel need to indicate a transaction success and then the target transaction result is a transaction success, and if the preliminary transaction results of $n/2+1$ transaction nodes in the channel that indicate a success do not exist, the target transaction result is a transaction failure, where n is the number of the transaction nodes. Under this transaction policy, the channel queries the preliminary transaction results returned by the transaction nodes; when at least $n/2+1$ preliminary transaction results indicate a transaction success, the target transaction result indicating a transaction success is output; and when at least $n/2+1$ preliminary transaction results indicate a transaction success do not exist, the target transaction result indicating a transaction failure is output.

In the foregoing embodiment, the channel obtains the target transaction result according to the preliminary transaction results returned by the transaction nodes, instead of relying on only one transaction node for processing, or obtains the target transaction result by intercommunication between the transaction nodes. The processing process is simple and the result is reliable.

It should be understood that although various steps in the flowchart of FIG. 2 are sequentially displayed as indicated by the arrows, these steps are not necessarily sequentially executed in the order indicated by the arrows. Unless explicitly stated herein, the execution of these steps is not subjected to strict order limitation, and these steps may be executed in other orders. Moreover, at least some of the steps in FIG. 2 may include a plurality of sub-steps or stages, which are not necessarily performed at the same time, but may be executed at different times. These sub-steps or stages are also not

necessarily sequentially executed, but may be performed alternately with at least some of other steps or sub-steps or stages of other steps.

Still referring to FIG. 1, a client, channels, and transaction nodes in FIG. 1 form a blockchain transaction data processing system.

5 The client is configured to receive a transaction data processing request, where the transaction data processing request carries a transaction type and transaction data; and the client transmits, according to the transaction type, the transaction data to a corresponding channel.

The channel is configured to transmit the transaction data to a transaction node therein.

10 The transaction node is configured to perform processing on the transaction data to obtain a preliminary transaction result, and feed back the preliminary transaction result to the channel.

The channel is further configured to perform, according to the preliminary transaction result, calculation to obtain a target transaction result, and return the target transaction result to the client.

The client is further configured to issue the target transaction result.

15 In one embodiment, the system further includes an organization. The organization is configured to allocate one or more transaction nodes to users registered in the organization, each transaction node corresponding to one transaction type; and the organization allocates the transaction nodes to different channels according to the transaction types.

In one embodiment, the organization sets corresponding transaction permissions for users registered in the organization.

20 The channel is further configured to acquire an organization identifier corresponding to the transaction node in the channel, and transmit the transaction data and the organization identifier to the corresponding transaction node in the channel.

The transaction node is further configured to verify the organization identifier, and after the verification of the organization identifier is passed, the transaction node verifies user permissions

corresponding to the transaction nodes; and after the verification of the user permissions is passed, the transaction node continues to perform processing on the transaction data to obtain a preliminary transaction result.

In one embodiment, the transaction nodes are a transaction node cluster, and the transaction
5 nodes in the transaction node cluster share a storage space.

The transaction node cluster is configured to detect whether a transaction node currently participating in a transaction has a fault; and if the transaction node currently participating in the transaction has a fault, the transaction node cluster selects a transaction node that has no fault. The selected transaction node that has no fault continues to process, according to information stored in the
10 shared storage space by the transaction node that has a fault, the transaction data to obtain a preliminary transaction result.

In one embodiment, the channel is further configured to determine whether the number of preliminary transaction results indicating a transaction success that are fed back by the transaction nodes participating in the transaction is greater than or equal to a preset value; when the number of the preliminary transaction results indicating a transaction success is greater than or equal to the preset
15 value, a target transaction result indicates a transaction success; and when the number of the preliminary transaction results indicating a transaction success is less than the preset value, the target transaction result indicates a transaction failure.

For specific definitions of the blockchain transaction data processing system, reference may be
20 made to the foregoing definition of the transaction data processing method, and details are not described herein again. Various modules in the blockchain transaction data processing system described above may be implemented fully or partially through software, hardware, and a combination thereof. Each of the foregoing modules may be embedded in or independent of a processor in a computer device, or may be stored in the memory in the computer device in the form of
25 software, so that the processor invokes operations corresponding to the above modules.

In one embodiment, a computer system is provided. The internal structure of the computer

system may be as shown in FIG. 3. The computer system includes a client, channels, and transaction nodes, where the client includes a first processor, a first memory, a first network interface and a first database which are connected by a system bus. The first processor is configured to provide computing and control capabilities. The first memory includes a non-volatile computer readable instruction storage medium and an internal memory. The non-volatile computer readable instruction storage medium stores an operating system, computer readable instructions, and a database. The internal memory provides an environment for the operation of the operating system and the computer readable instructions in the non-volatile computer readable instruction storage medium. The first network interface is configured to communicate with an external terminal through a network. The channel includes a second processor, a second memory, a second network interface and a second database which are connected by using a system bus. The second processor is configured to provide computing and control capabilities. The second memory includes a non-volatile computer readable instruction storage medium and an internal memory. The non-volatile computer readable instruction storage medium stores an operating system, computer readable instructions, and a database. The internal memory provides an environment for the operation of the operating system and the computer readable instructions in the non-volatile computer readable instruction storage medium. The second network interface is configured to communicate with an external terminal through a network. The transaction node includes a third processor, a third memory, a third network interface and a third database which are connected by using a system bus. The third processor is configured to provide computing and control capabilities. The third memory includes a non-volatile computer readable instruction storage medium and an internal memory. The non-volatile computer readable instruction storage medium stores an operating system, computer readable instructions, and a database. The internal memory provides an environment for the operation of the operating system and the computer readable instructions in the non-volatile computer readable instruction storage medium. The third network interface is configured to communicate with an external terminal through a network. When the computer readable instructions are executed by the first processor, the second processor, and the third processor, a blockchain transaction data processing method is implemented.

A person of ordinary skill in the art can understand that the structure shown in FIG. 3 is merely a block diagram of a partial structure related to the solution of this application, and does not constitute a limitation on the computer system to which the solution of this application is applied. A particular computer system may include more or fewer components than those shown in the drawing, or
5 combine some components, or have different component arrangements.

A computer device includes a memory and one or more processors. The memory stores a computer readable instruction therein. When the computer readable instruction is executed by the processor, the one or more processors execute the following steps: when a first processor executes the computer readable instruction, implementing the following steps: receiving a transaction data
10 processing request, the transaction data processing request carrying a transaction type and transaction data; transmitting, according to the transaction type, the transaction data to a corresponding channel, and issuing a target transaction result; when a second processor executes the computer readable instruction, implementing the following steps: transmitting the transaction data to a transaction node in the channel, performing, according to a preliminary transaction result, calculation to obtain a target
15 transaction result, and returning the target transaction result to the client; when a third processor executes the computer readable instruction, implementing the following steps: performing processing on the transaction data to obtain a preliminary transaction result, and feeding back the preliminary transaction result to the channel.

In one embodiment, the computer system further includes an organization, and the organization
20 includes a fourth processor, a fourth memory, a fourth network interface and a fourth database which are connected by using a system bus. The fourth processor is configured to provide computing and control capabilities. The fourth memory includes a non-volatile computer readable instruction storage medium and an internal memory. The non-volatile computer readable instruction storage medium stores an operating system, computer readable instructions, and a database. The internal memory
25 provides an environment for the operation of the operating system and the computer readable instructions in the non-volatile computer readable instruction storage medium. The fourth network interface is configured to communicate with an external terminal through a network.

The organization includes a fourth memory and one or more fourth processors. The memory stores a computer readable instruction therein. When the computer readable instruction is executed by the fourth processor, the one or more fourth processors execute the following steps: allocating one or more transaction nodes to users registered in the organization, each transaction node corresponding to one transaction type; and allocating the transaction nodes to different channels according to the transaction types.

In one embodiment, the transmitting the transaction data to a transaction node in the channel implemented when the second processor executes the computer readable instruction includes: acquiring an organization identifier corresponding to the transaction node in the channel, and transmitting the transaction data and the organization identifier to the corresponding transaction node in the channel. Before the performing processing on the transaction data to obtain a preliminary transaction result implemented when the third processor executes the computer readable instruction, the method further includes: verifying the organization identifier, and after the verification of the organization identifier is passed, verifying, by the transaction node, user permissions corresponding to the transaction nodes; and after the verification of the user permissions is passed, continuing to perform processing on the transaction data to obtain a preliminary transaction result.

In one embodiment, the transaction nodes are a transaction node cluster, and the transaction nodes in the transaction node cluster share a storage space. The performing processing on the transaction data to obtain a preliminary transaction result implemented when the third processor executes the computer readable instruction includes: detecting whether a transaction node currently participating in a transaction has a fault; if the transaction node currently participating in the transaction has a fault, selecting a transaction node that has no fault; and continuing to perform, according to information stored in the shared storage space by the transaction node that has a fault, processing on the transaction data to obtain a preliminary transaction result through the selected transaction node that has no fault.

In one embodiment, the performing, according to the preliminary transaction result, calculation to obtain a target transaction result implemented when the second processor executes the computer

readable instruction may include: determining whether the number of preliminary transaction results indicating a transaction success that are fed back by the transaction nodes participating in the transaction is greater than or equal to a preset value; when the number of the preliminary transaction results indicating a transaction success is greater than or equal to the preset value, a target transaction result indicating a transaction success; and when the number of the preliminary transaction results indicating a transaction success is less than the preset value, the target transaction result indicating a transaction failure.

One or more non-volatile computer readable storage media storing a computer readable instruction are provided. When the computer readable instruction is executed by one or more first processors, the one or more first processors execute the following steps: receiving a transaction data processing request, the transaction data processing request carrying a transaction type and transaction data; transmitting, according to the transaction type, the transaction data to a corresponding channel; and issuing a target transaction result. When the computer readable instruction is executed by one or more second processors, the one or more second processors execute the following steps: transmitting the transaction data to a transaction node in the channel; performing, according to a preliminary transaction result, calculation to obtain a target transaction result; and returning the target transaction result to the client. When the computer readable instruction is executed by one or more third processors, the one or more third processors execute the following steps: performing processing on the transaction data to obtain a preliminary transaction result, and feeding back the preliminary transaction result to the channel.

In one embodiment, when the computer readable instruction is executed by one or more fourth processors, the one or more fourth processors execute the following steps: allocating one or more transaction nodes to users registered in the organization, each transaction node corresponding to one transaction type; and allocating the transaction nodes to different channels according to the transaction types.

In one embodiment, the transmitting the transaction data to a transaction node in the channel implemented when the computer readable instruction is executed by one or more of the second

processors includes: acquiring an organization identifier corresponding to the transaction node in the channel; and transmitting the transaction data and the organization identifier to the corresponding transaction node in the channel. Before the performing processing on the transaction data to obtain a preliminary transaction result implemented when the computer readable instruction is executed by one or more of the third processors, the method further includes: verifying the organization identifier, and after the verification of the organization identifier is passed, verifying user permissions corresponding to the transaction nodes; and after the verification of the user permissions is passed, continuing to perform the transaction node performing processing on the transaction data to obtain a preliminary transaction result.

In one embodiment, the transaction nodes are a transaction node cluster, and the transaction nodes in the transaction node cluster share a storage space. The performing, by the transaction node, processing on the transaction data to obtain a preliminary transaction result implemented when the computer readable instruction is executed by one or more of the third processors includes: detecting whether a transaction node currently participating in a transaction has a fault; if the transaction node currently participating in the transaction has a fault, selecting, by the transaction node cluster, a transaction node that has no fault; and continuing to perform, according to information stored in the shared storage space by the transaction node that has a fault, processing on the transaction data to obtain a preliminary transaction result through the selected transaction node that has no fault.

In one embodiment, the performing, according to the preliminary transaction result, calculation to obtain a target transaction result implemented when the computer readable instruction is executed by one or more second processors include: determining whether the number of preliminary transaction results indicating a transaction success that are fed back by the transaction nodes participating in the transaction is greater than or equal to a preset value; when the number of the preliminary transaction results indicating a transaction success is greater than or equal to the preset value, a target transaction result indicates a transaction success; and when the number of the preliminary transaction results indicating a transaction success is less than the preset value, the target transaction result indicates a transaction failure.

A person of ordinary skill in the art can understand that all or part of processes for implementing the methods of the foregoing embodiments may be implemented through hardware related to computer readable instructions. The computer readable instructions may be stored in a non-volatile computer readable storage medium. When the computer readable instructions are executed, processes of the methods of the embodiments described above may be included. Any reference to a memory, storage, a database, or other media used in the various embodiments provided by this application may include a non-volatile memory and/or volatile memory. The non-volatile memory may include a Read-Only Memory (ROM), a Programmable ROM (PROM), an Electrically Programmable ROM (EPROM), an Electrically Erasable Programmable ROM (EEPROM) or a flash memory. The volatile memory may include an RAM) or an external cache memory. As an illustration and not a limitation, the RAM is available in a variety of formats, such as a Static RAM (SRAM), a Dynamic RAM (DRAM), a Synchronous DRAM (SDRAM), a Double Data Rate SDRAM (DDRSDRAM), an Enhanced SDRAM (ESDRAM), a Synchlink DRAM (SLDRAM), a Rambus Direct RAM (RDRAM), a Direct Rambus Dynamic RAM (DRDRAM), and a Rambus RAM (RDRAM).

The technical features of the foregoing embodiments may be arbitrarily combined. To make the description concise, all possible combinations of the technical features in the foregoing embodiments are not described. However, as long as there is no contradiction between the combinations of these technical features, all the combinations should fall within the scope of the specification.

The foregoing embodiments are merely illustrative of several embodiments of this application, and the description thereof is more specific and detailed, but should not be construed as limiting the invention patent scope. It should be noted that a person of ordinary skill in the art can further make several variations and improvements without departing from the idea of this application. These all fall within the protection scope of this application. Therefore, the patent protection scope of this application should be subject to the appended claims.

CLAIMS

1. A transaction data processing method, comprising:

receiving, by a client, a transaction data processing request, the transaction data processing request carrying a transaction type and transaction data; transmitting, by the client, according to the transaction type, the transaction data to a corresponding channel, wherein the client generates a transaction identifier according to the transaction type, selects the corresponding channel according to the transaction type, and transmits the transaction identifier and transaction data after security processing to the matching channel;

transmitting, by the channel, the transaction data to a transaction node therein;

performing, the transaction node, processing on the transaction data to obtain a preliminary transaction result, and feeding back the preliminary transaction result to the channel;

performing, by the channel, according to the preliminary transaction result, calculation to obtain a target transaction result, and returning the target transaction result to the client; and

issuing, by the client, the target transaction result,

wherein the client is independent of the transaction node and provides a transaction data processing request for outside services in a load-balancing mode.

2. The method according to claim 1, further comprising: allocating, by an organization, one or more transaction nodes to users registered in the organization, each transaction node corresponding to one transaction type; and

allocating, by the organization, the transaction nodes to different channels according to the transaction types.

3. The method according to claim 1, wherein the transmitting, by the channel, the transaction data to a transaction node therein comprises:

acquiring, by the channel, an organization identifier corresponding to the transaction node in the

channel, and transmitting the transaction data and the organization identifier to the corresponding transaction node in the channel;

before the performing, by the transaction node, processing on the transaction data to obtain a preliminary transaction result, the method further comprises:

5 verifying, by the transaction node, the organization identifier, and after the verification of the organization identifier is passed, verifying, by the transaction node, user permissions corresponding to the transaction nodes; and after the verification of the user permissions is passed, continuing to execute the transaction node performing processing on the transaction data to obtain a preliminary transaction result.

10 4. The method according to claim 1, wherein the transaction nodes are a transaction node cluster, and the transaction nodes in the transaction node cluster share a storage space;

the performing, by the transaction node, processing on the transaction data to obtain a preliminary transaction result comprises:

15 detecting, by the transaction node cluster, whether a transaction node currently participating in a transaction has a fault;

if the transaction node currently participating in the transaction has a fault, selecting, by the transaction node cluster, a transaction node that has no fault; and

20 continuing to process, by the selected transaction node that has no fault, according to information stored in the shared storage space by the transaction node that has a fault, the transaction data to obtain a preliminary transaction result.

5. The method according to claim 1, wherein the performing, by the channel, according to the preliminary transaction result, calculation to obtain a target transaction result comprises:

25 determining, by the channel, whether the number of preliminary transaction results indicating a transaction success that are fed back by the transaction nodes participating in the transaction is greater than or equal to a preset value; wherein

when the number of the preliminary transaction results indicating a transaction success is greater than or equal to the preset value, a target transaction result indicates a transaction success; and

when the number of the preliminary transaction results indicating a transaction success is less than the preset value, the target transaction result indicates a transaction failure.

5 6. A blockchain transaction data processing system, comprising a client, channels, and transaction nodes;

wherein the client is configured to receive a transaction data processing request, the transaction data processing request carrying a transaction type and transaction data; the client transmits, according to the transaction type, the transaction data to a corresponding channel, wherein the client
10 generates a transaction identifier according to the transaction type, selects the corresponding channel according to the transaction type, and transmits the transaction identifier and transaction data after security processing to the matching channel;

the channel is configured to transmit the transaction data to a transaction node therein;

the transaction node is configured to perform processing on the transaction data to obtain a
15 preliminary transaction result, and feed back the preliminary transaction result to the channel;

the channel is further configured to perform, according to the preliminary transaction result, calculation to obtain a target transaction result, and return the target transaction result to the client; and

the client is further configured to issue the target transaction result,

20 wherein the client is independent of the transaction node and provides a transaction data processing request for outside services in a load-balancing mode.

7. The system according to claim 6, further comprising an organization, wherein the organization is configured to allocate one or more transaction nodes to users registered in the organization, each transaction node corresponding to one transaction type; and the organization
25 allocates the transaction nodes to different channels according to the transaction types.

8. The system according to claim 6, wherein the organization sets corresponding transaction permissions for users registered in the organization;

the channel is further configured to acquire an organization identifier corresponding to the transaction node in the channel, and transmit the transaction data and the organization identifier to the
5 corresponding transaction node in the channel; and

the transaction node is further configured to verify the organization identifier, and after the verification of the organization identifier is passed, the transaction node verifies user permissions corresponding to the transaction nodes; and after the verification of the user permissions is passed, the transaction node continues to perform processing on the transaction data to obtain a preliminary
10 transaction result.

9. The system according to claim 6, wherein the transaction nodes are a transaction node cluster, and the transaction nodes in the transaction node cluster share a storage space;

the transaction node cluster is configured to detect whether a transaction node currently participating in a transaction has a fault;

15 if the transaction node currently participating in the transaction has a fault, the transaction node cluster selects a transaction node that has no fault; and

the selected transaction node that has no fault is configured to continue to process, according to information stored in the shared storage space by the transaction node that has a fault, the transaction data to obtain a preliminary transaction result.

20 10. The system according to claim 6, wherein the channel is further configured to determine whether the number of preliminary transaction results indicating a transaction success that are fed back by the transaction nodes participating in the transaction is greater than or equal to a preset value; wherein

when the number of the preliminary transaction results indicating a transaction success is greater
25 than or equal to the preset value, a target transaction result indicates a transaction success; and

when the number of the preliminary transaction results indicating a transaction success is less than the preset value, the target transaction result indicates a transaction failure.

11. A computer system, comprising a client, channels, and transaction nodes;

wherein the client comprises a first memory and one or more first processors; the memory stores
5 a computer readable instruction therein; when the computer readable instruction is executed by the first processor, the one or more first processors execute the following steps: receiving a transaction data processing request, the transaction data processing request carrying a transaction type and transaction data; transmitting, according to the transaction type, the transaction data to a
10 corresponding channel, wherein the client generates a transaction identifier according to the transaction type, selects the corresponding channel according to the transaction type, and transmits the transaction identifier and transaction data after security processing to the matching channel; and issuing a target transaction result;

the channel comprises a second memory and one or more second processors; the memory stores
15 a computer readable instruction therein; when the computer readable instruction is executed by the second processor, the one or more second processors execute the following steps: transmitting the transaction data to a transaction node in the channel; performing, according to a preliminary transaction result; calculation to obtain a target transaction result; and returning the target transaction result to the client, wherein the client is independent of the transaction node and provides a transaction data processing request for outside services in a load-balancing mode;

20 the transaction node comprises a third memory and one or more third processors; the memory stores a computer readable instruction therein; when the computer readable instruction is executed by the third processor, the one or more third processors execute the following steps: performing processing on the transaction data to obtain a preliminary transaction result; and feeding back the preliminary transaction result to the channel.

25 12. The computer system according to claim 11, further comprising an organization, wherein the organization comprises a fourth memory and one or more fourth processors; the memory stores a

computer readable instruction therein, and when the computer readable instruction is executed by the fourth processor, the one or more fourth processors execute the following steps:

allocating one or more transaction nodes to users registered in the organization, each transaction node corresponding to one transaction type; and

5 allocating the transaction nodes to different channels according to the transaction types.

13. The computer system according to claim 11, wherein the transmitting the transaction data to a transaction node in the channel implemented when the second processor executes the computer readable instruction comprises:

acquiring an organization identifier corresponding to the transaction node in the channel, and
10 transmitting the transaction data and the organization identifier to the corresponding transaction node in the channel;

before the performing processing on the transaction data to obtain a preliminary transaction result implemented when the third processor executes the computer readable instruction, the method further comprises:

15 verifying the organization identifier, and after the verification of the organization identifier is passed, verifying user permissions corresponding to the transaction nodes; and after the verification of the user permissions is passed, continuing to execute the transaction node performing processing on the transaction data to obtain a preliminary transaction result.

14. The computer system according to claim 11, wherein the transaction nodes are a
20 transaction node cluster, and the transaction nodes in the transaction node cluster share a storage space;

the performing processing on the transaction data to obtain a preliminary transaction result implemented when the third processor executes the computer readable instruction comprises:

detecting whether a transaction node currently participating in a transaction has a fault;

if the transaction node currently participating in the transaction has a fault, selecting, by the transaction node cluster, a transaction node that has no fault; and

continuing to perform, according to information stored in the shared storage space by the transaction node that has a fault, processing on the transaction data to obtain a preliminary transaction
5 result through the selected transaction node that has no fault.

15. The computer system according to claim 10, wherein the performing, according to the preliminary transaction result, calculation to obtain a target transaction result implemented when the second processor executes the computer readable instruction comprises:

determining whether the number of preliminary transaction results indicating a transaction
10 success that are fed back by the transaction nodes participating in the transaction is greater than or equal to a preset value; wherein

when the number of the preliminary transaction results indicating a transaction success is greater than or equal to the preset value, a target transaction result indicates a transaction success; and

when the number of the preliminary transaction results indicating a transaction success is less
15 than the preset value, the target transaction result indicates a transaction failure.

16. One or more non-volatile computer readable storage media storing a computer readable instruction, wherein when the computer readable instruction is executed by one or more first processors, the one or more first processors execute the following steps: receiving a transaction data processing request, the transaction data processing request carrying a transaction type and transaction
20 data; transmitting, according to the transaction type, the transaction data to a corresponding channel, wherein the client generates a transaction identifier according to the transaction type, selects the corresponding channel according to the transaction type, and transmits the transaction identifier and transaction data after security processing to the matching channel; and issuing a target transaction result;

25 when the computer readable instruction is executed by one or more second processors, the one or

more second processors execute the following steps: transmitting the transaction data to a transaction node in the channel; performing, according to a preliminary transaction result; calculation to obtain a target transaction result; and returning the target transaction result to the client, wherein the client is independent of the transaction node and provides a transaction data processing request for outside
5 services in a load-balancing mode;

when the computer readable instruction is executed by one or more third processors, the one or more third processors execute the following steps: performing processing on the transaction data to obtain a preliminary transaction result; and feeding back the preliminary transaction result to the channel.

10 17. The storage medium according to claim 16, wherein when the computer readable instruction is executed by one or more fourth processors, the one or more fourth processors execute the following steps:

allocating one or more transaction nodes to users registered in the organization, each transaction node corresponding to one transaction type; and

15 allocating the transaction nodes to different channels according to the transaction types.

18. The storage medium according to claim 16, wherein the transmitting the transaction data to a transaction node in the channel executed by one or more second processors when the computer readable instruction is executed by the one or more second processors comprises:

20 acquiring an organization identifier corresponding to the transaction node in the channel, and transmitting the transaction data and the organization identifier to the corresponding transaction node in the channel;

before the performing processing on the transaction data to obtain a preliminary transaction result executed by one or more third processors when the computer readable instruction is executed by the one or more third processors, the method further comprises:

25 verifying the organization identifier, and after the verification of the organization identifier is

passed, verifying user permissions corresponding to the transaction nodes; and after the verification of the user permissions is passed, continuing to execute the transaction node performing processing on the transaction data to obtain a preliminary transaction result.

19. The storage medium according to claim 16, wherein the transaction nodes are a transaction
5 node cluster, and the transaction nodes in the transaction node cluster share a storage space;

the performing processing on the transaction data to obtain a preliminary transaction result executed by one or more third processors when the computer readable instruction is executed by the one or more third processors comprises:

detecting whether a transaction node currently participating in a transaction has a fault;

10 if the transaction node currently participating in the transaction has a fault, selecting, by the transaction node cluster, a transaction node that has no fault; and

continuing to perform, according to information stored in the shared storage space by the transaction node that has a fault, processing on the transaction data to obtain a preliminary transaction result through the selected transaction node that has no fault.

15 20. The storage medium according to claim 16, wherein the performing, according to the preliminary transaction result, calculation to obtain a target transaction result executed by one or more second processors when the computer readable instruction is executed by the one or more second processors comprises:

20 determining whether the number of preliminary transaction results indicating a transaction success that are fed back by the transaction nodes participating in the transaction is greater than or equal to a preset value; wherein

when the number of the preliminary transaction results indicating a transaction success is greater than or equal to the preset value, a target transaction result indicates a transaction success; and

25 when the number of the preliminary transaction results indicating a transaction success is less than the preset value, the target transaction result indicates a transaction failure.