NODE RESTART SYSTEM FOR CONSORTIUM BLOCKCHAIN, METHOD, AND COMPUTER SYSTEM

TECHNICAL FIELD

[0001] This specification relates to the computer field, and in particular, to a node restart system for a consortium blockchain, a method, and a computer system.

BACKGROUND

[0002] With the development of computer technologies, blockchains are widely applied to various scenarios that require high credibility because of tamper-resistance and decentralization features of the blockchains. A consortium blockchain is a special blockchain, is usually jointly managed only by several participants, and is open only to members of a specific group and a limited quantity of third participants.

[0003] In the conventional technology, nodes in the consortium blockchain usually have different identities. Therefore, a user needs to select a specific node to submit a request, and the specific node handles the request submitted by the user. However, when the specific node is abnormal, some requests submitted by the user cannot be handled because another node cannot process a request received by the specific node. For example, in the cross-border payment field, a transaction application of a user is usually processed only by a node in a region in which the user is located, and a node in another region cannot process the application sent by the user in the region. When the node in the region is abnormal, none of applications sent by the user in the region can be processed, and the consortium blockchain does not have high availability. Therefore, how to ensure the high availability of the consortium blockchain becomes an urgent problem to be resolved.

[0004] This specification provides a node restart system for a consortium blockchain, a method, and a computer system.

SUMMARY

[0005] This specification provides a node restart system for a consortium blockchain, a method, and a computer system, to partially resolve the above-mentioned problem in the conventional technology.

[0006] This specification provides a node restart system for a consortium blockchain. The system includes a target node and a trusted storage node and a backup node that correspond to the target node. The target node is configured to: execute a service based on configured configuration data; upload a correspondence between an identifier of the target node and a communication protocol address of the target node to a smart contract in the consortium blockchain; and determine private data of the target node based on a preset update cycle, and update, based on the private data, private data that is of the target node and that is stored in the trusted storage node, where the private data includes data that is of the target node and on which consensus reaching is to be performed. The trusted storage node is configured to store the private data updated by the target node. The backup node is configured to: when detecting that the target node goes down, determine the identifier of the target node, obtain the configuration data, and configure the backup node based on the configuration data; obtain the private data of the target node from the trusted storage node based on the identifier of the target node; and update the correspondence in the smart contract to a communication protocol address of the backup node, to execute the service in place of the target node.

[0007] This specification provides a method performed by a target node in a consortium blockchain, including: executing a service based on configured configuration data; determining a correspondence between an identifier of the target node and a communication protocol address of the target node, and uploading the correspondence to a smart contract in the consortium blockchain for storage; and determining private data of the target node based on a preset update cycle, and updating, based on the private data, private data that is of the target node and that is stored in a trusted storage node, where the private data includes data that is of the target node and on which consensus reaching is to be performed.

[0008] This specification provides a method performed by a backup node in a consortium blockchain, including: when detecting that a target node corresponding to the backup node goes down, determining an identifier of the target node, obtaining configuration data, and configuring

the backup node based on the configuration data; obtaining private data of the target node from a trusted storage node based on the identifier of the target node, where the private data includes data that is of the target node and on which consensus reaching is to be performed; and updating, based on a communication protocol address of the backup node, a correspondence stored in a smart contract in the consortium blockchain, and executing a service in place of the target node, where the correspondence stored in the smart contract is a correspondence that is between the identifier of the target node and a communication protocol address of the target node and that is prestored in the smart contract by the target node.

[0009] This specification provides a computer system. The computer system is configured to implement a target node in a consortium blockchain, and includes a memory, a processor, and a computer program stored in the memory and capable of running on the processor. When the processor executes the program, the following steps are implemented: executing a service based on configured configuration data; determining a correspondence between an identifier of the target node and a communication protocol address of the target node, and uploading the correspondence to a smart contract in the consortium blockchain for storage; and determining private data of the target node based on a preset update cycle, and updating, based on the private data, private data that is of the target node and that is stored in a trusted storage node, where the private data includes data that is of the target node and on which consensus reaching is to be performed.

[0010] This specification provides a computer system. The computer system is configured to implement a backup node in a consortium blockchain, and includes a memory, a processor, and a computer program stored in the memory and capable of running on the processor. When the processor executes the program, the following steps are implemented: when detecting that a target node corresponding to the backup node goes down, determining an identifier of the target node, obtaining configuration data, and configuring the backup node based on the configuration data; obtaining private data of the target node from a trusted storage node based on the identifier of the target node, where the private data includes data that is of the target node and on which consensus reaching is to be performed; and updating, based on a communication protocol address of the backup node, a correspondence stored in a smart contract in the consortium blockchain, and executing a service in place of the target node, where the correspondence stored in the smart contract is a correspondence that is between the identifier of the target node and a communication protocol address of the target node and that is prestored in the smart contract by the target node.

[0011] This specification provides a computer-readable nonvolatile storage medium. The storage medium stores a computer program, and when the computer program is executed by a processor, the above-mentioned method is implemented.

[0012] The above-mentioned at least one technical solution used in this specification can achieve the following beneficial effects:

[0013] In the node restart system for a consortium blockchain described in this specification, the system includes a target node and a trusted storage node and a backup node that correspond to the target node. The target node is configured to back up data stored in the target node. The trusted storage node is configured to store data sent by the target node. The backup node is configured to: monitor a running status of the target node in real time, and when the target node goes down, obtain the data of the target node from the trusted storage node, to execute a corresponding service in place of the target node.

[0014] It can be seen from the above-mentioned system that data stored in each node is classified based on whether the data is common, common data of each node is pre-backed up by using a cloud server, and then non-common data of each node is updated in a trusted storage node of each node based on a preset cycle, so that when detecting that the target node goes down, a backup node can obtain corresponding data from the cloud server and the trusted storage node, and execute a service in place of the node that goes down. This ensures high availability of the consortium blockchain, reduces a data amount of backup data, and saves storage costs.

BRIEF DESCRIPTION OF DRAWINGS

[0015] The accompanying drawings described herein are used to provide a further understanding of this specification and constitute a part of this specification. The example embodiments of this specification and descriptions thereof are used to explain this specification, and do not constitute an improper limitation on this specification. In the accompanying drawings:

[0016] FIG. 1 is a schematic flowchart of a node restart system for a consortium blockchain according to this specification;

[0017] FIG. 2 is a schematic diagram of a node restart procedure for a consortium blockchain according to this specification;

[0018] FIG. 3 is a schematic flowchart of a node restart method for a consortium blockchain

according to this specification;

[0019] FIG. 4 is a schematic flowchart of a node restart method for a consortium blockchain according to this specification;

[0020] FIG. 5 is a schematic diagram of a consortium blockchain according to this specification;

[0021] FIG. 6 is a schematic diagram of a node restart procedure for a consortium blockchain according to this specification; and

[0022] FIG. 7 is a schematic diagram of a computer system corresponding to FIG. 2 and FIG. 3 according to this specification.

DESCRIPTION OF EMBODIMENTS

[0023] To make the objectives, technical solutions, and advantages of this specification clearer, the following clearly and comprehensively describes the technical solutions of this specification with reference to specific embodiments and corresponding accompanying drawings of this specification. Clearly, the described embodiments are merely some but not all of embodiments of this specification. All other embodiments obtained by a person of ordinary skill in the art based on the embodiments of this specification without creative efforts shall fall within the protection scope of the present invention.

[0024] In the node restart system for a consortium blockchain described in this specification, the system includes a plurality of target nodes, and further includes a trusted storage node and a backup node that correspond to each target node. The target node is configured to back up data stored in the target node. The trusted storage node is configured to store data sent by the target node. The backup node is configured to: monitor a running status of the target node in real time, and when the target node goes down, obtain the data of the target node from the trusted storage node, to execute a corresponding service in place of the target node.

[0025] The technical solutions provided in the embodiments of this specification are described in detail below with reference to the accompanying drawings.

[0026] FIG. 1 is a schematic flowchart of a node restart system for a consortium blockchain according to this specification. It should be noted that a smart contract in FIG. 1 is a device in which a smart contract in the consortium blockchain is located, and the smart contract is not a node

in the node restart system described in this specification. To facilitate detailed description of a system procedure, the smart contract is added to the schematic flowchart of the node restart system. Details are as follows:

[0027] S100: A target node executes a service based on configured configuration data.

[0028] In one or more embodiments of this specification, a specific device used to implement each node in the consortium blockchain is not limited, for example, a personal computer, a mobile terminal, and a server. However, each node is configured to execute a service in the consortium blockchain, and to ensure security of the consortium blockchain, is usually implemented by a server. Therefore, in this specification, descriptions are subsequently provided by using an example in which each node in the consortium blockchain is implemented by a server. The server can be a separate device or includes a plurality of devices, for example, a distributed server. This is not limited in this specification.

[0029] It should be noted that each node in the consortium blockchain can be the target node described in this specification. Therefore, for each target node, configuration data used to configure the target node can be directly obtained from a preset cloud server when the target node is configured. Certainly, the configuration data is common configuration data of each node in the consortium blockchain, for example, running environment data, consensus module code logic, communication module code logic, and other common data and code of each node, and can further include common data such as permission data of the node, a startup script, and a whitelist of nodes that can participate in the consortium blockchain. Examples are not listed one by one herein.

[0030] The target node executes the corresponding service based on the configured configuration data. Because configuration data of each node is common, the configuration data can be backed up in the cloud server. When the node goes down or a new node needs to be launched, the configuration data can be obtained by accessing the cloud server. Alternatively, a standardized image container can be made. When the node needs to be restarted or a new node needs to be launched, the image is directly deployed to a container instance.

[0031] A specific service executed by each target node is not limited in this specification, that is, any node that executes any service can be restarted by using the system provided in this specification.

[0032] S102: The target node uploads a correspondence between an identifier of the target node and a communication protocol address of the target node to a smart contract in the consortium

blockchain.

[0033] To communicate with another node in the consortium blockchain and execute the corresponding service, the target node needs to notify the another node of the communication protocol address of the target node. Therefore, the target node uploads the correspondence between the identifier of the target node and the communication protocol address of the target node to the smart contract in the consortium blockchain, and sends an announcement to the another node by using the smart contract.

[0034] Certainly, step S102 should be completed before each node in the consortium blockchain executes a corresponding service.

[0035] S104: The target node determines private data of the target node based on a preset update cycle, and updates, based on the private data, private data that is of the target node and that is stored in a trusted storage node, where the private data includes data that is of the target node and on which consensus reaching is to be performed.

[0036] S106: The trusted storage node stores the private data updated by the target node.

[0037] When the consortium blockchain starts to run, transaction data stored in the nodes in the consortium blockchain, received transaction requests, and other data are different. The data is data on which each node in the consortium blockchain does not perform consensus reaching with another node, or is data that is not processed. The data should be confidential and invisible to the another node, that is, is private data. Therefore, a trusted storage node should be selected to back up the data.

[0038] It should be noted that the trusted storage node can be a device that can communicate with another node, or can be a cloud server. This is not limited in this specification. Certainly, the trusted storage node can be a storage device or a cloud server corresponding to the target node, and is only configured to store the private data of the target node, or can be a storage device or a cloud server shared by some nodes in the consortium blockchain, or can be a storage device or a cloud service shared by nodes configured by a participant in the consortium blockchain. This is not limited in this specification.

[0039] Further, content of private data of the target node at different times is different. Therefore, when data of each node is backed up, the data that is backed up should be updated based on the preset update cycle, to ensure that when the target node goes down, no excessive data losses are caused when the target node is restarted.

[0040] Specifically, the preset cycle for updating the backup can be determined based on a speed of processing the transaction request by the target node and a speed of receiving the transaction request. Certainly, the preset cycle can vary with a time point. In one or more embodiments of this specification, how to specifically determine the preset cycle is not limited, provided that it is ensured that there are as few data losses as possible during node restart.

[0041] Then, when the preset cycle is reached, the target node can delete all data stored in the trusted storage node of the target node, and then store private data determined at a current moment in the trusted storage node.

[0042] S108: When detecting that the target node goes down, a backup node obtains the configuration data of the target node.

[0043] S110: The backup node configures the backup node by using the obtained configuration data, to execute the service.

[0044] S112: The backup node updates an address correspondence to an address of the backup node.

[0045] S114: The backup node executes the service in place of the target node based on the data stored in the storage node.

[0046] To implement fast node restart when the target node goes down, and reduce a loss caused because the node goes down, the backup node of the target node can be preset to monitor a running status of each node. It should be noted that to ensure security and privacy of the consortium blockchain, the backup node and the target node should belong to a same participant, or the backup node is a trusted node of the target node. Certainly, whether the backup node is a node in the consortium blockchain is not limited in this specification, and can be selected based on an actual requirement. Certainly, the backup node can alternatively be a standby server room of each node in the consortium blockchain, or is located on a same server as each node in the consortium blockchain, and is a space resource of the server. Further, a cloud computing device can be used as the backup node of the target node.

[0047] It should be noted that in one or more embodiments of this specification, a method, for example, a heartbeat detection mechanism, used by the backup node to monitor a running status of the target node is not limited. When the backup node detects that the target node goes down, the backup node should start to prepare for obtaining the common configuration data of each node.

[0048] When detecting that the target node goes down, the backup node can first obtain the

common configuration data of each node, then determine the trusted storage node of the target node based on the identifier of the target node, and obtain the private data of the target node from the trusted storage node of the target node. When it is determined that the configuration data and the private data of the target node are loaded into the backup node and the service can be normally executed, a request is sent to the smart contract, and the correspondence that is between the identifier of the target node and the communication protocol of the target node and that is stored in the smart contract is modified based on a communication protocol address of the backup node, so that in subsequent service execution, another node can determine, based on the correspondence stored in the smart contract, to communicate with the backup node to execute the corresponding service, and not to communicate with the original target node to restart the target node that goes down.

[0049] It should be noted that the backup node can prestore a correspondence between the identifier of the target node and the trusted storage node of the target node, so that when detecting that the target node goes down, the backup node can determine a specific trusted storage node from which the private data of the target node is to be obtained.

[0050] Based on the node restart system for a consortium blockchain described in FIG. 1, the system includes a plurality of target nodes, and further includes a trusted storage node and a backup node that correspond to each target node. The target node is configured to back up data stored in the target node. The trusted storage node is configured to store data sent by the target node. The backup node is configured to: monitor a running status of the target node in real time, and when the target node goes down, obtain the data of the target node from the trusted storage node, to execute a corresponding service in place of the target node.

[0051] It can be seen from the above-mentioned system that data stored in each node is classified based on whether the data is common, common data of each node is pre-backed up by using a cloud server, and then non-common data of each node is updated in a trusted storage node of each node based on a preset cycle. Then, when detecting that the target node goes down, a backup node can obtain corresponding data from the cloud server and the trusted storage node, and execute a service in place of the node that goes down. This ensures high availability of the consortium blockchain, reduces a data amount of backup data, and saves storage costs.

[0052] In addition, when backing up the data of the target node, the target node can further determine the configuration data based on stateless logical data in the target node, determine the

private data based on state data in the target node, and determine the correspondence between the identifier of the target node and the communication protocol address of the target node based on stateful shared data in the target node.

[0053] When the private data of the target node is determined, to enable the backup node to execute the corresponding service in place of the target node after obtaining the private data of the target node, the private data of the target node should further include at least a private key of the target node, so that the backup node can obtain the private key of the target node by obtaining the private data of the target node, then can complete identity verification by using the private key when executing the service in place of the target node, and can further modify, based on the private key of the target node, the correspondence stored in the smart contract.

[0054] Based on this, a public key of the target node should be further stored in the trusted storage node of the target node. In this case, when updating the data in the trusted storage node, the target node should further encrypt the private data by using the private key of the target node, and determine encrypted data; and send the encrypted data to the trusted storage node, and then the trusted storage node performs identity verification on the encrypted data based on the stored public key of target node, and when the identity verification succeeds, updates the data that is of the target node and that is stored in the trusted storage node of the target node.

[0055] Specifically, when the target node updates the data stored in the trusted storage node, to further reduce data that needs to be uploaded to the trusted storage node by the target node, reduce a time required for a backup, and save interface resources of each node and storage resources of the trusted storage node, the target node can further determine an identifier of data on which consensus reaching is performed from a previous update cycle to a current update cycle; delete data that is in the trusted storage node and that has an identifier the same as the identifier of the data on which consensus reaching is performed; then determine, for each piece of private data of the target node at the current moment, whether the trusted storage node stores data that has an identifier the same as an identifier of the private data; and if yes, update the data with the same identifier in the trusted storage node based on the private data; or if no, store the private data in the trusted storage node.

[0056] Further, to further reduce the data that needs to be uploaded, an identifier of each piece of private data at the current moment of the target node can be further determined as a first identifier and an identifier of each piece of data that is of the target node and that is stored in the

trusted storage node can be determined as a second identifier. Private data with different first identifiers and any second identifiers in all the pieces of private data is first backed up in the trusted storage node. Then, it is determined whether private data with a same first identifier and any second identifier in all the pieces of private data is data that is processed but on which consensus reaching is not performed. If no, data corresponding to the same identifier in the trusted storage node is deleted, and data corresponding to the same identifier in the private data is stored. If yes, the target node does not need to back up data corresponding to the same identifier to the trusted storage node.

[0057] It should be noted that in the above-mentioned embodiment, the configuration data of each node is prestored, and the private data of the target node is stored in the trusted storage node of the target node. Certainly, the configuration data of each node and the private data of the target node can be stored in the cloud server, can be stored in the trusted storage node, or can be stored in another trusted storage device. This is not limited in one or more embodiments of this specification.

[0058] In addition, the above-mentioned different backup manners for various types of data stored in the target node are merely classification manners provided in this embodiment of this specification. The various types of data can be further classified. For example, identity verification-related data of the target node is stored in another trusted cloud server, or identity data of nodes belonging to a same participant is stored in a same region, and transaction data is stored in the trusted storage node. This is not limited in this specification. Because there are many classification methods, examples are not listed one by one herein.

[0059] Further, if target nodes that support services in different regions are configured in the consortium blockchain by a participant of the consortium blockchain, especially when the consortium blockchain is applied to a scenario such as a cross-border payment or a cross-region service, for each backup node, the backup node can further monitor each target node and each backup node that are configured by a participant to which the backup node belongs; when detecting that both a target node and a backup node in a same region go down, encrypt an identifier of the regional target node by using a private key of the backup node, and determine encrypted data; send an obtaining instruction that carries the encrypted data to a trusted storage node of the regional target node, so that the trusted storage node of the regional target node performs identity verification on the encrypted data based on a public key of the backup node, receive private data that is of the regional target node and that is returned by the trusted storage node of the regional

target node, and obtain configuration data from the cloud server; and update, based on the communication protocol address of the backup node, the correspondence stored in the smart contract, execute the corresponding service in place of the target node, and execute a corresponding service in place of the regional target node.

[0060] It should be noted that a relatively large quantity of backup nodes are determined. Therefore, when a target node goes down, a plurality of backup nodes detect that the target node goes down. In this case, an appropriate backup node can be determined based on a determined downtime of the target node and a resource usage status of the backup node.

[0061] Specifically, each backup node can monitor a time when detecting that the target node goes down, and perform broadcasting in the nodes in the consortium blockchain. Certainly, an appropriate backup node can alternatively be determined based on a communication time between each backup node and the target node that goes down. This is not limited in this specification.

[0062] In addition, during node restart, the backup node does not limit a sequence of obtaining the configuration data and the private data of the target node, provided that it is ensured that when the backup node needs to execute a task in place of the target node, the data of the target node is stored in the backup node.

[0063] FIG. 2 is a schematic diagram of a node restart procedure for a consortium blockchain according to this specification. A server a in a region A and a server b in a region B constitute a consortium blockchain. A server c is a cloud server, a server d is a backup node of the server a, a solid line with an arrow is an interaction relationship between nodes in the consortium blockchain, and a dashed line with an arrow indicates interaction no longer performed after the server d executes a service in place of the server a. The server a periodically packages and stores data in the server c. The server d monitors a running status of the server a in the region A in real time. The server a and the server b in the region B interact with each other, to implement a service execution procedure. When detecting that the server a is faulty, the server d obtains data most recently updated by the server a from the server c, deploys the data on the server d, and then changes, by using a smart contract, an address of a server that interacts with the server d, to execute the service in place of the server a that goes down, and quickly restart the faulty node.

[0064] In addition, in one or more embodiments of this specification, a scenario to which the consortium blockchain is applied, for example, a cross-border payment and distributed server resource scheduling, is not limited. Because there are many applicable scenarios, examples are not

listed one by one in this specification.

[0065] Based on the node restart system for a consortium blockchain in FIG. 1, this specification further provides a node restart method for a consortium blockchain. FIG. 3 shows a node restart method for a consortium blockchain according to this specification. The method can specifically include the following steps.

[0066] S300: Obtain configuration data of a target node from a preset cloud server.

[0067] S302: Determine a correspondence between an identifier of the target node and a communication protocol address of the target node, and upload the correspondence to a smart contract in the consortium blockchain for storage.

[0068] Each node in the consortium blockchain has a different identity and therefore needs to execute a different service. Therefore, to enable another node to accurately communicate with the target node based on the identifier of the node when communication needs to be performed with the target, the target node can store a correspondence between the identifier of the target node and information such as the communication protocol address or a port of the target node in the smart contract.

[0069] Specifically, the identifier corresponding to the target node is determined, and the communication protocol address of the target node is determined. Then, after the correspondence between the identifier of the target node and the communication protocol address of the target node is determined, the correspondence is stored in the smart contract in the consortium blockchain, and is shared with another node in the consortium blockchain through consensus reaching by using the smart contract.

[0070] In addition, when target nodes that support services in different regions are configured in the consortium blockchain by a participant of the consortium blockchain, especially when nodes in the consortium blockchain need to interact with each other to execute a corresponding service across regions, to ensure data security and accurate and fast service execution, an identity of a participant to which the target node belongs can be further determined, and the identifier of the target node can be determined based on the identity and a region in which the target node executes the service.

[0071] S304: Determine private data of the target node based on a preset update cycle, and update, based on the private data, private data that is of the target node and that is stored in a trusted storage node, where the private data includes data that is of the target node and on which consensus

reaching is to be performed.

[0072] When the consortium blockchain starts to run, transaction data stored in the nodes in the consortium blockchain, received transaction requests, and other data are different. The data is data on which each node in the consortium blockchain does not perform consensus reaching with another node, or is data that is not processed. The data should be confidential and invisible to the another node, that is, is private data. Therefore, a trusted storage node should be selected to back up the data.

[0073] It should be noted that the trusted storage node can be a device that can communicate with another node, or can be a cloud server. This is not limited in this specification. Certainly, the trusted storage node can be a storage device or a cloud server corresponding to the target node, and is only configured to store the private data of the target node, or can be a storage device or a cloud server shared by some nodes in the consortium blockchain, or can be a storage device or a cloud service shared by nodes configured by a participant in the consortium blockchain. This is not limited in this specification.

[0074] Further, for each node serving as the target node, content of private data of the target node at different times is different. Therefore, when data of each node is backed up, the data that is backed up should be updated based on the preset update cycle, to ensure that when the target node goes down, no excessive data losses are caused when the target node is restarted.

[0075] Specifically, the preset cycle for updating the backup can be determined based on a speed of processing the transaction request by the target node and a speed of receiving the transaction request. Certainly, the preset cycle can vary with a time point. In one or more embodiments of this specification, how to specifically determine the preset cycle is not limited, provided that it is ensured that there are as few data losses as possible during node restart.

[0076] Then, when the preset cycle is reached, the target node can delete all data stored in the trusted storage node of the target node, and then store private data determined at a current moment in the trusted storage node.

[0077] To implement fast node restart when the target node goes down, and reduce a loss caused because the node goes down, a backup node of the target node can be preset to monitor a running status of each node. It should be noted that to ensure security and privacy of the consortium blockchain, the backup node and the target node should belong to a same participant, or the backup node is a trusted node of the target node. Certainly, whether the backup node is a node in the

consortium blockchain is not limited in this specification, and can be selected based on an actual requirement. Certainly, the backup node can alternatively be a standby server room of each node in the consortium blockchain, or is located on a same server as each node in the consortium blockchain, and is a space resource of the server. Further, a cloud computing device can be used as the backup node of the target node.

[0078] It should be noted that in one or more embodiments of this specification, a method used by the backup node to monitor a running status of the target node is not limited.

[0079] When detecting that the target node goes down, the backup node can first obtain common configuration data of each node, then determine the trusted storage node of the target node based on the identifier of the target node, and obtain the private data of the target node from the trusted storage node of the target node. When it is determined that the configuration data and the private data of the target node are loaded into the backup node and the service can be normally executed, a request is sent to the smart contract, and the correspondence that is between the identifier of the target node and the communication protocol of the target node and that is stored in the smart contract is modified based on a communication protocol address of the backup node, so that in subsequent service execution, another node can determine, based on the correspondence stored in the smart contract, to communicate with the backup node to execute the corresponding service, and not to communicate with the original target node to restart the target node that goes down.

[0080] Based on the node restart method for a consortium blockchain shown in FIG. 3, common configuration data of each target node in the blockchain is backed up in a preset server when the consortium blockchain is established, and a correspondence between an identifier of the target node and a communication protocol address of the target node is recorded in a smart contract. In addition, private data of the target node is determined based on a preset cycle, and data that is of the target node and that is stored in a trusted storage node is updated, so that when detecting that the target node goes down, a backup node of the target node can obtain the configuration data from the cloud server and the private data of the target node from the trusted storage node, to execute a corresponding service in place of the target node.

[0081] It can be seen from the above-mentioned method that data stored in each node is classified based on whether the data is common, common data of each node is pre-backed up by using a cloud server, and then non-common data of each node is updated in a trusted storage node

of each node based on a preset cycle, so that when detecting that the target node goes down, a backup node can obtain corresponding data from the cloud server and the trusted storage node, and execute a service in place of the node that goes down. This ensures high availability of the consortium blockchain, reduces a data amount of backup data, and saves storage costs.

[0082] When the private data of the target node is determined in step S304, to enable the backup node to execute the corresponding service in place of the target node after obtaining the private data of the target node, the private data of the target node should further include at least a private key of the target node, so that the backup node can obtain the private key of the target node by obtaining the private data of the target node, then can complete identity verification by using the private key when executing the service in place of the target node, and can further modify, based on the private key of the target node, the correspondence stored in the smart contract.

[0083] Based on this, a public key of the target node should be further stored in the trusted storage node of the target node. In this case, when updating the data in the trusted storage node, the target node should further encrypt the private data by using the private key of the target node, and determine encrypted data; and send the encrypted data to the trusted storage node, so that the trusted storage node performs identity verification on the encrypted data based on the stored public key of target node, and when the identity verification succeeds, updates the data that is of the target node and that is stored in the trusted storage node of the target node.

[0084] Specifically, when the target node updates the data stored in the trusted storage node, to further reduce data that needs to be uploaded to the trusted storage node by the target node, reduce a time required for a backup, and save interface resources of each node and storage resources of the trusted storage node, the target node can further determine an identifier of data on which consensus reaching is performed from a previous update cycle to a current update cycle; delete data that is in the trusted storage node and that has an identifier the same as the identifier of the data on which consensus reaching is performed; then determine, for each piece of private data of the target node at the current moment, whether the trusted storage node stores data that has an identifier the same as an identifier of the private data; and if yes, update the data with the same identifier in the trusted storage node based on the private data; or if no, store the private data in the trusted storage node.

[0085] Further, to further reduce the data that needs to be uploaded, an identifier of each piece of private data at the current moment of the target node can be further determined as a first

identifier and an identifier of each piece of data that is of the target node and that is stored in the trusted storage node can be determined as a second identifier. Private data with different first identifiers and any second identifiers in all the pieces of private data is first backed up in the trusted storage node. Then, it is determined whether private data with a same first identifier and any second identifier in all the pieces of private data is data that is processed but on which consensus reaching is not performed. If no, data corresponding to the same identifier in the trusted storage node is deleted, and data corresponding to the same identifier in the private data is stored. If yes, the target node does not need to back up data corresponding to the same identifier to the trusted storage node.

[0086] It should be noted that in the above-mentioned embodiment, the configuration data of each node is prestored in the cloud server, and the private data of the target node is stored in the trusted storage node of the target node. Certainly, the configuration data of each node and the private data of the target node can be stored in the cloud server, can be stored in the trusted storage node, or can be stored in another trusted storage device. This is not limited in one or more embodiments of this specification.

[0087] In addition, the above-mentioned different backup manners for various types of data stored in the target node are merely classification manners provided in this embodiment of this specification. The various types of data can be further classified. For example, identity verification-related data of the target node is stored in another trusted cloud server, or identity data of nodes belonging to a same participant is stored in a same region, and transaction data is stored in the trusted storage node. This is not limited in this specification. Because there are many classification methods, examples are not listed one by one herein.

[0088] FIG. 4 shows a node restart method for a consortium blockchain according to this specification. The method can specifically include the following steps.

[0089] S400: When detecting that a target node corresponding to a backup node goes down, determine an identifier of the target node, obtain configuration data, and configure the backup node based on the configuration data.

[0090] In one or more embodiments of this specification, a specific device that is the backup node is not limited, for example, a personal computer, a mobile terminal, and a server. However, the backup node and the target node should belong to a same participant, or the backup node is a trusted node of the target node. Certainly, whether the backup node is a node in the consortium blockchain is not limited in this specification, and can be selected based on an actual requirement.

The backup node can alternatively be a standby server room of each node in the consortium blockchain, or is located on a same server as each node in the consortium blockchain, and is a space resource of the server.

[0091] The backup node can monitor the target node in real time. Certainly, in one or more embodiments of this specification, a specific monitoring method, for example, a heartbeat detection mechanism, is not limited. When the backup node detects that the target node goes down, the backup node should start to prepare for obtaining common configuration data of each node.

[0092] It should be noted that the configuration data is common configuration data of each node, for example, running environment data, consensus module code logic, communication module code logic, and other common data and code of each node, can further include common data such as permission data of the node, a startup script, and a whitelist of nodes that can participate in the consortium blockchain, and can be backed up in the cloud server after each node is configured, or can be configuration data determined and stored in the cloud server when the consortium blockchain is constructed. Certainly, the configuration data can alternatively be stored in another device. This is not limited in this specification.

[0093] S402: Obtain private data of the target node from a trusted storage node based on the identifier of the target node, where the private data includes data that is of the target node and on which consensus reaching is to be performed.

[0094] To execute a corresponding service in place of the target node, the backup node should further obtain the private data of the target node. Specifically, the trusted storage node of the target node is determined based on the identifier of the target node, and then the private data of the target node is obtained from the trusted storage node. The private data is data on which each node in the consortium blockchain does not perform consensus reaching with another node, or is data that is not processed. The data should be confidential and invisible to the another node, and is updated by the target node in the trusted storage node based on a preset update cycle. A specific update method is described above. Details are not described herein again.

[0095] Certainly, to ensure security and privacy of data in the consortium blockchain, the trusted storage node should further need to perform identity verification on a request for obtaining data of the target node. Specifically, the trusted storage node should further store a public key of the backup node. When the backup node needs to send a data obtaining request to the trusted storage node, the backup node encrypts identity information of the target node by using a private

key of the backup node, and determines encrypted information; then sends an obtaining request that carries the encrypted information to the trusted storage node, so that the trusted storage node performs identity verification on the encrypted information based on the public key of the backup node; and then receives private data returned by the trusted storage node after the identity verification succeeds.

[0096] S404: Update, based on a communication protocol address of the backup node, a correspondence stored in the smart contract, and execute a service in place of the target node, where the correspondence stored in the smart contract is a correspondence that is between the identifier of the target node and a communication protocol address of the target node and that is prestored in the smart contract by the target node.

[0097] After the backup node obtains the configuration data and the private data of the target node, to notify, through broadcast, another node in the consortium blockchain that the target node goes down and currently the backup node executes the corresponding service in place of the target node, the backup node changes, based on the obtained private data of the target node, the correspondence that is between the identifier of the target node and the communication protocol address of the target node and that is stored in the smart contract.

[0098] Specifically, data that needs to be changed is encrypted based on a private key of the target node in the private data of the target node, and then an access request that carries the encrypted data is sent to the smart contract. After the smart contract completes identity verification on the access request based on a public key of the target node, the backup node can update the communication protocol address of the target node to the communication protocol address of the backup node.

[0099] Certainly, a dual verification manner can be further used to implement identity verification on each node by the smart contract, and implement identity verification on the target node and the backup node by the trusted storage node. This is not limited in this specification.

[0100] After updating the smart contract, the backup node can execute the corresponding service in place of the target node.

[0101] Based on the node restart method for a consortium blockchain shown in FIG. 4, common configuration data of each target node in the blockchain is backed up in a preset server when the consortium blockchain is established, and a correspondence between an identifier of the target node and a communication protocol address of the target node is recorded in a smart contract.

In addition, private data of the target node is determined based on a preset cycle, and data that is of the target node and that is stored in a trusted storage node is updated, so that when detecting that the target node goes down, a backup node of the target node can obtain the configuration data from the cloud server and the private data of the target node from the trusted storage node, to execute a corresponding service in place of the target node.

[0102] It can be seen from the above-mentioned method that data stored in each node is classified based on whether the data is common, common data of each node is pre-backed up by using a cloud server, and then non-common data of each node is updated in a trusted storage node of each node based on a preset cycle, so that when detecting that the target node goes down, a backup node can obtain corresponding data from the cloud server and the trusted storage node, and execute a service in place of the node that goes down. This ensures high availability of the consortium blockchain, reduces a data amount of backup data, and saves storage costs.

[0103] In addition, if target nodes that support services in different regions are configured in the consortium blockchain by a participant of the consortium blockchain, FIG. 5 is a schematic diagram of a consortium blockchain according to this specification. Two dashed-line circles represent two regions. Certainly, there can be a plurality of service regions. In each region, a plurality of target nodes are disposed by the participant, and target nodes in different regions can communicate with each other, to execute a service across regions. For each target node, a backup node and a trusted storage node that correspond to the target node are configured.

[0104] It should be noted that a backup node in a same region can be used to restart another target node in the region. When the consortium blockchain is applied to a scenario such as a cross-border payment or a cross-region service, for each backup node, the backup node can further monitor each target node and each backup node that are configured by a participant to which the backup node belongs, regardless of whether all of the nodes are located in a same region; when detecting that both a target node and a backup node in a same region go down, encrypt an identifier of the regional target node by using a private key of the backup node, and determine encrypted data; send an obtaining instruction that carries the encrypted data to a trusted storage node of the regional target node, so that the trusted storage node of the regional target node performs identity verification on the encrypted data based on a public key of the backup node, receive private data that is of the regional target node and that is returned by the trusted storage node of the regional target node, and obtain configuration data from the cloud server; and update, based on the

communication protocol address of the backup node, the correspondence stored in the smart contract, execute the corresponding service in place of the target node, and execute a corresponding service in place of the regional target node.

[0105] FIG. 6 is a schematic diagram of a node restart procedure for a consortium blockchain according to this specification. When a server e in a region E detects that both a target node and a backup node in a region A go down, a backup node, that is, the server e, in the region E can serve as a backup node of a server a in the region A, and temporarily execute a service in place of the server a. Certainly, if the method is applied to the cross-border payment field, the server e and the server a should belong to a same participant. A solid line with an arrow is an interaction relationship between nodes in the consortium blockchain, and a dashed line with an arrow indicates interaction no longer performed after the server e executes the service in place of the server a.

[0106] It should be noted that a relatively large quantity of backup nodes are determined. Therefore, when a target node goes down, a plurality of backup nodes detect that the target node goes down. In this case, an appropriate backup node can be determined based on a determined downtime of the target node and a resource usage status of the backup node.

[0107] Specifically, each backup node can monitor a time when detecting that the target node goes down, and perform broadcasting in the nodes in the consortium blockchain. Certainly, an appropriate backup node can alternatively be determined based on a communication time between each backup node and the target node that goes down. This is not limited in this specification.

[0108] It should be noted that in one or more embodiments provided in this specification, when a node in the consortium blockchain goes down, a restart method of the node is implemented through communication between a plurality of nodes, and decentralization of the consortium blockchain is implemented to some extent. Certainly, in the above-mentioned method, an entirety including the nodes in the consortium blockchain can be considered as a server cluster, and a running status of the node can be monitored by a management server of the server cluster. When a node in the blockchain goes down, the management server determines a backup node to be used to execute a corresponding service in place of the node that goes down. Certainly, nodes belonging to each participant can be used as a server cluster, and then a management server to which each participant belongs monitors each node of the participant in the blockchain, and the management server determines a backup node to be used to execute a corresponding service in place of the node that goes down.

[0109] The specification also provides a computer-readable non-transitory storage medium, wherein the storage medium stores a computer program, which, when executed by a processor, can be used to perform one or more steps of one or more methods described or illustrated herein or provides functionality described or illustrated herein. Herein, the computer-readable non-transitory storage medium or media may include one or more semiconductor-based or other integrated circuits (ICs) (such, as for example, field-programmable gate arrays (FPGAs) or application-specific ICs (ASICs)), hard disk drives (HDDs), hybrid hard drives (HHDs), optical discs, optical disc drives (ODDs), magneto-optical discs, magneto-optical drives, floppy diskettes, floppy disk drives (FDDs), magnetic tapes, solid-state drives (SSDs), RAM-drives, SECURE DIGITAL cards or drives, any other suitable computer-readable non-transitory storage media, or any suitable combination of two or more of these, where appropriate. A computer-readable non-transitory storage medium may be volatile, non-volatile, or a combination of volatile and non-volatile, where appropriate.

[0110] The specification also provides a computer system. FIG. 7 illustrates an example computer system 800. In particular embodiments, one or more computer systems 800 perform one or more steps of one or more methods described or illustrated herein. In particular embodiments, one or more computer systems 800 provide functionality described or illustrated herein. In particular embodiments, software running on one or more computer systems 800 performs one or more steps of one or more methods described or illustrated herein or provides functionality described or illustrated herein. Particular embodiments include one or more portions of one or more computer systems 800. Herein, reference to a computer system may encompass a computing device, and vice versa, where appropriate. Moreover, reference to a computer system may encompass one or more computer systems, where appropriate.

[0111] This disclosure contemplates any suitable number of computer systems 800. This disclosure contemplates computer system 800 taking any suitable physical form. As example and not by way of limitation, computer system 800 may be an embedded computer system, a system-on-chip (SOC), a single-board computer system (SBC) (such as, for example, a computer-on-module (COM) or system-on-module (SOM)), a desktop computer system, a laptop or notebook computer system, an interactive kiosk, a mainframe, a mesh of computer systems, a mobile telephone, a personal digital assistant (PDA), a server, a tablet computer system, or a combination of two or more of these. Where appropriate, computer system 800 may include one or more

computer systems 800; be unitary or distributed; span multiple locations; span multiple machines; span multiple data centers; or reside in a cloud, which may include one or more cloud components in one or more networks. Where appropriate, one or more computer systems 800 may perform without substantial spatial or temporal limitation one or more steps of one or more methods described or illustrated herein. As an example and not by way of limitation, one or more computer systems 800 may perform in real time or in batch mode one or more steps of one or more methods described or illustrated herein. One or more computer systems 800 may perform at different times or at different locations one or more steps of one or more methods described or illustrated herein, where appropriate.

[0112] In particular embodiments, computer system 800 includes a processor 802, memory 804, storage 806, an input/output (I/O) interface 808, a communication interface 810, and a bus 812. Although this disclosure describes and illustrates a particular computer system having a particular number of particular components in a particular arrangement, this disclosure contemplates any suitable computer system having any suitable number of any suitable components in any suitable arrangement.

[0113] In particular embodiments, processor 802 includes hardware for executing instructions, such as those making up a computer program. As an example and not by way of limitation, to execute instructions, processor 802 may retrieve (or fetch) the instructions from an internal register, an internal cache, memory 804, or storage 806; decode and execute them; and then write one or more results to an internal register, an internal cache, memory 804, or storage 806. In particular embodiments, processor 802 may include one or more internal caches for data, instructions, or addresses. This disclosure contemplates processor 802 including any suitable number of any suitable internal caches, where appropriate. As an example and not by way of limitation, processor 802 may include one or more instruction caches, one or more data caches, and one or more translation lookaside buffers (TLB s). Instructions in the instruction caches may be copies of instructions in memory 804 or storage 806, and the instruction caches may speed up retrieval of those instructions by processor 802. Data in the data caches may be copies of data of 80 in memory 804 or storage 806 for instructions executing at processor 802 to operate on; the results of previous instructions executed at processor 802 for access by subsequent instructions executing at processor 802 or for writing to memory 804 or storage 806; or other suitable data. The data caches may speed up read or write operations by processor 802. The TLBs may speed up virtual-address translation

for processor 802. In particular embodiments, processor 802 may include one or more internal registers for data, instructions, or addresses. This disclosure contemplates processor 802 including any suitable number of any suitable internal registers, where appropriate. Where appropriate, processor 802 may include one or more arithmetic logic units (ALUs); be a multi-core processor; or include one or more processors 802. Although this disclosure describes and illustrates a particular processor, this disclosure contemplates any suitable processor.

[0114] In particular embodiments, memory 804 includes main memory for storing instructions for processor 802 to execute or data for processor 802 to operate on. As an example and not by way of limitation, computer system 800 may load instructions from storage 806 or another source (such as, for example, another computer system 800) to memory 804. Processor 802 may then load the instructions from memory 804 to an internal register or internal cache. To execute the instructions, processor 802 may retrieve the instructions from the internal register or internal cache and decode them. During or after execution of the instructions, processor 802 may write one or more results (which may be intermediate or final results) to the internal register or internal cache. Processor 802 may then write one or more of those results to memory 804. In particular embodiments, processor 802 executes only instructions in one or more internal registers or internal caches or in memory 804 (as opposed to storage 806 or elsewhere) and operates only on data in one or more internal registers or internal caches or in memory 804 (as opposed to storage 806 or elsewhere). One or more memory buses (which may each include an address bus and a data bus) may couple processor 802 to memory 804. Bus 812 may include one or more memory buses, as described below. In particular embodiments, one or more memory management units (MMUs) reside between processor 802 and memory 804 and facilitate accesses to memory 804 requested by processor 802. In particular embodiments, memory 804 includes random access memory (RAM). This RAM may be volatile memory, where appropriate. Where appropriate, this RAM may be dynamic RAM (DRAM) or static RAM (SRAM). Moreover, where appropriate, this RAM may be single-ported or multi-ported RAM. This disclosure contemplates any suitable RAM. Memory 804 may include one or more memories 804, where appropriate. Although this disclosure describes and illustrates particular memory, this disclosure contemplates any suitable memory.

[0115] In particular embodiments, storage 806 includes mass storage for data or instructions. As an example and not by way of limitation, storage 806 may include a hard disk drive (HDD), a floppy disk drive, flash memory, an optical disc, a magneto-optical disc, magnetic tape, or a

Universal Serial Bus (USB) drive or a combination of two or more of these. Storage 806 may include removable or non-removable (or fixed) media, where appropriate. Storage 806 may be internal or external to computer system 800, where appropriate. In particular embodiments, storage 806 is non-volatile, solid-state memory. In particular embodiments, storage 806 includes read-only memory (ROM). Where appropriate, this ROM may be mask-programmed ROM, programmable ROM (PROM), erasable PROM (EPROM), electrically erasable PROM (EEPROM), electrically alterable ROM (EAROM), or flash memory or a combination of two or more of these. This disclosure contemplates mass storage 806 taking any suitable physical form. Storage 806 may include one or more storage control units facilitating communication between processor 802 and storage 806, where appropriate. Where appropriate, storage 806 may include one or more storages 806. Although this disclosure describes and illustrates particular storage, this disclosure contemplates any suitable storage.

[0116] In particular embodiments, I/O interface 808 includes hardware, software, or both, providing one or more interfaces for communication between computer system 800 and one or more I/O devices. Computer system 800 may include one or more of these I/O devices, where appropriate. One or more of these I/O devices may enable communication between a person and computer system 800. As an example and not by way of limitation, an I/O device may include a keyboard, keypad, microphone, monitor, mouse, printer, scanner, speaker, still camera, stylus, tablet, touch screen, trackball, video camera, another suitable I/O device or a combination of two or more of these. An I/O device may include one or more sensors. This disclosure contemplates any suitable I/O devices and any suitable I/O interfaces 808 for them. Where appropriate, I/O interface 808 may include one or more device or software drivers enabling processor 802 to drive one or more of these I/O devices. I/O interface 808 may include one or more I/O interfaces 808, where appropriate. Although this disclosure describes and illustrates a particular I/O interface, this disclosure contemplates any suitable I/O interface.

[0117] In particular embodiments, communication interface 810 includes hardware, software, or both providing one or more interfaces for communication (such as, for example, packet-based communication) between computer system 800 and one or more other computer systems 800 or one or more networks. As an example and not by way of limitation, communication interface 810 may include a network interface controller (NIC) or network adapter for communicating with an Ethernet or other wire-based network or a wireless NIC (WNIC) or wireless adapter for

communicating with a wireless network, such as a WI-FI network. This disclosure contemplates any suitable network and any suitable communication interface 810 for it. As an example and not by way of limitation, computer system 800 may communicate with an ad hoc network, a personal area network (PAN), a local area network (LAN), a wide area network (WAN), a metropolitan area network (MAN), or one or more portions of the Internet or a combination of two or more of these. One or more portions of one or more of these networks may be wired or wireless. As an example, computer system 800 may communicate with a wireless PAN (WPAN) (such as, for example, a BLUETOOTH WPAN), a WI-FI network, a WI-MAX network, a cellular telephone network (such as, for example, a Global System for Mobile Communications (GSM) network), or other suitable wireless network or a combination of two or more of these. Computer system 800 may include any suitable communication interface 810 for any of these networks, where appropriate. Communication interface 810 may include one or more communication interfaces 810, where appropriate. Although this disclosure describes and illustrates a particular communication interface, this disclosure contemplates any suitable communication interface.

[0118] In particular embodiments, bus 812 includes hardware, software, or both coupling components of computer system 800 to each other. As an example and not by way of limitation, bus 812 may include an Accelerated Graphics Port (AGP) or other graphics bus, an Enhanced Industry Standard Architecture (EISA) bus, a front-side bus (FSB), a HYPERTRANSPORT (HT) interconnect, an Industry Standard Architecture (ISA) bus, an INFINIBAND interconnect, a low-pin-count (LPC) bus, a memory bus, a Micro Channel Architecture (MCA) bus, a Peripheral Component Interconnect (PCI) bus, a PCI-Express (PCIe) bus, a serial advanced technology attachment (SATA) bus, a Video Electronics Standards Association local (VLB) bus, or another suitable bus or a combination of two or more of these. Bus 812 may include one or more buses 812, where appropriate. Although this disclosure describes and illustrates a particular bus, this disclosure contemplates any suitable bus or interconnect.

[0119] Certainly, in addition to software implementations, this specification does not exclude other implementations such as a logic device or a combination of software and hardware. In other words, an execution body of the following processing procedure is not limited to each logical unit, and can be hardware or a logic device.

[0120] The above-mentioned descriptions are merely embodiments of this specification, and are not intended to limit this specification. A person skilled in the art can make various changes

and variations to this specification. Any modification, equivalent replacement, improvement, etc. made without departing from the spirit and principle of this specification shall fall within the scope of the claims in this application.

CLAIMS

What is claimed is:

1. A node restart system for a consortium blockchain, wherein the system comprises a target node and a trusted storage node and a backup node that correspond to the target node;

the target node is configured to: execute a service based on configured configuration data; upload a correspondence between an identifier of the target node and a communication protocol address of the target node to a smart contract in the consortium blockchain; and determine private data of the target node based on a preset update cycle, and update, based on the private data, private data that is of the target node and that is stored in the trusted storage node, wherein the private data comprises data that is of the target node and on which consensus reaching is to be performed;

the trusted storage node is configured to store the private data updated by the target node; and

the backup node is configured to: when detecting that the target node goes down, determine the identifier of the target node, obtain the configuration data, and configure the backup node based on the configuration data; obtain the private data of the target node from the trusted storage node based on the identifier of the target node; and update the correspondence in the smart contract to a communication protocol address of the backup node, to execute the service in place of the target node.

2. The system according to claim 1, wherein the target node is configured to: determine data that exists at a current moment and on which consensus reaching is to be performed, wherein the data on which consensus reaching is to be performed comprises data that is not processed by the target node at the current moment and data that is processed but on which consensus reaching is not performed; and determine a private key of the target node, and use the private key of the target node and the data that is of the target node, that exists at the current moment, and on which consensus reaching is to be performed as the private data of the target node.

3. The system according to claim 2, wherein the target node is configured to: determine an identifier of data on which consensus reaching is performed from a previous update cycle to a current update cycle; delete data that is in the trusted storage node and that has an identifier the same as the identifier of the data on which consensus reaching is performed; determine, for each piece of private data, whether the trusted storage node stores data that has an identifier the same as an identifier of the private data; and if yes, update the data with the same identifier in the trusted

storage node based on the private data; or if no, store the private data in the trusted storage node.

4. The system according to claim 2, wherein the trusted storage node stores a public key of the target node;

the target node is configured to: encrypt the private data by using the private key of the target node, and determine encrypted data; and send the encrypted data to the trusted storage node; and

the trusted storage node is configured to: perform identity verification on the encrypted data based on the stored public key of the target node, and when the identity verification succeeds, update the data that is of the target node and that is stored in the trusted storage node.

5. The system according to claim 1, wherein target nodes that support services in different regions are configured in the consortium blockchain by a participant of the consortium blockchain;

the target node is configured to: determine an identity of a participant to which the target node belongs; and determine the identifier of the target node based on the identity and a region in which the target node executes the service; and

the backup node is configured to: when detecting that the target node goes down, determine, based on the identifier of the target node, the identity of the target node and the region in which the target node executes the service, and determine the trusted storage node of the target node based on the identity and the region in which the target node executes the service.

6. The system according to claim 1, wherein the backup node is configured to: determine the communication protocol address of the backup node; update the communication protocol address of the target node in the correspondence that is between the identifier of the target node and the communication protocol address of the target node and that is stored in the smart contract to the communication protocol address of the backup node; and in a service execution process, execute the service based on each communication protocol address stored in the smart contract.

7. The system according to claim 1, wherein the trusted node stores a public key of the backup node;

the backup node is configured to: encrypt identity information of the target node by using a private key of the backup node, and determine encrypted information; send an obtaining request that carries the encrypted information to the trusted storage node; and receive private data returned by the trusted storage node after identity verification succeeds; and

the trusted storage node is configured to perform identity verification on the encrypted information based on the public key of the backup node.

8. The system according to claim 7, wherein target nodes that support services in different regions are configured in the consortium blockchain by a participant of the consortium blockchain;

the backup node is configured to: monitor each target node and each backup node that are configured by a participant to which the backup node belongs; when detecting that both a target node and a backup node in a same region go down, encrypt an identifier of the regional target node by using the private key of the backup node, and determine encrypted data; send an obtaining instruction that carries the encrypted data to a trusted storage node of the regional target node, receive private data that is of the regional target node and that is returned by the trusted storage node of the regional target node, and obtain the configuration data; and update, based on the communication protocol address of the backup node, the correspondence stored in the smart contract, execute the corresponding service in place of the target node, and execute a corresponding service in place of the regional target node; and

the trusted storage node of the regional target node is configured to perform identity verification on the encrypted data based on the public key of the backup node.

9. The system according to any one of claims 1 to 8, wherein the target node determines the configuration data based on stateless logical data in the target node, determines the private data based on state data in the target node, and determines the correspondence between the identifier of the target node and the communication protocol address of the target node based on stateful shared data in the target node.

10. A method performed by a target node in a consortium blockchain, comprising:

executing a service based on configured configuration data;

determining a correspondence between an identifier of the target node and a communication protocol address of the target node, and uploading the correspondence to a smart contract in the consortium blockchain for storage; and

determining private data of the target node based on a preset update cycle, and updating, based on the private data, private data that is of the target node and that is stored in a trusted storage node, wherein the private data comprises data that is of the target node and on which consensus reaching is to be performed.

11. The method according to claim 10, wherein the determining private data of the target node specifically comprises:

determining data that is of the target node, that exists at a current moment, and on which

consensus reaching is to be performed, wherein the data on which consensus reaching is to be performed comprises data that is not processed by the target node at the current moment and data that is processed but on which consensus reaching is not performed; and

determining a private key of the target node, and using the private key of the target node and the data that is of the target node, that exists at the current moment, and on which consensus reaching is to be performed as the private data of the target node.

12. The method according to claim 11, wherein the updating, based on the private data, private data that is of the target node and that is stored in a trusted storage node specifically comprises:

determining an identifier of data on which consensus reaching is performed from a previous update cycle to a current update cycle;

deleting data that is in the trusted storage node and that has an identifier the same as the identifier of the data on which consensus reaching is performed;

determining, for each piece of private data, whether the trusted storage node stores data that has an identifier the same as an identifier of the private data; and

if yes, updating the data with the same identifier in the trusted storage node based on the private data; or

if no, storing the private data in the trusted storage node.

13. The method according to claim 11, wherein the trusted storage node stores a public key of the target node; and

the updating, based on the private data, private data that is of the target node and that is stored in a trusted storage node specifically comprises:

encrypting the private data by using the private key of the target node, and determining encrypted data; and

sending the encrypted data to the trusted storage node, so that the trusted storage node performs identity verification on the encrypted data based on the stored public key of the target node, and when the identity verification succeeds, updates the data that is of the target node and that is stored in the trusted storage node.

14. The method according to claim 10, wherein target nodes that support services in different regions are configured in the consortium blockchain by a participant of the consortium blockchain; and

before the determining a correspondence between an identifier of the target node and a

communication protocol address of the target node, the method further comprises:

determining an identity of a participant to which the target node belongs; and

determining the identifier of the target node based on the identity and a region in which the target node executes the service.

15. A method performed by a backup node in a consortium blockchain, comprising:

when detecting that a target node corresponding to the backup node goes down, determining an identifier of the target node, obtaining configuration data, and configuring the backup node based on the configuration data;

obtaining private data of the target node from a trusted storage node based on the identifier of the target node, wherein the private data comprises data that is of the target node and on which consensus reaching is to be performed; and

updating, based on a communication protocol address of the backup node, a correspondence stored in a smart contract in the consortium blockchain, and executing a service in place of the target node, wherein the correspondence stored in the smart contract is a correspondence that is between the identifier of the target node and a communication protocol address of the target node and that is prestored in the smart contract by the target node.

16. The method according to claim 15, wherein the updating, based on a communication protocol address of the backup node, a correspondence stored in a smart contract in the consortium blockchain specifically comprises:

determining the communication protocol address of the backup node; and

updating the communication protocol address of the target node in the correspondence that is between the identifier of the target node and the communication protocol address of the target node and that is stored in the smart contract to the communication protocol address of the backup node.

17. The method according to claim 15, wherein the trusted storage node stores a public key of the backup node; and

the obtaining private data of the target node from a trusted storage node specifically comprises:

encrypting identity information of the target node by using a private key of the backup node, and determining encrypted information;

sending an obtaining request that carries the encrypted information to the trusted storage node, so that the trusted storage node performs identity verification on the encrypted information based on the public key of the backup node; and

receiving private data returned by the trusted storage node after the identity verification succeeds.

18. The method according to claim 17, wherein target nodes that support services in different regions are configured in the consortium blockchain by a participant of the consortium blockchain; and

the method further comprises:

monitoring each target node and each backup node that are configured by a participant to which the backup node belongs;

when detecting that both a target node and a backup node in a same region go down, encrypting an identifier of the regional target node by using the private key of the backup node, and determining encrypted data;

sending an obtaining instruction that carries the encrypted data to a trusted storage node of the regional target node, so that the trusted storage node of the regional target node performs identity verification on the encrypted data based on the public key of the backup node;

receiving private data that is of the regional target node and that is returned by the trusted storage node of the regional target node, and obtaining the configuration data from a cloud server; and

updating, based on the communication protocol address of the backup node, the correspondence stored in the smart contract, executing the corresponding service in place of the target node, and executing a corresponding service in place of the regional target node.

19. A computer system, configured to implement a target node in a consortium blockchain and comprising a memory, a processor, and a computer program stored in the memory and capable of running on the processor, wherein when the processor executes the program, the following steps are implemented:

executing a service based on configured configuration data;

determining a correspondence between an identifier of the target node and a communication protocol address of the target node, and uploading the correspondence to a smart contract in the consortium blockchain for storage; and

determining private data of the target node based on a preset update cycle, and updating, based on the private data, private data that is of the target node and that is stored in a trusted storage node, wherein the private data comprises data that is of the target node and on which consensus

reaching is to be performed.

20. A computer system, applied to implement a backup node in a consortium blockchain and comprising a memory, a processor, and a computer program stored in the memory and capable of running on the processor, wherein when the processor executes the program, the following steps are implemented:

when detecting that a target node corresponding to the backup node goes down, determining an identifier of the target node, obtaining configuration data, and configuring the backup node based on the configuration data;

obtaining private data of the target node from a trusted storage node based on the identifier of the target node, wherein the private data comprises data that is of the target node and on which consensus reaching is to be performed; and

updating, based on a communication protocol address of the backup node, a correspondence stored in a smart contract in the consortium blockchain, and executing a service in place of the target node, wherein the correspondence stored in the smart contract is a correspondence that is between the identifier of the target node and a communication protocol address of the target node and that is prestored in the smart contract by the target node.

21. A computer-readable nonvolatile storage medium, wherein the storage medium stores a computer program, and when the computer program is executed by a processor, the method according to any one of claims 10 to 18 is implemented.