

METHOD AND APPARATUS FOR CONFIGURING LOCAL CONSENSUS AND COMPUTER-READABLE STORAGE MEDIUM

FIELD OF THE DISCLOSURE

[0001] The present disclosure relates to the field of block chain technology, and in particular, to a method for configuring a block chain-based local consensus, an apparatus for configuring a block chain-based local consensus, and a corresponding tangible computer-readable storage medium.

BACKGROUND

[0002] Block chain technology is a decentralization-based peer-to-peer network technology that combines cryptographic principles with consensus mechanisms to ensure data coherence and continuity of distributed nodes, thereby achieving the purpose for information immediate validation, traceability, tamper resistance, and block resistance, in turn to create a set of private, high-efficient, and secure shared value systems.

[0003] Block chains are generally classified into public block chains, consortium block chains, and private block chains according to the access rights. Among them, the public block chain refers to the block chain to which anyone can access according to an agreement and participate in the consensus; the consortium block chain refers to the block chain which is controlled by the preselected node during its consensus; the private block chain refers to a block chain whose all permissions are in one organization, and which is controlled by the organization arbitrarily.

[0004] A consensus algorithm refers to a set of protocols operated and followed by a plurality of nodes participated in the block chain, to ensure that the request operation (sometimes also comprising an execution result) submitted to the block chain can reach an agreement among the plurality of nodes in the block chain.

[0005] In the existing block chain technology, the consensus algorithms selected by different block chain platforms also have their own merits and differences. From the view of technical indicators, different consensus algorithms have large differences in indicators, such as system availability, extensibility and consensus confirmation speed, etc., and thus there is no perfect consensus algorithm to meet all scenarios. Therefore, in the practice of

the actual block chain, it is often necessary to select a suitable consensus algorithm according to the actual usage scenario and technical indicators.

SUMMARY

[0006] Since the block chain platforms currently in the market substantially support one single consensus algorithm, even if a consensus module is a pluggable consensus module, it allows only one consensus algorithm to run at the same time, so the flexibility of the existing block chain platforms is poor to be not able to satisfy multiple different service scenarios simultaneously. In addition, in some scenarios, in reality, a subset of nodes needs to be selected to compose a sub-chain to run some particular block chain services, and other irrelevant nodes are not allowed to receive messages, which cannot be achieved in the existing block chain platforms.

[0007] For the above-described problems, a first aspect of the present disclosure discloses a method for configuring a block chain-based local consensus, the method comprising:

[0008] implementing an initialization of a plurality of nodes; and

[0009] creating a local consensus instance for a set of nodes selected from the plurality of nodes.

[0010] The method for configuring a block chain-based local consensus according to the present disclosure can select, from the plurality of nodes, relevant nodes that are a part of a plurality of nodes, thereby establishing a local consensus instance among the selected nodes, in turn to ensure that the local consensus instance can select its desired consensus algorithm, so as to optimize the network and improve the network service quality.

[0011] In one embodiment in accordance with the present disclosure, creating a local consensus instance for a set of nodes selected from the plurality of nodes further comprises:

[0012] selecting the set of nodes from the plurality of nodes according to a service demand running on the local consensus instance to be created;

[0013] broadcasting a first transaction command for creating the local consensus instance to all the nodes in the block chain via one of the plurality of nodes; and

[0014] all the nodes in the block chain responding to the first transaction command to create the local consensus instance.

[0015] In this way, the administrator of the block chain can select the set of nodes from the plurality of nodes according to the service demand running on the local consensus instance to be created, so that the local consensus instance is established among the selected set of nodes, and other irrelevant nodes do not need to or are not able to participate in the local consensus instance. On one hand, this will ensure the security and privacy of communication among the set of nodes, on the other hand, it is also possible to ensure that the created local consensus instance is relatively free to select its desired consensus algorithm.

[0016] In one embodiment in accordance with the first transaction command comprises information indicating a node list of the set of nodes, information related to a local consensus identifier specified by an administrator of the block chain, and information related to a consensus algorithm used by the created local consensus instance.

[0017] In one embodiment in accordance with the present disclosure, the consensus algorithm comprises at least one of the following algorithms: proof of work consensus algorithm; proof of stake consensus algorithm; delegated proof of stake consensus algorithm; Practical Byzantine Fault Tolerance consensus algorithm; and/or BFT-RAFT consensus algorithm. Among them, proof of work (POW) consensus algorithm, such as used by Bitcoin, is mostly used in public block chains; proof of stake (POS) consensus algorithm, which is tried by Ethereum, is mostly used in public block chains; delegated proof of stake (DPOS) consensus algorithm which is a variant of POS consensus algorithm is mostly used in the public block chain; Practical Byzantine Fault Tolerance (PBFT) consensus algorithm is mostly used in the consortium block chains; and in 2014, a newly proposed distributed consensus algorithm--BFT-RAFT consensus algorithm, which is similar to PBFT, is mostly used in distributed systems and consortium block chains.

[0018] In one embodiment in accordance with the present disclosure, the information related to a consensus algorithm used by the created local consensus instance is determined based on the service demand running on the local consensus instance to be created and the consensus algorithm supported by the block chain.

[0019] In an embodiment in accordance with the present disclosure, the method further comprises one or more communication channels being created by the local consensus instance according to the type of a consensus algorithm used by the local consensus instance. In an embodiment in accordance with the present disclosure, the method further comprises a message processing thread for receiving and processing message transmitted from other nodes being initiated by the created local consensus instance.

[0020] In one embodiment in accordance with the present disclosure, the method further comprises:

[0021] broadcasting a second transaction command for deleting the local consensus instance to all the nodes in the block chain after completing the service demand running on the local consensus instance; and

[0022] all the nodes in the block chain responding to the second transaction command to delete the local consensus instance.

[0023] In this way, after completing the service demand running on the local consensus instance, the administrator of the block chain can delete the local consensus instance in time, thereby releasing the corresponding network resources, so as to utilize the block chain network resource optimally.

[0024] In one embodiment in accordance with the present disclosure, implementing an initialization of a plurality of nodes further comprises:

[0025] configuring network addresses and ports for the plurality of nodes;

[0026] generating node public and private keys; and

[0027] initiating and running a node block chain program.

[0028] Furthermore, the second aspect of the present disclosure discloses a tangible computer-readable storage medium, the storage medium comprising instructions that, when executed, cause a processor of the computer at least to:

[0029] implement an initialization of a plurality of nodes; and

[0030] create a local consensus instance for a set of nodes selected from the plurality of nodes.

[0031] In one embodiment in accordance with the present disclosure, to create a local consensus instance for a set of nodes selected from the plurality of nodes further comprises:

[0032] select the set of nodes from the plurality of nodes according to a service demand running on the local consensus instance to be created;

[0033] broadcast a first transaction command for creating the local consensus instance to all the nodes in the block chain via one of the plurality of nodes; and

[0034] all the nodes in the block chain responding to the second transaction command to delete the local consensus instance.

[0035] In one embodiment in accordance with the present disclosure, when executed, the instructions cause the processor of the computer further to:

[0036] broadcast a second transaction command for deleting the local consensus instance to all the nodes in the block chain after completing the service demand running on the local consensus instance; and

[0037] cause all the nodes in the block chain to respond to the second transaction command to delete the local consensus instance.

[0038] Finally, a third aspect of the present disclosure discloses an apparatus for configuring a block chain-based local consensus, the apparatus comprising:

[0039] an initialization module configured to implement an initialization of a plurality of nodes; and

[0040] a local consensus configuration module configured to create a local consensus instance for a set of nodes selected from the plurality of nodes.

[0041] In one embodiment in accordance with the present disclosure, the local consensus configuration module is further configured to:

[0042] select the set of nodes from the plurality of nodes according to a service demand running on the local consensus instance to be created;

[0043] broadcast a first transaction command for creating the local consensus instance to all the nodes in the block chain via one of the plurality of nodes; and

[0044] cause all the nodes in the block chain to respond to the first transaction command to create the local consensus instance.

[0045] In one embodiment in accordance with the present disclosure, the first transaction command comprises information indicating a node list of the set of nodes, information related to a local consensus identifier specified by an administrator of the block chain, and information related to a consensus algorithm used by the created local consensus instance.

[0046] In one embodiment in accordance with the present disclosure, the consensus algorithm comprises at least one of the following algorithms:

[0047] proof of work consensus algorithm;

[0048] proof of stake consensus algorithm;

[0049] delegated proof of stake consensus algorithm;

[0050] Practical Byzantine Fault Tolerance consensus algorithm; and/or

[0051] BFT-RAFT consensus algorithm.

[0052] In one embodiment in accordance with the present disclosure, the information related to a consensus algorithm used by the created local consensus instance is determined based on the service demand running on the local consensus instance to be created and the consensus algorithm supported by the block chain.

[0053] In one embodiment in accordance with the present disclosure, the local consensus configuration module is further configured to:

[0054] broadcast a second transaction command for deleting the local consensus instance to all the nodes in the block chain after completing the service demand running on the local consensus instance; and

[0055] cause all the nodes in the block chain to respond to the first transaction command to create the local consensus instance.

[0056] Method, apparatus, and corresponding computer-readable storage medium for configuring a block chain-based local consensus according to the present disclosure can select, from the plurality of nodes, the relevant nodes that are a part of a plurality of nodes, thereby establishing a local consensus instance among the selected nodes, in turn to

ensure that the local consensus instance can select its desired consensus algorithm, so as to optimize the network and improve the network service quality.

BRIEF DESCRIPTION OF THE DRAWINGS

[0057] The features, advantages, and other aspects of various embodiments of the present disclosure will become more apparent in conjunction with the drawings and the following detailed description, here illustrates several embodiments of the present disclosure by way of example and not limitation, in the drawings:

[0058] FIG. 1 is a schematic diagram of a network structure 100 of a block chain in the prior art;

[0059] FIG. 2 is a schematic diagram of a network structure 200 of a block chain according to the present disclosure;

[0060] FIG. 3 is a flowchart of a method 300 for configuring a block chain-based local consensus according to the present disclosure; and

[0061] FIG. 4 is a schematic diagram of an apparatus 400 for configuring a block chain-based local consensus according to the present disclosure; and

[0062] FIG. 5 illustrates a schematic diagram of a plurality of embodiments 500 for a local consensus implemented according to the present disclosure.

DETAILED DESCRIPTION

[0063] Various exemplary embodiments of the present disclosure are described in detail below with reference to the accompanying drawings. The flowchart and block diagrams in the figures illustrate the architecture, functionality, and operation of methods and systems in accordance with various embodiments of the present disclosure which are possibly implemented. It should be noted that each block of the flowchart or block diagrams, which may represent one module, a section of a program, or a portion of a code, may comprise one or more components for implementing executable instructions of the specified logical function in various embodiments. It should also be noted that in some implementations as alternatives, the functions noted in the blocks may occur in an order other than as noted in the drawings. For example, two blocks represented in succession may in fact be executed substantially in parallel, or they may sometimes be executed in

the reverse order, depending on the function involved. It should also be noted that each block of the flowcharts and/or block diagrams, and combinations of blocks in the flowcharts and/or block diagrams, can be implemented using dedicated hardware-based systems that perform the specified functions or operations, or it can be implemented using a combination of dedicated hardware and computer instructions.

[0064] The terms "comprising", "including", and the like, as used herein, should be construed as open-ended terms, i.e., "comprising/including but not limited to," indicating that other content may also be comprised. The term "based on" is "based at least in part on". The term "one embodiment" means "at least one embodiment"; the term "another embodiment" means "at least one other embodiment", and the like.

[0065] Techniques, methods, and devices known to those of ordinary skill in the relevant art may not be discussed in detail, but in an appropriate case, the techniques, methods and devices should be considered as part of the specification. For the connection between the units in the drawings is merely for convenience of description, which represents that at least the units at both ends of the connection communicate with each other, and is not intended to limit that the communication between unconnected units cannot be achieved.

[0066] FIG. 1 is a schematic diagram of a network structure 100 of a block chain in the prior art. As can be seen from FIG. 1, the block chain platform 100 includes, but is not limited to, terminals 101, 102, 103, 104, 105, 106, 107, 108, and 109 connected by a network, wherein the terminals 101, 103, 104 are connected to other block chain terminals through the wireless network, while the terminals 102, 105, 106, 107, 108, and 109 are connected to the other block chain terminals through the wired network. However, there are following technical problems in such a block chain platform, namely the block chain platforms currently in the market substantially support one single consensus algorithm, even if a consensus module is a pluggable consensus module, it allows only one consensus algorithm to run at the same time, so the flexibility of the existing block chain platforms is poor to be not able to satisfy multiple different service scenarios simultaneously. The actual demand corresponding to the above is that in some scenarios, in reality, a subset of nodes needs to be selected to compose a sub-chain to run some particular block chain service, and other irrelevant nodes are not allowed to receive messages, which cannot be achieved in the existing block chain platforms.

[0067] In addition, from the consensus algorithms selected by specific block chain platforms, they also have their own merits and differences, from the view of technical indicators, different consensus algorithms have large differences in indicators, such as system availability, extensibility, and consensus confirmation speed, etc., and thus there is no perfect consensus algorithm to meet all scenarios. Therefore, in the practice of the actual block chain, it is often necessary to select a suitable consensus algorithm according to the actual usage scenario and technical indicators.

[0068] As a result of research, researchers in the present disclosure have found that on a large block chain platform, relevant nodes that need to participate in a specific service can be selected from these nodes according to a service demand, and a local consensus instance can be established among these relevant nodes, which enables to ensure the security and privacy of communications among the local consensus instances. In addition, a consensus algorithm can be selected for the local consensus instance individually, such that the selection of the consensus algorithm running on the local consensus instance is more pertinent to be able to optimize the operation of the block chain.

[0069] The concept according to the present disclosure will be described below with reference to FIG. 2. For example, in the block chain platform shown in FIG. 1, a bank customer A (i.e., illustrated node 201) and an insurance customer B (illustrated node 206), and another insurance customer C (illustrated node 208) establish one sub-chain, i.e., a local consensus instance. At this time, the administrator of the block chain can specify an identifier of the local consensus instance (e.g., "reimbursement platform"), and select, from these nodes, the relevant nodes (i.e., nodes 201, 206, and 208, represented by block 220) which need to participate in the specific service for the local consensus instance, and select a corresponding consensus algorithm (e.g., a Practical Byzantine Fault Tolerance consensus algorithm) for the local consensus instance, however, those skilled in the art should appreciate that the identifier of the local consensus instance, the selected relevant nodes (i.e., nodes 201, 206, and 208), and the selected Practical Byzantine Fault Tolerance consensus algorithm here are all exemplary and non-limited, and other nodes, other consensus algorithms, and identifiers that specify other local consensus instances can also be selected when they meet the requirements. The administrator of the block chain then broadcasts a first transaction command for creating the local consensus instance to all the nodes in the block chain through any of the nodes in the block chain,

and all the nodes in the block chain respond to the first transaction command to create the local consensus instance. As a result, the local consensus instance according to the present disclosure has been created, and it can be seen that the first transaction command includes information indicating a node list of the set of nodes (i.e., information indicating to include the nodes 201, 206, and 208), information related to a local consensus identifiers specified by an administrator of the block chain (i.e., information related to for example, "reimbursement platform"), and information related to a consensus algorithm used by the created local consensus instance (i.e., for example, information related to Practical Byzantine Fault Tolerance consensus algorithm). A method for configuring a block chain-based local consensus according to the present disclosure can select from the plurality of nodes the relevant nodes that are a part of the plurality of nodes, thereby establishing a local consensus instance among the selected nodes, in turn to ensure that this local consensus instance can select its desired consensus algorithm, so as to optimize the network and improve the network service quality.

[0070] After creating the local consensus instance, the local consensus instance creates one or more communication channels according to the type of consensus algorithm (e.g., Practical Byzantine Fault Tolerance consensus algorithm) used by the local consensus instance. Subsequently, the created local consensus instance initiates a message processing thread for receiving and processing message transmitted from other nodes. After completing the service demand running on the local consensus instance, a second transaction command for deleting the local consensus instance is broadcasted to all the nodes in the block chain, and all the nodes in the block chain respond to the second transaction command to delete the local consensus instance. In this manner, after completing the service demand running on the local consensus instance, the administrator of the block chain can delete the local consensus instance in time, thereby releasing the corresponding network resources, and thus to utilize this block chain network resource optimally.

[0071] A method 300 for configuring a block chain-based local consensus according with the present disclosure is further described below with reference to FIG. 3, the method 300 includes the following steps, i.e., firstly, in the step 310 of the method, implementing an initialization of a plurality of nodes; then in the step 320 of the method, creating a local consensus instance for a set of nodes selected from the plurality of nodes.

In one embodiment, the step 320 of the method further includes selecting the set of nodes from the plurality of nodes according to a service demand running on the local consensus instance to be created; broadcasting a first transaction command for creating the local consensus instance to all the nodes in the block chain via one of the plurality of nodes; and all the nodes in the block chain responding to the first transaction command to create the local consensus instance. The first transaction command includes information indicating a node list of the set of nodes, information related to a local consensus identifier specified by an administrator of the block chain, and information related to a consensus algorithm used by the created local consensus instance. The method step 310 further comprises configuring network addresses and ports for the plurality of nodes; generating node public and private keys; and initiating and running a node block chain program.

[0072] The above-described consensus algorithm includes at least one of the following algorithms: proof of work consensus algorithm; proof of stake consensus algorithm; delegated proof of stake consensus algorithm; Practical Byzantine Fault Tolerance consensus algorithm; and/or BFT-RAFT consensus algorithm. Among them, proof of work (POW) consensus algorithm, such as used by Bitcoin, is mostly used in public block chains; proof of stake (POS) consensus algorithm, which is tried by Ethereum, is mostly used in public block chains; delegated proof of stake (DPOS) consensus algorithm which is a variant of POS consensus algorithm is mostly used in the public block chain; Practical Byzantine Fault Tolerance (PBFT) consensus algorithm is mostly used in the consortium block chains; and in 2014, a newly proposed distributed consensus algorithm--BFT-RAFT consensus algorithm, which is similar to PBFT, is mostly used in distributed systems and consortium block chains.

[0073] More specifically, the information relevant to a consensus algorithm used by the created local consensus instance is determined based on the service demand running on the local consensus instance to be created and the consensus algorithm supported by the block chain.

[0074] The above configuration method can be implemented by means of a tangible computer-readable storage medium, the storage medium comprising instructions that, when executed, cause a processor of the computer to at least implement an initialization of a plurality of nodes; and create a local consensus instance for a set of nodes selected

from the plurality of nodes. In addition, when executed, the instructions enable the processor of the computer to select the set of nodes from the plurality of nodes according to a service demand running on the local consensus instance to be created, broadcast a first transaction command for creating the local consensus instance to all the nodes in the block chain via one of the plurality of node, and all the nodes in the block chain responding to the second transaction command to delete the local consensus instance. In one embodiment according to the present disclosure, when executed, the instructions cause the processor of the computer further to broadcast a second transaction command for deleting the local consensus instance to all the nodes in the block chain after completing the service demand running on the local consensus instance; and cause all the nodes in the block chain to respond to the second transaction command to delete the local consensus instance.

[0075] Alternatively, the above configuration method can be implemented by a computer program product. The computer program product may comprise a computer-readable storage medium having computer readable program instructions thereon for carrying out aspects of the present disclosure. The computer-readable storage medium may be a tangible device that can hold and store instructions used by an instruction execution device. The computer-readable storage medium may be, but not limited to, an electrical storage device, a magnetic storage device, an optical storage device, an electromagnetic storage device, a semiconductor storage device, or any suitable combination of the above. More specific examples (non-exhaustive lists) of the computer-readable storage media comprise portable computer disks, hard disks, random access memory (RAM), read only memory (ROM), erasable programmable read only memory (EPROM or flash memory), static random access memory (SRAM), portable compact disk read-only memory (CD-ROM), digital versatile disk (DVD), memory stick, floppy disk, mechanical coding device, for example, punch-out cards or groove inner protrusion structures on which instructions are stored, as well as any suitable combination of the above. The computer-readable storage media as used here is not to be construed as an instantaneous signal per se, such as radio waves or other freely propagating electromagnetic waves, electromagnetic waves propagating through waveguides or other transmission media (e.g., optical pulses through optical fiber cables), or electrical signals propagating through the wired transmission.

[0076] FIG. 4 is a schematic diagram of an apparatus 400 for configuring a block chain-based local consensus according to the present disclosure. It can be seen from the figure that the apparatus 400 for configuring a block chain-based local consensus includes an initialization module 410 configured to implement an initialization of a plurality of nodes. In addition, the apparatus 400 for configuring a block chain-based local consensus further includes a local consensus configuration module 420 configured to create a local consensus instance for a set of nodes selected from the plurality of nodes. In addition, the local consensus configuration module 420 can be further configured to select the set of nodes from the plurality of nodes according to a service demand running on the local consensus instance to be created, broadcast a first transaction command for creating the local consensus instance to all the nodes in the block chain via one of the plurality of nodes; and cause all the nodes in the block chain to respond to the first transaction command to create the local consensus instance. The first transaction command includes information indicating a node list of the set of nodes, information related to a local consensus identifier specified by an administrator of the block chain, and information related to a consensus algorithm used by the created local consensus instance. The consensus algorithm includes at least one of the following algorithms: proof of work consensus algorithm, proof of stake consensus algorithm, delegated proof of stake consensus algorithm, Practical Byzantine Fault Tolerance consensus algorithm, and/or BFT-RAFT consensus algorithm. In one embodiment according to the present disclosure, the local consensus configuration module is further configured to broadcast a second transaction command for deleting the local consensus instance to all the nodes in the block chain after completing the service demand running on the local consensus instance; and cause all the nodes in the block chain to respond to the first transaction command to create the local consensus instance.

[0077] Method, apparatus, and corresponding computer-readable storage medium for configuring a block chain-based local consensus according to the present disclosure can select from the plurality of nodes the relevant nodes that are a part of a plurality of nodes, thereby establishing a local consensus instance among the selected nodes, in turn to ensure that the local consensus instance can select its desired consensus algorithm, so as to optimize the network and improve the network service quality.

[0078] An embodiment 500 of a plurality of local consensuses implemented by the proposed method according to the present disclosure is further described below in conjunction with FIG. 5. It can be seen from the figure that the transmission layer 510 is a peer to peer network connection layer which is responsible for maintaining the physical connections between nodes. The node layer 520 is used for maintaining the network address, the port, and the public and private keys information of each node and completing handshake, signature and the verification processing. Then, multiple channels can be created on the channel layer 530, among multiple nodes, only channels with the same ID can communicate with each other, each local consensus instance can dynamically create and delete channels, while the ID of each channel cannot be repeated, which means that the ID of each channel is unique. On the consensus algorithm layer, however, each consensus algorithm specifies a set of nodes, creates one or more channels to communicate with the corresponding channels of other nodes, and processes the received channel messages. In addition, the created consensus algorithms (e.g., first local consensus 541, second local consensus 542, third local consensus 543, and fourth local consensus 544) may be created, modified, and/or deleted dynamically.

[0079] In the view of the network establishment, the establishment of a local consensus instance includes two phases. In the first phase, an initialization of a plurality of nodes is implemented. Implementing an initialization of a plurality of nodes includes configuring the node network address and port, generating node public and private keys, and initiating and running a node block chain program. Then, in the second phase, a local consensus will be initiated. Initiating a local consensus, however, comprising determining the string ID of the local consensus, and the ID cannot be repeated; then determining the participated set of nodes and the consensus algorithm of the local consensus, it should be noted that the consensus algorithm must be a consensus algorithm supported by the platform; then, using the identity of the administrator to initiate one transaction command for creating a local consensus, meanwhile submitting the string ID, the set of nodes, and the consensus algorithm of the local consensus. The transaction is automatically created on the selected nodes after the network-wide conformation, and the local consensus instance is initiated; the newly created local consensus instances create one or more communication channels (channels) to communicate with each other, and the newly created local consensuses initiate message processing threads to receive and process the

messages transmitted from other nodes. Lastly, if the local consensus finishes a task and needs to stop, a transaction command for deleting the local consensus only needs to be initiated using the identity of the administrator.

[0080] Compared with the prior art, the method of the present invention has the advantages of the following aspects:

[0081] Firstly, the method can initiate / stop one new local consensus without redeployment or restarting, just needs to select the nodes to participate in the local consensus and select a suitable consensus algorithm, create the local consensus dynamically through the creating new local consensus command supported by the platform, and utilize the local consensus in time. When not needed, the specified local consensus is stopped by the stopping the local consensus command supported by the platform.

[0082] Secondly, the method can simultaneously support a plurality of local consensus which are running synchronically, and the local consensuses are completely isolated from each other logically and are not able to affect each other: i.e., local consensus are not visible to each other, so different local consensus can select completely different consensus algorithms arbitrarily, which has greater flexibility.

[0083] Finally, the local consensus supported by this method can be free to select a completely different set of nodes. The message of each local consensus is strictly limited to be passed among the specified set of nodes, other irrelevant nodes will not receive the consensus message, which has better privacy. It can be well satisfied that some nodes are running in a scenario that belongs to a small range of the block chain application, which has stronger data privacy and security.

[0084] It should be noted that although several apparatuses or sub-apparatus of the device mentioned in the above detailed description, such division is merely exemplary and not mandatory. In fact, according to the embodiments of the present disclosure, the features and functions of two or more above-described apparatus may be embodied in one apparatus. Conversely, the features and functions of one apparatus described above may be further divided into a plurality of apparatuses to embody.

[0085] The foregoing descriptions are merely optional embodiments of the present disclosure, and are not intended to limit the embodiments of the present disclosure. For

those skilled in the art, various modifications and changes may be made to the embodiments of the present disclosure. Any modification, equivalent replacement, improvement and the like made within the spirit and principle of the embodiments of the present disclosure should be included in the protection scope of the embodiments of the present disclosure.

[0086] Although the embodiments of the present disclosure have been described with reference to several specific embodiments, it should be understood that the embodiments of the present disclosure are not limited to the specific embodiments disclosed. The embodiments of the present disclosure are intended to cover various modifications and equivalent arrangements included within the spirit and scope of the appended claims. The scope of the following claims is to be accorded the broadest interpretation so as to encompass all such modifications and equivalent structures and functions.

CLAIMS

What is claimed is:

1. A method for configuring a plurality of block chain-based local consensuses to run synchronically, the method comprising:

implementing an initialization of a plurality of nodes;

creating a plurality of local consensus instances each for a specific set of nodes selected from the plurality of nodes according to a service demand running on each of the plurality of local consensus instances;

for each of the plurality of local consensus instances, selecting a type of consensus algorithm; and

creating, by each of the plurality of local consensus instances, a plurality of communication channels according to the type of consensus algorithm used by the corresponding local consensus instance, each communication channel having a channel ID, wherein only channels with the same channel ID can communicate with each other.

2. The method of claim 1, wherein creating the plurality of local consensus instances each for a specific set of nodes selected from the plurality of nodes further comprises:

for each of the plurality of local consensus instances, broadcasting a first transaction command for creating the local consensus instance to all the nodes in the block chain via one of the plurality of nodes; and

all the nodes in the block chain responding to each of the first transaction commands to create each of the plurality of local consensus instances.

3. The method of claim 2, wherein the first transaction command for each of the plurality of local consensus instances includes information indicating a node list of the set of nodes, information related to a local consensus identifier specified by an administrator of the block chain, and information related to a consensus algorithm used by each of the plurality of created local consensus instances.

4. The method according to claim 3, wherein the consensus algorithm comprises at least one of the following algorithms:

- proof of work consensus algorithm;
- proof of stake consensus algorithm;
- delegated proof of stake consensus algorithm;
- Practical Byzantine Fault Tolerance consensus algorithm; and/or
- BFT-RAFT consensus algorithm.

5. The method according to claim 3, wherein the information related to a consensus algorithm used by each of the plurality of created local consensus instances is determined based on the service demand running on each of the local consensus instances to be created and the consensus algorithms supported by the block chain.

6. The method of claim 1, wherein the method further comprises:
a message processing thread for receiving and processing message transmitted from other nodes being initiated by each of the plurality of created local consensus instances.

7. The method according to claim 1, wherein the method further comprises:
for each of the plurality of local consensus instances, broadcasting a second transaction command for deleting the local consensus instance to all the nodes in the block chain after completing the service demand running on the local consensus instance; and
all the nodes in the block chain responding to each of the second transaction commands to delete each of the plurality of local consensus instances.

8. The method of claim 1, wherein implementing an initialization of a plurality of nodes further comprises:

- configuring network addresses and ports for the plurality of nodes;
- generating node public and private keys; and
- initiating and running a node block chain program.

9. A tangible computer-readable storage medium for configuring a plurality of block chain-based local consensus to run synchronically, the storage medium comprising instructions that, when executed, cause a processor of the computer to at least:

- implementing an initialization of a plurality of nodes;
- creating a plurality of local consensus instances each for a specific set of nodes selected from the plurality of nodes according to a service command running on each of the plurality of local consensus instances;
- for each of the plurality of local consensus instances, selecting a type of consensus algorithm; and
- creating, by each of the plurality of local consensus instances, a plurality of communication channels according to the type of consensus algorithm used by the corresponding local consensus instance, each communication channel having a channel ID, wherein only channels with the same channel ID can communicate with each other.

10. The computer-readable storage medium of claim 9, wherein to create a plurality of local consensus instances each for a specific set of nodes selected from the plurality of nodes further comprises:

- for each of the plurality of local consensus instances, broadcast a first transaction command for creating the local consensus instance to all the nodes in the block chain via one of the plurality of nodes; and

- all the nodes in the block chain responding to each of the first transaction commands to create each of the plurality of local consensus instances.

11. The computer-readable storage medium of claim 9, wherein when executed, the instructions cause the processor of the computer further to:

- for each of the plurality of local consensus instances, broadcast a second transaction command for deleting the local consensus instance to all the nodes in the block chain after completing the service demand running on the local consensus instance; and

- cause all the nodes in the block chain to respond to each of the second transaction commands to delete each of the plurality of local consensus instances.

12. An apparatus for configuring a plurality of block chain-based local consensuses to run synchronically, the apparatus comprising:

- an initialization module configured to implement an initialization of a plurality of nodes; and

- a local consensus configuration module configured to:

create a plurality of local consensus instances each for a specific set of nodes selected from the plurality of nodes according to a service command running on each of the plurality of local consensus instances;

selecting a type of consensus algorithm for each of the plurality of local consensus instances; and

creating, by each of the plurality of local consensus instances, a plurality of communication channels according to the type of consensus algorithm used by the corresponding local consensus instance, each communication channel having a channel ID, wherein only channels with the same channel ID can communicate with each other.

13. The apparatus of claim 12, wherein the local consensus configuration module is further configured to:

for each of the plurality of local consensus instances, broadcast a first transaction command for creating the local consensus instance to all the nodes in the block chain via one of the plurality of nodes; and

cause all the nodes in the block chain to respond to each of the plurality of first transaction commands to create each of the plurality of local consensus instances.

14. The apparatus of claim 13, wherein the first transaction commands for each of the plurality of local consensus instances comprises information indicating a node list of the set of nodes, information related to a local consensus identifier specified by an administrator of the block chain, and information related to a consensus algorithm used by each of the plurality of created local consensus instances.

15. The apparatus of claim 14, wherein the consensus algorithm comprises at least one of the following algorithms:

proof of work consensus algorithm;

proof of stake consensus algorithm;

delegated proof of stake consensus algorithm;

Practical Byzantine Fault Tolerance consensus algorithm; and/or

BFT-RAFT consensus algorithm.

16. The apparatus of claim 14, wherein the information related to a consensus algorithm used by each of the plurality of created local consensus instances is determined based on the service demand running on each of the local consensus instances to be created and the consensus algorithms supported by the block chain.

17. The apparatus of claim 12, wherein the local consensus configuration module is further configured to:

for each of the plurality of local consensus instances, broadcast a second transaction command for deleting the local consensus instance to all the nodes in the block chain after completing the service demand running on the local consensus instance; and

cause all the nodes in the block chain to respond to each of the second transaction commands to delete each of the plurality of local consensus instances.