METHODS AND APPARATUSES FOR PROCESSING TRANSACTIONS BASED ON DEDICATED BLOCKCHAIN NODE DEVICE

TECHNICAL FIELD

[0001] The present disclosure relates the field of information technologies, and in particular to methods and apparatuses for processing transactions based on a dedicated blockchain node device.

BACKGROUND

[0002] Blockchain technology (also referred to as a distributed ledger technology) is a decentralized distributed database technology having characteristics such as openness, transparency, immutability and trustability, and thus it is applicable to many application scenarios with high demands for data reliability. At present, a transaction is usually processed by a Central Processing Unit (CPU) of a node (for example, performing consensus for a to-be-executed transaction, executing a transaction and writing the executed transaction into a blockchain and the like).

[0003] Based on the existing technology, a method of processing a blockchain transaction with a higher efficiency is needed.

SUMMARY

[0004] In order to solve the lower efficiency of the existing method of processing a blockchain transaction, the embodiments of the present disclosure provide methods and apparatuses for processing transactions based on a dedicated blockchain node device. The following technical solution is adopted.

[0005] According to a first aspect of embodiments of the present disclosure, provided is a method of processing a transaction based on a dedicated blockchain node device. The dedicated blockchain node device includes a Central Processing Unit (CPU) and a smart network card and is any node of a blockchain network. The method includes: when the dedicated blockchain node device accepts a transaction through the smart network card, writing, by the smart network card, the accepted transaction into a network card cache; performing, by the smart network card, consensus interaction with other nodes of the blockchain network based on the transaction in the network card cache in a process that the dedicated blockchain node device participates in a blockchain consensus; determining and sending, by the smart network card, a to-be-executed transaction list to the CPU based on a result of the consensus; and executing, by the CPU, transaction based on the to-be-executed transaction list, and completing transaction chaining.

[0006] According to a second aspect of embodiments of the present disclosure, provided is a method of processing a transaction based on a dedicated blockchain node device, which is applied to a smart network card of the dedicated blockchain node device. The dedicated blockchain node device further includes a Central Processing Unit (CPU) and is any node of a blockchain network. The method includes: when the dedicated blockchain node device accepts a transaction through the smart network card, writing the accepted transaction into a network card cache; performing consensus interaction with other nodes of the blockchain network based on the transaction in the network card cache in a process that the dedicated blockchain node device participates in a blockchain consensus; and determining and sending a to-be-executed transaction list to the CPU based on a result of the consensus.

[0007] According to a third aspect of embodiments of the present disclosure, provided is another method of processing a transaction based on a dedicated blockchain node device, which is applied to a Central Processing unit (CPU) of the dedicated blockchain node device. The dedicated blockchain node device further includes a smart network card and is any node of a blockchain network. The method includes: executing, by the CPU, transaction based on a to-be-executed transaction list sent by the smart network card, and completing transaction chaining.

[0008] According to a fourth aspect of embodiments of the present disclosure, provided is an apparatus for processing a transaction based on a dedicated blockchain node device, which is applied to a smart network card of the dedicated blockchain node device. The dedicated blockchain node device further includes a Central Processing Unit (CPU) and is any node of a blockchain network. The apparatus includes: a caching module, configured to, when the dedicated blockchain node device accepts a transaction through the smart network card, write the accepted transaction into a network card cache; a consensus interacting module, configured to perform consensus interaction with other nodes of the blockchain network based on the transaction in the network card cache in a process that the dedicated blockchain node device participates in a blockchain consensus; and a sending module, configured to determine and send a to-be-executed transaction list to the CPU based on a result of the consensus.

[0009] According to a fifth aspect of embodiments of the present disclosure, provided is an apparatus for processing a transaction based on a dedicated blockchain node device, which is applied to a Central Processing unit (CPU) of the dedicated blockchain node device. The dedicated blockchain node device further includes a smart network card and is any node of a blockchain network. The apparatus includes: a chaining executing module, configured to execute transaction based on a to-be-executed transaction list sent by the smart network card

and complete transaction chaining.

[0010] In the technical solutions according to the embodiments of the present disclosure, the dedicated blockchain node device includes a CPU and a smart network card, the smart network card is a network card having a built-in processor or microprocessor and can perform data computation and processing. The smart network card can replace the CPU to actually participate in a blockchain consensus on behalf of the dedicated blockchain node device, and send a to-be-executed transaction list to the CPU for transaction execution and transaction chaining based on a result of the consensus.

[0011] According to the embodiments of the present disclosure, the following technical effects can be achieved by transferring operations such as blockchain consensus needing frequent network interaction with other nodes from the CPU to the smart network card:

[0012] 1. The smart network card is specifically in charge of blockchain consensus, while the CPU focuses on transaction execution and transaction chaining. In this case, the operation efficiency of the dedicated blockchain node device serving as a node can be improved, and the CPU can also execute more transactions within a unit time, thereby improving throughput.

[0013] 2. In a blockchain consensus process, the network traffic does not need to flow through the CPU but is received by the smart network card for direct processing and feedback. Thus, the CPU does not participate in the consensus process, so that the feedback delay of the dedicated blockchain node device in the consensus process is reduced.

[0014] It should be understood that the above general descriptions and the subsequent detailed descriptions are merely examples and illustrative, and are not intended to limit the embodiments of the present disclosure.

[0015] In addition, any of the embodiments of the present disclosure is not needed to achieve all effects above-mentioned.

BRIEF DESCRIPTION OF THE DRAWINGS

[0016] To describe the technical solutions in the embodiments of the present disclosure or in the existing technology more clearly, drawings needed in descriptions of the embodiments of the present disclosure or the existing technology will be briefly introduced below. It is clear that the drawings described below are merely some embodiments of the present disclosure and other drawings can also be obtained by those of ordinary skill in the art based on these drawings.

[0017] FIG. 1 is a structural schematic diagram of a dedicated blockchain node device according to embodiments of the present disclosure.

[0018] FIG. 2 is a structural schematic diagram of a blockchain system according to

embodiments of the present disclosure.

[0019] FIG. 3 is a flowchart of a method of processing a transaction based on a dedicated blockchain node device according to embodiments of the present disclosure.

[0020] FIG. 4 is a flowchart of a method of synchronizing data based on a dedicated blockchain node device according to embodiments of the present disclosure.

[0021] FIG. 5 is a flowchart of a method of transferring a transaction based on a dedicated blockchain node device according to embodiments of the present disclosure.

[0022] FIG. 6 is a flowchart of a method of identifying a replay transaction based on a dedicated blockchain node device according to embodiments of the present disclosure.

[0023] FIG. 7 is a flowchart of a method of identifying a to-be-filtered transaction based on a dedicated blockchain node device according to embodiments of the present disclosure.

[0024] FIG. 8 is a structural schematic diagram of an apparatus for processing a transaction based on a dedicated blockchain node device according to embodiments of the present disclosure.

[0025] FIG. 9 is a structural schematic diagram of an apparatus for processing a transaction based on a dedicated blockchain node device according to embodiments of the present disclosure.

[0026] FIG.10 is a structural schematic diagram of an apparatus for synchronizing data based on a dedicated blockchain node device according to embodiments of the present disclosure.

[0027] FIG.11 is a structural schematic diagram of an apparatus for synchronizing data based on a dedicated blockchain node device according to embodiments of the present disclosure.

[0028] FIG. 12 is a structural schematic diagram of an apparatus for transferring a transaction based on a dedicated blockchain node device according to embodiments of the present disclosure.

[0029] FIG. 13 is a structural schematic diagram of an apparatus for identifying a replay transaction based on a dedicated blockchain node device according to embodiments of the present disclosure.

[0030] FIG. 14 is a structural schematic diagram of an apparatus for identifying a to-be-filtered transaction based on a dedicated blockchain node device according to embodiments of the present disclosure.

[0031] FIG. 15 is a structural schematic diagram of a computer device for configuring a method of embodiments of the present disclosure.

DETAILED DESCRIPTION OF THE EMBODIMENTS

[0032] In the early stage of development of the blockchain technology, users mostly add their own personal computer (PC) and laptop computer, and the like into a blockchain network to become a blockchain node in the blockchain network. At this time, the stage can be called 1.0 architecture era of the blockchain network, in which behaviors of users to participate in the blockchain network are autonomous and the users also need to perform autonomous maintenance, for example, perform maintenance and configuration, and so on for their devices, for example, PC, participating in the blockchain network. Along with continuous development of the blockchain technology, especially along with increasing needs of users for infrastructures with high performance and high availability, the blockchain network develops into 2.0 architecture era based on a cloud service. In the 2.0 architecture era, cloud service providers provide the infrastructures with high performance and high availability to the users based on higher-performance servers and cloud computing, so as to configure and form blockchain nodes needed by the users. To satisfy the needs of users for privatization, security, and the like of the blockchain network, architecture needs to be upgraded further to the blockchain network, thereby realizing 3.0 architecture era based on a dedicated blockchain node device (blockchain all-in-one machine).

[0033] Software and hardware integration can be realized for the dedicated blockchain node device. When a dedicated blockchain node device is published, a provider not only provides hardware devices of the dedicated blockchain node device to users but also provides software configurations for realizing deep optimization of the hardware devices integrated into the dedicated blockchain node device, thereby realizing the above-mentioned software-hardware integration.

[0034] Hardware optimization can be realized for the dedicated blockchain node device. For example, a dedicated smart contract processing chip can be deployed on the dedicated blockchain node device. For example, the smart contract processing chip can be a Field Programmable Gate Array (FPGA) chip, or another type of chip to improve a processing efficiency for a smart contract. A hardware root-of-trust key can be deployed on the smart contract processing chip, for example, the hardware root-of-trust key can be pre-programmed by the provider into the smart contract processing chip and the provider can also know a public key corresponding to the hardware root-of-trust key (for example, the public key is disclosed). Therefore, the smart contract processing chip can send negotiation information to the provider and sign the negotiation information by using the hardware root-of-trust key, so that the provider can verify the signature based on the corresponding public key; and, after the

signature is successfully verified, it is ensured that the smart contract processing chip and the provider obtain the same key through negotiation based on the above-mentioned negotiation information respectively. The negotiated key can include a file deployment key, and thus the provider can encrypt and transmit a binary image file needed by the blockchain node to the smart contract processing chip based on the file deployment key, and the smart contract processing chip can decrypt and deploy the binary image file based on the file deployment key. The negotiated key can include a service secret deployment key, and thus the provider can encrypt and transmit a node private key, a service root key, and the like of the blockchain node to the smart contract processing chip based on the service secret deployment key, and the smart contract processing chip can obtain and deploy the node private key, the service root key, and the like based on the service secret deployment key to satisfy the privacy transaction needs in a blockchain scenario. For example, the node private key corresponds to a node public key, and thus a client device can perform encrypted transmission for a blockchain transaction by using the node public key, and the blockchain node can perform decryption through the node private key. The service root key is a symmetric key that can be used to perform encrypted storage for service data such as contract codes and values of contract statuses. The service root key may not be directly used, and the smart contract processing chip can perform encryption and decryption through a derivation key of the service root key to reduce a security risk of the service root key. Through reliable management for the node private key and the service root key (or its derivation key), data is always in an encrypted state unless processed by the smart contract processing chip. Therefore, the smart contract processing chip actually forms a Trusted Execution Environment (TEE) of hardware on the dedicated blockchain node device, so as to ensure the data needing privacy protection such as transactions, contract codes, and contract statuses not to be leaked.

[0035] For another example, a smart network card can be deployed on the dedicated blockchain node device. In addition to realizing functions of a traditional network card, the smart network card can also replace or assist a CPU of the dedicated blockchain node device to perform partial functions so as to offload computation of the CPU. Especially, the operations with intensive network I/O can be transferred from the CPU to the smart network card to perform, so that the CPU can process more computation-intensive operations, such as transaction executing and storage processing. Compared with other components (e.g., CPU) in the dedicated blockchain node device, the smart network card is closer to the network on both physical level and logical level, so the smart network card always fetches data transmitted in the network preferentially. Therefore, with no storage access or a small amount of storage access is involved, the smart network card can process the data with a relatively

higher processing efficiency, a relatively smaller delay and a relatively larger throughput, so as to achieve a higher performance benefit with a relatively lower cost. For example, in a consensus algorithm, there is almost no need to access storage except in the cases of a change of the network state, addition and deletion of the node, a change of consensus configuration, and the like. Therefore, the consensus operation can be completed by the smart network card and only need to inform the CPU of a consensus result. Therefore, the CPU is not needed to directly participate in the consensus process, thereby significantly improving consensus efficiency. Similarly, a similar effect can be achieved in transferring transactions by the smart network card and achieving block synchronization by the smart network card on a newly-added blockchain node and the like, which will not be repeated herein. Furthermore, after receiving transactions, the smart network card can identify or filter out a replay transaction by comparing the received transactions with historical transactions, for example, comparing data fields such as sender information of transaction, destination address, time stamp, and hash value and the like. The smart network card can also perform content analysis for the received transactions, so as to filter out illegal transactions or predefined undesired transactions, or the like as a supplementation to layer-2 or layer-3 packet filtering implemented by a switch.

[0036] For another example, a cryptographic acceleration card which is also called a high-speed cryptographic card can be deployed on the dedicated blockchain node device. The cryptographic acceleration card can realize total encrypted memory, defend against side channel attacks by hardware reinforcement, and also realize physical protection against approaches such as probe and laser, having very high security. For example, the cryptographic acceleration card used on the dedicated blockchain node device can have level-2 qualification from the state cryptography administration, level-3 qualification from the state cryptography administration, or the like. When the cryptographic acceleration card is deployed, the hardware roof-of-trust key described above can be maintained in the cryptographic acceleration card, and the cryptographic acceleration card can perform a signature operation based on the hardware roof-of-trust key and replace or assist the smart contract processing chip to complete the operations such as key negotiation described above. Similarly, the cryptographic acceleration card can be used to maintain a public key so that the cryptographic acceleration card can realize an operation of signature verification based on the maintained public key. In short, at least a part of operations relating to key management, encryption and decryption, signature verification, and the like on the dedicated blockchain node device can be handed over to the cryptographic acceleration card, so that very high security can be obtained and task offloading can also be realized for the CPU of the dedicated blockchain node device

or the above-mentioned smart contract processing chip, thereby improving the processing efficiency.

[0037] Software optimization can be realized for the dedicated blockchain node device. For example, a certificate authority service can be built in the dedicated blockchain node device to realize automatic certificate issuing, node identity authentication, automatic blockchain construction and automatic adding of blockchain nodes, thereby realizing plug and play of the dedicated blockchain node device. In this case, a user can realize fast deployment of the dedicated blockchain node device. In addition to quickly constructing a private blockchain network between a plurality of dedicated blockchain node devices, the dedicated blockchain node device can integrate a standardized on-cloud service interface to enable the dedicated blockchain node device to automatically connect to on-cloud service, thereby realizing hybrid deployment between the dedicated blockchain node device and the cloud-deployed blockchain node to construct a hybrid blockchain network. The dedicated blockchain node device can also integrate a standardized cross-chain service interface to enable the dedicated blockchain node device to realize cross-chain services based on a standardized cross-chain protocol or standardized cross-chain service, thereby greatly expanding the application scenarios of the dedicated blockchain node device, and satisfying the cross-chain needs of users. For example, cross-chain data interaction between different blockchain networks is achieved, and for another example, cross-chain data interaction between the blockchain network and an off-chain computing node and the like is achieved (for example, the off-chain computing node shares a computation task for a blockchain node), and the like.

[0038] The technical solution according to each embodiment of the present disclosure will be described in detail below in combination with the drawings.

[0039] FIG. 1 is a structural schematic diagram of a dedicated blockchain node device according to embodiments of the present disclosure. As shown in FIG. 1, the dedicated blockchain node device includes a Central Processing Unit (CPU) and a smart network card. Different from a traditional network card only supporting network communication, the smart network card of the present disclosure is a network card with a built-in processor or microprocessor (e.g., an ARM processor), and thus has a certain data computing and processing capability. Certainly, it can be understood that the dedicated blockchain node device as a node of a blockchain network also includes a memory not shown in FIG. 1. The memory is used to store a blockchain itself, or the memory is also used to store a state database for an Ethereum blockchain.

[0040] In addition, the dedicated blockchain node device can further include other hardware, such as a smart contract processing chip and a cryptographic acceleration card.

[0041] FIG. 2 is a structural schematic diagram of a blockchain system according to embodiments of the present disclosure. The blockchain system includes a blockchain network. In the blockchain network, at least one node can serve as a dedicated blockchain node device. That is, the dedicated blockchain node device as a node can interact with other nodes (common nodes) which are not dedicated blockchain node devices, or can interact with other nodes which are also dedicated blockchain node devices. For convenience of description, the technical solutions will be described with a single dedicated blockchain node device herein.

[0042] FIG. 3 is a flowchart of a method of processing a transaction based on a dedicated blockchain node device according to embodiments of the present disclosure. The method includes the following steps:

[0043] At step S300, when the dedicated blockchain node device accepts a transaction through a smart network card, the smart network card writes the accepted transaction into a network card cache.

[0044] It is well known that a blockchain network usually processes a transaction in three stages, i.e. a transaction acceptance stage, a consensus stage and a transaction execution and chaining stage. Firstly, the above-mentioned three stages will be briefly introduced herein.

[0045] In the transaction acceptance stage, each node in the blockchain network accepts and caches a transaction to prepare for subsequent consensus and transaction execution. Specifically, a client device constructs a transaction and submits the transaction to any node in the blockchain network, so that the node directly accepts and caches the transaction, and broadcasts the transaction to the entire network to enable other nodes to indirectly accept the transaction. The dedicated blockchain node device accepting the transaction through the smart network card at step 300 can refer to that the dedicated blockchain node device directly accepts the transaction, or the dedicated blockchain node device indirectly accepts the transaction. In short, regardless of direct acceptance or indirect acceptance, it is important that most nodes in the blockchain network will cache the same transaction. It can be understood that in an actual application, a small number of nodes may not receive the transaction due to shutdown or network problem, which, however, will not affect the operation of a distributed blockchain network.

[0046] In the consensus stage, various nodes of the blockchain network perform interaction based on a consensus algorithm (e.g., Byzantine fault tolerance algorithm) to achieve message consistency, that is, to reach a consensus on which accepted transactions are to be executed this time. It can be understood that in the actual application, not every node participates in the consensus. However, because the consensus algorithm has fault tolerance, a consensus result will not be affected.

[0047] In the transaction execution and chaining stage, each node will fetch a transaction from the cache for execution based on the consensus result, and the fetched transaction is then packaged into a block and written into the blockchain (i.e., transaction chaining) after execution.

[0048] The cache in above-mentioned introduction usually refers to a cache of a CPU of a node. In the embodiments of the present disclosure, because a smart network card is configured for the node, the cache can include a network card cache and a CPU cache. In the transaction acceptance stage, the smart network card writes a transaction into the network card cache so as to perform subsequent blockchain consensus based on the transaction in the network card cache.

[0049] At step S302, when the dedicated blockchain node device participates in the blockchain consensus, the smart network card performs consensus interaction with other nodes of the blockchain network based on the transaction in the network card cache.

[0050] In the consensus stage, the smart network card replaces the CPU to actually participate in the consensus and perform consensus interaction with other nodes. Network traffic generated in the consensus process directly enters the smart network card, and the smart network card processes the network traffic based on the consensus algorithm and directly feeds back a processing result. Therefore, the network traffic will not flow through the CPU.

[0051] At step S304, the smart network card determines a to-be-executed transaction list based on a result of the consensus, and sends the to-be-executed transaction list to the CPU.

[0052] It is well known that the blockchain network performs consensus again and again according to a certain consensus trigger condition. For example, the consensus trigger condition can be that one consensus is performed every ten minutes, which means that the blockchain network determines one batch of to-be-executed transactions every ten minutes, and packages this batch of to-be-executed transactions into a block for chaining. The consensus result actually refers to those to-be-executed transactions determined in one blockchain consensus.

[0053] It should be noted herein that an access operation for blockchain storage (the blockchain and the state database) generally needs to be performed by the CPU for efficiency. In an actual application, Practical Byzantine Fault Tolerance (PBFT) algorithm is usually adopted, but the blockchain storage does not need to be accessed (the blockchain and the state database) for a host change operation involved in the PBFT algorithm process. Therefore, the CPU neither actually participates in the consensus process nor adversely affects efficiency of storage access.

[0054] Specifically, the to-be-executed transaction list described at step S304 can be a list formed by the to-be-executed transactions determined by the present blockchain consensus, or can be a list formed by transaction identifiers (e.g., transaction hashes) of the to-be-executed transactions determined by the present blockchain consensus.

[0055] When the to-be-executed transaction list includes transaction hashes, the smart network card also needs to send the to-be-executed transactions to the CPU. In an actual application, the smart network card can send the accepted transactions to the CPU while writing the accepted transactions into the network card cache in the transaction acceptance stage. The CPU writes the received transactions into the CPU cache.

[0056] At step S306, the CPU executes transaction based on the to-be-executed transaction list and completes transaction chaining.

[0057] In embodiments of the present disclosure, in the process that the dedicated blockchain node device participates in the blockchain consensus, the smart network card sends consensus configuration modification information to the CPU if receiving the consensus configuration modification information. The CPU can modify a consensus configuration locally stored in the dedicated blockchain node device based on the consensus configuration modification information.

[0058] Because the blockchain storage needs to be accessed for the operation of modifying the consensus configuration (for example, adding or deleting nodes; for another example, determining a capacity size of one batch of to-be-executed transactions, i.e. a batch size, in one consensus), the operation needs to be performed by the CPU for efficiency of storage access. Normally, if the modification of consensus configuration is not involved in the consensus process, the CPU will not participate in the consensus.

[0059] In addition, execution results generated by the CPU by executing transaction based on the to-be-executed transaction list are sent to the smart network card, and these execution results are reference information needed by the smart network card for participating in the next consensus. For example, the CPU executes a transaction initiated by an account A, which consumes 100 GAS of the account A with only 50 GAS left in the account A. This means that the account A cannot provide sufficient GAS for other transactions initiated by the account A. In this case, the CPU sends this execution result to the smart network card, and the smart network card will not tend to determine other transactions initiated by the account A as to-be-executed transactions in the next consensus.

[0060] According to the method shown in FIG. 3, the dedicated blockchain node device includes a CPU and a smart network card, and the smart network card is a network card having a built-in processor or microprocessor and can perform data computing and processing.

The smart network card replaces the CPU to actually participate in a blockchain consensus on behalf of the dedicated blockchain node device, and sends a to-be-executed transaction list to the CPU for transaction execution and transaction chaining based on a result of the consensus.

[0061] According to the embodiments of the present disclosure, the following technical effects can be achieved by transferring operations such as blockchain consensus needing frequent network interaction with other nodes from the CPU to the smart network card:

[0062] 1. The smart network card is specifically in charge of blockchain consensus, while the CPU focuses on transaction execution and transaction chaining. In this case, the operation efficiency of the dedicated blockchain node device serving as a node can be improved, and the CPU can also execute more transactions within a unit time, thereby improving throughput.

[0063] 2. In a blockchain consensus process, the network traffic does not need to flow through the CPU but is received by the smart network card for direct processing and feedback. Thus, the CPU does not participate in the consensus process, so that the feedback delay of the dedicated blockchain node device in the consensus process is reduced.

[0064] FIG. 4 is a flowchart of a method of synchronizing data based on a dedicated blockchain node device according to embodiments of the present disclosure. The method includes the following steps:

[0065] At step S400, a smart network card inquires other nodes about whether there is to-be-synchronized block data.

[0066] Generally, when there is a newly-added node in a blockchain network, the node needs to pull block data from other nodes for synchronization. In addition, in other cases, sometimes a node also needs to pull block data from other nodes for synchronization, which will not be repeated herein.

[0067] To determine whether to perform data synchronization currently, the node usually needs to frequently inquire other nodes about whether there is to-be-synchronized block data, which will involve frequent I/O operations. In embodiments of the present disclosure, the smart network card is in charge of inquiring other nodes about whether there is to-be-synchronized block data, so as to offload this part of operation burden of a CPU, thereby improving the processing efficiency of the CPU.

[0068] At step S402, if determining that there is the to-be-synchronized block data, the smart network card pulls the to-be-synchronized block data from other nodes.

[0069] After determining that there is to-be-synchronized block data in other nodes, the node needs to pull the to-be-synchronized block data from other nodes. This means that the node needs to maintain a process of pulling data from network and take time to wait for completion of data pulling. If this part of operation is handed over to the smart network card

for execution, operation burden can also be offloaded for the CPU, thereby improving the operation efficiency of the CPU.

[0070] At step S404, the smart network card provides the to-be-synchronized block data to the CPU.

[0071] Specifically, the smart network card can directly send the to-be-synchronized block data to the CPU, or writes the to-be-synchronized block data into a public cache between the smart network card and the CPU, and the CPU can fetch the to-be-synchronized block data from the public cache. For example, the CPU can fetch the to-be-synchronized block data from the public cache for data synchronization in an idle time.

[0072] At step S406, the CPU completes data synchronization based on the to-be-synchronized block data.

[0073] After obtaining the to-be-synchronized block data, the smart network card sends the to-be-synchronized block data to the CPU. At this time, the CPU only needs to write the to-be-synchronized block data into a local blockchain of the node without consuming excessive resource and time.

[0074] According to the method shown in FIG. 4, a dedicated blockchain node device includes a CPU and a smart network card, and the smart network card is a network card having a built-in processor or microprocessor and can perform data computing and processing. The smart network card can replace the CPU to actually participate in a blockchain consensus on behalf of the dedicated blockchain node device, and sends a to-be-executed transaction list to the CPU for transaction execution and transaction chaining based on a result of the consensus.

[0075] According to the embodiments of the present disclosure, the following technical effects can be achieved by transferring operations such as blockchain consensus needing frequent network interaction with other nodes from the CPU to the smart network card:

[0076] 1. The smart network card is specifically in charge of blockchain consensus, while the CPU focuses on transaction execution and transaction chaining. In this case, the operation efficiency of the dedicated blockchain node device serving as a node is improved, and the CPU can also execute more transactions within a unit time, thereby improving throughput.

[0077] 2. In a blockchain consensus process, the network traffic does not need to flow through the CPU but is received by the smart network card for direct processing and feedback. Thus, the CPU does not participate in the consensus process, so that the feedback delay of the dedicated blockchain node device in the consensus process is reduced.

[0078] FIG. 5 is a flowchart of a method of transferring a transaction based on a dedicated blockchain node device according to embodiments of the present disclosure. The method

includes the following steps:

[0079] At step S500, when the dedicated blockchain node device accepts a transaction through a smart network card, the smart network card determines other to-be-transferred nodes.

[0080] At step S502, the smart network card transfers the transaction to the to-be-transferred nodes.

[0081] In the transaction acceptance stage, in addition to transferring the accepted transaction, the smart network card can also write the accepted transaction into a network card cache to implement the method shown in FIG. 3.

[0082] Generally, before transferring and caching the accepted transaction, the smart network card can perform legality verification for the transaction. The legality verification usually involves some legality items to be verified based on a blockchain protocol (for example, signature verification of an account initiating a transaction).

[0083] If the legality verification fails, the smart network card generally does not transfer the transaction.

[0084] The smart network card can discard a transaction that fails in legality verification. In addition, it should be noted that in an actual application, the smart network card can also retain a transaction that fails in legality verification, and write the transaction into the network card cache and send the transaction to a CPU for the following reason: when a transaction fails to pass a legality verification performed by a node, it does not necessarily mean that the transaction is illegal but can mean that the transaction loses part of data in network transmission, or an error can occur to the verification of the node. For a blockchain network as a distributed database, wrong transaction verification results obtained by a small number of nodes shall not affect the execution of the transaction. Therefore, even if a transaction fails to pass the legality verification performed by a node, the node can still retain the transaction which can be subsequently determined as a to-be-executed transaction by the entire network through consensus in the consensus stage.

[0085] In addition, in embodiments of the present disclosure, if a transaction passes the legality verification performed by the smart network card , the smart network card can mark the transaction. Further, the smart network card can send the marked transaction to the CPU. Generally, the CPU can determine whether the transaction passes legality verification before writing the transaction into a CPU cache. If the transaction has a mark, it indicates that the smart network card already successfully performs verification, and the CPU does not need to perform verification again. If the transaction does not have a mark, it indicates that the smart network card unsuccessfully performs verification, and the CPU can choose to record that the

transaction fails to pass the verification, or to perform verification on the transaction again.

[0086] According to the method shown in FIG. 5, a dedicated blockchain node device includes a CPU and a smart network card, and the smart network card is a network card having a built-in processor or microprocessor and can perform data computing and processing. The smart network card can replace the CPU to perform transaction transferring in the transaction acceptance stage.

[0087] According to the embodiments of the present disclosure, the following technical effects can be achieved by transferring operations such as transaction transferring needing frequent network interaction with other nodes from the CPU to the smart network card:

[0088] 1. The smart network card is specifically in charge of transaction transferring to offload this part of operation burden for the CPU. In this case the operation efficiency of the dedicated blockchain node device serving as a node is improved, and the CPU can also execute more transactions within a unit time, thereby improving throughput.

[0089] 2. In a blockchain transferring process, the network traffic does not need to flow through the CPU but is received by the smart network card for direct processing and feedback so that the feedback delay of the dedicated blockchain node device is reduced.

[0090] FIG. 6 is a flowchart of a method of identifying a replay transaction based on a dedicated blockchain node device according to embodiments of the present disclosure. The method includes the following steps:

[0091] At step S600, when the dedicated blockchain node device accepts a transaction through a smart network card, the smart network card compares currently accepted transaction with historically-accepted transactions in a network card cache.

[0092] At step S602, if a comparison result indicates that there is a historically-accepted transaction same as the currently accepted transaction, the smart network card determines the currently accepted transaction as a replay transaction.

[0093] In embodiments of the present disclosure, the smart network card can compare the currently accepted transaction with all historically-accepted transactions one by one to determine whether the currently accepted transaction is a replay transaction.

[0094] In addition, considering that some historically-accepted transactions are too old to be replayed, the currently accepted transaction can be compared with recent historically-accepted transactions.

[0095] Specifically, the smart network card can write each accepted transaction into a transaction identification pool in the network card cache, and take out the historically-accepted transactions with an acceptance time length greater than a designated time length from the transaction identification pool. The acceptance time length refers to a

time length between an acceptance time point and a current time point. The smart network card can compare the currently accepted transaction with the historically accepted transactions in the transaction identification pool.

[0096] Further, in an actual application, a transaction usually has a valid time length. If a transaction acceptance time is too long from now, the transaction will become invalid. It is meaningless to replay those historically accepted transactions that are too old. Therefore, the above-mentioned designated time length can be specifically set to be no less than a valid time length of a transaction.

[0097] In embodiments of the present disclosure, the smart network card can classify those identified replay transactions into a replay transaction set. The replay transaction set does not participate in the blockchain consensus, or the replay transaction set is not sent to the CPU. Therefore, the replay transaction can be filtered. In addition, the smart network card can also discard the replay transaction.

[0098] According to the method shown in FIG. 6, a dedicated blockchain node device includes a CPU and a smart network card, and the smart network card is a network card having a built-in processor or microprocessor and can perform data computing and processing. The smart network card can replace the CPU to perform replay transaction identification.

[0099] According to the embodiments of the present disclosure, the following technical effects can be achieved:

[00100] 1. The smart network card is specifically in charge of identifying a replay transaction to offload this part of operation burden for the CPU. In this case, the operation efficiency of the dedicated blockchain node device serving as a node is improved and the CPU can also execute more transactions within a unit time, thereby improving throughput.

[00101] 2. The replay transaction can be identified and filtered at the smart network card thus the replay transaction generally will not flow through the CPU, so that the replay transaction can be identified and filtered more rapidly.

[00102] FIG. 7 is a flowchart of a method of identifying a to-be-filtered transaction based on a dedicated blockchain node device according to embodiments of the present disclosure. The method includes the following steps:

[00103] At step S700, when the dedicated blockchain node device accepts a transaction through a smart network card, the smart network card determines whether the currently accepted transaction satisfies a predetermined filtering condition.

[00104] At step S702, if the smart network card determines that the currently accepted transaction satisfies the predetermined filtering condition, the smart network card determines the currently accepted transaction as a to-be-filtered transaction.

[00105] The smart network card can discard the identified to-be-filtered transaction, or retain and classify the to-be-filtered transaction into a to-be-filtered transaction set. The to-be-filtered transaction set does not participate in a blockchain consensus, or is not sent to the CPU. Therefore, the to-be-filtered transaction can be filtered.

[00106] In addition, the to-be-filtered transactions mainly involve the following two cases:

[00107] 1. A service operation type corresponding to a service operation realized by a transaction is not a service operation type in charged by the present node (the dedicated blockchain node device), nor belong to a service operation type set corresponding to the present node. For example, due to limited transaction processing capability, the present node will only selectively execute transactions of some specific service operation types. For another example, some service operation types are written into a blacklist of the present node, and the present node refuses to execute the transactions of these service operation types.

[00108] In addition, it should be noted that for a blockchain network of such distributed architecture, failure of a small number of nodes to execute some transactions will not affect a world state of the entire blockchain network, and the world state will still be synchronized to each node.

[00109] 2. A service operation type corresponding to a service operation realized by a transaction is a service operation type in the charge of the present node, and belongs to a service operation type set corresponding to the present node. However, specific contents of the service operation realized by the transaction are illegal, nor belong to legal operation contents corresponding to the service operation type. For example, a transaction is used to realize a query operation, but the query operation is specifically used to perform repeated meaningless queries for a particular parameter value, and therefore the transaction actually belongs to an attack transaction for a blockchain network and is illegal. For another example, a transaction is used to perform a transfer operation but the balance of the transferor is insufficient, and the transaction is also illegal. For another example, an account of a transaction initiator has no operation authority, and thus the transaction is also illegal.

[00110] In addition, according to the method shown in FIG. 7, a more complex filtering condition can be configured for the smart network card. For example, the following filtering condition can be configured:

[00111] (A particular time period) and (initiated by account A) and (sending to account B or C for operation).

[00112] According to the method shown in FIG. 7, a dedicated blockchain node device includes a CPU and a smart network card, and the smart network card is a network card having a built-in processor or microprocessor and can perform data computing and processing.

The smart network card can replace the CPU to perform to-be-filtered transaction identification.

[00113] According to the embodiments of the present disclosure, the following technical effects can be achieved:

[00114] 1. The smart network card is specifically in charge of identifying a to-be-filtered transaction to offload this part of operation burden for the CPU. In this case, the operation efficiency of the dedicated blockchain node device serving as a node is improved and the CPU can also execute more transactions within a unit time, thereby improving throughput.

[00115] 2. The to-be-filtered transaction can be identified and filtered at the smart network card and thus the to-be-filtered transaction generally will not flow through the CPU, so that the to-be-filtered transaction is identified and filtered more rapidly.

[00116] FIG. 8 is a structural schematic diagram of an apparatus for processing a transaction based on a dedicated blockchain node device according to embodiments of the present disclosure. The apparatus is applied to a smart network card of the dedicated blockchain node device, and the dedicated blockchain node device further includes a CPU and is any node of a blockchain network. The apparatus includes: a caching module 801, configured to, when the dedicated blockchain node device accepts a transaction through the smart network card, write the accepted transaction into a network card cache; a consensus interacting module 802, configured to perform consensus interaction with other nodes of the blockchain network based on the transaction in the network card cache in a process that the dedicated blockchain node device participates in a blockchain consensus; and a sending module 803, configured to determine and send a to-be-executed transaction list to the CPU based on a result of the consensus.

[00117] FIG. 9 is a structural schematic diagram of an apparatus for processing a transaction based on a dedicated blockchain node device according to embodiments of the present disclosure. The apparatus is applied to a CPU of the dedicated blockchain node device, and the dedicated blockchain node device further includes a smart network card and is any node of a blockchain network. The apparatus includes: a chaining executing module 901, configured to execute transaction based on a to-be-executed transaction list sent by the smart network card and complete transaction chaining.

[00118] FIG. 10 is a structural schematic diagram of an apparatus for synchronizing data based on a dedicated blockchain node device according to embodiments of the present disclosure. The apparatus is applied to a smart network card of the dedicated blockchain node device, and the dedicated blockchain node device further includes a CPU and is any node of a blockchain network. The apparatus includes: an inquiring module 1001, configured to inquire

other nodes about whether there is to-be-synchronized block data; a pulling module 1002, configured to pull the to-be-synchronized block data from other nodes if determining that there is the to-be-synchronized block data; and a providing module 1003, configured to provide the to-be-synchronized block data for the CPU.

[00119] FIG. 11 is a structural schematic diagram of an apparatus for synchronizing data based on a dedicated blockchain node device according to embodiments of the present disclosure. The apparatus is applied to a CPU of the dedicated blockchain node device, and the dedicated blockchain node device further includes a smart network card and is any node of a blockchain network. The apparatus includes: a synchronizing module 1101, configured to complete data synchronization based on to-be-synchronized block data.

[00120] FIG. 12 is a structural schematic diagram of an apparatus for transferring a transaction based on a dedicated blockchain node device according to embodiments of the present disclosure. The apparatus is applied to a smart network card of the dedicated blockchain node device, and the dedicated blockchain node device further includes a CPU and is any node of a blockchain network. The apparatus includes: a determining module 1201, configured to determine other to-be-transferred nodes when the dedicated blockchain node device accepts a transaction through the smart network card; and a transferring module 1202, configured to transfer the transaction to the to-be-transferred nodes.

[00121] FIG. 13 is a structural schematic diagram of an apparatus for identifying a replay transaction based on a dedicated blockchain node device according to embodiments of the present disclosure. The apparatus is applied to a smart network card of the dedicated blockchain node device, and the dedicated blockchain node device further includes a CPU and is any node of a blockchain network. The apparatus includes: an identifying module 1301, configured to, when the dedicated blockchain node device accepts a transaction through the smart network card, identify currently accepted transaction, including: comparing the currently accepted transaction with historically accepted transactions in a network card cache; and if a comparison result indicates that there is a historically accepted transaction same as the currently accepted transaction, determining the currently accepted transaction as a replay transaction.

[00122] FIG. 14 is a structural schematic diagram of an apparatus for identifying a to-be-filtered transaction based on a dedicated blockchain node device according to embodiments of the present disclosure. The apparatus is applied to a smart network card of the dedicated blockchain node device, and the dedicated blockchain node device further includes a CPU and is any node of a blockchain network. The apparatus includes: an identifying module 1401, configured to, when the dedicated blockchain node device accepts a

transaction through the smart network card, identify currently accepted transaction, including: determining whether the currently accepted transaction satisfies a predetermined filtering condition; and if determining that the currently accepted transaction satisfies the predetermined filtering condition, determining the currently accepted transaction as a to-be-filtered transaction.

[00123] Embodiments of the present disclosure further provide a computer device at least including a memory, a CPU, a smart network card, and computer programs that are stored on the memory and run on the smart network card. When executing the programs, the smart network card implements the functions of each method in the present disclosure.

[00124] FIG. 15 is a hardware structural schematic diagram of a more specific computer device according to embodiments of the present disclosure. The device can include: a processor 1510, a memory 1520, an input/output interface 1530, a communication interface 1540 and a bus 1550. The processor 1510, the memory 1520, the input/output interface 1530 and the communication interface 1540 realize communication connection with each other in the device through the bus 1550.

[00125] The processor 1510 can be implemented by a general central processing unit (CPU), a microprocessor, an application specific integrated circuit (ASIC), or one or more integrated circuits, or the like to execute relevant programs, so as to implement the technical solutions according to the embodiments of the present disclosure.

[00126] The memory 1520 can be implemented by a read only memory (ROM), a random access memory (RAM), a static storage device, a dynamic storage device, or the like. The memory 1520 can store an operating system and other application programs. When the technical solutions according to the embodiments of the present disclosure are implemented by software or firmware, relevant program codes stored in the memory 1520 are invoked and executed by the processor 1510.

[00127] The input/output interface 1530 is used to connect an inputting/outputting module, so as to realize information input and output. The inputting/outputting module can be configured as a component (not shown) in the device, or can be externally connected to the device to provide corresponding functions. The input device can include a keyboard, a mouse, a touch screen, a microphone, various sensors, and the like, and the output device can include a display, a speaker, a vibrator, an indicator light, and the like.

[00128] The communication interface 1540 is used to connect a communication module (not shown), so as to realize communication interaction between the device and other devices. The communication module can realize communication in a wired manner (such as a USB and a network cable) or in a wireless manner (such as a mobile network, WIFI and Bluetooth).

[00129] The bus 1550 includes a channel transmitting information between different components (such as the processor 1510, the memory 1520, the input/output interface 1530 and the communication interface 1540) of the device.

[00130] It should be noted that although the above-mentioned device only shows the processor 1510, the memory 1520, the input/output interface 1530, the communication interface 1540 and the bus 1550, the device can further include other components needed for normal operation in a specific implementation process. In addition, persons skilled in the art can understand that the above-mentioned device can only include components needed for the solution according to the embodiments of the present disclosure, rather than all components shown in the drawings.

[00131] Embodiments of the present disclosure further provide a computer readable storage medium storing computer programs, and the programs are executed by a smart network card to implement functions of each method in the present disclosure.

[00132] The computer readable medium includes permanent, non-permanent, mobile and non-mobile media, which can realize information storage by any method or technology. The information can be computer readable instructions, data structures, program modules and other data. Examples of the computer storage medium include but not limited to: a phase change random access memory (PRAM), a Static Random Access Memory (SRAM), a Dynamic Random Access Memory (DRAM), and other types of RAMs, Read-Only Memories (ROM), Electrically-Erasable Programmable Read-Only Memories (EEPROM), Flash Memories, or other memory technologies, CD-ROMs, digital versatile discs (DVD) or other optical storages, cassette type magnetic tapes, magnetic disk storages, or other magnetic storage devices or any other non-transmission mediums for storing information accessible by computing devices. According to the definition of the specification, the computer readable medium does not include transitory computer readable media, such as modulated data signals and carriers.

[00133] It can be known from descriptions of the above embodiments that persons skilled in the art can clearly understand that the embodiments of the present disclosure can be implemented by means of software and a necessary general hardware platform. Based on such understanding, the technical solutions of embodiments of the present disclosure essentially or a part contributing to the existing technology can be embodied in the form of a software product, and the computer software product can be stored in a storage medium, such as a ROM/RAM, a diskette or a compact disk, and includes several instructions for enabling a computer device (such as a personal computer, a server or a network device) to execute the methods of different embodiments or some parts of the embodiments of the present

disclosure.

[00134] The systems, methods, modules or units described in the above-mentioned embodiments can be specifically implemented by a computer chip or an entity, or can be implemented by a product with a particular function. A typical implementing device can be a computer, and the computer can specifically be a personal computer, a laptop computer, a cellular phone, a camera phone, a smart phone, a personal digital assistant, a media player, a navigation device, an email transceiver, a game console, a tablet computer, a wearable device, or a combination of any several devices of the above devices.

[00135] The embodiments in the present disclosure are described in a progressive manner, each embodiment focuses on differences from other embodiments, and same or similar parts among the embodiments can be referred to each other. Especially, because apparatus embodiments are basically similar to method embodiments, simple descriptions are made to the apparatus embodiments, and relevant parts can be referred to part of the descriptions of the method embodiments. The apparatus embodiments described above are merely illustrative, where modules described as separate members can be or does not have to be physically separated, and functions of different modules can be implemented in the same or several software and/or hardware during implementation of the embodiments of the present disclosure. Part or all of the modules can also be selected according to actual needs to achieve the objectives of the solution of the embodiment. Persons of ordinary skill in the art can understand and implement the solutions without creative work.

[00136] The above-mentioned descriptions are merely specific embodiments of the present disclosure. It should be noted that persons of ordinary skill in the art can also make several improvements and modifications without departing from the principles of the embodiments of the present disclosure, and these improvements and modifications shall also be included in the scope of protection of embodiments of the present disclosure.

CLAIMS

1. A method of processing a transaction based on a dedicated blockchain node device, wherein the dedicated blockchain node device comprises a Central Processing Unit, CPU, and a smart network card and is any node of a blockchain network, and the method comprises:

when the dedicated blockchain node device accepts a transaction through the smart network card, writing (S300), by the smart network card, the accepted transaction into a network card cache, sending, by the smart network card, the accepted transaction to the CPU and writing, by the CPU, the accepted transaction into a CPU cache;

performing (S302), by the smart network card, a consensus interaction with other nodes of the blockchain network based on the transaction in the network card cache in a process that the dedicated blockchain node device participates in a blockchain consensus;

determining and sending (S304), by the smart network card, a to-be-executed transaction list to the CPU based on a result of the blockchain consensus, wherein the to-be-executed transaction list comprises a transaction identifier of each to-be-executed transaction determined based on the present blockchain consensus; and

executing (S306), by the CPU, transaction based on the to-be-executed transaction list and completing transaction chaining.

2. The method according to claim 1, wherein the to-be-executed transaction list includes a transaction hash of each to-be-executed transaction determined based on the present blockchain consensus and wherein, when the dedicated blockchain node device accepts the transaction through the smart network card, the smart network card sends the accepted transaction to the CPU while writing the accepted transaction into the network card cache.

3. The method according to claim 1, further comprising:

in the process that the dedicated blockchain node device participates in the blockchain consensus, sending, by the smart network card, consensus configuration modification information to the CPU if receiving the consensus configuration modification information; and

modifying, by the CPU, a consensus configuration locally stored in the dedicated blockchain node device based on the consensus configuration modification information.

4. The method according to claim 1, further comprising:

sending, by the CPU, an execution result generated by executing transaction based on the to-be-executed transaction list to the smart network card; and

receiving, by the smart network card, the execution result as reference information of a next blockchain consensus.

5. A dedicated blockchain node device, comprising a Central Processing Unit, CPU, and a smart network card, and the dedicated blockchain node device being any node of a blockchain network, wherein the dedicated blockchain node device is configured to carry out the method of any of the preceding claims.

6. A blockchain system, comprising a blockchain network, wherein at least one node of the blockchain network is the dedicated blockchain node device according to claim 5.