

BLOCKCHAIN-BASED DATA PROCESSING METHOD AND APPARATUS, AND DEVICE, STORAGE MEDIUM AND PROGRAM PRODUCT

RELATED APPLICATION

[0001] The present application claims priority to Chinese Patent Application No. 202211319128.5, entitled "DATA PROCESSING METHOD BASED ON BLOCKCHAIN, DEVICE, and READABLE STORAGE MEDIUM" and filed with the China National Intellectual Property Administration on October 26, 2022.

FIELD OF THE TECHNOLOGY

[0002] The present application relates to the field of internet technologies, and in particular, to a data processing method and apparatus based on a blockchain, a device, a storage medium, and a program product.

BACKGROUND OF THE DISCLOSURE

[0003] With the rapid development of network technologies and the emphasis of enterprises on data security, blockchains are widely used. When a large number of resources need to be transferred, a virtual resource transfer interface on a blockchain needs to be called regularly and continuously, and a transaction on a chain, such as salary transfer between employers and employees, and personal tax transfer between employees and the tax bureau, needs to be formed in each cycle.

[0004] Through the stream-resource transfer method, shortcomings of periodic resource transfer on a chain can be effectively solved. For example, periodic resources may be transactions submitted for each resource transfer or may be handling fees paid for each transaction. However, with the widespread applications of the stream-resource transfer method, the number of stream-resource transfers on a chain increases day by day, resulting in an increase in the complexity of stream-resource transfers on the chain, thereby reducing the efficiency of stream-resource transfers.

SUMMARY

[0005] Embodiments of the present application provide a data processing method and apparatus based on a blockchain, a device, a storage medium, and a program product, which can

reduce the complexity of a stream-resource transfer channel on a blockchain, thereby improving the efficiency of stream-resource transfer.

[0006] According to an aspect, an embodiment of the present application provides a data processing method based on a blockchain, executable by a computer device. The method includes:

creating a target stream-resource transfer channel for a stream-resource channel creation request in the blockchain based on a stream-resource transfer contract in the blockchain, the stream-resource channel creation request including a target object address pair, and the target stream-resource transfer channel providing, for the target object address pair, a function of performing stream-resource transfer based on the stream-resource transfer contract;

obtaining, in the blockchain according to the target object address pair, a historical stream-resource transfer channel including a historical object address pair, the target object address pair and the historical object address pair sharing at least one identical object address;

combining the historical stream-resource transfer channel and the target stream-resource transfer channel to obtain an update stream-resource transfer channel, the update stream-resource transfer channel including an update object address pair; and

performing, according to the update stream-resource transfer channel, stream-resource transfer processing associated with the update object address pair.

[0007] According to another aspect, an embodiment of the present application provides a data processing apparatus based on a blockchain. The apparatus includes:

a channel creation module, configured to create a target stream-resource transfer channel for a stream-resource channel creation request in the blockchain based on a stream-resource transfer contract in the blockchain, the stream-resource channel creation request including a target object address pair, and the target stream-resource transfer channel providing, for the target object address pair, a function of performing stream-resource transfer based on the stream-resource transfer contract;

a first obtaining module, configured to obtain, in the blockchain according to the target object address pair, a historical stream-resource transfer channel including a historical object address pair, the target object address pair and the historical object address pair sharing at least one identical object address; a first processing module, configured to combine the historical stream-resource transfer channel and the target stream-resource transfer channel to obtain an update stream-resource transfer channel, the update stream-resource transfer channel including an update object address pair; and

a second processing module, configured to perform, according to the update stream-resource transfer channel, stream-resource transfer processing associated with the update object address pair.

[0008] According to another aspect, an embodiment of the present application provides a computer device, including: a processor, a memory, and a network interface;

the processor being connected to the memory and the network interface, the network interface being configured to provide a data communication function, the memory being configured to store computer programs, and the processor being configured to call the computer programs so that the computer device performs the method according to the embodiments of the present application.

[0009] According to another aspect, an embodiment of the present application provides a computer-readable storage medium, storing computer programs, the computer programs being loaded and executed by a processor to perform the method according to the embodiments of the present application.

[0010] According to another aspect, an embodiment of the present application provides a computer program product, including computer programs, and the computer programs being stored in a computer-readable storage medium. A processor of a computer device reads the computer programs from the computer-readable storage medium, and the processor executes the computer programs, so that the computer device performs the method according to the embodiments of the present application.

BRIEF DESCRIPTION OF THE DRAWINGS

[0011] To describe the technical solutions of the embodiments of the present application more clearly, the following briefly introduces the accompanying drawings required for describing the embodiments. Apparently, the accompanying drawings in the following description show only some embodiments of the present application, and a person of ordinary skill in the art may still derive other drawings from these accompanying drawings without creative efforts.

[0012] FIG. 1 is a schematic diagram of a system architecture according to an embodiment of the present application;

[0013] FIG. 2a is a first schematic diagram of a data processing scenario based on a blockchain according to an embodiment of the present application;

[0014] FIG. 2b is a second schematic diagram of a data processing scenario based on a blockchain according to an embodiment of the present application;

[0015] FIG. 3 is a first schematic flowchart of a data processing method based on a blockchain according to an embodiment of the present application;

[0016] FIG. 4 is a third schematic diagram of a data processing scenario based on a blockchain according to an embodiment of the present application;

[0017] FIG. 5 is a fourth schematic diagram of a data processing scenario based on a blockchain according to an embodiment of the present application;

[0018] FIG. 6 is a schematic interaction diagram of a data processing method based on a blockchain according to an embodiment of the present application;

[0019] FIG. 7 is a second schematic flowchart of a data processing method based on a blockchain according to an embodiment of the present application;

[0020] FIG. 8 is a third schematic flowchart of a data processing method based on a blockchain according to an embodiment of the present application;

[0021] FIG. 9 is a schematic structural diagram of a data processing apparatus based on a blockchain according to an embodiment of the present application; and

[0022] FIG. 10 is a schematic structural diagram of a computer device according to an embodiment of the present application.

DESCRIPTION OF EMBODIMENTS

[0023] The technical solutions of the embodiments of the disclosure will be described below clearly and comprehensively in conjunction with accompanying drawings of the embodiments of the disclosure. Apparently, the embodiments described are merely some of, rather than all of, the embodiments of the disclosure. All other embodiments obtained by a person of ordinary skill in the art based on the embodiments of the present application without creative efforts shall fall within the protection scope of the present application.

[0024] To facilitate understanding, some nouns are described below:

[0025] 1. Blockchain: In a narrow sense, a blockchain is a chained data structure with blocks as the basic unit, and digital summaries are used in blocks to verify previously obtained transaction history, which is suitable for needs of tamper resistance and scalability in distributed accounting scenarios. In a broad sense, a blockchain also refers to a distributed accounting

technology implemented through a blockchain structure, including distributed consensus, privacy and security protection, point-to-point communication technologies, network protocols, smart contracts, or the like.

[0026] The goal of the blockchain is to achieve a distributed data recording ledger that only allows addition and no deletion. The basic structure of the underlying ledger is a linear linked list. The linked list includes "blocks" connected in series. A hash value of a previous block is recorded in a subsequent block. Whether each block (and transactions in the block) is legal can be quickly verified by calculating the hash value. If a node in a network proposes to add a new block, consensus on the block needs to be reached through the consensus mechanism.

[0027] 2. Block: A block is a data packet that carries transaction data on the blockchain network. A block is a data structure marked with a timestamp and a hash value corresponding to the previous block. The block verifies and confirms a transaction in the block through the consensus mechanism of the network. A block includes a block header and a block body. The block header can record meta-information of a current block, including data such as a current version number, a hash value corresponding to a previous block, a timestamp, a random number, and a hash value of a Merkle root.

[0028] The block body can record detailed data generated over a period of time, including all transaction records or other information verified by the current block and generated during a block creation process, and can be understood as a form of representation of a ledger. In addition, the detailed data of the block body may include that a unique Merkle root is generated through the hashing process of the Merkle tree and recorded in the block header.

[0029] A previous block is also called a parent block, and the blockchain achieves temporal ordering by recording the hash value corresponding to the block and the hash value corresponding to the parent block in the block header.

[0030] 3. Blockchain node: A blockchain network divides nodes into consensus nodes (which may also be referred to as core nodes) and synchronization nodes (which may include data nodes and light nodes). The consensus node is responsible for the consensus service of the entire blockchain network. The synchronization node is responsible for synchronizing ledger information of the consensus node, that is, synchronizing the latest block data. The internal structure of either the consensus node or the synchronization node includes network communication components, because the blockchain network is essentially a peer-to-peer (P2P) network and P2P components are needed to communicate with other nodes in the blockchain network. Resources and services in the blockchain network are scattered on various nodes, and the transmission of information and the

implementation of services are performed directly between nodes without the intervention of intermediate links or centralized servers (third parties).

[0031] 4. Public key and private key: A public key and a private key are a key pair (that is, one public key and one private key) obtained through an algorithm. The public key is the public part of the key pair, and the private key is the non-public part. The public key is commonly used to encrypt data, verify digital signatures, or the like. This algorithm can ensure that the obtained key pair is unique. When using this key pair, if one of the keys is used to encrypt a piece of data, the other key needs to be used to decrypt the data. For example, if the public key is used to encrypt the data, the private key needs to be used to decrypt the data, and if the private key is used to encrypt the data, the public key needs to be used to decrypt the data; otherwise, the decryption does not succeed.

[0032] 5. Asymmetric signature: A signature algorithm includes two keys: the public key and the private key. The public key and the private key are a pair. If the private key is used to sign the data, the corresponding public key needs to be used to verify the signature of the data. Because the signature process and the signature verification process use two different keys, this algorithm is called asymmetric signature. The basic process of implementing confidential information exchange through asymmetric signature may be: A party A generates a key pair and makes the public key public. When needing to send a message to another party (a party B), the party A uses its own private key to sign the confidential message and then sends the message to the party B. The party B then uses the public key of the party A to verify the signed message.

[0033] 6. Smart contract: A smart contract is a computer protocol designed to communicate, verify, or execute contracts in an information-based manner. In the blockchain system, a smart contract is a code that may be understood and executed by each node of the blockchain. Any logic may be executed and a result may be obtained. In practical applications, a smart contract is managed and tried through transactions on the blockchain. Each transaction is equivalent to a remote procedure call (RPC) request of the blockchain system. If a smart contract is equivalent to an executable program, the blockchain is equivalent to an operating system that provides a running environment. The blockchain may include multiple contracts that are distinguished by contract identities (ID), identification numbers, or names. In the embodiments of the present application, a stream-resource transfer contract is a smart contract.

[0034] 7. Stream-resource transfer: that is, stream payment. Stream-resource transfer is implemented through smart contracts and programming of digital assets (also referred to as virtual resources). Stream-resource transfer defines a directional relationship between two object addresses

(a stream-resource transfer-out address and a stream-resource transfer-in address). A stream-resource is created from the stream-resource transfer-out address, and continues to flow to the stream-resource transfer-in address according to a transfer rate per unit time (for example, per second or per hour).

[0035] FIG. 1 is a schematic diagram of a system architecture according to an embodiment of the present application. As shown in FIG. 1, the system architecture may include a service server 101, a terminal device cluster 100, and a blockchain network.

[0036] The blockchain network may include a blockchain node cluster 10 with consensus permission, and the blockchain node cluster 10 may include one or more blockchain nodes. This embodiment of the present application does not limit the number of blockchain nodes in the blockchain node cluster 10. As shown in FIG. 1, the blockchain node cluster 10 may include a blockchain node 10a, a blockchain node 10b, a blockchain node 10c, and a blockchain node 10d. When working normally, each blockchain node may receive a transaction request (for example, a stream-resource channel creation request in this embodiment of the present application) sent by off-chain devices (for example, the service server 101 and the terminal device in the terminal device cluster 100 in this embodiment of the present application), and generate a block based on the received transaction request and then put the block on the chain.

[0037] In specific implementations of the present application, related data such as user information (for example, a stream-resource channel creation request and a target object address pair) is involved, and when the embodiments of the present application are applied to a specific product or technology, it is necessary to obtain user permission or consent, and the collection, usage, and processing of related data need to comply with relevant laws, regulations and standards of relevant countries and regions.

[0038] To ensure data interoperability between various blockchain nodes, there may be a data connection between blockchain nodes. For example, there is a data connection between the blockchain node 10a and the blockchain node 10c, there is a data connection between the blockchain node 10a and the blockchain node 10d, and there is a data connection between the blockchain node 10b and the blockchain node 10c.

[0039] Data or block transmission may be performed between blockchain nodes through the data connection. The data connection between the blockchain nodes may be based on node identifiers. Each blockchain node in the blockchain network has a corresponding node identifier, and each blockchain node may store node identifiers of other blockchain nodes that are connected to the blockchain node, so that obtained data or generated blocks may be subsequently broadcast to

other blockchain nodes based on node identifiers of the other blockchain nodes. For example, the blockchain node 10a may maintain a node identifier list, which stores node names and node identifiers of other blockchain nodes, as shown in Table 1.

Table 1

Node name	Node identifier
Blockchain node 10b	AAAAA
Blockchain node 10c	BBBBB
...	...
Blockchain node 10d	CCCCC

[0040] The node identifier may be an internet protocol (IP) address, and any other information that may be configured for identifying the node in the blockchain network. Table 1 only uses the IP address as an example for description.

[0041] Assuming that the node identifier of the blockchain node 10a is FFFFFFFF, the blockchain node 10a may send a stream-resource channel creation request to the blockchain node 10d through the node identifier CCCCC, and the blockchain node 10d may determine, through the node identifier FFFFFFFF, that the stream-resource channel creation request is sent by the blockchain node 10a. Similarly, the blockchain node 10d may send, to the blockchain node 10c through the node identifier BBBBB, a block for which consensus is to be reached, and the blockchain node 10c may determine, through the node identifier CCCCC, that the block for which consensus is to be reached is sent by the blockchain node 10d. Data transmission between other nodes is similar and is not described one by one.

[0042] A connection method of the data connection is not limited, and may be direct or indirect connection through wired communication, may be direct or indirect connection through wireless communication, or may be other connection methods. The present application is not limited herein.

[0043] Each blockchain node (for example, the blockchain node 10a, the blockchain node 10b, the blockchain node 10c, and the blockchain node 10d) in the blockchain node cluster 10 may be used to maintain the same blockchain network, and this embodiment of the present application does not limit the type of the blockchain network, which may be a private chain, a public chain, or a consortium chain including a private chain and a public chain. Any two blockchain nodes in the blockchain node cluster 10 may form a point-to-point (P2P) network, and the point-to-point

network may adopt the P2P protocol. In a distributed system, any device such as a server or a terminal may join to become a blockchain node.

[0044] The system may include one or more terminal devices, and this embodiment of the present application does not limit the number of terminal devices in the terminal device cluster 100. As shown in FIG. 1, the terminal device cluster 100 includes a terminal device 100a, a terminal device 100b, a terminal device 100c, and a terminal device 100d. There may be a communication connection in the terminal device cluster 100, for example, there may be a communication connection between the terminal device 100c and the terminal device 100a. Any terminal device in the terminal device cluster 100 may have a communication connection with any blockchain node in the blockchain node cluster 10. For example, there is a communication connection between the blockchain node 10a and the terminal device 100b. There is a communication connection between the blockchain node 10d and the terminal device 100c. Any terminal device in the terminal device cluster 100 may have a communication connection with the service server 101. For example, there may be a communication connection between the terminal device 100a and the service server 101. A connection method of the communication connection is not limited, and may be direct or indirect connection through wired communication, may be direct or indirect connection through wireless communication, or may be other methods. The present application is not limited herein.

[0045] Each terminal device in the terminal device cluster 100 shown in FIG. 1 may be installed with a client. When the client is run in each terminal device, the client may exchange data with the blockchain node cluster 10 shown in FIG. 1, that is, the above communication connection, and may also exchange data with the background server (for example, the service server 101) corresponding to the client. The client may be a video application, a social networking application, an instant messaging application, an office software application, a navigation application, a shopping application, a financial management application, a business application, a browser, and other application clients that have the function of sending request messages. The client may be an independent client or an embedded sub-client integrated in a client (for example, a social networking client, an education client, or a multimedia client), which is not limited herein.

[0046] The service server 101 may be a background server corresponding to the client installed on the terminal device in the terminal device cluster 100. The system may include one or more service servers 101, and the number of service servers 101 is not limited herein. Taking a digital asset application as an example, each terminal device may perform, through a client corresponding to the digital asset application, data transmission with a background server corresponding to the digital asset application. For example, the terminal device may upload a

stream-resource channel creation request to the background server through the client of the digital asset application, and then the background server may transmit the stream-resource channel creation request to the blockchain network. Another feasible manner is that the terminal device may directly transmit the stream-resource channel creation request to the blockchain network through the client of the digital asset application.

[0047] The blockchain nodes in FIG. 1 include but are not limited to mobile terminals or servers. The server may be an independent physical server, or a server cluster composed of multiple physical servers or a distributed system, or may be a cloud server that provides basic cloud computing services such as cloud services, cloud databases, cloud computing, cloud functions, cloud storage, and network services, cloud communications, middleware services, domain name services, security services, content delivery networks (CDN), big data, and artificial intelligence platforms. The mobile terminals include but are not limited to mobile phones, computers, intelligent voice interaction devices, smart home appliances, vehicle-mounted terminals, aircraft, or the like. The mobile terminal and the server may be connected directly or indirectly through wired or wireless methods, which are not limited in the embodiments of the present application.

[0048] Further, FIG. 2a is a first schematic diagram of a data processing scenario based on a blockchain according to an embodiment of the present application. An implementation process of this data processing scenario may be performed in a blockchain node. The blockchain node may be any blockchain node in the blockchain node cluster 10 in the embodiment corresponding to FIG. 1. FIG. 2a takes the blockchain node 10a as an example for description. As shown in FIG. 2a, the blockchain node 10a obtains a stream-resource channel creation request 20a. In this embodiment of the present application, the stream-resource channel creation request 20a is, for example, "an address d1 transfers 5 virtual resources to an address d2 per day". The first target object address having a stream-resource transfer-out attribute is, for example, the address d1, the second target object address having a stream-resource transfer-in attribute is, for example, the address d2, and the target stream-resource transfer rate is, for example, transferring 5 virtual resources per day.

[0049] After obtaining the stream-resource channel creation request 20a, the blockchain node 10a calls a stream-resource transfer contract 20b in the blockchain. The stream-resource transfer contract 20b is a smart contract and may implement stream-resource transfer on the chain. Based on the stream-resource transfer contract 20b, the blockchain node 10a creates a target stream-resource transfer channel 201c in the blockchain for the stream-resource channel creation request 20a. For simplicity and clarity, FIG. 2a shows the stream-resource transfer rate in an abbreviated manner, including: the target stream-resource transfer rate (that is, 5 virtual resources are

transferred per day) is abbreviated as 5/day, and the historical stream-resource transfer rate (that is, 10 virtual resources are transferred per day) is abbreviated as 10/day. The symbol "→" in FIG. 2a has a directional meaning, for example, the symbol "→" in the target stream-resource transfer channel 201c indicates that the address d1 has the stream-resource transfer-out attribute, and the address d2 has the stream-resource transfer-in attribute.

[0050] In this embodiment of the present application, the target stream-resource transfer channel 201c provides a function of stream-resource transfer based on the stream-resource transfer contract 20b for the target object address pair in the stream-resource channel creation request 20a, that is, the blockchain node 10a may perform stream-resource transfer between target object address pairs based on the stream-resource transfer contract 20b. The target object address pair in FIG. 2a includes the address d1 and the address d2.

[0051] According to the target object address pair, the blockchain node 10a may obtain the historical stream-resource transfer channel including the historical object address pair from the blockchain, where object addresses in the target object address pair and object addresses in the historical object address pair share an identical object address. The historical stream-resource transfer channel provides the function of stream-resource transfer based on the stream-resource transfer contract 20b for a historical object address pair. That is, based on the stream-resource transfer contract 20b, the blockchain node 10a may perform stream-resource transfer between the historical object address pairs. The process of obtaining the historical stream-resource transfer channel is not described in detail herein. Refer to the description of operation S102 in the embodiment corresponding to FIG. 3 below.

[0052] Refer to FIG. 2a again. In this embodiment of the present application, the historical stream-resource transfer channel 201d is "address d2→address d1, 10/day". For the specific meaning of the historical stream-resource transfer channel 201d, refer to the above description of the meaning of the target stream-resource transfer channel 201c. Details are not described again herein. The historical object address pair in FIG. 2a includes a third historical object address having a stream-resource transfer-out attribute, and a fourth historical object address having a stream-resource transfer-in attribute, that is, the address d2 and the address d1 in FIG. 2a.

[0053] The blockchain node 10a determines the historical stream-resource transfer rate of the historical stream-resource transfer channel 201d (the example in FIG. 2a is 10/day), and determines the target stream-resource transfer rate of the target stream-resource transfer channel 201c (the example in FIG. 2a is 5/day). The blockchain node 10a may combine the historical stream-resource transfer channel 201d and the target stream-resource transfer channel 201c

according to the target object address pair, the historical object address pair, the historical stream-resource transfer rate, and the target stream-resource transfer rate, to obtain the update stream-resource transfer channel. Specifically, the blockchain node 10a obtains the first target object address having the stream-resource transfer-out attribute and the second target object address having the stream-resource transfer-in attribute in the target object address pair, such as the address d1 and the address d2 as shown in FIG. 2a. At the same time, the blockchain node 10a obtains a third historical object address having a stream-resource transfer-out attribute, and a fourth historical object address having a stream-resource transfer-in attribute in the historical object address pair, such as the address d2 and the address d1 in FIG. 2a.

[0054] The blockchain node 10a may determine the association between the historical object address pair and the target object address pair. In FIG. 2a, the association may be a first association 20j. The first association 20j may represent that the first target object address is the same as the fourth historical object address, and the second target object address is the same as the third historical object address. In this scenario, the blockchain node 10a compares the historical stream-resource transfer rate with the target stream-resource transfer rate. In FIG. 2a, the historical stream-resource transfer rate (for example, 10/day) is greater than the target stream-resource transfer rate (for example, 5/day). Therefore, the blockchain node 10a may set the transfer state of the target stream-resource transfer channel 201c to the stop transfer state, as shown in FIG. 2a, and obtain the target stream-resource transfer channel 202c. "Stop" in the target stream-resource transfer channel 202c indicates the stop transfer state.

[0055] At the same time, the blockchain node 10a determines the first stream-resource transfer rate difference between the historical stream-resource transfer rate and the target stream-resource transfer rate, for example, identified as a stream-resource transfer rate difference a. In FIG. 2a, the stream-resource transfer rate difference a may be equal to 5/day, and is abbreviated as "difference a=5/day". Further, the blockchain node 10a updates the historical stream-resource transfer rate of the historical stream-resource transfer channel 201d (10/day in FIG. 2a) to the stream-resource transfer rate difference a, so that the historical stream-resource transfer channel 202d may be obtained.

[0056] This embodiment of the present application does not limit the total number of target stream-resource transfer channels, which is determined according to the actual application scenario and may be one or more. FIG. 2a takes one target stream-resource transfer channel (that is, the target stream-resource transfer channel 201a) as an example for description. If there are multiple

target stream-resource transfer channels, processing procedures corresponding to the target stream-resource transfer channels are the same.

[0057] This embodiment of the present application does not limit the total number of historical stream-resource transfer channels, which is determined according to the actual application scenario and may be one or more. FIG. 2a takes one historical stream-resource transfer channel (that is, a historical stream-resource transfer channel 201d) as an example for description. If there are multiple historical stream-resource transfer channels, the processing procedure of the remaining historical stream-resource transfer channel is the same as that of the historical stream-resource transfer channel 201d in FIG. 2a, that is, an object address in a historical object address pair included in the remaining historical stream-resource transfer channel and the object address in the target object address pair also share an identical object address. Therefore, the object address in the historical object address pair included in the remaining historical stream-resource transfer channel may be combined with the newly generated target stream-resource transfer channel 202c. Optionally, the object address in the historical object address pair included in the remaining historical stream-resource transfer channel may also be combined with the newly generated historical stream-resource transfer channel 202d. A combination process is the same as the process described above.

[0058] Further, the blockchain node 10a generates an update stream-resource transfer channel according to the historical stream-resource transfer channel updated to the stream-resource transfer rate difference a and the target stream-resource transfer channel set to the stop transfer state. In FIG. 2a, the blockchain node 10a may determine the target stream-resource transfer channel 202c and the historical stream-resource transfer channel 202d as an update stream-resource transfer channel 20e.

[0059] The embodiment corresponding to FIG. 2a is used to describe a scenario in which the association between the historical object address pair and the target object address pair is the first association, and the historical stream-resource transfer rate is greater than the target stream-resource transfer rate. The subsequent processing procedure of the blockchain node when the historical stream-resource transfer rate is less than the target stream-resource transfer rate is the same as that when the historical stream-resource transfer rate is greater than the target stream-resource transfer rate. Specifically, the blockchain node may set the transfer state of the historical stream-resource transfer channel to the stop transfer state; determine a stream-resource transfer rate difference b between the target stream-resource transfer rate and the historical stream-resource transfer rate; update the target stream-resource transfer rate of the target stream-resource transfer

channel to the stream-resource transfer rate difference b; and generate an update stream-resource transfer channel according to the target stream-resource transfer channel updated to the stream-resource transfer rate difference b and the historical stream-resource transfer channel set to the stop transfer state.

[0060] In addition, there is another scenario in which the historical stream-resource transfer rate is equal to the target stream-resource transfer rate. In this case, the blockchain node sets the transfer state of the historical stream-resource transfer channel to the stop transfer state, and sets the transfer state of the target stream-resource transfer channel to the stop transfer state; and generates an update stream-resource transfer channel according to the target stream-resource transfer channel set to the stop transfer state and the historical stream-resource transfer channel set to the stop transfer state.

[0061] Further, the blockchain node performs, according to the update stream-resource transfer channel, stream-resource transfer processing associated with the update object address pair. The process is not described in detail herein. Refer to the description of operation S104 in the embodiment corresponding to FIG. 3 below.

[0062] In the embodiment corresponding to FIG. 2a, the channel cycle of the stream-resource transfer channel (including the target stream-resource transfer channel and the historical stream-resource transfer channel) is not considered. Therefore, the end timestamp of the stream-resource transfer channel, that is, the expiration timestamp of the stream-resource transfer channel, is not considered. Therefore, FIG. 2a may be applied to stream-resource transfer channels not set with a channel cycle and stream-resource transfer channels with the same end timestamp.

[0063] If the stream-resource transfer channel is set with a channel cycle, and the end timestamps corresponding to the target stream-resource transfer channel and the historical stream-resource transfer channel are different, refer to FIG. 2b. FIG. 2b is a second schematic diagram of a data processing scenario based on a blockchain according to an embodiment of the present application. In FIG. 2b, "address d1→address d2, 5/day, January 1, 2024" is used as an example of the target stream-resource transfer channel 201g, and "address d2→address d1, 10/day, January 1, 2023" is used as an example of the historical stream-resource transfer channel 201f.

[0064] In this embodiment of the present application, when combining the target stream-resource transfer channel and the historical stream-resource transfer channel, the start timestamps respectively corresponding to the target stream-resource transfer channel and the historical stream-resource transfer channel do not need to be considered, as long as the historical stream-resource transfer channel is still in a corresponding channel cycle of the historical stream-resource transfer

channel (which may include the start timestamp and the end timestamp), and the target stream-resource transfer channel is still in a corresponding channel cycle of the target stream-resource transfer channel (which may include the start timestamp and the end timestamp). Therefore, none of the target stream-resource transfer channel 201g and the historical stream-resource transfer channel 201f in FIG. 2b is marked with a start timestamp.

[0065] The end timestamp of the target stream-resource transfer channel 201g may be referred to as a first end timestamp, such as January 1, 2024 in FIG. 2b. The end timestamp of the historical stream-resource transfer channel 201f may be referred to as a second end timestamp, such as January 1, 2023 in FIG. 2b. For corresponding meanings of the target stream-resource transfer channel 201g and the historical stream-resource transfer channel 201f, refer to the description in FIG. 2a. In FIG. 2b, the second end timestamp is smaller than the first end timestamp. A valid time period of the combination (which may be equal to the first time period herein) is from a combination timestamp (that is, a combination moment of the stream-resource transfer channel) to the second end timestamp, that is, after the second end timestamp, the combination of the target stream-resource transfer channel 201g and the historical stream-resource transfer channel 201f becomes invalid.

[0066] The blockchain node may determine the first time period (equivalent to a time period in which the combination is valid) and the second time period (equivalent to a time period in which the combination is invalid) according to the first end timestamp (January 1, 2024 in FIG. 2b), the second end timestamp (January 1, 2023 in FIG. 2b), and a combination timestamp 20h (for example, October 24, 2022 in FIG. 2b). As shown in FIG. 2b, the first time period is (October 24, 2022, January 1, 2023), and the second time period is (January 2, 2023, January 1, 2024). Further, the blockchain node sets the transfer state of the target stream-resource transfer channel 201g in the first time period to the stop transfer state, to obtain a target stream-resource transfer channel 202g; and sets the transfer state of the target stream-resource transfer channel 201g in the second time period to a continuous transfer state, and maintains the target stream-resource transfer rate of the target stream-resource transfer channel 201g in the second time period, to obtain a target stream-resource transfer channel 203g. January 2, 2023 in the target stream-resource transfer channel 203g represents the minimum timestamp of the second time period.

[0067] The procedure of processing the historical stream-resource transfer channel 201f by the blockchain node is the same as that of processing the historical stream-resource transfer channel 201f by the blockchain node in FIG. 2a. Therefore, details are not described herein.

[0068] Further, the blockchain node may determine, as an update stream-resource transfer channel 20i, the historical stream-resource transfer channel updated to the stream-resource transfer rate difference a, the target stream-resource transfer channel that is in the first time period and that is set to the stop transfer state, and the target stream-resource transfer channel that is in the second time period and that maintains the target stream-resource transfer rate, that is, determine the historical stream-resource transfer channel 202f, the target stream-resource transfer channel 202g, and the target stream-resource transfer channel 203g as the update stream-resource transfer channel 20i.

[0069] The embodiment corresponding to FIG. 2b is further detailed description of the embodiment corresponding to FIG. 2a from the perspective of the end timestamp of the stream-resource transfer channel, and the above describes a scenario in which the second end timestamp is smaller than the first end timestamp. Another scenario is that the second end timestamp is greater than the first end timestamp. In this case, the blockchain node updates the historical stream-resource transfer rate of the historical stream-resource transfer channel in the third time period to the stream-resource transfer rate difference a. The maximum timestamp of the third time period is the second end timestamp. The third time period has the same meaning as the first time period. At the same time, the blockchain node maintains the historical stream-resource transfer rate of the historical stream-resource transfer channel in the fourth time period. The minimum timestamp of the fourth time period is greater than the second end timestamp, and the fourth time period and the third time period are adjacent time periods. The fourth time period has the same meaning as the second time period. Further, the blockchain node generates the update stream-resource transfer channel according to the historical stream-resource transfer channel that is in the third time period and that is updated to the stream-resource transfer rate difference a, the historical stream-resource transfer channel that is in the fourth time period and that maintains the historical stream-resource transfer rate, and the target stream-resource transfer channel set to the stop transfer state.

[0070] Based on FIG. 2a and FIG. 2b, it can be seen that in the present application, when the association between the target object address pair and the historical object address pair is the first association, the historical stream-resource transfer channel and the target stream-resource transfer channel may be combined, to obtain the update stream-resource transfer channel, which can simplify stream-resource transfer between object address pairs (including the target object address pair and the historical object address pair), and therefore can improve the efficiency of stream-resource transfer.

[0071] Further, FIG. 3 is a first schematic flowchart of a data processing method based on a blockchain according to an embodiment of the present application. The data processing method based on a blockchain may be executed by a computer device corresponding to a blockchain node. The blockchain node may be any blockchain node in the blockchain node cluster 10 in the embodiment corresponding to FIG. 1. FIG. 3 takes the blockchain node 10a as an example for description. As shown in FIG. 3, the data processing method based on a blockchain may at least include the following operation S101 to operation S104.

[0072] Step S101: Create a target stream-resource transfer channel for a stream-resource channel creation request in the blockchain based on a stream-resource transfer contract in the blockchain, the stream-resource channel creation request including a target object address pair, and the target stream-resource transfer channel providing, for the target object address pair, a function of performing stream-resource transfer based on the stream-resource transfer contract.

[0073] Specifically, before operation S101, the method further includes: obtaining the stream-resource channel creation request initiated by a first target object address; where the target object address pair includes the first target object address having a stream-resource transfer-out attribute; calling the stream-resource transfer contract in the blockchain according to the stream-resource channel creation request; when the target stream-resource transfer channel is successfully created, determining, based on the stream-resource transfer contract, a to-be-transferred target stream-resource corresponding to the target stream-resource transfer channel; and transferring the to-be-transferred target stream-resource from the first target object address to a contract address of the stream-resource transfer contract.

[0074] In the embodiments of the present application, a manner in which the blockchain node obtains the stream-resource channel creation request is not limited, and may be set according to an actual application scenario. The stream-resource channel creation request may be sent to the blockchain node in a point-to-point manner by a terminal device (for example, the terminal device 100a in FIG. 1) logged into the first target object address, or may be first sent by the terminal device to a corresponding service server (for example, the service server 101 in FIG. 1) and then forwarded by the service server to the blockchain node.

[0075] The stream-resource channel creation request includes a target object address pair, that is, a first target object address having a stream-resource transfer-out attribute, and a second target object address having a stream-resource transfer-in attribute. In the embodiments of the present application, other data included in the stream-resource channel creation request is not limited, and may be set according to an actual application scenario.

[0076] For example, the stream-resource channel creation request may include the target stream-resource transfer rate. In this case, the blockchain node may obtain the target stream-resource transfer rate in the stream-resource channel creation request. Alternatively, the request does not include the target stream-resource transfer rate when including the channel cycle (from the start timestamp to the end timestamp), the to-be-transferred target stream-resource, and the transfer type (for example, a fixed amount flow). In this case, the blockchain node may determine the target stream-resource transfer rate based on the channel cycle, the to-be-transferred target stream-resource, and the transfer type.

[0077] For example, the stream-resource channel creation request may include the to-be-transferred target stream-resource. In this case, the blockchain node may obtain the to-be-transferred target stream-resource in the stream-resource channel creation request. Alternatively, the request does not include the to-be-transferred target stream-resource when including the channel cycle (from the start timestamp to the end timestamp), the target stream-resource transfer rate, and the transfer type (for example, a fixed amount flow). In this case, the blockchain node may determine the to-be-transferred target stream-resource based on the channel cycle, the target stream-resource transfer rate, and the transfer type.

[0078] To transfer stream-resources, the blockchain node transfers the to-be-transferred target stream-resource from the first target object address to a contract address of the stream-resource transfer contract.

[0079] Step S102: Obtain, in the blockchain according to the target object address pair, a historical stream-resource transfer channel including a historical object address pair, the target object address pair and the historical object address pair sharing at least one identical object address.

[0080] Herein, the historical stream-resource transfer channel provides the function of stream-resource transfer based on the stream-resource transfer contract for a historical object address pair.

[0081] Specifically, the target object address pair includes a first target object address having a stream-resource transfer-out attribute, and a second target object address having a stream-resource transfer-in attribute. The historical object address pair includes a first historical object address pair, a second historical object address pair, and a third historical object address pair. A first stream-resource transfer channel including the first historical object address pair is obtained in the blockchain; where the first historical object address pair includes a first historical object address that has the stream-resource transfer-in attribute and that is the same as the first target object

address. A second stream-resource transfer channel including the second historical object address pair is obtained in the blockchain; where the second historical object address pair includes a second historical object address that has the stream-resource transfer-out attribute and that is the same as the second target object address. The first historical object address pair is different from the second historical object address pair. A third stream-resource transfer channel including the third historical object address pair is obtained in the blockchain; where the third historical object address pair is the same as the target object address pair. At least one of the first stream-resource transfer channel, the second stream-resource transfer channel, and the third stream-resource transfer channel is determined as the historical stream-resource transfer channel.

[0082] The blockchain node traverses created stream-resource transfer channels in the blockchain based on the stream-resource transfer contract, and determines a stream-resource transfer channel that meets the condition among the created stream-resource transfer channels as the historical stream-resource transfer channel. The condition may be divided into three types:

[0083] 1. A first stream-resource transfer channel includes a first historical object address, where the first historical object address has a stream-resource transfer-in attribute and is the same as the first target object address.

[0084] 2. A second stream-resource transfer channel includes a second historical object address, where the second historical object address has a stream-resource transfer-out attribute and is the same as the second target object address.

[0085] 3. A third stream-resource transfer channel includes a third historical object address pair, where the third historical object address pair is the same as the target object address pair.

[0086] The historical stream-resource transfer channel 201d and the historical stream-resource transfer channel 201f in FIG. 2a and FIG. 2b may be divided into the first type of condition or the second type of condition.

[0087] Step S103: Combine the historical stream-resource transfer channel and the target stream-resource transfer channel to obtain an update stream-resource transfer channel, the update stream-resource transfer channel including an update object address pair.

[0088] Specifically, a historical stream-resource transfer rate of the historical stream-resource transfer channel is determined, and a target stream-resource transfer rate of the target stream-resource transfer channel is determined; and the historical stream-resource transfer channel and the target stream-resource transfer channel are combined according to the target object address

pair, the historical object address pair, the historical stream-resource transfer rate, and the target stream-resource transfer rate, to obtain the update stream-resource transfer channel.

[0089] A specific process of combining the historical stream-resource transfer channel and the target stream-resource transfer channel according to the target object address pair, the historical object address pair, the historical stream-resource transfer rate, and the target stream-resource transfer rate, to obtain the update stream-resource transfer channel may include: obtaining a first target object address having a stream-resource transfer-out attribute, and a second target object address having a stream-resource transfer-in attribute in the target object address pair; obtaining a fifth historical object address having the stream-resource transfer-out attribute and a sixth historical object address having the stream-resource transfer-in attribute in the historical object address pair; setting the transfer state of the target stream-resource transfer channel to the stop transfer state in response to the first target object address being the same as the fifth historical object address, and the second target object address being the same as the sixth historical object address; determining a stream-resource transfer rate sum of the historical stream-resource transfer rate and the target stream-resource transfer rate; updating the historical stream-resource transfer rate of the historical stream-resource transfer channel to the stream-resource transfer rate sum; generating the update stream-resource transfer channel according to the historical stream-resource transfer channel updated to the stream-resource transfer rate sum and the target stream-resource transfer channel set to the stop transfer state. For example, the stream-resource transfer rate sum is identified as a stream-resource transfer rate sum c.

[0090] A specific process of updating the historical stream-resource transfer rate of the historical stream-resource transfer channel to the stream-resource transfer rate sum may include: obtaining a first end timestamp of the target stream-resource transfer channel, and obtaining a third end timestamp of the historical stream-resource transfer channel; updating the historical stream-resource transfer rate of the historical stream-resource transfer channel in a fifth time period to the stream-resource transfer rate sum in response to the third end timestamp being less than the first end timestamp; where the maximum timestamp of the fifth time period is the third end timestamp; updating the historical stream-resource transfer rate of the historical stream-resource transfer channel in a sixth time period to the target stream-resource transfer rate; where the maximum timestamp of the sixth time period is the first end timestamp, and the sixth time period and the fifth time period are adjacent time periods; and a specific process of generating the update stream-resource transfer channel according to the historical stream-resource transfer channel updated to the stream-resource transfer rate sum and the target stream-resource transfer channel set to the stop transfer state may include: generating the update stream-resource transfer channel according to the

historical stream-resource transfer channel that is in the fifth time period and that is updated to the stream-resource transfer rate sum, the historical stream-resource transfer channel that is in the sixth time period and that is updated to the target stream-resource transfer rate, and the target stream-resource transfer channel set to the stop transfer state.

[0091] In response to a transfer rate unit of the historical stream-resource transfer rate being different from a transfer rate unit of the target stream-resource transfer rate, the historical stream-resource transfer rate is adjusted according to the transfer rate unit of the target stream-resource transfer rate, to obtain an adjusted stream-resource transfer rate; where a transfer rate unit of the adjusted stream-resource transfer rate is the same as the transfer rate unit of the target stream-resource transfer rate; and the historical stream-resource transfer channel and the target stream-resource transfer channel are combined according to the target object address pair, the historical object address pair, the adjusted stream-resource transfer rate, and the target stream-resource transfer rate, to obtain the update stream-resource transfer channel.

[0092] FIG. 4 is a third schematic diagram of a data processing scenario based on a blockchain according to an embodiment of the present application. As shown in FIG. 4, the blockchain node 10a determines the historical stream-resource transfer rate of the historical stream-resource transfer channel 401d (the example in FIG. 4 is 5/half a day), and determines the target stream-resource transfer rate of the target stream-resource transfer channel 401c (the example in FIG. 4 is 5/day). For the corresponding meanings of the historical stream-resource transfer channel 401d and the target stream-resource transfer channel 401c, refer to the description in the embodiment corresponding to FIG. 2a. Details are not described herein again.

[0093] The transfer rate unit of the historical stream-resource transfer rate is half a day, and the transfer rate unit of the target stream-resource transfer rate is per day. Therefore, the transfer rate unit of the historical stream-resource transfer rate is different from the transfer rate unit of the target stream-resource transfer rate. In this case, the blockchain node 10a may adjust the historical stream-resource transfer rate according to the transfer rate unit of the target stream-resource transfer rate to obtain the adjusted stream-resource transfer rate (which may be referred to as the first adjusted stream-resource transfer rate for distinguishing). The adjusted stream-resource transfer rate is 10/day in FIG. 4.

[0094] Optionally, the blockchain node 10a may adjust the target stream-resource transfer rate according to the transfer rate unit of the historical stream-resource transfer rate to obtain the second adjusted stream-resource transfer rate.

[0095] In this embodiment of the present application, the time unit of the transfer rate unit is not limited and may be set according to an actual application scenario, and includes but is not limited to second, minute, hour, day, month, and the like.

[0096] Further, the blockchain node 10a may combine the historical stream-resource transfer channel 401d and the target stream-resource transfer channel 401c according to the target object address pair, the historical object address pair, the adjusted stream-resource transfer rate (which is the historical stream-resource transfer rate if the transfer rate unit of the historical stream-resource transfer rate is the same as the transfer rate unit of the target stream-resource transfer rate), and the target stream-resource transfer rate, to obtain an update stream-resource transfer channel. Specifically, the blockchain node 10a obtains the first target object address having the stream-resource transfer-out attribute and the second target object address having the stream-resource transfer-in attribute in the target object address pair, such as the address d1 and the address d2 as shown in FIG. 4. At the same time, the blockchain node 10a obtains a fifth historical object address having a stream-resource transfer-out attribute, and a sixth historical object address having a stream-resource transfer-in attribute in the historical object address pair, such as the address d1 and the address d2 in FIG. 4.

[0097] The blockchain node 10a may determine the association between the historical object address pair and the target object address pair, and when combining the historical stream-resource transfer channel and the target stream-resource transfer channel, may determine the update object address pair based on the association. In FIG. 4, the association may be a second association 40j. The second association 40j may represent that the first target object address is the same as the fifth historical object address, and the second target object address is the same as the sixth historical object address. In this case, the update object address pair is {address d1, address d2}. In this scenario, the blockchain node 10a may set the transfer state of the target stream-resource transfer channel 401c to the stop transfer state, as shown in FIG. 4, and obtain the target stream-resource transfer channel 402c. "Stop" in the target stream-resource transfer channel 402c indicates the stop transfer state.

[0098] At the same time, the blockchain node 10a determines the stream-resource transfer rate sum c of the adjusted stream-resource transfer rate and the target stream-resource transfer rate. In FIG. 4, the stream-resource transfer rate sum c is 15/day and is abbreviated as "sum c=15/day in FIG. 4. Further, the blockchain node 10a updates the historical stream-resource transfer rate of the historical stream-resource transfer channel 401d (5/half a day in FIG. 4) to the stream-resource transfer rate sum c, so that the historical stream-resource transfer channel 402d may be obtained.

[0099] Optionally, the blockchain node 10a may set the transfer state of the historical stream-resource transfer channel 401d to the stop transfer state, and update the historical stream-resource transfer rate (5/day in FIG. 4) of the target stream-resource transfer channel 401c to the stream-resource transfer rate sum c.

[00100] FIG. 4 takes one target stream-resource transfer channel (that is, the target stream-resource transfer channel 401c) and one historical stream-resource transfer channel (that is, the historical stream-resource transfer channel 401d) as an example for description. If there are multiple historical stream-resource transfer channels, the processing procedure of the remaining historical stream-resource transfer channel is the same as that of the historical stream-resource transfer channel 401d in FIG. 4, that is, an object address in a historical object address pair included in the remaining historical stream-resource transfer channel and the object address in the target object address pair also share an identical object address. Therefore, the object address in the historical object address pair included in the remaining historical stream-resource transfer channel may be combined with the newly generated historical stream-resource transfer channel 402d. The association between the historical object address pair included in the remaining historical stream-resource transfer channel and the target object address pair may be the first association 20j in FIG. 2a, or may be the second association 40j described in this operation, or may be the third association in the embodiment corresponding to FIG. 7 below, or may be the fourth association in the embodiment corresponding to FIG. 8 below. That is, historical object address pairs included in different historical stream-resource transfer channels may have different associations with the target object address pair.

[00101] Further, the blockchain node 10a generates an update stream-resource transfer channel according to the historical stream-resource transfer channel updated to the stream-resource transfer rate sum c and the target stream-resource transfer channel set to the stop transfer state. In FIG. 4, the blockchain node 10a may determine the target stream-resource transfer channel 402c and the historical stream-resource transfer channel 402d as an update stream-resource transfer channel 40e.

[00102] In the embodiment corresponding to FIG. 4, the channel cycle of the stream-resource transfer channel (including the target stream-resource transfer channel and the historical stream-resource transfer channel) is not considered. Therefore, the end timestamp of the stream-resource transfer channel, that is, the expiration timestamp of the stream-resource transfer channel, is not considered. Therefore, FIG. 4 may be applied to stream-resource transfer channels not set with a channel cycle and stream-resource transfer channels with the same end timestamp.

[00103] If the stream-resource transfer channel is set with a channel cycle, and the end timestamps corresponding to the target stream-resource transfer channel and the historical stream-resource transfer channel are different, refer to FIG. 5. FIG. 5 is a fourth schematic diagram of a data processing scenario based on a blockchain according to an embodiment of the present application. In FIG. 5, "address d1→address d2, 5/day, January 1, 2024" is used as an example of the target stream-resource transfer channel 401g, and "address d1→address d2, 5/half a day, January 1, 2023" is used as an example of the historical stream-resource transfer channel 201f. For the corresponding meanings of the target stream-resource transfer channel 401g and the historical stream-resource transfer channel 401f, refer to the description of the target stream-resource transfer channel 201g and the historical stream-resource transfer channel 201f in FIG. 2b. Details are not described herein. None of the target stream-resource transfer channel 401g and the historical stream-resource transfer channel 401f in FIG 5 is marked with a start timestamp.

[00104] The end timestamp of the target stream-resource transfer channel 401g may be referred to as a first end timestamp, such as January 1, 2024 in FIG. 5. The end timestamp of the historical stream-resource transfer channel 401f may be referred to as a third end timestamp, such as January 1, 2023 in FIG. 5. In FIG. 5, the third end timestamp is smaller than the first end timestamp. A valid time period of the combination (which may be equal to the fifth time period herein) is from a combination timestamp (that is, a combination moment of the stream-resource transfer channel) to the third end timestamp, that is, after the third end timestamp, the combination of the target stream-resource transfer channel 401g and the historical stream-resource transfer channel 401f becomes invalid.

[00105] The blockchain node may determine the fifth time period (equivalent to a time period in which the combination is valid) and the sixth time period (equivalent to a time period in which the combination is invalid) according to the first end timestamp (January 1, 2024 in FIG. 5), the third end timestamp (January 1, 2023 in FIG. 5), and a combination timestamp 40h (for example, October 24, 2022 in FIG. 5). As shown in the example of FIG. 5, the fifth time period is (October 24, 2022, January 1, 2023), and the sixth time period is (January 2, 2023, January 1, 2024). Further, the blockchain node updates the historical stream-resource transfer rate of the historical stream-resource transfer channel 401f in the fifth time period to the stream-resource transfer rate sum c, so that the historical stream-resource transfer channel 402f may be obtained; and updates the historical stream-resource transfer rate of the historical stream-resource transfer channel 401f in the sixth time period to the target stream-resource transfer rate, so that the historical stream-resource transfer channel 403f may be obtained. January 2, 2023 in the historical stream-

resource transfer channel 403f represents the minimum timestamp of the sixth time period. Since the time unit in the example is per day, the unit of the timestamp is also per day.

[00106] Further, the blockchain node may generate an update stream-resource transfer channel 40i according to the historical stream-resource transfer channel that is in the fifth time period and that is updated to the stream-resource transfer rate sum c, the historical stream-resource transfer channel that is in the sixth time period and that is updated to the target stream-resource transfer rate, and the target stream-resource transfer channel set to the stop transfer state, that is, determine the historical stream-resource transfer channel 402f, the target stream-resource transfer channel 402g, and the historical stream-resource transfer channel 403f as the update stream-resource transfer channel 40i. Since the historical object address pair and the target object address pair are the same, the processing procedure of the blockchain node when the third end timestamp is greater than the first end timestamp is the same as that when the third end timestamp is less than the first end timestamp. Therefore, details are not described.

[00107] Step S104: Perform, according to the update stream-resource transfer channel, stream-resource transfer processing associated with the update object address pair.

[00108] Specifically, after the target stream-resource transfer channel is successfully created, the blockchain node transfers, from the first target object address to a contract address based on the stream-resource transfer contract, a to-be-transferred target stream-resource corresponding to the target stream-resource transfer channel. Similarly, the blockchain node transfers, from the historical object address having the stream-resource transfer-out attribute to a contract address based on the stream-resource transfer contract, a to-be-transferred historical stream-resource corresponding to the historical stream-resource transfer channel.

[00109] Since the contract address possesses the to-be-transferred target stream-resource and the to-be-transferred historical stream-resource, the blockchain node may continuously transfer virtual resources according to the update stream-resource transfer rate corresponding to the update stream-resource transfer channel, for example, the stream-resource transfer rate sum c (that is, 15/day) in the historical stream-resource transfer channel 402d in FIG. 4, and for another example, the stream-resource transfer rate difference a (that is, 5/day) in the historical stream-resource transfer channel 202d in FIG. 2a. The blockchain node continuously performs transfer processing on virtual resources, but does not transfer the resource in real time to the object address (abbreviated as an update transfer-in object address) having the stream-resource transfer-in attribute in the update object address pair. Instead, when the channel cycle corresponding to the update stream-resource transfer channel ends, or when the update transfer-in object address initiates a

stream-resource extraction request, the blockchain node performs settlement processing on the stream-resource, and when the settlement succeeds, transfers a stream-resource settlement result to the update transfer-in object address.

[00110] According to the above embodiment, since the update object address pair is obtained based on the combination of the target object address pair and the historical object address pair, stream-resource transfer processing respectively corresponding to the historical stream-resource transfer channel and the target stream-resource transfer channel may be implemented through the update stream-resource transfer channel. Therefore, the present application can reduce the complexity of the stream-resource transfer channel in the blockchain, thereby improving the efficiency of stream-resource transfer.

[00111] FIG. 6 is a schematic interaction diagram of a data processing method based on a blockchain according to an embodiment of the present application. As shown in FIG. 6:

[00112] Step A: A transfer-out terminal device sends a stream-resource channel creation request to a service server. The transfer-out terminal device is a terminal device logged into an object address (for example, the first target object address in the example of the present application) having a stream-resource transfer-out attribute, such as the terminal device 100a in FIG. 1.

[00113] Step B: The service server sends the stream-resource channel creation request to a blockchain node.

[00114] The service server is a background server corresponding to a client installed on the transfer-out terminal device. Specifically, on the premise that the first target object address has permission, the service server may sign the stream-resource channel creation request based on a private key of the first target object address, to obtain a first signature message, call the stream-resource transfer contract on the blockchain, create a stream-resource transfer interface, and send the stream-resource channel creation request carrying the first signature message to the blockchain node. Optionally, the transfer-out terminal device can directly send the stream-resource channel creation request to the blockchain node.

[00115] Step C: The blockchain node generates a target stream-resource transfer channel.

[00116] Step D: The blockchain node obtains a historical stream-resource transfer channel.

[00117] Based on the stream-resource transfer contract, the blockchain node can not only create the target stream-resource transfer channel, but also traverse stream-resource transfer channels that have been created in the blockchain, to obtain the historical stream-resource transfer channel.

[00118] Step E: The blockchain node combines the target stream-resource transfer channel and the historical stream-resource transfer channel to generate an update stream-resource transfer channel, and determines, based on a target stream-resource transfer rate and a historical stream-resource transfer rate, an update stream-resource transfer rate corresponding to the update stream-resource transfer channel.

[00119] Step F: The blockchain node continuously transfers stream-resources according to the update stream-resource transfer channel. As the block height on the chain increases, the update stream-resource transfer channel transfers digital assets on the blockchain.

[00120] Step G: A transfer-in terminal device sends a stream-resource extraction request to the blockchain node.

[00121] The transfer-in terminal device is a terminal device logged into an object address (for example, the second target object address in the example of the present application) having a stream-resource transfer-in attribute, such as the terminal device 100b in FIG. 1. Similarly, the stream-resource extraction request may be forwarded to the blockchain node through the service server.

[00122] Step H: The blockchain node determines a stream-resource transfer channel according to the stream-resource extraction request, and performs stream-resource settlement processing, to obtain a stream-resource settlement result 1.

[00123] In the example of the present application, the stream-resource settlement method is not limited and may be set according to an actual application scenario, and includes but is not limited to: based on a time (up to an obtaining timestamp of the stream-resource extraction request) for continuous transfer of stream-resources through the stream-resource transfer channel, and based on a block height difference corresponding to the stream-resource transfer channel. The block height difference is a difference between a block height added when the stream-resource transfer channel is generated and a block height added by the stream-resource extraction request.

[00124] Step I: The blockchain node performs transfer-in processing on the stream-resource settlement result 1.

[00125] Step J: When the stream-resource transfer channel expires, stop stream-resource transfer processing.

[00126] Object address pairs corresponding to the stream-resource transfer channel may respectively initiate transaction requests to retrieve their own resources, and finally the stream-resource transfer channel is canceled.

[00127] To sum up, since the update object address pair is obtained based on the combination of the target object address pair and the historical object address pair, stream-resource transfer processing respectively corresponding to the historical stream-resource transfer channel and the target stream-resource transfer channel may be implemented through the update stream-resource transfer channel. Therefore, the present application can reduce the complexity of the stream-resource transfer channel in the blockchain, thereby improving the efficiency of stream-resource transfer.

[00128] Further, FIG. 7 is a second schematic flowchart of a data processing method according to an embodiment of the present application. As shown in FIG. 7, the process of the data processing method includes the following operation S201 to operation S207, and operation S201 to operation S207 are a specific embodiment of operation S103 in the embodiment corresponding to FIG. 3.

[00129] Step S201: Obtain a first target object address having a stream-resource transfer-out attribute, and a second target object address having a stream-resource transfer-in attribute in the target object address pair.

[00130] Step S202: Obtain a seventh historical object address having the stream-resource transfer-out attribute, and an eighth historical object address having the stream-resource transfer-in attribute in the historical object address pair.

[00131] For specific implementation process of operation S201 and operation S202, refer to operation S103 in the corresponding embodiment in FIG. 3, and details are not described herein.

[00132] Step S203: Compare the historical stream-resource transfer rate with the target stream-resource transfer rate in response to the first target object address being the same as the eighth historical object address, and the second target object address being different from the seventh historical object address.

[00133] Specifically, the blockchain node may determine the association between the historical object address pair and the target object address pair. In this embodiment of the present application, the association may be a third association. The third association may represent that the first target object address is the same as the eighth historical object address, and the second target object address is different from the seventh historical object address. In this scenario, the blockchain node compares the historical stream-resource transfer rate with the target stream-resource transfer rate. If the historical stream-resource transfer rate is greater than the target stream-resource transfer rate, operation S204 to operation S207 are executed.

[00134] In another scenario, if the historical stream-resource transfer rate is less than the target stream-resource transfer rate, the blockchain node determines a stream-resource transfer rate difference e between the target stream-resource transfer rate and the historical stream-resource transfer rate; updates the target stream-resource transfer rate of the target stream-resource transfer channel to the stream-resource transfer rate difference e ; and updates the eighth historical object address in the historical stream-resource transfer channel to the second target object address; generates the update stream-resource transfer channel according to the target stream-resource transfer channel updated to the stream-resource transfer rate difference e and the historical stream-resource transfer channel updated to the second target object address.

[00135] In another scenario, if the historical stream-resource transfer rate is equal to the target stream-resource transfer rate, the blockchain node sets the transfer state of the historical stream-resource transfer channel to the stop transfer state; updates the first target object address in the target stream-resource transfer channel to the seventh historical object address; and generates an update stream-resource transfer channel according to the target stream-resource transfer channel updated to the seventh historical object address and the historical stream-resource transfer channel set to the stop transfer state.

[00136] Step S204: Determine a second stream-resource transfer rate difference between the historical stream-resource transfer rate and the target stream-resource transfer rate in response to the historical stream-resource transfer rate being greater than the target stream-resource transfer rate.

[00137] For example, the second stream-resource transfer rate difference is the stream-resource transfer rate difference d .

[00138] Step S205: Update the historical stream-resource transfer rate of the historical stream-resource transfer channel to the second stream-resource transfer rate difference.

[00139] Specifically, a first end timestamp of the target stream-resource transfer channel is obtained, and a fourth end timestamp of the historical stream-resource transfer channel is obtained; the historical stream-resource transfer rate of the historical stream-resource transfer channel in a seventh time period is updated to the second stream-resource transfer rate difference in response to the fourth end timestamp being greater than the first end timestamp; where the maximum timestamp of the seventh time period is the first end timestamp, and the historical stream-resource transfer rate of the historical stream-resource transfer channel in the eighth time period is maintained; where the maximum timestamp of the eighth time period is the fourth end timestamp, and the eighth time period and the seventh time period are adjacent time periods.

[00140] Step S206: Update the first target object address in the target stream-resource transfer channel to the seventh historical object address.

[00141] Specifically, a first end timestamp of the target stream-resource transfer channel is obtained, and a fourth end timestamp of the historical stream-resource transfer channel is obtained; the first target object address in the target stream-resource transfer channel in the ninth time period is updated to the seventh historical object address in response to the fourth end timestamp being less than the first end timestamp; where the maximum timestamp of the ninth time period is the fourth end timestamp. the first target object address of the target stream-resource transfer channel in a tenth time period is maintained; where the maximum timestamp of the tenth time period is the first end timestamp, and the tenth time period and the ninth time period are adjacent time periods.

[00142] Step S207: Generate the update stream-resource transfer channel according to the historical stream-resource transfer channel updated to the second stream-resource transfer rate difference and the target stream-resource transfer channel updated to the seventh historical object address.

[00143] Specifically, the update stream-resource transfer channel is generated according to the historical stream-resource transfer channel that is in the seventh time period and that is updated to the second stream-resource transfer rate difference, the historical stream-resource transfer channel that is in the eighth time period and that maintains the historical stream-resource transfer rate, and the target stream-resource transfer channel updated to the seventh historical object address.

[00144] Specifically, the update stream-resource transfer channel is generated according to the historical stream-resource transfer channel updated to the second stream-resource transfer rate difference, the target stream-resource transfer channel that is in the ninth time period and that is updated to the seventh historical object address, and the target stream-resource transfer channel that is in the tenth time period and that maintains the first target object address.

[00145] To sum up, since the update object address pair is obtained based on the combination of the target object address pair and the historical object address pair, stream-resource transfer processing respectively corresponding to the historical stream-resource transfer channel and the target stream-resource transfer channel may be implemented through the update stream-resource transfer channel. Therefore, the present application can reduce the complexity of the stream-resource transfer channel in the blockchain, thereby improving the efficiency of stream-resource transfer.

[00146] Further, FIG. 8 is a third schematic flowchart of a data processing method according to an embodiment of the present application. As shown in FIG. 8, the process of the data processing method includes the following operation S301 to operation S307, and operation S301 to operation S307 are a specific embodiment of operation S103 in the embodiment corresponding to FIG. 3.

[00147] Step S301: Obtain a first target object address having a stream-resource transfer-out attribute, and a second target object address having a stream-resource transfer-in attribute in the target object address pair.

[00148] Step S302: Obtain a ninth historical object address having the stream-resource transfer-out attribute and a tenth historical object address having the stream-resource transfer-in attribute in the historical object address pair.

[00149] Step S303: Compare the historical stream-resource transfer rate with the target stream-resource transfer rate in response to the first target object address being different from the tenth historical object address, and the second target object address being the same as the ninth historical object address.

[00150] Specifically, The blockchain node may determine the association between the historical object address pair and the target object address pair. In this embodiment of the present application, the association may be a fourth association. The fourth association may represent that the first target object address is different from the tenth historical object address, and the second target object address is the same as the ninth historical object address. In this scenario, the blockchain node compares the historical stream-resource transfer rate with the target stream-resource transfer rate. If the historical stream-resource transfer rate is greater than the target stream-resource transfer rate, operation S304 to operation S307 are executed.

[00151] In another scenario, if the historical stream-resource transfer rate is less than the target stream-resource transfer rate, the blockchain node determines a stream-resource transfer rate difference g between the target stream-resource transfer rate and the historical stream-resource transfer rate; updates the target stream-resource transfer rate of the target stream-resource transfer channel to the stream-resource transfer rate difference g ; and updates the ninth historical object address in the historical stream-resource transfer channel to the first target object address; and generates the update stream-resource transfer channel according to the target stream-resource transfer channel updated to the stream-resource transfer rate difference g and the historical stream-resource transfer channel updated to the first target object address.

[00152] In another scenario, if the historical stream-resource transfer rate is equal to the target stream-resource transfer rate, the blockchain node sets the transfer state of the historical stream-resource transfer channel to the stop transfer state; updates the second target object address in the target stream-resource transfer channel to the tenth historical object address; and generates an update stream-resource transfer channel according to the target stream-resource transfer channel updated to the tenth historical object address and the historical stream-resource transfer channel set to the stop transfer state.

[00153] Step S304: Determine a third stream-resource transfer rate difference between the historical stream-resource transfer rate and the target stream-resource transfer rate in response to the historical stream-resource transfer rate being greater than the target stream-resource transfer rate.

[00154] For example, the third stream-resource transfer rate difference is a stream-resource transfer rate difference f .

[00155] Step S305: Update the historical stream-resource transfer rate of the historical stream-resource transfer channel to the third stream-resource transfer rate difference.

[00156] Specifically, a first end timestamp of the target stream-resource transfer channel is obtained, and a fifth end timestamp of the historical stream-resource transfer channel is obtained; the historical stream-resource transfer rate of the historical stream-resource transfer channel in an eleventh time period is updated to the third stream-resource transfer rate difference in response to the fifth end timestamp being greater than the first end timestamp; where the maximum timestamp of the eleventh time period is the first end timestamp, the historical stream-resource transfer rate of the historical stream-resource transfer channel in the twelfth time period is maintained; where the maximum timestamp of the twelfth time period is the fifth end timestamp, and the twelfth time period and the eleventh time period are adjacent time periods.

[00157] Step S306: Update the second target object address in the target stream-resource transfer channel to the tenth historical object address.

[00158] Specifically, a first end timestamp of the target stream-resource transfer channel is obtained, and a fifth end timestamp of the historical stream-resource transfer channel is obtained; the second target object address in the target stream-resource transfer channel in the thirteenth time period is updated to the tenth historical object address in response to the fifth end timestamp being less than the first end timestamp; where the maximum timestamp of the thirteenth time period is the fifth end timestamp, the second target object address of the target stream-resource transfer channel in a fourteenth time period is maintained; where the maximum timestamp of the fourteenth time

period is the first end timestamp, and the fourteenth time period and the thirteenth time period are adjacent time periods.

[00159] Step S307: Generate the update stream-resource transfer channel according to the historical stream-resource transfer channel updated to the third stream-resource transfer rate difference and the target stream-resource transfer channel updated to the tenth historical object address.

[00160] Specifically, the update stream-resource transfer channel is generated according to the historical stream-resource transfer channel that is in the eleventh time period and that is updated to the third stream-resource transfer rate difference, the historical stream-resource transfer channel that is in the twelfth time period and that maintains the historical stream-resource transfer rate, and the target stream-resource transfer channel updated to the tenth historical object address.

[00161] Specifically, the update stream-resource transfer channel is generated according to the historical stream-resource transfer channel updated to the third stream-resource transfer rate difference, the target stream-resource transfer channel that is in the thirteenth time period and that is updated to the tenth historical object address, and the target stream-resource transfer channel that is in the fourteenth time period and that maintains the second target object address.

[00162] To sum up, since the update object address pair is obtained based on the combination of the target object address pair and the historical object address pair, stream-resource transfer processing respectively corresponding to the historical stream-resource transfer channel and the target stream-resource transfer channel may be implemented through the update stream-resource transfer channel. Therefore, the present application can reduce the complexity of the stream-resource transfer channel in the blockchain, thereby improving the efficiency of stream-resource transfer.

[00163] Further, FIG. 9 is a schematic structural diagram of a data processing apparatus 1 based on a blockchain according to an embodiment of the present application. The data processing apparatus based on a blockchain may run on a master node with consensus permission in a blockchain network. The data processing apparatus 1 based on a blockchain may be configured to execute corresponding operations in the method provided by the embodiments of the present application. As shown in FIG. 9, the data processing apparatus 1 based on a blockchain may include: a channel creation module 11, a first obtaining module 12, a first processing module 13, and a second processing module 14.

[00164] The channel creation module 11 is configured to create a target stream-resource transfer channel for a stream-resource channel creation request in the blockchain based on a stream-resource transfer contract in the blockchain, the stream-resource channel creation request including a target object address pair, and the target stream-resource transfer channel providing, for the target object address pair, a function of performing stream-resource transfer based on the stream-resource transfer contract.

[00165] The first obtaining module 12 is configured to obtain, in the blockchain according to the target object address pair, a historical stream-resource transfer channel including a historical object address pair, the target object address pair and the historical object address pair sharing at least one identical object address.

[00166] The first processing module 13 is configured to combine the historical stream-resource transfer channel and the target stream-resource transfer channel to obtain an update stream-resource transfer channel, the update stream-resource transfer channel including an update object address pair.

[00167] The second processing module 14 is configured to perform, according to the update stream-resource transfer channel, stream-resource transfer processing associated with the update object address pair.

[00168] For specific implementations of functions of the channel creation module 11, the first obtaining module 12, the first processing module 13, and the second processing module 14, refer to operation S101 to operation S104 of the above corresponding embodiment in FIG. 3. Details are not described herein.

[00169] Referring to FIG. 9 again, the data processing apparatus 1 based on a blockchain may further include: a second obtaining module 15, a contract calling module 16, and a resource transfer-out module 17.

[00170] The second obtaining module 15 is configured to obtain the stream-resource channel creation request initiated by a first target object address. The target object address pair includes the first target object address having a stream-resource transfer-out attribute.

[00171] The contract calling module 16 is configured to call the stream-resource transfer contract in the blockchain according to the stream-resource channel creation request.

[00172] The contract calling module 16 is further configured to: when the target stream-resource transfer channel is successfully created, determine, based on the stream-resource transfer

contract, a to-be-transferred target stream-resource corresponding to the target stream-resource transfer channel.

[00173] The resource transfer-out module 17 is configured to transfer the to-be-transferred target stream-resource from the first target object address to a contract address of the stream-resource transfer contract.

[00174] For specific implementations of functions of the second obtaining module 15, the contract calling module 16, and the resource transfer-out module 17, refer to operation S101 of the above corresponding embodiment in FIG. 3. Details are not described herein.

[00175] Referring to FIG. 9 again, the target object address pair includes a first target object address having a stream-resource transfer-out attribute, and a second target object address having a stream-resource transfer-in attribute. The historical object address pair includes a first historical object address pair, a second historical object address pair, and a third historical object address pair.

[00176] The first obtaining module 12 may include: a first obtaining unit 121, a second obtaining unit 122, a third obtaining unit 123, and a first determining unit 124.

[00177] The first obtaining unit 121 is configured to obtain, in the blockchain, a first stream-resource transfer channel including the first historical object address pair; where the first historical object address pair includes a first historical object address that has the stream-resource transfer-in attribute and that is the same as the first target object address.

[00178] The second obtaining unit 122 is configured to obtain, in the blockchain, a second stream-resource transfer channel including the second historical object address pair; where the second historical object address pair includes a second historical object address that has the stream-resource transfer-out attribute and that is the same as the second target object address.

[00179] The third obtaining unit 123 is configured to obtain, in the blockchain, a third stream-resource transfer channel including the third historical object address pair; where the third historical object address pair is the same as the target object address pair.

[00180] The first determining unit 124 is configured to determine at least one of the first stream-resource transfer channel, the second stream-resource transfer channel, and the third stream-resource transfer channel as the historical stream-resource transfer channel.

[00181] For specific implementations of functions of the first obtaining unit 121, the second obtaining unit 122, the third obtaining unit 123, and the first determining unit 124, refer to operation S102 of the above corresponding embodiment in FIG. 3. Details are not described herein.

[00182] Referring to FIG. 9 again, the first processing module 13 may include: a second determining unit 131 and a channel combining unit 132.

[00183] The second determining unit 131 is configured to determine a historical stream-resource transfer rate of the historical stream-resource transfer channel, and determine a target stream-resource transfer rate of the target stream-resource transfer channel.

[00184] The channel combining unit 132 is configured to combine the historical stream-resource transfer channel and the target stream-resource transfer channel according to the target object address pair, the historical object address pair, the historical stream-resource transfer rate, and the target stream-resource transfer rate, to obtain the update stream-resource transfer channel.

[00185] For specific implementations of functions of the second determining unit 131 and the channel combining unit 132, refer to operation S103 of the above corresponding embodiment in FIG. 3. Details are not described herein.

[00186] Referring to FIG. 9 again, the channel combining unit 132 may include: a first obtaining subunit 1321, a first comparison subunit 1322, a first setting subunit 1323, a first determining subunit 1324, a first update subunit 1325, and a first generation subunit 1326.

[00187] The first obtaining subunit 1321 is configured to obtain a first target object address having a stream-resource transfer-out attribute, and a second target object address having a stream-resource transfer-in attribute in the target object address pair.

[00188] The first obtaining subunit 1321 is further configured to obtain a third historical object address having the stream-resource transfer-out attribute and a fourth historical object address having the stream-resource transfer-in attribute in the historical object address pair.

[00189] The first comparison subunit 1322 is configured to compare the historical stream-resource transfer rate with the target stream-resource transfer rate in response to the first target object address being the same as the fourth historical object address, and the second target object address being the same as the third historical object address.

[00190] The first setting subunit 1323 is configured to set a transfer state of the target stream-resource transfer channel to a stop transfer state in response to the historical stream-resource transfer rate being greater than the target stream-resource transfer rate.

[00191] The first determining subunit 1324 is configured to determine a first stream-resource transfer rate difference between the historical stream-resource transfer rate and the target stream-resource transfer rate.

[00192] The first update subunit 1325 is configured to update the historical stream-resource transfer rate of the historical stream-resource transfer channel to the first stream-resource transfer rate difference.

[00193] The first generation subunit 1326 is configured to generate the update stream-resource transfer channel according to the historical stream-resource transfer channel updated to the first stream-resource transfer rate difference and the target stream-resource transfer channel set to the stop transfer state.

[00194] For specific implementations of functions of the first obtaining subunit 1321, the first comparison subunit 1322, the first setting subunit 1323, the first determining subunit 1324, the first update subunit 1325, and the first generation subunit 1326, refer to the above corresponding embodiment in FIG. 2a. Details are not described herein.

[00195] Referring to FIG. 9 again, the first setting subunit 1323 may include: a first ending subunit 13231, a state setting subunit 13232, and a first maintaining subunit 13233.

[00196] The first ending subunit 13231 is configured to obtain a first end timestamp of the target stream-resource transfer channel, and obtain a second end timestamp of the historical stream-resource transfer channel.

[00197] The state setting subunit 13232 is configured to set the transfer state of the target stream-resource transfer channel in a first time period to the stop transfer state in response to the second end timestamp being less than the first end timestamp; where the maximum timestamp of the first time period is the second end timestamp.

[00198] The first maintaining subunit 13233 is configured to set the transfer state of the target stream-resource transfer channel in a second time period to a continuous transfer state, and maintain the target stream-resource transfer rate of the target stream-resource transfer channel in the second time period; where the maximum timestamp of the second time period is the first end timestamp, and the second time period and the first time period are adjacent time periods.

[00199] The first generation subunit 1326 is specifically configured to generate the update stream-resource transfer channel according to the historical stream-resource transfer channel updated to the first stream-resource transfer rate difference, the target stream-resource transfer channel that is in the first time period and that is set to the stop transfer state, and the target stream-resource transfer channel that is in the second time period and that maintains the target stream-resource transfer rate.

[00200] For specific implementations of functions of the first ending subunit 13231, the state setting subunit 13232, the first maintaining subunit 13233, and the first generation subunit 1326, refer to the above corresponding embodiment in FIG. 2b. Details are not described herein.

[00201] Referring to FIG. 9 again, the first update subunit 1325 may include: a second ending subunit 13251, a first rate subunit 13252, and a second maintaining subunit 13253.

[00202] The second ending subunit 13251 is configured to obtain the second end timestamp of the historical stream-resource transfer channel and obtain the first end timestamp of the target stream-resource transfer channel.

[00203] The first rate subunit 13252 is configured to update the historical stream-resource transfer rate of the historical stream-resource transfer channel in a third time period to the first stream-resource transfer rate difference in response to the second end timestamp being greater than the first end timestamp; where the maximum timestamp of the third time period is the second end timestamp.

[00204] The second maintaining subunit 13253 is configured to maintain the historical stream-resource transfer rate of the historical stream-resource transfer channel in the fourth time period; where the minimum timestamp of the fourth time period is greater than the second end timestamp, and the fourth time period and the third time period are adjacent time periods.

[00205] The first generation subunit 1326 is specifically configured to generate the update stream-resource transfer channel according to the historical stream-resource transfer channel that is in the third time period and that is updated to the first stream-resource transfer rate difference, the historical stream-resource transfer channel that is in the fourth time period and that maintains the historical stream-resource transfer rate, and the target stream-resource transfer channel set to the stop transfer state.

[00206] For specific implementations of functions of the second ending subunit 13251, the first rate subunit 13252, the second maintaining subunit 13253, and the first generation subunit 1326, refer to the above corresponding embodiment in FIG. 2b. Details are not described herein.

[00207] Referring to FIG. 9 again, the first setting subunit 1323 is further configured to set a transfer state of the historical stream-resource transfer channel to a stop transfer state in response to the historical stream-resource transfer rate being less than the target stream-resource transfer rate.

[00208] The first determining subunit 1324 is further configured to determine a stream-resource transfer rate difference b between the target stream-resource transfer rate and the historical stream-resource transfer rate.

[00209] The first update subunit 1325 is further configured to update the target stream-resource transfer rate of the target stream-resource transfer channel to the stream-resource transfer rate difference b.

[00210] The first generation subunit 1326 is further configured to generate an update stream-resource transfer channel according to the target stream-resource transfer channel updated to the stream-resource transfer rate difference b and the historical stream-resource transfer channel set to the stop transfer state.

[00211] For specific implementations of functions of the first setting subunit 1323, the first determining subunit 1324, the first update subunit 1325, and the first generation subunit 1326, refer to the above corresponding embodiment in FIG. 2a. Details are not described herein.

[00212] Referring to FIG. 9 again, the first setting subunit 1323 is further configured to set a transfer state of the historical stream-resource transfer channel to a stop transfer state, and set a transfer state of the target stream-resource transfer channel to a stop transfer state in response to the historical stream-resource transfer rate being equal to the target stream-resource transfer rate.

[00213] The first generation subunit 1326 is further configured to generate an update stream-resource transfer channel according to the target stream-resource transfer channel set to the stop transfer state and the historical stream-resource transfer channel set to the stop transfer state.

[00214] For specific implementations of functions of the first setting subunit 1323 and the first generation subunit 1326, refer to the above corresponding embodiment in FIG. 2a. Details are not described herein.

[00215] Referring to FIG. 9 again, the channel combining unit 132 may include: a second obtaining subunit 2321, a second setting subunit 2322, a second determining subunit 2323, a second update subunit 2324, and a second generation subunit 2325.

[00216] The second obtaining subunit 2321 is configured to obtain a first target object address having a stream-resource transfer-out attribute, and a second target object address having a stream-resource transfer-in attribute in the target object address pair.

[00217] The second obtaining subunit 2321 is further configured to obtain a fifth historical object address having the stream-resource transfer-out attribute and a sixth historical object address having the stream-resource transfer-in attribute in the historical object address pair.

[00218] The second setting subunit 2322 is configured to set the transfer state of the target stream-resource transfer channel to the stop transfer state in response to the first target object

address being the same as the fifth historical object address, and the second target object address being the same as the sixth historical object address.

[00219] The second determining subunit 2323 is configured to determine a stream-resource transfer rate sum of the historical stream-resource transfer rate and the target stream-resource transfer rate.

[00220] The second update subunit 2324 is configured to update the historical stream-resource transfer rate of the historical stream-resource transfer channel to the stream-resource transfer rate sum.

[00221] The second generation subunit 2325 is configured to generate the update stream-resource transfer channel according to the historical stream-resource transfer channel updated to the stream-resource transfer rate sum and the target stream-resource transfer channel set to the stop transfer state.

[00222] For specific implementations of functions of the second obtaining subunit 2321, the second setting subunit 2322, the second determining subunit 2323, the second update subunit 2324, and the second generation subunit 2325, refer to operation S103 in the above corresponding embodiment in FIG. 3. Details are not described herein.

[00223] Referring to FIG. 9 again, the second update subunit 2324 may include: a third ending subunit 23241, a second rate subunit 23242, and a third rate subunit 23243.

[00224] The third ending subunit 23241 is configured to obtain a first end timestamp of the target stream-resource transfer channel, and obtain a third end timestamp of the historical stream-resource transfer channel.

[00225] The second rate subunit 23242 is configured to update the historical stream-resource transfer rate of the historical stream-resource transfer channel in a fifth time period to the stream-resource transfer rate sum in response to the third end timestamp being less than the first end timestamp; where the maximum timestamp of the fifth time period is the third end timestamp.

[00226] The third rate subunit 23243 is configured to update the historical stream-resource transfer rate of the historical stream-resource transfer channel in a sixth time period to the target stream-resource transfer rate; where the maximum timestamp of the sixth time period is the first end timestamp, and the sixth time period and the fifth time period are adjacent time periods.

[00227] The second generation subunit 2325 is specifically configured to generate the update stream-resource transfer channel according to the historical stream-resource transfer channel that is in the fifth time period and that is updated to the stream-resource transfer rate sum, the historical

stream-resource transfer channel that is in the sixth time period and that is updated to the target stream-resource transfer rate, and the target stream-resource transfer channel set to the stop transfer state.

[00228] For specific implementations of functions of the third ending subunit 23241, the second rate subunit 23242, the third rate subunit 23243, and the second generation subunit 2325, refer to operation S103 in the above corresponding embodiment in FIG. 3. Details are not described herein.

[00229] Referring to FIG. 9 again, the channel combining unit 132 may include: a third obtaining subunit 3321, a second comparison subunit 3322, a third determining subunit 3323, a third update subunit 3324, a fourth update subunit 3325, and a third generation subunit 3326.

[00230] The third obtaining subunit 3321 is configured to obtain a first target object address having a stream-resource transfer-out attribute, and a second target object address having a stream-resource transfer-in attribute in the target object address pair.

[00231] The third obtaining subunit 3321 is further configured to obtain a seventh historical object address having the stream-resource transfer-out attribute, and an eighth historical object address having the stream-resource transfer-in attribute in the historical object address pair.

[00232] The second comparison subunit 3322 is configured to compare the historical stream-resource transfer rate with the target stream-resource transfer rate in response to the first target object address being the same as the eighth historical object address, and the second target object address being different from the seventh historical object address.

[00233] The third determining subunit 3323 is configured to determine a second stream-resource transfer rate difference between the historical stream-resource transfer rate and the target stream-resource transfer rate in response to the historical stream-resource transfer rate being greater than the target stream-resource transfer rate.

[00234] The third update subunit 3324 is configured to update the historical stream-resource transfer rate of the historical stream-resource transfer channel to the second stream-resource transfer rate difference.

[00235] The fourth update subunit 3325 is configured to update the first target object address in the target stream-resource transfer channel to the seventh historical object address.

[00236] The third generation subunit 3326 is configured to generate the update stream-resource transfer channel according to the historical stream-resource transfer channel updated to the

second stream-resource transfer rate difference and the target stream-resource transfer channel updated to the seventh historical object address.

[00237] For specific implementations of functions of the third obtaining subunit 3321, the second comparison subunit 3322, the third determining subunit 3323, the third update subunit 3324, the fourth update subunit 3325, and the third generation subunit 3326, refer to operation S201 to operation S207 in the above corresponding embodiment in FIG. 7. Details are not described herein.

[00238] Referring to FIG. 9 again, the third update subunit 3324 may include: a fourth ending subunit 33241, a fourth rate subunit 33242, and a third maintaining subunit 33243.

[00239] The fourth ending subunit 33241 is configured to obtain a first end timestamp of the target stream-resource transfer channel, and obtain a fourth end timestamp of the historical stream-resource transfer channel.

[00240] The fourth rate subunit 33242 is configured to update the historical stream-resource transfer rate of the historical stream-resource transfer channel in a seventh time period to the second stream-resource transfer rate difference in response to the fourth end timestamp being greater than the first end timestamp; where the maximum timestamp of the seventh time period is the first end timestamp.

[00241] The third maintaining subunit 33243 is configured to maintain the historical stream-resource transfer rate of the historical stream-resource transfer channel in the eighth time period; where the maximum timestamp of the eighth time period is the fourth end timestamp, and the eighth time period and the seventh time period are adjacent time periods.

[00242] The third generation subunit 3326 is specifically configured to generate the update stream-resource transfer channel according to the historical stream-resource transfer channel that is in the seventh time period and that is updated to the second stream-resource transfer rate difference, the historical stream-resource transfer channel that is in the eighth time period and that maintains the historical stream-resource transfer rate, and the target stream-resource transfer channel updated to the seventh historical object address.

[00243] For specific implementations of functions of the fourth ending subunit 33241, the fourth rate subunit 33242, the third maintaining subunit 33243, and the third generation subunit 3326, refer to operation S205 in the above corresponding embodiment in FIG. 7. Details are not described herein.

[00244] Referring to FIG. 9 again, the fourth update subunit 3325 may include: a fifth ending subunit 33251, a first address subunit 33252, and a fourth maintaining subunit 33253.

[00245] The fifth ending subunit 33251 is configured to obtain a first end timestamp of the target stream-resource transfer channel, and obtain a fourth end timestamp of the historical stream-resource transfer channel.

[00246] The first address subunit 33252 is configured to update the first target object address in the target stream-resource transfer channel in the ninth time period to the seventh historical object address in response to the fourth end timestamp being less than the first end timestamp; where the maximum timestamp of the ninth time period is the fourth end timestamp.

[00247] The fourth maintaining subunit 33253 is configured to maintain the first target object address of the target stream-resource transfer channel in a tenth time period; where the maximum timestamp of the tenth time period is the first end timestamp, and the tenth time period and the ninth time period are adjacent time periods.

[00248] The third generation subunit 3326 is configured to generate the update stream-resource transfer channel according to the historical stream-resource transfer channel updated to the second stream-resource transfer rate difference, the target stream-resource transfer channel that is in the ninth time period and that is updated to the seventh historical object address, and the target stream-resource transfer channel that is in the tenth time period and that maintains the first target object address.

[00249] For specific implementations of functions of the fifth ending subunit 33251, the first address subunit 33252, the fourth maintaining subunit 33253, and the third generation subunit 3326, refer to operation S206 in the above corresponding embodiment in FIG. 7. Details are not described herein.

[00250] Referring to FIG. 9 again, the third determining subunit 3323 is further configured to determine a stream-resource transfer rate difference e between the target stream-resource transfer rate and the historical stream-resource transfer rate in response to the historical stream-resource transfer rate being less than the target stream-resource transfer rate.

[00251] The third update subunit 3324 is further configured to update the target stream-resource transfer rate of the target stream-resource transfer channel to the stream-resource transfer rate difference e .

[00252] The fourth update subunit 3325 is further configured to update the eighth historical object address in the historical stream-resource transfer channel to the second target object address.

[00253] The third generation subunit 3326 is further configured to generate the update stream-resource transfer channel according to the target stream-resource transfer channel updated to

the stream-resource transfer rate difference e and the historical stream-resource transfer channel updated to the second target object address.

[00254] For specific implementations of functions of the third determining subunit 3323, the third update subunit 3324, the fourth update subunit 3325, and the third generation subunit 3326, refer to operation S203 in the above corresponding embodiment in FIG. 7. Details are not described herein.

[00255] Referring to FIG. 9 again, the channel combining unit 132 may further include: a third setting subunit 3327.

[00256] The third setting subunit 3327 is configured to set a transfer state of the historical stream-resource transfer channel to a stop transfer state in response to the historical stream-resource transfer rate being equal to the target stream-resource transfer rate.

[00257] The fourth update subunit 3325 is further configured to update the first target object address in the target stream-resource transfer channel to the seventh historical object address.

[00258] The third generation subunit 3326 is further configured to generate an update stream-resource transfer channel according to the target stream-resource transfer channel updated to the seventh historical object address and the historical stream-resource transfer channel set to the stop transfer state.

[00259] For specific implementations of functions of the third setting subunit 3327, the fourth update subunit 3325, and the third generation subunit 3326, refer to operation S203 in the above corresponding embodiment in FIG. 7. Details are not described herein.

[00260] Referring to FIG. 9 again, the channel combining unit 132 may include: a fourth obtaining subunit 4321, a third comparison subunit 4322, a fourth determining subunit 4323, a fifth update subunit 4324, a sixth update subunit 4325, and a fourth generation subunit 4326.

[00261] The fourth obtaining subunit 4321 is configured to obtain a first target object address having a stream-resource transfer-out attribute, and a second target object address having a stream-resource transfer-in attribute in the target object address pair.

[00262] The fourth obtaining subunit 4321 is further configured to obtain a ninth historical object address having the stream-resource transfer-out attribute and a tenth historical object address having the stream-resource transfer-in attribute in the historical object address pair.

[00263] The third comparison subunit 4322 is configured to compare the historical stream-resource transfer rate with the target stream-resource transfer rate in response to the first target

object address being different from the tenth historical object address, and the second target object address being the same as the ninth historical object address.

[00264] The fourth determining subunit 4323 is configured to determine a third stream-resource transfer rate difference between the historical stream-resource transfer rate and the target stream-resource transfer rate in response to the historical stream-resource transfer rate being greater than the target stream-resource transfer rate.

[00265] The fifth update subunit 4324 is configured to update the historical stream-resource transfer rate of the historical stream-resource transfer channel to the third stream-resource transfer rate difference.

[00266] The sixth update subunit 4325 is configured to update the second target object address in the target stream-resource transfer channel to the tenth historical object address.

[00267] The fourth generation subunit 4326 is configured to generate the update stream-resource transfer channel according to the historical stream-resource transfer channel updated to the third stream-resource transfer rate difference and the target stream-resource transfer channel updated to the tenth historical object address.

[00268] For specific implementations of functions of the fourth obtaining subunit 4321, the third comparison subunit 4322, the fourth determining subunit 4323, the fifth update subunit 4324, the sixth update subunit 4325, and the fourth generation subunit 4326, refer to operation S301 to operation S307 in the above corresponding embodiment in FIG. 8. Details are not described herein.

[00269] Referring to FIG. 9 again, the fifth update subunit 4324 may include: a sixth ending subunit 43241, a fifth rate subunit 43242, and a fifth maintaining subunit 43243.

[00270] The sixth ending subunit 43241 is configured to obtain a first end timestamp of the target stream-resource transfer channel, and obtain a fifth end timestamp of the historical stream-resource transfer channel.

[00271] The fifth rate subunit 43242 is configured to update the historical stream-resource transfer rate of the historical stream-resource transfer channel in an eleventh time period to the third stream-resource transfer rate difference in response to the fifth end timestamp being greater than the first end timestamp; where the maximum timestamp of the eleventh time period is the first end timestamp.

[00272] The fifth maintaining subunit 43243 is configured to maintain the historical stream-resource transfer rate of the historical stream-resource transfer channel in the twelfth time period;

where the maximum timestamp of the twelfth time period is the fifth end timestamp, and the twelfth time period and the eleventh time period are adjacent time periods.

[00273] The fourth generation subunit 4326 is specifically configured to generate the update stream-resource transfer channel according to the historical stream-resource transfer channel that is in the eleventh time period and that is updated to the third stream-resource transfer rate difference, the historical stream-resource transfer channel that is in the twelfth time period and that maintains the historical stream-resource transfer rate, and the target stream-resource transfer channel updated to the tenth historical object address.

[00274] For specific implementations of functions of the sixth ending subunit 43241, the fifth rate subunit 43242, the fifth maintaining subunit 43243, and the fourth generation subunit 4326, refer to operation S305 in the above corresponding embodiment in FIG. 8. Details are not described herein.

[00275] Referring to FIG. 9 again, the sixth update subunit 4325 may include: a seventh ending subunit 43251, a second address subunit 43252, and a sixth maintaining subunit 43253.

[00276] The seventh ending subunit 43251 is configured to obtain a first end timestamp of the target stream-resource transfer channel, and obtain a fifth end timestamp of the historical stream-resource transfer channel.

[00277] The second address subunit 43252 is configured to update the second target object address in the target stream-resource transfer channel in the thirteenth time period to the tenth historical object address in response to the fifth end timestamp being less than the first end timestamp; where the maximum timestamp of the thirteenth time period is the fifth end timestamp.

[00278] The sixth maintaining subunit 43253 is configured to maintain the second target object address of the target stream-resource transfer channel in a fourteenth time period; where the maximum timestamp of the fourteenth time period is the first end timestamp, and the fourteenth time period and the thirteenth time period are adjacent time periods.

[00279] The fourth generation subunit 4326 is specifically configured to generate the update stream-resource transfer channel according to the historical stream-resource transfer channel updated to the third stream-resource transfer rate difference, the target stream-resource transfer channel that is in the thirteenth time period and that is updated to the tenth historical object address, and the target stream-resource transfer channel that is in the fourteenth time period and that maintains the second target object address.

[00280] For specific implementations of functions of the seventh ending subunit 43251, the second address subunit 43252, the sixth maintaining subunit 43253, and the fourth generation subunit 4326, refer to operation S306 in the above corresponding embodiment in FIG. 8. Details are not described herein.

[00281] Referring to FIG. 9 again, the fourth determining subunit 4323 is further configured to determine a stream-resource transfer rate difference g between the target stream-resource transfer rate and the historical stream-resource transfer rate in response to the historical stream-resource transfer rate being less than the target stream-resource transfer rate.

[00282] The fifth update subunit 4324 is further configured to update the target stream-resource transfer rate of the target stream-resource transfer channel to the stream-resource transfer rate difference g .

[00283] The sixth update subunit 4325 is further configured to update the ninth historical object address in the historical stream-resource transfer channel to the first target object address.

[00284] The fourth generation subunit 4326 is further configured to generate the update stream-resource transfer channel according to the target stream-resource transfer channel updated to the stream-resource transfer rate difference g and the historical stream-resource transfer channel updated to the first target object address.

[00285] For specific implementations of functions of the fourth determining subunit 4323, the fifth update subunit 4324, the sixth update subunit 4325, and the fourth generation subunit 4326, refer to operation S303 in the above corresponding embodiment in FIG. 8. Details are not described herein.

[00286] Referring to FIG. 9 again, the channel combining unit 132 may further include: a fourth setting subunit 4327.

[00287] The fourth setting subunit 4327 is configured to set a transfer state of the historical stream-resource transfer channel to a stop transfer state in response to the historical stream-resource transfer rate being equal to the target stream-resource transfer rate.

[00288] The sixth update subunit 4325 is further configured to update the second target object address in the target stream-resource transfer channel to the tenth historical object address.

[00289] The fourth generation subunit 4326 is further configured to generate an update stream-resource transfer channel according to the target stream-resource transfer channel updated to the tenth historical object address and the historical stream-resource transfer channel set to the stop transfer state.

[00290] For specific implementations of functions of the fourth setting subunit 4327, the sixth update subunit 4325, and the fourth generation subunit 4326, refer to operation S303 in the above corresponding embodiment in FIG. 8. Details are not described herein.

[00291] Referring to FIG. 9 again, the first processing module 13 may further include: a unit adjustment unit 133.

[00292] The unit adjustment unit 133 is configured to: in response to a transfer rate unit of the historical stream-resource transfer rate being different from a transfer rate unit of the target stream-resource transfer rate, adjust the historical stream-resource transfer rate according to the transfer rate unit of the target stream-resource transfer rate, to obtain an adjusted stream-resource transfer rate; where a transfer rate unit of the adjusted stream-resource transfer rate is the same as the transfer rate unit of the target stream-resource transfer rate; and

[00293] the channel combining unit 132 is specifically configured to combine the historical stream-resource transfer channel and the target stream-resource transfer channel according to the target object address pair, the historical object address pair, the adjusted stream-resource transfer rate, and the target stream-resource transfer rate, to obtain the update stream-resource transfer channel.

[00294] For specific implementations of functions of the unit adjustment unit 133 and the channel combining unit 132, refer to operation S103 of the above corresponding embodiment in FIG. 3. Details are not described herein.

[00295] To sum up, since the update object address pair is obtained based on the combination of the target object address pair and the historical object address pair, stream-resource transfer processing respectively corresponding to the historical stream-resource transfer channel and the target stream-resource transfer channel may be implemented through the update stream-resource transfer channel. Therefore, the present application can reduce the complexity of the stream-resource transfer channel in the blockchain, thereby improving the efficiency of stream-resource transfer.

[00296] Further, FIG. 10 is a schematic structural diagram of a computer device according to an embodiment of the present application. As shown in FIG. 10, the computer device 1000 may include: at least one processor 1001, for example, a CPU, at least one network interface 1004, a user interface 1003, a memory 1005, and at least one communications bus 1002. The communications bus 1002 is configured to implement connection and communication between these components. In some embodiments, the user interface 1003 may include a display and a keyboard. In an

embodiment, the network interface 1004 may include a standard wired interface and wireless interface. The memory 1005 may be a high-speed RAM memory, or may be a non-volatile memory, for example, at least one magnetic disk memory. In an embodiment, the memory 1005 may be at least one storage apparatus located remotely from the foregoing processor 1001. As shown in FIG. 10, the memory 1005 used as a computer storage medium may include an operating system, a network communications module, a user interface module, and a device-control application program.

[00297] In the computer device 1000 shown in FIG. 10, the network interface 1004 may provide a network communication function. The user interface 1003 is mainly configured to provide an input interface for a user. The processor 1001 may be configured to invoke a device-control application program stored in the memory 1005, to perform the following operations:

[00298] creating a target stream-resource transfer channel for a stream-resource channel creation request in the blockchain based on a stream-resource transfer contract in the blockchain, the stream-resource channel creation request including a target object address pair, and the target stream-resource transfer channel providing, for the target object address pair, a function of performing stream-resource transfer based on the stream-resource transfer contract;

[00299] obtaining, in the blockchain according to the target object address pair, a historical stream-resource transfer channel including a historical object address pair, the target object address pair and the historical object address pair sharing at least one identical object address;

[00300] combining the historical stream-resource transfer channel and the target stream-resource transfer channel to obtain an update stream-resource transfer channel, the update stream-resource transfer channel including an update object address pair; and

[00301] performing, according to the update stream-resource transfer channel, stream-resource transfer processing associated with the update object address pair.

[00302] The computer device 1000 described in this embodiment of the present application may execute the descriptions of the data processing method or apparatus based on a blockchain in the above embodiments. Details are not described again herein. In addition, the description of beneficial effects of the same method are not described herein again.

[00303] An embodiment of the present application further provides a computer-readable storage medium. The computer-readable storage medium stores a computer program. When the computer program is executed by a processor, the descriptions of the data processing method or apparatus based on a blockchain in the above embodiments are implemented. Details are not

described again herein. In addition, the description of beneficial effects of the same method are not described herein again.

[00304] The computer-readable storage medium may be the data processing apparatus based on a blockchain in any one of the above embodiments or an internal storage unit of the computer device, such as a hard disk or a memory of the computer device. The computer-readable storage medium may also be an external storage device of the computer device, such as a plug-in hard disk equipped on the computer device, a smart media card (SMC), a secure digital (SD) card, a flash card, or the like. Further, the computer readable storage medium may also include both the internal storage unit and the external storage device of the computer device. The computer-readable storage medium is configured to store the computer program and other programs and data required by the computer device. The computer-readable storage medium may also be configured to temporarily store data that has been output or will be output.

[00305] An embodiment of the present application further provides a computer program product, including computer programs, and the computer programs being stored in a computer-readable storage medium. The processor of the computer device reads the computer program from the computer-readable storage medium, and the processor executes the computer program, so that the computer device may execute the descriptions of the data processing method or apparatus based on a blockchain in the above embodiments. Details are not described again herein. In addition, the description of beneficial effects of the same method are not described herein again.

[00306] In the specification and claims and the accompanying drawings of the embodiments of the present application, the terms such as "first" and "second" are used to distinguish between different objects, but are not used to describe a particular sequence of the objects. In addition, the terms "include" and any variant thereof are intended to cover a non-exclusive inclusion. For example, a process, method, apparatus, product, or device that includes a series of operations or units is not limited to the listed operations or modules; and instead, further includes an operation or module that is not listed in an embodiment, or further includes another operation or unit that is intrinsic to the process, method, apparatus, product, or device in an embodiment.

[00307] A person of ordinary skill in the art may understand that, units and algorithm operations of the examples described in the foregoing disclosed embodiments may be implemented by electronic hardware, computer software, or a combination thereof. To clearly describe the interchangeability between the hardware and the software, the foregoing has generally described compositions and operations of each example based on functions. Whether the functions are executed in a mode of hardware or software depends on particular applications and design

constraint conditions of the technical solutions. A person skilled in the art may use different methods to implement the described functions for each particular application, but it is not considered that the implementation goes beyond the scope of the embodiments of the present application.

[00308] What is disclosed above is merely exemplary embodiments of the present application, and certainly is not intended to limit the scope of the claims of the present application. Therefore, equivalent variations made in accordance with the claims of the present application shall fall within the scope of the present application.

CLAIMS

What is claimed is:

1. A data processing method based on a blockchain, executable by a computer device, comprising:

creating a target stream-resource transfer channel for a stream-resource channel creation request in the blockchain based on a stream-resource transfer contract in the blockchain, the stream-resource channel creation request comprising a target object address pair, and the target stream-resource transfer channel providing, for the target object address pair, a function of performing stream-resource transfer based on the stream-resource transfer contract;

obtaining, in the blockchain according to the target object address pair, a historical stream-resource transfer channel comprising a historical object address pair, the target object address pair and the historical object address pair sharing at least one identical object address;

combining the historical stream-resource transfer channel and the target stream-resource transfer channel to obtain an update stream-resource transfer channel, the update stream-resource transfer channel comprising an update object address pair; and

performing, according to the update stream-resource transfer channel, stream-resource transfer processing associated with the update object address pair.

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

obtaining the stream-resource channel creation request initiated by a first target object address; the target object address pair comprising the first target object address having a stream-resource transfer-out attribute;

calling the stream-resource transfer contract in the blockchain according to the stream-resource channel creation request;

when the target stream-resource transfer channel is successfully created, determining, based on the stream-resource transfer contract, a to-be-transferred target stream-resource corresponding to the target stream-resource transfer channel; and

transferring the to-be-transferred target stream-resource from the first target object address to a contract address of the stream-resource transfer contract.

3. The method according to claim 1, wherein the target object address pair comprises a first target object address having a stream-resource transfer-out attribute, and a second target object address having a stream-resource transfer-in attribute; the historical object address pair comprises a

first historical object address pair, a second historical object address pair, and a third historical object address pair; and

the obtaining, in the blockchain according to the target object address pair, a historical stream-resource transfer channel comprising a historical object address pair comprises:

obtaining, in the blockchain, a first stream-resource transfer channel comprising the first historical object address pair; wherein the first historical object address pair comprises a first historical object address that has the stream-resource transfer-in attribute and that is the same as the first target object address;

obtaining, in the blockchain, a second stream-resource transfer channel comprising the second historical object address pair; wherein the second historical object address pair comprises a second historical object address that has the stream-resource transfer-out attribute and that is the same as the second target object address;

obtaining, in the blockchain, a third stream-resource transfer channel comprising the third historical object address pair; wherein the third historical object address pair is the same as the target object address pair; and

determining at least one of the first stream-resource transfer channel, the second stream-resource transfer channel, and the third stream-resource transfer channel as the historical stream-resource transfer channel.

4. The method according to claim 1, wherein the combining the historical stream-resource transfer channel and the target stream-resource transfer channel to obtain an update stream-resource transfer channel comprises:

determining a historical stream-resource transfer rate of the historical stream-resource transfer channel, and determining a target stream-resource transfer rate of the target stream-resource transfer channel; and

combining the historical stream-resource transfer channel and the target stream-resource transfer channel according to the target object address pair, the historical object address pair, the historical stream-resource transfer rate, and the target stream-resource transfer rate, to obtain the update stream-resource transfer channel.

5. The method according to claim 4, wherein the combining the historical stream-resource transfer channel and the target stream-resource transfer channel according to the target object address pair, the historical object address pair, the historical stream-resource transfer rate, and the target stream-resource transfer rate, to obtain the update stream-resource transfer channel

comprises:

obtaining a first target object address having a stream-resource transfer-out attribute, and a second target object address having a stream-resource transfer-in attribute in the target object address pair;

obtaining a third historical object address having the stream-resource transfer-out attribute and a fourth historical object address having the stream-resource transfer-in attribute in the historical object address pair;

comparing the historical stream-resource transfer rate with the target stream-resource transfer rate in response to the first target object address being the same as the fourth historical object address, and the second target object address being the same as the third historical object address;

setting a transfer state of the target stream-resource transfer channel to a stop transfer state in response to the historical stream-resource transfer rate being greater than the target stream-resource transfer rate;

determining a first stream-resource transfer rate difference between the historical stream-resource transfer rate and the target stream-resource transfer rate, and updating the historical stream-resource transfer rate of the historical stream-resource transfer channel to the first stream-resource transfer rate difference; and

generating the update stream-resource transfer channel according to the historical stream-resource transfer channel updated to the first stream-resource transfer rate difference and the target stream-resource transfer channel set to the stop transfer state.

6. The method according to claim 5, wherein the setting a transfer state of the target stream-resource transfer channel to a stop transfer state comprises:

obtaining a first end timestamp of the target stream-resource transfer channel, and obtaining a second end timestamp of the historical stream-resource transfer channel;

setting the transfer state of the target stream-resource transfer channel in a first time period to the stop transfer state in response to the second end timestamp being less than the first end timestamp; wherein the maximum timestamp of the first time period is the second end timestamp; and

setting the transfer state of the target stream-resource transfer channel in a second time period to a continuous transfer state, and maintaining the target stream-resource transfer rate of the target stream-resource transfer channel in the second time period; wherein the maximum timestamp of the

second time period is the first end timestamp, and the second time period and the first time period are adjacent time periods; and

the generating the update stream-resource transfer channel according to the historical stream-resource transfer channel updated to the first stream-resource transfer rate difference and the target stream-resource transfer channel set to the stop transfer state comprises:

generating the update stream-resource transfer channel according to the historical stream-resource transfer channel updated to the first stream-resource transfer rate difference, the target stream-resource transfer channel that is in the first time period and that is set to the stop transfer state, and the target stream-resource transfer channel that is in the second time period and that maintains the target stream-resource transfer rate.

7. The method according to claim 5, wherein the updating the historical stream-resource transfer rate of the historical stream-resource transfer channel to the first stream-resource transfer rate difference comprises:

obtaining a second end timestamp of the history stream-resource transfer channel, and obtaining a first end timestamp of the target stream-resource transfer channel;

updating the historical stream-resource transfer rate of the historical stream-resource transfer channel in a third time period to the first stream-resource transfer rate difference in response to the second end timestamp being greater than the first end timestamp; wherein the maximum timestamp of the third time period is the second end timestamp; and

maintaining the historical stream-resource transfer rate of the historical stream-resource transfer channel in the fourth time period; wherein the minimum timestamp of the fourth time period is greater than the second end timestamp, and the fourth time period and the third time period are adjacent time periods; and

the generating the update stream-resource transfer channel according to the historical stream-resource transfer channel updated to the first stream-resource transfer rate difference and the target stream-resource transfer channel set to the stop transfer state comprises:

generating the update stream-resource transfer channel according to the historical stream-resource transfer channel that is in the third time period and that is updated to the first stream-resource transfer rate difference, the historical stream-resource transfer channel that is in the fourth time period and that maintains the historical stream-resource transfer rate, and the target stream-resource transfer channel set to the stop transfer state.

8. The method according to claim 4, wherein the combining the historical stream-resource

transfer channel and the target stream-resource transfer channel according to the target object address pair, the historical object address pair, the historical stream-resource transfer rate, and the target stream-resource transfer rate, to obtain the update stream-resource transfer channel comprises:

obtaining a first target object address having a stream-resource transfer-out attribute, and a second target object address having a stream-resource transfer-in attribute in the target object address pair;

obtaining a fifth historical object address having the stream-resource transfer-out attribute and a sixth historical object address having the stream-resource transfer-in attribute in the historical object address pair;

setting the transfer state of the target stream-resource transfer channel to the stop transfer state in response to the first target object address being the same as the fifth historical object address, and the second target object address being the same as the sixth historical object address;

determining a stream-resource transfer rate sum of the historical stream-resource transfer rate and the target stream-resource transfer rate, and updating the historical stream-resource transfer rate of the historical stream-resource transfer channel to the stream-resource transfer rate sum; and

generating the update stream-resource transfer channel according to the historical stream-resource transfer channel updated to the stream-resource transfer rate sum and the target stream-resource transfer channel set to the stop transfer state.

9. The method according to claim 8, wherein the updating the historical stream-resource transfer rate of the historical stream-resource transfer channel to the stream-resource transfer rate sum comprises:

obtaining a first end timestamp of the target stream-resource transfer channel, and obtaining a third end timestamp of the historical stream-resource transfer channel;

updating the historical stream-resource transfer rate of the historical stream-resource transfer channel in a fifth time period to the stream-resource transfer rate sum in response to the third end timestamp being less than the first end timestamp; wherein the maximum timestamp of the fifth time period is the third end timestamp; and

updating the historical stream-resource transfer rate of the historical stream-resource transfer channel in a sixth time period to the target stream-resource transfer rate; wherein the maximum timestamp of the sixth time period is the first end timestamp, and the sixth time period and the fifth time period are adjacent time periods; and

the generating the update stream-resource transfer channel according to the historical stream-resource transfer channel updated to the stream-resource transfer rate sum and the target stream-resource transfer channel set to the stop transfer state comprises:

generating the update stream-resource transfer channel according to the historical stream-resource transfer channel that is in the fifth time period and that is updated to the stream-resource transfer rate sum, the historical stream-resource transfer channel that is in the sixth time period and that is updated to the target stream-resource transfer rate, and the target stream-resource transfer channel set to the stop transfer state.

10. The method according to claim 4, wherein the combining the historical stream-resource transfer channel and the target stream-resource transfer channel according to the target object address pair, the historical object address pair, the historical stream-resource transfer rate, and the target stream-resource transfer rate, to obtain the update stream-resource transfer channel comprises:

obtaining a first target object address having a stream-resource transfer-out attribute, and a second target object address having a stream-resource transfer-in attribute in the target object address pair;

obtaining a seventh historical object address having the stream-resource transfer-out attribute, and an eighth historical object address having the stream-resource transfer-in attribute in the historical object address pair;

comparing the historical stream-resource transfer rate with the target stream-resource transfer rate in response to the first target object address being the same as the eighth historical object address, and the second target object address being different from the seventh historical object address;

determining a second stream-resource transfer rate difference between the historical stream-resource transfer rate and the target stream-resource transfer rate in response to the historical stream-resource transfer rate being greater than the target stream-resource transfer rate, and updating the historical stream-resource transfer rate of the historical stream-resource transfer channel to the second stream-resource transfer rate difference;

updating the first target object address in the target stream-resource transfer channel to the seventh historical object address; and

generating the update stream-resource transfer channel according to the historical stream-resource transfer channel updated to the second stream-resource transfer rate difference and the

target stream-resource transfer channel updated to the seventh historical object address.

11. The method according to claim 10, wherein the updating the historical stream-resource transfer rate of the historical stream-resource transfer channel to the second stream-resource transfer rate difference comprises:

obtaining a first end timestamp of the target stream-resource transfer channel, and obtaining a fourth end timestamp of the historical stream-resource transfer channel;

updating the historical stream-resource transfer rate of the historical stream-resource transfer channel in a seventh time period to the second stream-resource transfer rate difference in response to the fourth end timestamp being greater than the first end timestamp; wherein the maximum timestamp of the seventh time period is the first end timestamp; and

maintaining the historical stream-resource transfer rate of the historical stream-resource transfer channel in the eighth time period; wherein the maximum timestamp of the eighth time period is the fourth end timestamp, and the eighth time period and the seventh time period are adjacent time periods; and

the generating the update stream-resource transfer channel according to the historical stream-resource transfer channel updated to the second stream-resource transfer rate difference and the target stream-resource transfer channel updated to the seventh historical object address comprises:

generating the update stream-resource transfer channel according to the historical stream-resource transfer channel that is in the seventh time period and that is updated to the second stream-resource transfer rate difference, the historical stream-resource transfer channel that is in the eighth time period and that maintains the historical stream-resource transfer rate, and the target stream-resource transfer channel updated to the seventh historical object address.

12. The method according to claim 10, wherein the updating the first target object address in the target stream-resource transfer channel to the seventh historical object address comprises:

obtaining a first end timestamp of the target stream-resource transfer channel, and obtaining a fourth end timestamp of the historical stream-resource transfer channel;

updating the first target object address in the target stream-resource transfer channel in the ninth time period to the seventh historical object address in response to the fourth end timestamp being less than the first end timestamp; wherein the maximum timestamp of the ninth time period is the fourth end timestamp; and

maintaining the first target object address of the target stream-resource transfer channel in a

tenth time period; wherein the maximum timestamp of the tenth time period is the first end timestamp, and the tenth time period and the ninth time period are adjacent time periods; and

the generating the update stream-resource transfer channel according to the historical stream-resource transfer channel updated to the second stream-resource transfer rate difference and the target stream-resource transfer channel updated to the seventh historical object address comprises:

generating the update stream-resource transfer channel according to the historical stream-resource transfer channel updated to the second stream-resource transfer rate difference, the target stream-resource transfer channel that is in the ninth time period and that is updated to the seventh historical object address, and the target stream-resource transfer channel that is in the tenth time period and that maintains the first target object address.

13. The method according to claim 4, wherein the combining the historical stream-resource transfer channel and the target stream-resource transfer channel according to the target object address pair, the historical object address pair, the historical stream-resource transfer rate, and the target stream-resource transfer rate, to obtain the update stream-resource transfer channel comprises:

obtaining a first target object address having a stream-resource transfer-out attribute, and a second target object address having a stream-resource transfer-in attribute in the target object address pair;

obtaining a ninth historical object address having the stream-resource transfer-out attribute and a tenth historical object address having the stream-resource transfer-in attribute in the historical object address pair;

comparing the historical stream-resource transfer rate with the target stream-resource transfer rate in response to the first target object address being different from the tenth historical object address, and the second target object address being the same as the ninth historical object address;

determining a third stream-resource transfer rate difference between the historical stream-resource transfer rate and the target stream-resource transfer rate in response to the historical stream-resource transfer rate being greater than the target stream-resource transfer rate, and updating the historical stream-resource transfer rate of the historical stream-resource transfer channel to the third stream-resource transfer rate difference;

updating the second target object address in the target stream-resource transfer channel to the tenth historical object address; and

generating the update stream-resource transfer channel according to the historical stream-

resource transfer channel updated to the third stream-resource transfer rate difference and the target stream-resource transfer channel updated to the tenth historical object address.

14. The method according to claim 13, wherein the updating the historical stream-resource transfer rate of the historical stream-resource transfer channel to the third stream-resource transfer rate difference comprises:

obtaining a first end timestamp of the target stream-resource transfer channel, and obtaining a fifth end timestamp of the historical stream-resource transfer channel;

updating the historical stream-resource transfer rate of the historical stream-resource transfer channel in an eleventh time period to the third stream-resource transfer rate difference in response to the fifth end timestamp being greater than the first end timestamp; wherein the maximum timestamp of the eleventh time period is the first end timestamp; and

maintaining the historical stream-resource transfer rate of the historical stream-resource transfer channel in a twelfth time period; wherein the maximum timestamp of the twelfth time period is the fifth end timestamp, and the twelfth time period and the eleventh time period are adjacent time periods; and

the generating the update stream-resource transfer channel according to the historical stream-resource transfer channel updated to the third stream-resource transfer rate difference and the target stream-resource transfer channel updated to the tenth historical object address comprises:

generating the update stream-resource transfer channel according to the historical stream-resource transfer channel that is in the eleventh time period and that is updated to the third stream-resource transfer rate difference, the historical stream-resource transfer channel that is in the twelfth time period and that maintains the historical stream-resource transfer rate, and the target stream-resource transfer channel updated to the tenth historical object address.

15. The method according to claim 13, wherein the updating the second target object address in the target stream-resource transfer channel to the tenth historical object address comprises:

obtaining a first end timestamp of the target stream-resource transfer channel, and obtaining a fifth end timestamp of the historical stream-resource transfer channel;

updating the second target object address in the target stream-resource transfer channel in a thirteenth time period to the tenth historical object address in response to the fifth end timestamp being less than the first end timestamp; wherein the maximum timestamp of the thirteenth time period is the fifth end timestamp; and

maintaining the second target object address of the target stream-resource transfer channel in a fourteenth time period; wherein the maximum timestamp of the fourteenth time period is the first end timestamp, and the fourteenth time period and the thirteenth time period are adjacent time periods; and

the generating the update stream-resource transfer channel according to the historical stream-resource transfer channel updated to the third stream-resource transfer rate difference and the target stream-resource transfer channel updated to the tenth historical object address comprises:

generating the update stream-resource transfer channel according to the historical stream-resource transfer channel updated to the third stream-resource transfer rate difference, the target stream-resource transfer channel that is in the thirteenth time period and that is updated to the tenth historical object address, and the target stream-resource transfer channel that is in the fourteenth time period and that maintains the second target object address.

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

in response to a transfer rate unit of the historical stream-resource transfer rate being different from a transfer rate unit of the target stream-resource transfer rate, adjusting the historical stream-resource transfer rate according to the transfer rate unit of the target stream-resource transfer rate, to obtain an adjusted stream-resource transfer rate; wherein a transfer rate unit of the adjusted stream-resource transfer rate is the same as the transfer rate unit of the target stream-resource transfer rate; and

the combining the historical stream-resource transfer channel and the target stream-resource transfer channel according to the target object address pair, the historical object address pair, the historical stream-resource transfer rate, and the target stream-resource transfer rate, to obtain the update stream-resource transfer channel comprises:

combining the historical stream-resource transfer channel and the target stream-resource transfer channel according to the target object address pair, the historical object address pair, the adjusted stream-resource transfer rate, and the target stream-resource transfer rate, to obtain the update stream-resource transfer channel.

17. A data processing apparatus based on a blockchain, comprising:

a channel creation module, configured to create a target stream-resource transfer channel for a stream-resource channel creation request in the blockchain based on a stream-resource transfer contract in the blockchain, the stream-resource channel creation request comprising a target object address pair, and the target stream-resource transfer channel providing, for the target object address

pair, a function of performing stream-resource transfer based on the stream-resource transfer contract;

a first obtaining module, configured to obtain, in the blockchain according to the target object address pair, a historical stream-resource transfer channel comprising a historical object address pair, the target object address pair and the historical object address pair sharing at least one identical object address; a first processing module, configured to combine the historical stream-resource transfer channel and the target stream-resource transfer channel to obtain an update stream-resource transfer channel, the update stream-resource transfer channel comprising an update object address pair; and

a second processing module, configured to perform, according to the update stream-resource transfer channel, stream-resource transfer processing associated with the update object address pair.

18. A computer device, comprising: a processor, a memory, and a network interface;

the processor being connected to the memory and the network interface, the network interface being configured to provide a data communication function, the memory being configured to store computer programs, and the processor being configured to call the computer programs so that the computer device performs the method according to any one of claims 1 to 16.

19. A computer-readable storage medium, storing computer programs, and the computer programs being loaded and executed by a processor, so that a computer device having the processor performs the method according to any one of claims 1 to 16.

20. A computer program product, comprising a computer program stored in a computer-readable storage medium, the computer program being read and executed by a processor, so that a computer device having the processor performs the method according to any one of claims 1 to 16.