CLAIMS

1. A data archiving method for archiving blocks of a blockchain, executable by a computer device and comprising:

finding, in a local space when an archiving task for a target block header in a block header chain of the blockchain exists, a current state tree that the target block header points to and a current state tree that each of one or more first block headers before the target block header points to, wherein for each block header of the target block header and the one or more first block headers, the current state tree that the block header points to represents a state tree that the block header points to when the archiving task exists;

characterized by the method further comprising: reconstructing, according to current state trees which are found and node reuse information between the current state trees, a new state tree that the target block header points to, wherein the new state tree that the target block header points to independently stores state data corresponding to the target block header, and no node reuse exists between the new state tree that the target block header points to and the respective current state tree that each first block header points to;

reconstructing, according to the new state tree that the target block header points to and target state data corresponding to each of one or more second block headers after the target block header, a new state tree that the second block header points to, wherein node reuse occurs between one of new state trees corresponding to a block header and a new state tree that a block header next to the corresponding block header points to; and

archiving the current state trees stored in the local space to a target archiving server as to-be-archived data, and deleting the current state trees in the local space after successful archiving;

wherein the current state tree and the new state tree are Merkle Patricia Tries

(MPTs).

2. The method according to claim 1, wherein the reconstructing, according to current state trees which are found and node reuse information between the current state trees, a new state tree that the target block header points to comprises:

determining at least one reused tree node from the current state trees according to the node reuse information between the current state trees, wherein the reused tree node represents a tree node reused in the current state tree that the target block header points to and a current state tree that one of the one or more first block headers points to;

copying the at least one reused tree node, and copying tree nodes in the current state tree that the target block header points to; and

reconstructing, according to tree nodes obtained by copying, the new state tree that the target block header points to.

3. The method according to claim 1, wherein

a quantity of the second block headers is M, M being a positive integer, and

wherein the reconstructing, according to the new state tree that the target block header and target state data corresponding to each of one or more second block headers after the target block header, a new state tree that the second block header points to comprises: performing the following processing for an mth second block header:

generating a new leaf node based on target state data corresponding to the mth second block header, wherein the target state data corresponding to the mth second block header represents at least one piece of state data, different from each piece of state data corresponding to a previous block header of the mth second block header, in a plurality of pieces of state data corresponding to the mth second block header, and $m \in [1, M]$;

storing the target state data corresponding to the mth second block header to the

new leaf node; and

reconstructing, according to the new leaf node storing the target state data and a new state tree that the previous block header of the mth second block header points to, a new state tree that the mth second block header points to, wherein at least one leaf node is reused in the new state tree that the mth second block header points to and the new state tree that the previous block header points to, and the previous block header is the target block header when m is 1.

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

for each of the one or more second block headers after the target block header, obtaining transaction data corresponding to the second block header, the transaction data being used for indicating a transaction operation on at least one piece of state data; and

sequentially performing transaction operations indicated by transaction data corresponding to the one or more second block headers in order of block heights of the one or more second block headers from low to high, according to the state data stored in the new state tree that the target block header points to, to obtain the target state data corresponding to each second block header.

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

traversing the one or more second block headers after the target block header, and determining a current second block header that is currently traversed; and

copying target state data corresponding to the current second block header from a current state tree that the current second block header points to.

6. The method according to claim 1, wherein the archiving the current state trees stored in the local space to a target archiving server as to-be-archived data comprises:

determining a block number of a first block header with a smallest block height in the one or more first block headers as a starting block number;

determining a block number of a first block header with a largest block height in

the one or more first block headers as an ending block number;

determining the current state trees stored in the local space as the to-be-archived data, and generating an archiving request according to the to-be-archived data, the starting block number, and the ending block number; and

transmitting the archiving request to the target archiving server, wherein the target archiving server performs a hash operation on the to-be-archived data according to the archiving request to obtain a to-be-archived data hash value, and returns the to-be-archived data hash value after storing the to-be-archived data hash value, the to-be-archived data, the starting block number, and the ending block number.

7. The method according to claim 6, further comprising:

determining, in response to receiving the to-be-archived data hash value transmitted by the target archiving server, that the to-be-archived data is successfully archived; and

recording the to-be-archived data hash value, an archiving timestamp, the starting block number, the ending block number, and a server identifier of the target archiving server to an archiving record in the local space.

8. The method according to claim 7, further comprising:

reading the ending block number from the archiving record in the local space in response to a data query request for a specified block header;

generating, when a block number of the specified block header is less than or equal to the ending block number, a specified data query request according to the to-be-archived data hash value in the archiving record in the local space and the block number of the specified block header;

transmitting the specified data query request to the target archiving server, wherein the target archiving server determines and returns state data corresponding to the specified block header according to the to-be-archived data hash value carried in the specified data query request, the to-be-archived data hash value indicates the

current state trees in the to-be-archived data and the block number of the specified block header; and

receiving the state data that is returned by the target archiving server and that corresponds to the specified block header, and outputting the state data corresponding to the specified block header.

9. The method according to claim 8, further comprising:

determining, in the local space when the block number of the specified block header is greater than the ending block number, a new state tree that the specified block header points to, and determining a new state tree that each of one or more block headers before the specified block header points to; and

obtaining state data corresponding to the specified block header according to new state trees which are determined, and outputting the obtained state data.

10. The method according to any one of claims 1 to 9, wherein

the computer device communicates with a plurality of archiving servers, and

wherein the method further comprises:

for each of the plurality of archiving servers,

obtaining an archiving performance parameter of the archiving server, the archiving performance parameter comprising at least one of a remaining data storage capacity, a transmission bandwidth, and a security coefficient, and the security coefficient being used for measuring a probability of data of the archiving server being stolen, and

determining an archiving priority of the archiving server according to the archiving performance parameter; and

determining an archiving server with a highest archiving priority among the plurality of archiving servers as the target archiving server.

11. The method according to claim 10, wherein

the archiving performance parameter comprises the transmission bandwidth, and

wherein the obtaining an archiving performance parameter of the archiving server comprises:

transmitting a bandwidth detection request to the archiving server, the bandwidth detection request being used for instructing the archiving server to return target data of a specified data volume; and

receiving the target data returned by the archiving server, and determining a transmission bandwidth of the archiving server based on time consumed by the archiving server to return the target data.

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

determining a current block header in a block header chain when the archiving task for the target block header exists, the current block header representing a block header that points to a state tree in the local space and that has a smallest block height;

determining a latest block header in the block header chain, the latest block header representing a block header that points to a state tree in the local space and that has a largest block height; and

performing, when a block height of the target block header is greater than that of the current block header and less than that of the latest block header, the operation of finding, in a local space, a current state tree that the target block header points to and a current state tree that each of one or more first block headers before the target block header points to,

wherein the first block headers are the current block header and at least one block header between the current block header and the target block header, and the second block headers are the latest block header and at least one block header between the latest block header and the target block header.

13. A data archiving apparatus for archiving blocks of a blockchain, comprising:

a search unit, configured to find, in a local space when an archiving task for a

target block header in a block header chain of the blockchain exists, a current state tree that the target block header points to and a current state tree that each of one or more first block headers before the target block header points to, wherein for each block header of the target block header and the one or more first block headers, the current state tree that the block header points to represents a state tree that the block header points to when the archiving task exists;

characterized by the apparatus further comprising: a first reconstruction unit, configured to reconstruct, according to current state trees which are found and node reuse information between the current state trees, a new state tree that the target block header points to, wherein the new state tree that the target block header points to independently stores state data corresponding to the target block header, and no node reuse exists between the new state tree that the target block header points to and the respective current state tree that each first block header points to;

a second reconstruction unit, configured to reconstruct, according to the new state tree that the target block header points to and target state data corresponding to each of one or more second block headers after the target block header, a new state tree that the second block header points to, wherein node reuse occurs between one of new state trees corresponding to a block header and a new state tree that a block header next to the corresponding block header points to; and

an archiving unit, configured to archive the current state trees stored in the local space to a target archiving server as to-be-archived data, and delete the current state trees in the local space after successful archiving;

wherein the current state tree and the new state tree are Merkle Patricia Tries (MPTs).

14. A computer device, the computer device comprising an input device and an output device, the computer device further comprising a processor and a storage medium, and the processor being configured to obtain one or more instructions stored in the storage medium to perform the method according to any one of claims 1 to 12.

15. A computer-readable storage medium, the computer-readable storage medium storing one or more instructions that, when executed, performing the method according to any one of claims 1 to 12.

16. A computer program product, the computer program product comprising a computer program that, when executed by a processor, performing the method according to any one of claims 1 to 12.