CLAIMS

What is claimed is:

1. A data processing method, performed by a blockchain node, wherein the blockchain node stores a first block file, the first block file stores blocks on a blockchain, each of the blocks stores one or more transaction records, the method comprising:

creating, on the blockchain node, a second block file that is different from the first block file, wherein the second block file stores block heads of all the blocks adjacent to each other according to a sequence of the blocks in the first block file;

for each block in the first block file,

detecting a to-be-maintained transaction record in the block;

selecting, from a hash tree of the block, nodes used for proving that the to-be-maintained transaction record belongs to the block;

constructing a node maintenance tree according to the selected nodes;

storing the to-be-maintained transaction record at a corresponding leaf node of the node maintenance tree;

updating the second block file by storing, in an associated manner, a block head of the block and the node maintenance tree storing the to-be-maintained transaction record; and

deleting the first block file after the update processing for all of the blocks in the first block file is completed.

2. The method according to claim 1, wherein the selecting, from a hash tree of the block, nodes used for proving that the transaction record belongs to the block comprises:

performing, in response to that the to-be-maintained transaction record is detected in the block, the operation of selecting, from a hash tree of the block, nodes used for proving that the transaction record belongs to the block; and

the method further comprises:

updating, in response to that no to-be-maintained transaction record is detected in the block, the second block file by storing the block head of the block to be adjacent to the block head of a previous block.

3. The method according to claim 1, wherein the transaction record is a transaction record used

for account reconciliation; and the detecting a to-be-maintained transaction record in the block comprises:

comparing a transaction record in the block with a transaction record in an unsuccessful account reconciliation set; and

determining, in response to that the transaction record in the block is found in the unsuccessful account reconciliation set, the transaction record as the to-be-maintained transaction record.

4. The method according to claim 3, wherein the unsuccessful account reconciliation set comprises an uncompleted account reconciliation set; and the method further comprises:

storing, in response to receiving an account reconciliation call request transmitted by a first account reconciliation device, a first transaction record on which account reconciliation is to be performed to a block in the first block file through a smart contract, the first transaction record being carried in the account reconciliation call request;

obtaining, through the smart contract, an account reconciliation association identifier corresponding to the first transaction record;

searching an uncompleted account reconciliation set created on a blockchain for a second transaction record of a second account reconciliation device, wherein the second transaction record corresponds to the account reconciliation association identifier; and

adding the first transaction record to the uncompleted account reconciliation set in response to not finding the second transaction record.

5. The method according to claim 4, wherein the unsuccessful account reconciliation set further comprises an account reconciliation error set; and the method further comprises:

removing the second transaction record from the uncompleted account reconciliation set in response to finding the second transaction record;

comparing the first transaction record with the second transaction record; and

adding the first transaction record and the second transaction record to the account reconciliation error set in response to a comparison result indicating inconsistent.

6. The method according to claim 1, wherein the selecting, from a hash tree of the block, nodes used for proving that the transaction record belongs to the block comprises:

determining, in the hash tree of the block, a leaf node corresponding to the transaction record;

selecting a first ancestor node and a sibling leaf node of the leaf node from the hash tree; and

selecting, for each non-sibling leaf node of the leaf node, a second ancestor node nearest to a root node among all ancestor nodes of the non-sibling leaf node; and

the constructing a node maintenance tree according to the selected nodes comprises:

constructing a node maintenance tree according to the leaf node, the first ancestor node, the sibling leaf node, and the second ancestor node.

7. The method according to claim 1, wherein the hash tree is a double-layer hash tree, a hash value of a root node of the double-layer hash tree is obtained by performing a hash operation on a series of connected hash values of all leaf nodes, and the selected nodes are all nodes of the hash tree.

8. The method according to any one of claims 1 to 7, further comprising:

obtaining a second storage position of the second block file; and

modifying a position index of the block from pointing to a first storage position of the first block file to pointing to the second storage position.

9. A data processing apparatus, applied to a blockchain node, wherein the blockchain node stores a first block file, the first block file stores blocks on a blockchain, each of the blocks stores one or more transaction records, the apparatus comprising:

a block file creating module, configured to create a second block file on the blockchain node, wherein the second block file stores block heads of all the blocks adjacent to each other according to a sequence of the blocks in the first block file;

a node maintenance module, configured to: detect, for each block in a first block file, a to-be-maintained transaction record in the block; select, from a hash tree of the block, nodes used for proving that the to-be-maintained transaction record belongs to the block; and construct a node maintenance tree according to the selected nodes;

a transaction storage module, configured to store the to-be-maintained transaction record to a corresponding leaf node of the node maintenance tree;

a block update module, configured to update the second block file by storing, in an associated manner, a block head of the block and the node maintenance tree storing the to-be-maintained transaction record; and

a deletion module, configured to delete the first block file after the update processing for all of the blocks in the first block file is completed.

10. The apparatus according to claim 9, wherein

the node maintenance module is configured to perform, in response to that the to-be-maintained transaction record is detected in the block, the operation of selecting, from a hash tree of the block, nodes used for proving that the transaction record belongs to the block; and

the node maintenance module is further configured to instruct, in response to that no to-be-maintained transaction record is detected in the block, the block update module to update the second block file by storing the block head of the block to be adjacent to the block head of the previous block.

11. The apparatus according to claim 9, wherein the transaction record is a transaction record used for account reconciliation; and the node maintenance module is further configured to compare a transaction record in the block with a transaction record in an unsuccessful account reconciliation set; and determine, in response to that the transaction record in the block is comprised in the unsuccessful account reconciliation set, the transaction record as the to-be-maintained transaction record.

12. The apparatus according to claim 11, wherein the unsuccessful account reconciliation set comprises an uncompleted account reconciliation set; and the apparatus further comprises:

a real-time account reconciliation module, configured to store, in response to receiving an account reconciliation call request transmitted by a first account reconciliation device, a first transaction record on which account reconciliation is to be performed to the block in the first block file through a smart contract, the first transaction record being carried in the account reconciliation call request; obtain, through the smart contract, an account reconciliation association identifier corresponding to the first transaction record; search an uncompleted account reconciliation set created on a blockchain for a second transaction record of a second account reconciliation device, wherein the second transaction record corresponds to the account reconciliation association identifier; and add the first transaction record to the uncompleted account reconciliation set in response to not finding the second transaction record.

13. The apparatus according to claim 12, wherein the unsuccessful account reconciliation set further comprises an account reconciliation error set; and the real-time account reconciliation module is further configured to: remove the second transaction record from the uncompleted account reconciliation set in response to finding the second transaction record; compare the first transaction record with the second transaction record; and add the first transaction record and the second transaction record to the account reconciliation error set in response to a comparison result

indicating inconsistent.

14. The apparatus according to claim 9, wherein the node maintenance module is further configured to: determine, in the hash tree of the block, a leaf node corresponding to the transaction record; select a first ancestor node and a brother leaf node of the leaf node from the hash tree; select, for each non-brother leaf node of the leaf node, a second ancestor node nearest to a root node among all ancestor nodes of the non-brother leaf node; and construct a node maintenance tree according to the leaf node, the first ancestor node, the brother leaf node, and the second ancestor node.

15. The apparatus according to claim 9, wherein the hash tree is a double-layer hash tree, a hash value of a root node of the double-layer hash tree is obtained by performing a hash operation on a series of connected hash values of all leaf nodes, and the selected nodes are all nodes of the hash tree.

16. The apparatus according to claim 15, wherein the block update module is further configured to obtain a second storage position of the second block file; and modify a position index of the block from pointing to a first storage position of the first block file to pointing to the second storage position.

17. A computer device, comprising a memory and a processor, the memory storing a computer program, the computer program, when executed by the processor, causing the processor to perform the operations of the method according to any one of claims 1 to 8.

18. A computer-readable storage medium, the computer-readable storage medium storing a computer program, the computer program, when executed by a processor, causing the processor to perform the operations of the method according to any one of claims 1 to 8.