

CLAIMS

1. 1. A method for updating databases that store blockchain data, the method comprising:

receiving a transaction associated with a blockchain network;

determining, after the transaction is performed, an updated account state of a blockchain account involved in the transaction;

~~adding, to a history state object database and after~~appending, based on the updated account state, a current block ~~is appended~~ to a blockchain associated with the blockchain network ~~based on~~by performing a consensus algorithm;

adding, to a current database, state information of the current block to a current state tree in the current database, wherein the state information of the current block comprises a block identifier (ID) of the current block, and the current state tree of the current database is a location-addressed state tree that comprises a node value associated with the block ID of the current block;

adding, to a history state object database, the updated account state, a hash value of the updated account state, an account identifier (ID) of the blockchain account, and ~~at~~the block ID of the current block; and

updating, based on the hash value of the updated account state, the account ID of the blockchain account, and the block ID of the current block, a state tree stored in a history state database~~-,~~ wherein the state tree of the history state database comprises a key-value pair (KVP) that is formed by the hash value of the updated account state and the account ID of the blockchain account.

2. -The method of claim 1, wherein the history state object database stores a mapping between an account ID, a block ID, a hash value of an account state, and an account state of each blockchain account of the blockchain network and an account state of ~~the~~a corresponding blockchain account.

3. -The method of claim 1, wherein the consensus algorithm is based on one of proof of work (PoW), proof of stake (PoS), and practical Byzantine fault tolerance (PBFT).
4. -The method of ~~any previous~~ claim 1, wherein the state tree is a content-addressed state tree.
5. -The method of claim 4, wherein the content-addressed state tree is a history state tree of a fixed-depth Merkle tree (FDMT), a sparse Merkle Tree (SMT), or a Merkle Patricia tree (MPT).
6. -The method of ~~any previous~~ claim 1, wherein updating the state tree includes adding one or more branch nodes and a leaf node under a state root associated with the current block and store the hash value of the updated account state, the account ID of the blockchain account in the leaf node.
7. -The method of claim 6, wherein the leaf node stores hash values of account states and account IDs corresponding to a plurality of blockchain accounts of the blockchain network.
8. -The method of claim 7, wherein the account IDs identify relative locations of corresponding hash values of account states of the plurality of blockchain accounts stored in the leaf node.
- 9.- The method of claim 7~~-or 8~~, wherein an address of the leaf node is a hash value generated based on ~~the~~ account states corresponding to ~~the~~a plurality of blockchain nodes.
10. -The method of ~~any previous~~ claim 1, wherein the consensus algorithm is performed after one or more of existence, validity, and authenticity of the transaction are verified.
- 11.- A computer-implemented system for updating databases that store blockchain data, comprising:

one or more processors; and

one or more computer-readable memories coupled to the one or more processors and having instructions stored thereon that are executable by the one or more processors to perform ~~the method of any of one or more operations comprising: claims 1 to 10.~~

receiving a transaction associated with a blockchain network;

determining, after the transaction is performed, an updated account state of a blockchain account involved in the transaction;

appending, based on the updated account state, a current block to a blockchain associated with the blockchain network by performing a consensus algorithm;

adding, to a current database, state information of the current block to a current state tree in the current database, wherein the state information of the current block comprises a block identifier (ID) of the current block, and the current state tree of the current database is a location-addressed state tree that comprises a node value associated with the block ID of the current block;

adding, to a history state object database, the updated account state, a hash value of the updated account state, an account identifier (ID) of the blockchain account, and the block ID of the current block; and

updating, based on the hash value of the updated account state, the account ID of the blockchain account, and the block ID of the current block, a state tree stored in a history state database, wherein the state tree of the history state database comprises a key-value pair (KVP) that is formed by the hash value of the updated account state and the account ID of the blockchain account.

12.—An apparatus The computer-implemented system of claim 11, wherein the history state object database stores a mapping between an account ID, a block ID, a hash value of an account state, and an account state of each blockchain account of the blockchain network and an account state of a corresponding blockchain account.

13. The computer-implemented system of claim 11, wherein updating the state tree includes adding one or more branch nodes and a leaf node under a state root associated with

the current block and store the hash value of the updated account state, the account ID of the blockchain account in the leaf node.

14. The computer-implemented system of claim 13, wherein the leaf node stores hash values of account states and account IDs corresponding to a plurality of blockchain accounts of the blockchain network.

15. The computer-implemented system of claim 14, wherein the account IDs identify relative locations of corresponding hash values of account states of the plurality of blockchain accounts stored in the leaf node.

16. The computer-implemented system of claim 14, wherein an address of the leaf node is a hash value generated based on account states corresponding to a plurality of blockchain nodes.

17. A non-transitory, computer-readable medium for updating databases that store blockchain data, the apparatus non-transitory, computer-readable medium storing one or more instructions executable by a computer system to perform operations comprising a plurality of modules for performing the method of any of claims 1 to 10.:

receiving a transaction associated with a blockchain network;
determining, after the transaction is performed, an updated account state of a blockchain account involved in the transaction;

appending, based on the updated account state, a current block to a blockchain associated with the blockchain network by performing a consensus algorithm;

adding, to a current database, state information of the current block to a current state tree in the current database, wherein the state information of the current block comprises a block identifier (ID) of the current block, and the current state tree of the current database is a location-addressed state tree that comprises a node value associated with the block ID of the current block;

adding, to a history state object database, the updated account state, a hash value of the updated account state, an account identifier (ID) of the blockchain account, and the block ID of the current block; and

updating, based on the hash value of the updated account state, the account ID of the blockchain account, and the block ID of the current block, a state tree stored in a history state database, wherein the state tree of the history state database comprises a key-value pair (KVP) that is formed by the hash value of the updated account state and the account ID of the blockchain account.

18. The non-transitory, computer-readable medium of claim 17, wherein the history state object database stores a mapping between an account ID, a block ID, a hash value of an account state, and an account state of each blockchain account of the blockchain network and an account state of a corresponding blockchain account.

19. The non-transitory, computer-readable medium of claim 17, wherein updating the state tree includes adding one or more branch nodes and a leaf node under a state root associated with the current block and store the hash value of the updated account state, the account ID of the blockchain account in the leaf node.

20. The non-transitory, computer-readable medium of claim 19, wherein the leaf node stores hash values of account states and account IDs corresponding to a plurality of blockchain accounts of the blockchain network.