

CLAIMS

1. A computer-implemented method for performing a map iteration by a network node of a blockchain network, the method comprising:

receiving, by a blockchain node in the blockchain network, a request to obtain a plurality of keys included in a map by the network node, the map i) being stored in the blockchain node, and ii) storing a plurality of key-value pairs that comprise blockchain transaction data generated by the blockchain network and comprise the plurality of keys and a plurality of values corresponding to the plurality of keys;

maintaining, by the blockchain node in the blockchain network, data representing a forest that stores the plurality of keys that are stored in the map, the forest comprising a plurality of trees, each tree comprising up to a respective plurality of storage nodes, each storage node storing a subset of the plurality of keys, wherein a storage location of a key of a key-value pair on a storage node of the forest is based on one or more modulo values calculated based on one or more hash values of the key;

traversing, by the blockchain node in the blockchain network, the forest to retrieve the plurality of keys stored in the forest; and

returning, by the blockchain node in the blockchain network, the plurality of keys.

2. The computer-implemented method of claim 1, wherein the traversing the forest to retrieve the plurality of keys stored in the forest comprises:

iterating each storage node of the forest; and

retrieving one or more keys that are stored in the each storage node of the forest.

3. The computer-implemented method of any preceding claim, wherein the traversing the forest comprises at least one of:

performing a depth-first search on the forest; or

performing a breadth-first search on the forest.

4. The computer-implemented method of any preceding claim, wherein each tree of the plurality of trees comprises a plurality of levels, and wherein each level comprises one or more storage nodes.

5. The computer-implemented method of claim 4, wherein the plurality of levels correspond to a plurality of different hash functions, and wherein each key is stored in the forest based on one or more hash values of the each key that are computed using one or more hash functions of the plurality of different hash functions.
6. The computer-implemented method of claim 5, wherein the method further comprises:
 - receiving a first request to add a first key of a first key-value pair to the forest in response to the first key-value pair being stored in the map;
 - determining a first storage node of the forest to insert the first key based on one or more first hash values of the first key that are computed using one or more first hash functions of the plurality of different hash functions; and
 - storing the first key to the first storage node.
7. The computer-implemented method of claim 5, wherein the method further comprises:
 - receiving a second request to delete a second key of a second key-value pair from the forest in response to the second key-value pair being deleted from the map;
 - determining a second storage node of the forest that stores the second key based on one or more second hash values of the second key that are computed using one or more second hash functions of the plurality of different hash functions; and
 - deleting the second key from the second storage node.
8. The computer-implemented method of claim 5, wherein the method further comprises:
 - receiving more than one request to add respective keys of key-value pairs to the forest in response to the key-value pairs being stored in the map;
 - determining different storage nodes of the forest to insert the respective keys based on one or more hash values of the respective keys that are computed using one or more hash functions of the plurality of different hash functions; and
 - storing the respective keys to the different storage nodes concurrently.

9. The computer-implemented method of any preceding claim, wherein each tree of the plurality of trees comprises a plurality of leaf storage nodes and one or more non-leaf storage nodes, and wherein each of the one or more non-leaf storage nodes corresponds to a configurable number of child nodes.
10. The computer-implemented method of any preceding claim, wherein each storage node of the forest stores a configurable number of keys.
11. The computer-implemented method of any preceding claim, wherein maintaining data representing the forest that stores the plurality of keys that are stored in the map comprises one or more of:
- generating a forest data structure to store the plurality of keys;
 - adding a new key into the forest in response to a corresponding key-value pair is stored in the map;
 - deleting an existing key from the forest in response to a corresponding key-value pair is removed from the map; or
 - modifying an existing key in the forest in response to a corresponding key-value pair is changed in the map.
12. The computer-implemented method of any preceding claim, wherein the key of the key-value pair is stored on the storage node of the forest based on:
- determining the one or more hash values of the key using one or more hash functions;
 - calculating the one or more modulo values based on a modulo operation on each of the one or more hash values of the key;
 - determining the storage node of the forest based on the one or more modulo values;
 - and
 - storing the key into the storage node of the forest.
13. An apparatus for implementing a blockchain-based map iteration system, the apparatus comprising a plurality of modules for performing the computer-implemented method of any one of claims 1 to 12.

14. A blockchain-based map iteration system, 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 computer-implemented method of any one of claims 1 to 12.