

WHAT IS CLAIMED IS:

- 1 1. A method comprising:
2 broadcasting, by a node computer, a propose message comprising a
3 new block for a blockchain to a plurality of node computers in a network shard;
4 receiving, by the node computer, at least one vote message from the
5 plurality of node computers, the at least one vote message indicating verification of
6 the new block;
7 after a predetermined amount of time, receiving, by the node computer,
8 a plurality of pre-commit messages comprising at least the new block and a block
9 certificate created based on the at least one vote message; and
10 if a number of received pre-commit messages of the plurality of pre-
11 commit messages is greater than a quorum value associated with the network shard,
12 committing, by the node computer, the new block to the blockchain.
- 1 2. The method of claim 1, wherein the propose message
2 comprises the new block, a view number, and a previous block certificate.
- 1 3. The method of claim 2, wherein the plurality of node computers
2 verify the previous block certificate prior to broadcasting the at least one vote
3 message.
- 1 4. The method of claim 1, wherein the plurality of node computers
2 in the network shard are a first plurality of node computers in a first shard
3 maintaining the blockchain which is a first blockchain, wherein the quorum value is a
4 first quorum value, and wherein a second plurality of node computers in a second
5 shard maintaining a second blockchain is associated with a second quorum value.
- 1 5. The method of claim 4 further comprising:
2 generating, by the node computer, a shutdown message based on an
3 inactivity timer associated with the first network shard; and
4 transmitting, by the node computer, the shutdown message to the
5 second plurality of node computers in the second network shard, wherein the second
6 network shard is a network reference shard, wherein the reference network shard
7 determines to shutdown the first network shard and distributes the first plurality of
8 node computers to a plurality of additional shards.

1 6. The method of claim 4, wherein prior to broadcasting the
2 propose message, the method further comprises:
3 receiving, by the node computer, a commit interaction request
4 message from a client device, the commit interaction request message comprising
5 interaction data and one or more proof-of-inclusions created by one or more shards,
6 wherein at least one proof-of-inclusion of the one or more proof-of-inclusions is
7 created by the second network shard;
8 verifying, by the node computer, the one or more proof-of-inclusions;
9 and
10 if the one or more proof-of-inclusions are valid, including, by the node
11 computer, at least the interaction data in the new block.

1 7. The method of claim 1, wherein the new block comprises a
2 block header and a block body, wherein the block header is a tuple comprising at
3 least an accumulator value.

1 8. The method of claim 7, wherein after committing the new block
2 to the blockchain the method further comprises:
3 removing, by the node computer, the block body from a memory of the
4 node computer.

1 9. The method of claim 1, wherein the plurality of node computers
2 in the network shard includes 20 to 200 node computers, and wherein the quorum
3 value is in a range of 25 to 40.

1 10. The method of claim 1 further comprising:
2 after receiving, the plurality of pre-commit messages, verifying, by the
3 node computer, the plurality of pre-commit messages.

1 11. A node computer comprising:
2 a processor; and
3 a computer readable medium coupled to the processor, the computer
4 readable medium comprising code, executable by the processor, to implement a
5 method comprising:

6 broadcasting a propose message comprising a new block for a
7 blockchain to a plurality of node computers in a network shard;
8 receiving at least one vote message from the plurality of node
9 computers, the at least one vote message indicating verification of the new
10 block;
11 after a predetermined amount of time, receiving a plurality of
12 pre-commit messages comprising at least the new block and a block
13 certificate created based on the at least one vote message; and
14 if a number of received pre-commit messages of the plurality of
15 pre-commit messages is greater than a quorum value associated with the
16 network shard, committing the new block to the blockchain.

1 12. The node computer of claim 11, wherein the propose message
2 comprises the new block, a view number, and a previous block certificate, wherein
3 each pre-commit message of the plurality of pre-commit messages comprises the
4 view number, and wherein the method further comprises:
5 after receiving, the plurality of pre-commit messages, verifying the
6 plurality of pre-commit messages using at least the view number in each pre-commit
7 message.

1 13. The node computer of claim 11, wherein prior to broadcasting
2 the propose message, the method further comprises:
3 receiving a commit interaction request message from a client device,
4 the commit interaction request message comprising interaction data and one or more
5 proof-of-inclusions;
6 verifying, by the node computer, the one or more proof-of-inclusions;
7 and
8 if the one or more proof-of-inclusions are valid, including, by the node
9 computer, at least the interaction data in the new block.

1 14. The node computer of claim 11, wherein the network shard is a
2 network reference shard and the node computer is a first node computer and is a
3 current leader of the reference network shard maintain a first blockchain, wherein a
4 second plurality of node computers in a second shard maintain a second blockchain,
5 and wherein the method further comprises:
6 receiving one or more shutdown messages from the second plurality of
7 node computers;
8 if a number of received shutdown messages of the one or more
9 shutdown messages is greater than shutdown threshold associated with the second
10 network shard, assigning each node computer of the second plurality of node
11 computers to a null network shard in a node-to-shard table for a present epoch; and
12 reassigning new transactions from the second network shard to a third
13 shard.

1 15. The node computer of claim 14, wherein a network comprises a
2 plurality of shards including the reference network shard and the second network
3 shard, and wherein the method further comprises:
4 broadcasting a propose table update message comprising the node-to-
5 shard table to the plurality of node computers in the reference network shard;
6 performing a commitment process to commit the reference network
7 shard to the node-to-shard table; and
8 if the reference network shard commits to the node-to-shard table,
9 broadcasting the node-to-shard table to the network, wherein after receiving the
10 node-to-shard table, the second plurality of node computers stop processing new
11 blocks for the second blockchain.

1 16. The node computer of claim 11, wherein the method further
2 comprises:
3 prior to broadcasting the propose message, generating the new block,
4 wherein the new block comprises a block header and a block body.

1 17. The node computer of claim 16, wherein the block header is a
2 tuple comprising at least an accumulator value, a block height, an epoch number, a
3 root of a Merkle tree, and a digest of a previous block header of a previous block in
4 the blockchain, wherein generating the new block comprises:

5 determining the accumulator value using an accumulator over one or
6 more latest amounts of the blockchain.

1 18. The node computer of claim 17, wherein after committing the
2 new block to the blockchain the method further comprises:

3 removing the block body from a memory of the node computer.

1 19. The node computer of claim 11, wherein the plurality of node
2 computers in the network shard are a first plurality of node computers in a first shard
3 maintaining the blockchain which is a first blockchain, wherein the quorum value is a
4 first quorum value, and wherein a second plurality of node computers in a second
5 shard maintaining a second blockchain is associated with a second quorum value,
6 wherein the first network shard and the second network shard include a same
7 number of node computers, and wherein the first quorum value is equal to the
8 second quorum value.

1 20. The node computer of claim 11, wherein the quorum value is a
2 predetermined value.

1 21. A method comprising:

2 providing, by a client device, a first proof request comprising a request
3 for a first proof-of-inclusion from a first shard maintaining a first blockchain;

4 receiving, by the client device, the first proof-of-inclusion from a first
5 node computer of the first network shard;

6 providing, by the client device, a second proof request comprising a
7 request for a second proof-of-inclusion from a second shard maintaining a second
8 blockchain;

9 receiving, by the client device, the second proof-of-inclusion from a
10 second node computer of the second network shard;

11 generating, by the client device, a commit interaction request message
12 comprising interaction data and the first proof-of-inclusion and the second proof-of-
13 inclusion; and

14 providing, by the client device, the commit interaction request message
15 to a third shard maintaining a third blockchain, wherein a third node computer of the
16 third shard verifies the first proof-of-inclusion and the second proof-of-inclusion and if
17 the first proof-of-inclusion and the second proof-of-inclusion are valid, includes at
18 least the interaction data in a new block of the third blockchain.

1 22. The method of claim 21, wherein the first proof-of-inclusion
2 comprises a first Merkle-proof, a first block header, and a first block commitment
3 proof, and wherein the second proof-of-inclusion comprises a second Merkle-proof, a
4 second block header, and a second block commitment proof.

1 23. The method of claim 22, wherein the third node computer
2 verifies the first proof-of-inclusion and the second proof-of-inclusion includes the third
3 node computer verifies the first Merkle-proof, the first block commitment proof, the
4 second Merkle-proof, and the second block commitment proof.

1 24. The method of claim 21, wherein the first proof-of-inclusion is a
2 proof that a first amount in the first blockchain is available, wherein the second proof-
3 of-inclusion is a proof that a second amount in the second blockchain is available,
4 and wherein the method further comprises:
5 obtaining, by the client device, the interaction data comprising at least
6 a third amount.

1 25. The method of claim 24, wherein the third node computer
2 determines that a total of the first amount and the second amount exceed the third
3 amount.

1 26. The method of claim 21 further comprising:
2 receiving, by the client device, an interaction response message from
3 the third node computer of the third shard, wherein the interaction response
4 message indicates that the interaction data is included in the third blockchain.

1 27. The method of claim 21:
2 providing, by the client device, a third proof request comprising a
3 request for a third proof-of-inclusion from a fourth shard maintaining a fourth
4 blockchain; and

5 receiving, by the client device, the third proof-of-inclusion from a fourth
6 node computer of the fourth shard.

1 28. The method of claim 27, wherein the commit interaction request
2 message comprises the interaction data, the first proof-of-inclusion, the second
3 proof-of-inclusion, and the third proof-of-inclusion.

1 29. The method of claim 28, wherein the third node computer of the
2 third shard verifies the third proof-of-inclusion.

1 30. The method of claim 21, wherein the new block comprises a
2 block header and a block body, wherein the block header is a tuple comprising at
3 least an accumulator value.