

CLAIMS:

1. A computer-implemented consensus method to be implemented on a blockchain maintained by a number (N) of nodes, wherein one of the nodes acts as a primary node and the other (N-1) nodes act as backup nodes, and the method is performed by one of the backup nodes, the method comprising:

obtaining a pre-prepare message from the primary node;

multicasting a prepare message to at least some of the primary node and the other (N-2) backup nodes, the prepare message indicating an acceptance of the pre-prepare message;

obtaining (Q-1) or more prepare messages respectively from (Q-1) or more of the backup nodes, wherein Q (quorum) is $(N+F+1)/2$ rounded up to the nearest integer, and F is $(N-1)/3$ rounded down to the nearest integer;

storing at least a minimal amount of consensus messages for recovery after one or more of the N nodes crash, wherein the minimal amount of consensus messages comprises the pre-prepare message and at least (Q-1) of the (Q-1) or more prepare messages;

multicasting a commit message to at least some of the primary node and the other backup nodes, the commit message indicating that the one backup node agrees to the (Q-1) or more prepare messages; and

obtaining, respectively from Q or more nodes among the primary node and the backup nodes, Q or more commit messages each indicating that a corresponding node of the Q or more nodes agrees to (Q-1) or more prepare messages received by the corresponding node.

2. The method of claim 1, wherein:

before obtaining the pre-prepare message from the primary node, the method further comprises obtaining one or more transaction requests from at least one of: a client, the primary node, or one or more of the other backup nodes;

the pre-prepare message comprises an order of one or more transactions corresponding to the one or more transaction requests;

the multicast commit message indicates that the one backup node agrees to the order; and

the method further comprises packing the one or more transactions into a local copy of the blockchain maintained by the one backup node according to the order.

3. The method of claim 1, wherein the minimal amount of consensus messages comprise:

only the pre-prepare message and the (Q-1) prepare messages.

4. The method of claim 1, wherein:

the (Q-1) or more prepare messages include the multicast prepare message;
and

the Q or more commit messages include the multicast commit message.

5. The method of claim 1, after multicasting the commit message, further comprising:

performing a system restart; and

loading at least the stored minimal amount of consensus messages.

6. The method of claim 1, after storing at least the minimal amount of consensus messages and before multicasting the commit message, further comprising:

performing a system restart; and

loading at least the stored minimal amount of consensus messages, wherein the loading comprises loading the pre-prepare message and the at least (Q-1) of the (Q-1) or more prepare messages.

7. The method of claim 6, after storing at least the minimal amount of consensus messages and before multicasting the commit message, further comprising:

multicasting a view change message comprising the loaded pre-prepare message and the loaded at least (Q-1) of the (Q-1) or more prepare messages.

8. The method of claim 7, after storing at least the minimal amount of consensus messages and before multicasting the commit message, further comprising:

obtaining from a new primary node a new view message indicating that the new primary node has received Q or more view change messages each indicating that a node that sent one of the Q or more view change messages agrees to the view change message;

multicasting another prepare message to at least some of the backup nodes including the new primary node, the another prepare message indicating an acceptance of the new view message; and

obtaining another (Q-1) or more prepare messages respectively from (Q-1) or more of the backup nodes.

9. The method of claim 7, after storing at least the minimal amount of consensus messages and before multicasting the commit message, further comprising:

obtaining, from each node of Q or more of the backup nodes, a view change message indicating that the each node of Q or more of the backup nodes agrees to the view change message;

multicasting to at least some of the backup nodes a new view message indicating that the one backup node acting as a new primary node has received the Q or more view change messages; and

obtaining another (Q-1) or more prepare messages respectively from (Q-1) or more of the backup nodes.

10. The method of claim 6, wherein performing the system restart comprises: performing the system restart without triggering a view change.

11. A consensus system for maintaining a blockchain, wherein a number of N nodes maintain the blockchain with one of the N nodes acting as a primary node and the other (N-1) nodes acting as backup nodes, the consensus system acting as one of the (N-1) backup nodes and comprising one or more processors and one or more non-transitory computer-readable memories coupled to the one or more processors

and configured with instructions executable by the one or more processors to cause the system to perform operations comprising:

- obtaining a pre-prepare message from the primary node;

- multicasting a prepare message to at least some of the primary node and the other (N-2) backup nodes, the prepare message indicating an acceptance of the pre-prepare message;

- obtaining (Q-1) or more prepare messages respectively from (Q-1) or more of the backup nodes, wherein Q (quorum) is $(N+F+1)/2$ rounded up to the nearest integer, and F is $(N-1)/3$ rounded down to the nearest integer;

- storing at least a minimal amount of consensus messages for recovery after one or more of the N nodes crash, wherein the minimal amount of consensus messages comprises the pre-prepare message and at least (Q-1) of the (Q-1) or more prepare messages;

- multicasting a commit message to at least some of the primary node and the other backup nodes, the commit message indicating that the one backup node agrees to the (Q-1) or more prepare messages; and

- obtaining, respectively from Q or more nodes among the primary node and the backup nodes, Q or more commit messages each indicating that a corresponding node of the Q or more nodes agrees to (Q-1) or more prepare messages received by the corresponding node.

12. The system of claim 11, wherein, after storing at least the minimal amount of consensus messages and before multicasting the commit message, the operations further comprise:

- performing a system restart; and

- loading at least the stored minimal amount of consensus messages, wherein the loading comprises loading the pre-prepare message and the at least (Q-1) of the (Q-1) or more prepare messages.

13. The system of claim 12, wherein, after storing at least the minimal amount of consensus messages and before multicasting the commit message, the operations further comprise:

- multicasting a view change message comprising the loaded pre-prepare message and the loaded at least (Q-1) of the (Q-1) or more prepare messages.

14. The system of claim 13, wherein, after storing at least the minimal amount of consensus messages and before multicasting the commit message, the operations further comprise:

obtaining from a new primary node a new view message indicating that the new primary node has received Q or more view change messages each indicating that a node that sent one of the Q or more view change messages agrees to the view change message;

multicasting another prepare message indicating an acceptance of the new view message to at least some of the backup nodes including the new primary node; and

obtaining another $(Q-1)$ or more prepare messages respectively from $(Q-1)$ or more of the backup nodes.

15. The system of claim 13, wherein, after storing at least the minimal amount of consensus messages and before multicasting the commit message, the operations further comprise:

obtaining, from each node of Q or more of the backup nodes, a view change message indicating that the each node of Q or more of the backup nodes agrees to the view change message;

multicasting to at least some of the backup nodes a new view message indicating that the one backup node acting as a new primary node has received the Q or more view change messages; and

obtaining another $(Q-1)$ or more prepare messages respectively from $(Q-1)$ or more of the backup nodes.

16. A non-transitory computer-readable storage medium for maintaining a blockchain, wherein a number of N nodes maintain the blockchain with one of the N nodes acting as a primary node and the other $(N-1)$ nodes acting as backup nodes, the storage medium being associated with one of the $(N-1)$ backup nodes and configured with instructions executable by one or more processors to cause the one or more processors to perform operations comprising:

obtaining a pre-prepare message from the primary node;

multicasting a prepare message to at least some of the primary node and the other (N-2) backup nodes, the prepare message indicating an acceptance of the pre-prepare message;

obtaining (Q-1) or more prepare messages respectively from (Q-1) or more of the backup nodes, wherein Q (quorum) is $(N+F+1)/2$ rounded up to the nearest integer, and F is $(N-1)/3$ rounded down to the nearest integer;

storing at least a minimal amount of consensus messages for recovery after one or more of the N nodes crash, wherein the minimal amount of consensus messages comprises the pre-prepare message and at least (Q-1) of the (Q-1) or more prepare messages;

multicasting a commit message to at least some of the primary node and the other backup nodes, the commit message indicating that the one backup node agrees to the (Q-1) or more prepare messages; and

obtaining, respectively from Q or more nodes among the primary node and the backup nodes, Q or more commit messages each indicating that a corresponding node of the Q or more nodes agrees to (Q-1) or more prepare messages received by the corresponding node.

17. The storage medium of claim 16, wherein, after storing at least the minimal amount of consensus messages and before multicasting the commit message, the operations further comprise:

performing a system restart; and

loading at least the stored minimal amount of consensus messages, wherein the loading comprises loading the pre-prepare message and the at least (Q-1) of the (Q-1) or more prepare messages.

18. The storage medium of claim 17, wherein, after storing at least the minimal amount of consensus messages and before multicasting the commit message, the operations further comprise:

multicasting a view change message comprising the loaded pre-prepare message and the loaded at least (Q-1) of the (Q-1) or more prepare messages

19. The storage medium of claim 18, wherein, after storing at least the minimal amount of consensus messages and before multicasting the commit message, the operations further comprise:

- obtaining from a new primary node a new view message indicating that the new primary node has received Q or more view change messages each indicating that a node that sent one of the Q or more view change messages agrees to the view change message;

- multicasting another prepare message indicating an acceptance of the new view message to at least some of the backup nodes including the new primary node; and

- obtaining another (Q-1) or more prepare messages respectively from (Q-1) or more of the backup nodes.

20. The storage medium of claim 18, wherein, after storing at least the minimal amount of consensus messages and before multicasting the commit message, the operations further comprise:

- obtaining, from each node of Q or more of the backup nodes, a view change message indicating that the each node of Q or more of the backup nodes agrees to the view change message;

- multicasting to at least some of the backup nodes a new view message indicating that the one backup node acting as a new primary node has received the Q or more view change messages; and

- obtaining another (Q-1) or more prepare messages respectively from (Q-1) or more of the backup nodes.