

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 the primary node, the method comprising:

multicasting a pre-prepare message to at least some of the backup nodes;

obtaining (Q-1) or more prepare messages respectively from (Q-1) or more of the backup nodes, wherein the prepare messages each indicates an acceptance of the pre-prepare message by the corresponding backup node, 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;

after the one or more of the N nodes crash, loading at least the stored minimal amount of consensus messages;

based on the loaded at least the stored minimal amount of consensus messages, multicasting a commit message to at least some of the backup nodes, the commit message indicating that the primary 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 multicasting the pre-prepare message to at least some of the backup nodes, the method further comprises obtaining one or more transaction requests from at least one of: one or more clients or one or more of the backup nodes.

3. The method of claim 2, wherein:

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

the commit message indicates that the primary node that sent the commit message agrees to the order.

4. The method of claim 3, further comprising:

packing the one or more transactions into a local copy of the blockchain maintained by the primary node according to the order.

5. The method of claim 1, wherein:

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

6. The method of claim 1, wherein the minimal amount of consensus messages comprises:

only the pre-prepare message and the at least (Q-1) of the (Q-1) or more prepare messages.

7. The method of claim 1, further comprising performing a system restart without triggering a view change after the one or more of the N nodes crash.

8. 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 the primary node 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:

multicasting a pre-prepare message to at least some of the backup nodes;

obtaining (Q-1) or more prepare messages respectively from (Q-1) or more of the backup nodes, wherein the prepare messages each indicates an acceptance of the pre-prepare message by the corresponding backup node, 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;

after the one or more of the N nodes crash, loading at least the stored minimal amount of consensus messages;

based on the loaded at least the stored minimal amount of consensus messages, multicasting a commit message to at least some of the backup nodes, the commit message indicating that the primary 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.

9. The system of claim 8, wherein:

before multicasting the pre-prepare message to at least some of the backup nodes, the operations further comprise obtaining one or more transaction requests from at least one of: one or more clients or one or more of the backup nodes;

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

the commit message indicates that the primary node that sent the commit message agrees to the order; and

the operations further comprise packing the one or more transactions into a local copy of the blockchain maintained by the primary node according to the order.

10. The system of claim 8, wherein the minimal amount of consensus messages comprises:

only the pre-prepare message and the at least (Q-1) of the (Q-1) or more prepare messages.

11. The system of claim 8, wherein the operations further comprise performing a system restart without triggering a view change after the one or more of the N nodes crash.

12. 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 the primary node and configured with instructions executable by one or more processors to cause the one or more processors to perform operations comprising:

- multicasting a pre-prepare message to at least some of the backup nodes;

- obtaining (Q-1) or more prepare messages respectively from (Q-1) or more of the backup nodes, wherein the prepare messages each indicates an acceptance of the pre-prepare message by the corresponding backup node, 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;

- after the one or more of the N nodes crash, loading at least the stored minimal amount of consensus messages;

- based on the loaded at least the stored minimal amount of consensus messages, multicasting a commit message to at least some of the backup nodes, the commit message indicating that the primary 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.

13. The storage medium of claim 12, wherein:

- before multicasting the pre-prepare message to at least some of the backup nodes, the operations further comprise obtaining one or more transaction requests from at least one of: one or more clients or one or more of the backup nodes;

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

- the commit message indicates that the primary node that sent the commit message agrees to the order; and

- the operations further comprise packing the one or more transactions into a local copy of the blockchain maintained by the primary node according to the order.

14. The storage medium of claim 12, wherein the minimal amount of consensus messages comprises:

only the pre-prepare message and the at least $(Q-1)$ of the $(Q-1)$ or more prepare messages.

15. The storage medium of claim 12, wherein the operations further comprise performing a system restart without triggering a view change after the one or more of the N nodes crash.