Cheetah

Consensus in One Round of Voting

Pactus Team
info@pactus.org

June 2024

1 License

2 Motivation

Many Byzantine agreement or consensus protocols assume that a certain percentage of the replicas in the protocol are faulty or Byzantine and that the adversary has control of the network. These assumptions are crucial for designing a Byzantine fault-tolerant protocol. However, in practice, if not all, then most of the replicas are not faulty, and the network is healthy. Here, we explain that we can design a protocol that can tolerate faulty replicas and a faulty network while also being efficient in a non-faulty environment.

3 The Protocol

The Cheetah consensus algorithm operates with $N = 3f + 1$ replicas, where f is the maximum number of replicas that may be faulty or Byzantine. For

example, if there is one faulty replica, the algorithm requires at least 3 non-faulty replicas to maintain optimal resiliency, making the minimum number of replicas 4.

Messages are denoted as $\langle m \rangle$ tuples, and a message signed by node i is represented as $\langle m \rangle_{\sigma_i}$. We use an *aggregate signature scheme* for signing the message that allows each party to justify its vote for a particular value by a single aggregated signature generated from n signatures for the same message m.

$$\sigma_{agg} = Agg(\sigma_1, \sigma_2, \ldots, \sigma_n)$$

The Cheetah consensus algorithm has two main paths: the **gracious path** and the **change proposer path**.

Gracious Path

The gracious block creation path in the Cheetah consensus algorithm includes these steps: **Propose**, **Precommit**, and **Commit** [1]. The protocol proceeds in rounds $r = 0, 1, 2, \ldots$.

Propose Step

In each round r, one replica acts as the proposer, and the others act as verifiers. The proposer p collects transactions and creates a proposed block B. It signs and broadcasts a proposal message with the proposed block piggybacked to all the replicas. The proposal message has this form:

$$\langle \langle \text{PROPOSE}, h, r \rangle_{\sigma_p}, B \rangle$$

where:

- B is the proposed block
- h indicates the block height
- r is an assigned round number, which is zero for the first round

[1] The Practical Byzantine Fault Tolerance is based on three steps: pre-prepare, prepare, and commit.

Precommit Step

If replica i accepts the proposal, it enters the *precommit* step and signs and broadcasts a *precommit* message to all other replicas. Otherwise, it does nothing. The precommit message has this form:

$$\langle \text{PRECOMMIT}, h, r, d \rangle_{\sigma_i}$$

where:

- d is the digest or hash of the proposed block B

Commit Step

At any time in the protocol, if replica i receives $3f + 1$ precommit messages from other replicas (including its own), it becomes **committed** and enters the commit step.

In the commit step, replica i can create a certificate for the proposed block and broadcasts the *block-announce* to the network. The block-announce message has this form:

$$\langle \text{BLOCK-ANNOUNCE}, h, r, B, C \rangle$$

where:

- C is the **full certificate** for the proposed block

Replicas can move to the next height and clear the message logs after receiving a valid block-announce message, even if their timer has expired.

Figure 1 shows the operation of the algorithm in the gracious path. Replica 1 is the proposer, and other replicas attest to the proposed block.

Change Proposer Path

If the network, proposer, or some replicas are faulty, replicas might not receive responses from all $3f+1$ replicas, causing them to become stuck. The change proposer phase is triggered by timeouts and prevents replicas from waiting indefinitely for the proposal to be committed.

The change proposer phase involves a binary agreement. The subject of the agreement is whether there is a quorum certificate for the proposed block. A **quorum certificate** can be issued with at least $2f + 1$ precommit messages. Each replica begins with an initial value $v_i \in \{0, 1\}$, and in the end, all non-faulty replicas decide on either 0 or 1.

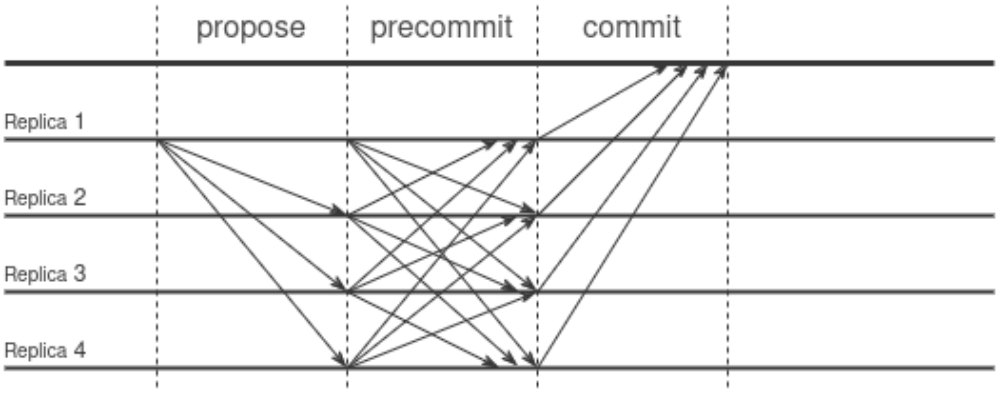

Figure 1: Gracious Path

If a correct replica has at least $2f + 1$ precommit messages from other replicas (including its own), it can created a quorum certificate for the proposal and set the initial value 0 (No). Otherwise, the initial value should be set to 1 (Yes),

The protocol is biased toward 0. Thus, even if all honest parties start with 1, they may still decide on 0 if they obtain the corresponding validating data for 0 during the agreement protocol.

The outcome of the binary agreement determines whether the proposer should be changed or not. If the replicas agree not to change the proposer, it means there is a valid certificate signed by the majority of replicas. Therefore, they can move to the *commit* step and commit the proposed block. The block-announce message in this case should be in this form:

$$\langle \text{BLOCK-ANNOUNCE}, h, r, B, c, P \rangle$$

where:

- c is the quorum certificate for the proposed block
- P is the proof for the decided value 0 in the binary agreement

Figure 2 shows the communication patterns for the Change Proposer path when one replica is faulty.

If the majority of the replicas agree to change the proposer, then they increase the round number and move to the propose step.

Figure 3 shows the communication patterns for the Change Proposer path when the proposer is faulty.

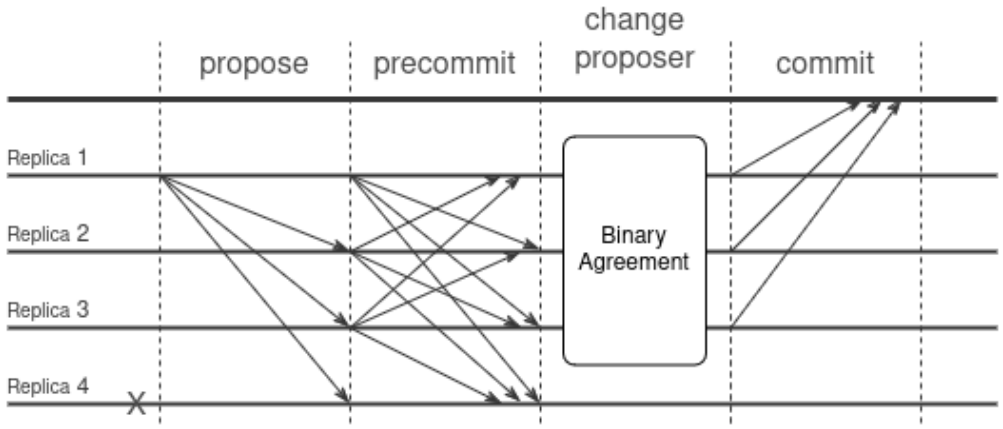

Figure 2: Change Proposer Path - Decided No

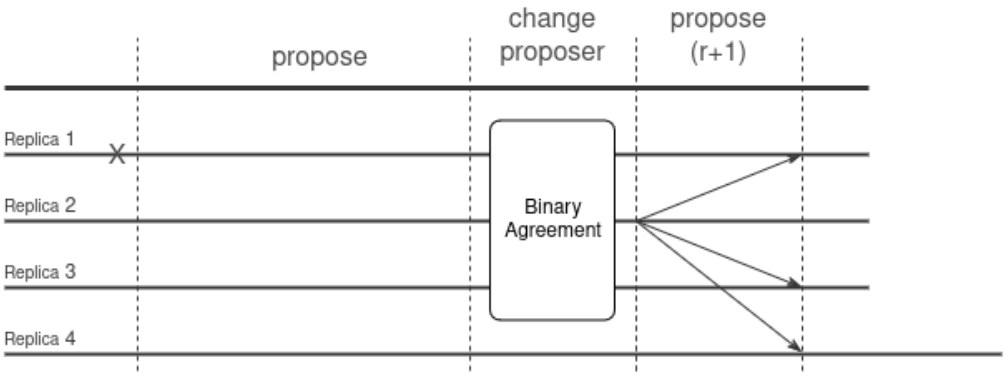

Figure 3: Change Proposer Path - Decided Yes

Binary Agreement

The binary agreement is a *binary validated Byzantine agreement protocol biased towards 0* such that the following condition holds:

Biased External Validity: If at least $f + 1$ honest replicas propose 0, then any honest replica that terminates for h, r decides 0.[2]

The binary agreement in the Cheetah protocol is a fully asynchronous Byzantine agreement protocol that provides liveness when the replicas are unable to decide to move forward.

[2]Validated Byzantine Agreement are explained in section 4 at Secure and Efficient Asynchronous Broadcast Protocols paper

If the timer of a replica expires in round r, the replica starts the binary agreement protocol. During this agreement, even if they don't have the proposal, honest replicas may decide to vote 0 if they obtain a valid quorum certificate for the proposed block.

The binary agreement includes these three steps: **Pre-vote**, **Main-vote**, and **Decide**. The protocol proceeds in rounds $r_{cp} = 0, 1, 2, \ldots$. [3]

Pre-vote Step

In the Pre-vote step, each replica casts a pre-vote for a value $b \in \{0, 1\}$ and broadcasts a pre-vote message to the network. The pre-vote message has this form:

$$\langle\langle \text{CP:PRE-VOTE}, h, r, r_{cp}, b\rangle_{\sigma_i}, \text{justification}\rangle$$

The first round is a special round where each replica starts with an initial value. If the replica has a valid and proper quorum certificate for the proposed block, its initial value is 0; otherwise, its initial value is 1.

$$b = \begin{cases} 0 & \text{has a quorum certificate for the proposed block,} \\ 1 & \text{otherwise.} \end{cases}$$

If a correct replica has a valid proposal and has already sent the precommit message but does not yet have a valid quorum certificate, it should wait before broadcasting the pre-vote message too early. It should wait until it either receives $f + 1$ pre-votes from other replicas with value 1 or receives $2f + 1$ precommit and pre-vote messages from other replicas.

In the next rounds, each replica selects $2f+1$ properly justified main-votes from round $r - 1$ and

$$b = \begin{cases} 0 & \text{if there is a main-vote for 0,} \\ 1 & \text{if there is a main-vote for 1,} \\ 0 \text{ (biased)} & \text{if all main-votes are abstain.} \end{cases}$$

These pre-votes must be supported with appropriate justifications. In the first round, the justification for value 0 is a quorum certificate, while the justification for value 1 is nil.

In the next rounds, a pre-vote for b may be justified in two ways:

[3] Our protocol is highly inspired by the Random Oracles in Constantinople paper.

- **Hard**: that is the quorum certificate for

$$\langle \text{CP:PRE-VOTE}, h, r, r_{cp} - 1, b \rangle$$

- **Soft**: that is the quorum certificate for

$$\langle \text{CP:MAIN-VOTE}, h, r, r_{cp} - 1, \text{abstain} \rangle$$

Main-vote Step

After collecting $2f + 1$ valid and justified pre-votes, each replica casts a main-vote $v \in \{0, 1, \text{abstain}\}$ and broadcasts the main-vote message to the network. The main-vote message has this form:

$$\langle \langle \text{CP:MAIN-VOTE}, h, r, r_{cp}, v \rangle_{\sigma_i}, \text{justification} \rangle$$

The main-vote value v is determined as follows:

$$v = \begin{cases} 0 & \text{if there are } 2f + 1 \text{ pre-votes for 0,} \\ 1 & \text{if there are } 2f + 1 \text{ pre-votes for 1,} \\ \text{abstain} & \text{if there are pre-votes for 0 and 1.} \end{cases}$$

These main-votes must be justified with an appropriate justification. A main-vote for v may be justified in two ways:

- **Non-conflicting**: that is the quorum certificate for

$$\langle \text{CP:PRE-VOTE}, h, r, r_{cp}, b \rangle$$

- **Conflicting**: that consists of the justifications for the two conflicting pre-votes.

Decide Step

After collecting $2f + 1$ valid and justified main-votes, each replica examines these votes. If all votes are for a value $b \in \{0, 1\}$, then the replica decides b and broadcasts the decided message to the network. Otherwise, it moves to the next round. The decided message has this form:

$$\langle \langle \text{CP:DECIDED}, h, r, r_{cp}, b \rangle, \text{justification} \rangle$$

The justification in the decided message is a combination of all main-vote signatures for the value b.

4 Optimization

The ABBA protocol can be optimized so that it can be decided in just one step:

If a replica collects $2f+1$ Pre-Votes for value 0 in any round, it can decide on value 0 and directly publish a BLOCK-ANNOUNCE message.

5 Summary

The Cheetah protocol is optimized for a healthy environment. If all replicas are non-faulty and the network conditions are favorable, the protocol can decide on a proposed block in just one round of voting, within a fraction of a second.

If fewer than f replicas are faulty or if the network is too slow for the Precommit message to be delivered on time to some replicas, the protocol can still decide on the proposed block in the first step in the binary agreement. In this scenario, the Cheetah protocol performs comparably to PBFT.

In the rare case where up to f replicas are faulty or Byzantine, the protocol remains safe, though performance may degrade. However, since such situations are rare in production environments, this overhead is considered reasonable.