Fast Paxos

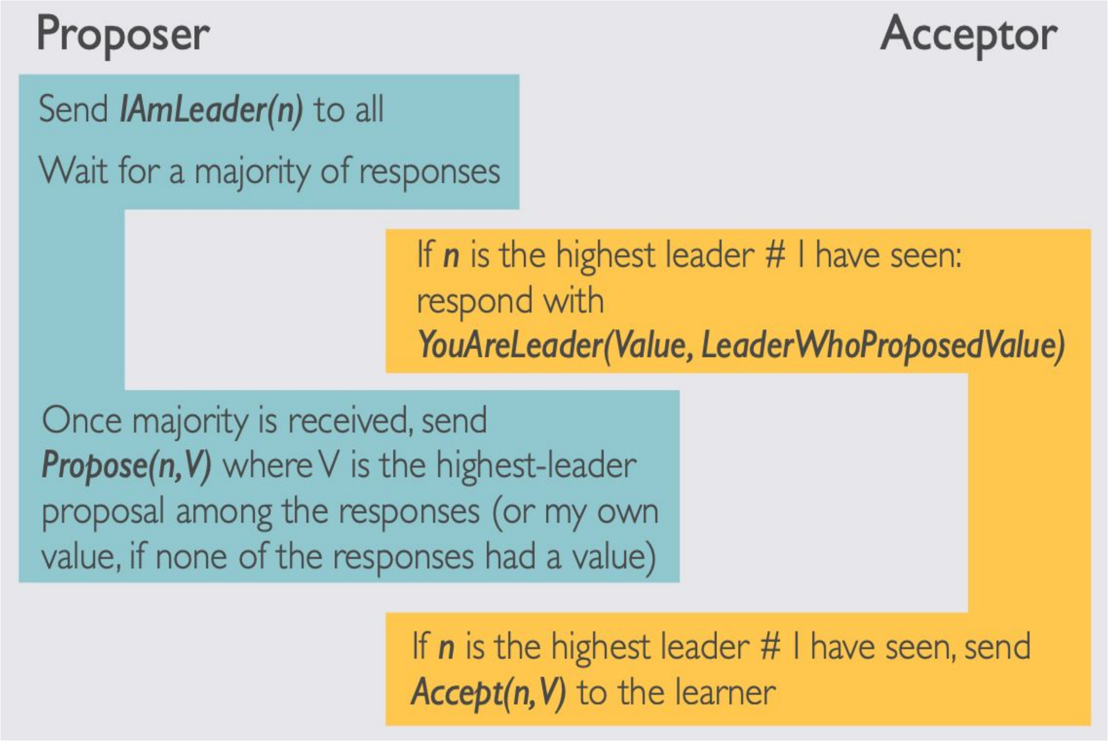
Presented by Tianji Cong

11/1/2021

Background

- A traditional consensus algorithm like Paxos requires three message delays in a client-server system with synchrony
- Paxos is popular as the requirement of three message delays has been shown optimal in practice

Classic Paxos Recap



Class version of Paxos

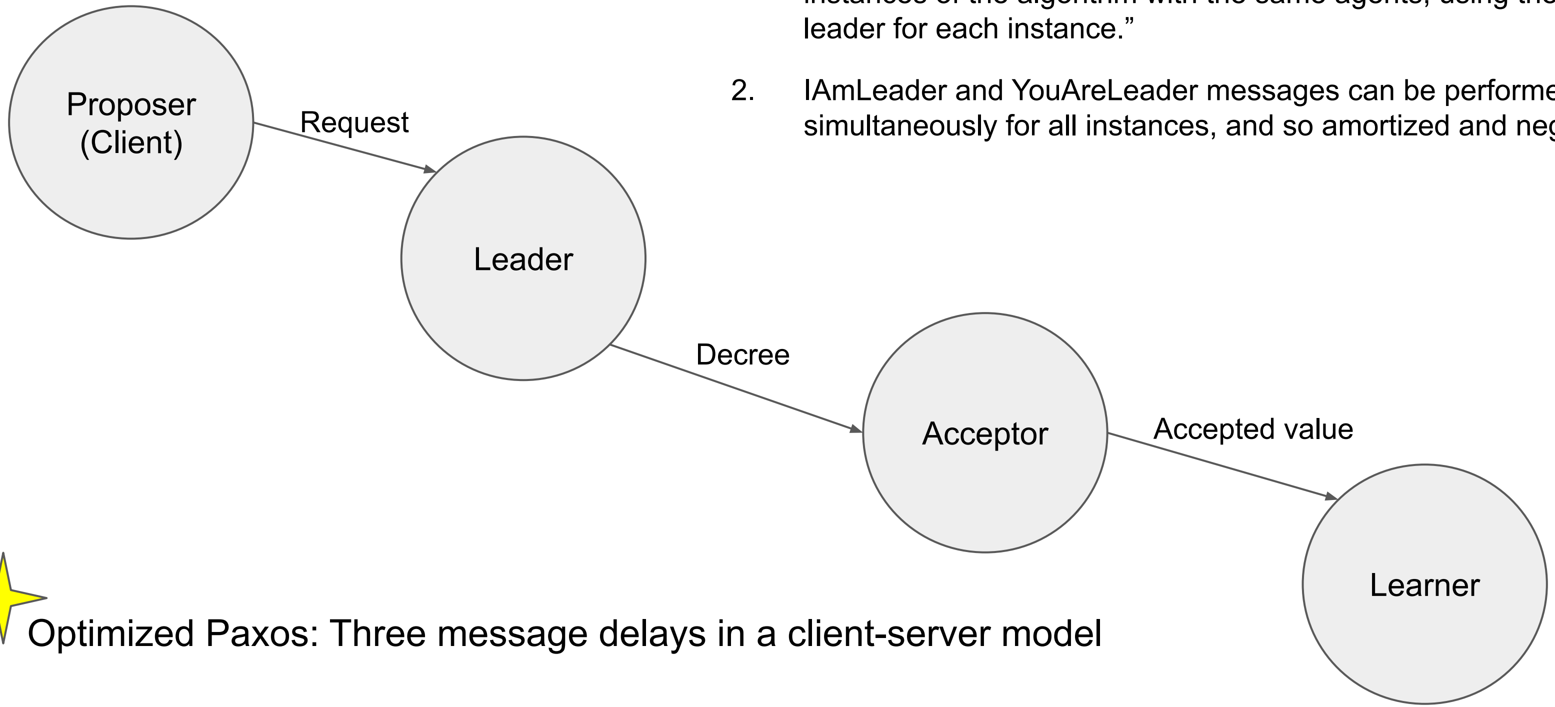
- Three roles: proposer, acceptor, learner
- Does not terminate
- New leader with a higher leader number has to propose the same value that has been learned

Screenshot from EECS 591 lecture slides (Fall 2021) compiled by Prof. Manos Kapritsos

Paxos in Original Paper

Observations / Optimizations:

1. “In most applications, the system executes a sequence of instances of the algorithm with the same agents, using the same leader for each instance.”
2. IAmLeader and YouAreLeader messages can be performed simultaneously for all instances, and so amortized and negligible



Optimized Paxos: Three message delays in a client-server model

Fast Paxos: Extension of Paxos

Paxos: Three message delays in a client-server model

Challenge: Can we do better?

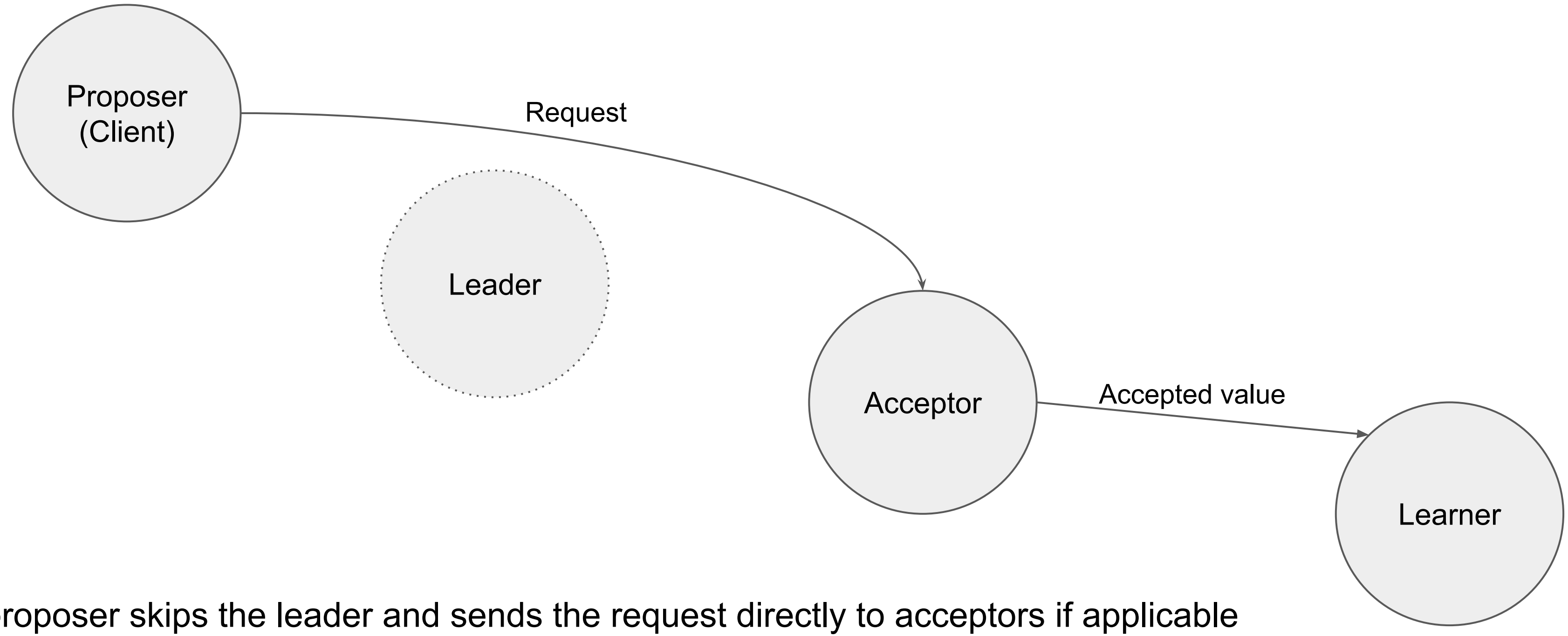
I’ll be live when the network is synchronous while keeping safe all the time

Paxos

I’ll cheat when there is no concurrent requests; otherwise, remain as Paxos

Fast Paxos

Key Idea of Fast Paxos

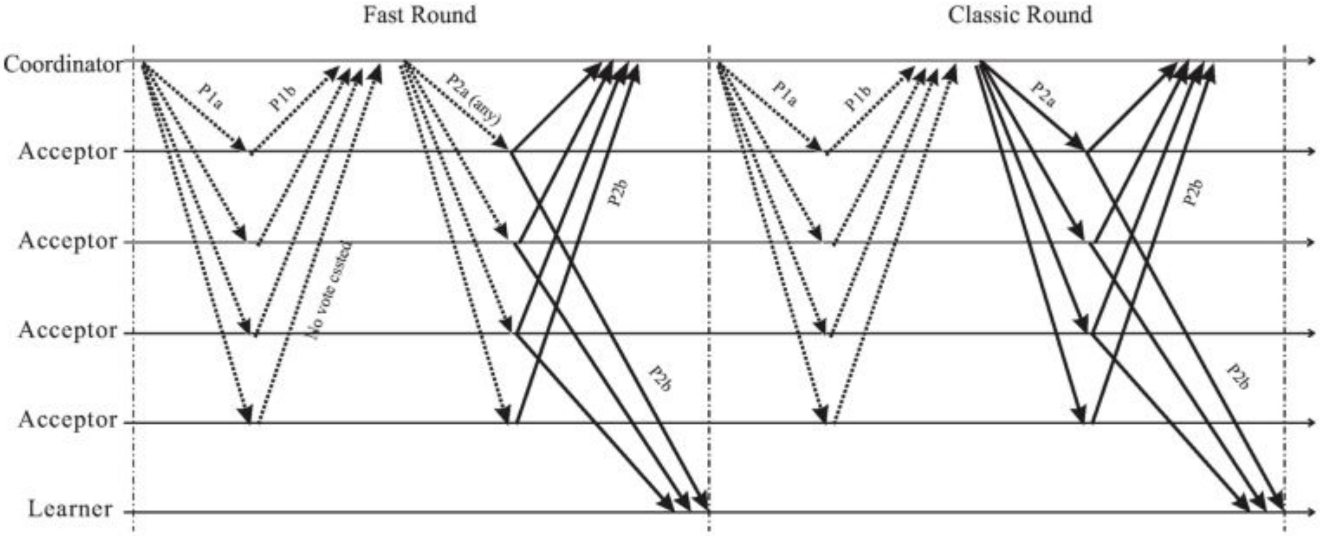


The proposer skips the leader and sends the request directly to acceptors if applicable

Fast Paxos Overview

- A round (An execution) is either a fast round or a classic round
- In a fast round (with Leader elected)
    - Leader sends a special message (“any” message) to acceptors without proposing any value if no value was learned before
    - Acceptors receiving “any” message accept proposed values directly from a client and send it to learners
    - A learner learns a value if it receives a quorum of acceptors
- The classic round is the same as an execution of classic paxos

Fast Round vs. Classic Round



Screenshot from: Zhao, Wenbing. "Fast paxos made easy: Theory and implementation." *International Journal of Distributed Systems and Technologies (IJDST)* 6.1 (2015): 15-33

How Fast Paxos Helps?

When could a fast round be executed?

1) A Leader elected as the system starts running **AND**
2) No value has been proposed, so Leader can send “any” message

Thus,

- ➔ Allow a single “any” message for all instances of Fast Paxos
- ➔ Eliminate a message delay from Leader to Acceptors that proposes decree

A classic run takes over if a consensus cannot be reached in the fast run

No Free Lunch

- Even when there is no failure
  - A simple majority is not enough to tell if a value has been learnt in the previous round
  - Need more than 3/4 of the acceptors in the fast quorum
- Fast Paxos “degrades” to classic Paxos
  - Failure of Leader
  - Client requests collision
    - Multiple clients send concurrent requests
    - It’s possible that different Acceptors accept different values in the fast round
    - Learners get confused when seeing different values in the quorum

Quorum Requirement

- Must guarantee to learn the same value as in previous rounds
  - Cannot have two quorums in the same round that accept different values **WHILE**
  - Two quorums and a majority set of acceptors (following Leader) have empty intersection
- Quorum requirements (for any round numbers *i* and *j*)
  - Any *i*-quorum and any *j*-quorum have non-empty intersection
  - If *j* is a fast round #, then any *i*-quorum and any two *j*-quorums have a non-empty intersection

Choosing Quorums

- Let N be the number of acceptors, and choose F and E such that
  1) Any set of at least $N - F$ acceptors is a classic quorum **AND**
  2) Any set of at least $N - E$ acceptors is a fast quorum
- Can always assume $E \leq F$
  - Any *i*-quorum and any *j*-quorum have non-empty intersection
    - ➔ $N > 2F$
  - Any two fast quorums and any classic or fast quorum have a non-empty intersection
    - ➔ $N > 2E + F$
- For fixed N, two ways to choose E and F to maximize one or the other
  - Maximize E ⇒ $E = F = \lceil N / 3 \rceil - 1$
  - Maximize F ⇒ $F = \lceil N / 2 \rceil - 1$ and $E = \lfloor N / 4 \rfloor$

Collision Recovery

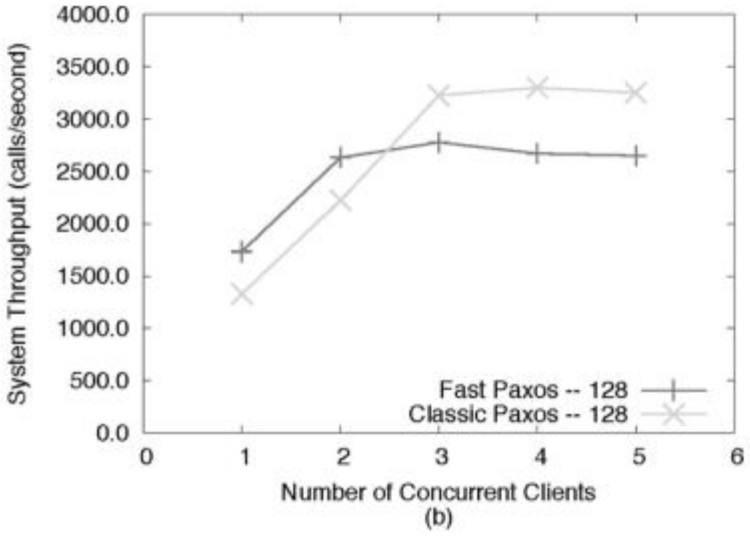
Collision: In a fast round, Acceptors receive concurrent requests in different orders

Recovery: Leader can begin a classic round after detecting a collision

- ➔ Coordinated Recovery
    - Suppose round *i* is a fast round, round *i* and *i+1* have the same leader
    - Leader (as a learner) receives *p2b* message from acceptors
    - Leader reuses *p2b* message in round *i* as *p1b* message in round *i+1*
    - The rest procedure of a classic round follows as usual

Is Fast Paxos Practical?

- No performance evaluation in the original paper
- Seems brittle as the scenario of concurrent client requests is not uncommon



Screenshot from: Zhao, Wenbing. "Fast paxos made easy: Theory and implementation." *International Journal of Distributed Systems and Technologies (IJDST)* 6.1 (2015): 15-33

Discussion Time