Consensus Problem

• A set of processes to achieve a single value

• Asynchrony with Non-Byzantine failures
  • Communications can be reordered, duplicated, lost, but never corrupted
  • Processes can fail, restart, but never behave incorrectly
Consensus Problem

- Processes as agents in client-server model Paxos

Client
- Proposer
- Coordinator/Leader
- Accepter
- Learner

Server
Consensus Problem

• Classic Paxos
  • 3 message delays to reach consensus:
    Proposer — Coordinator — Acceptor — Learner

• Faster?
“Fast” Paxos

- Impossible to shrink message delays to 2
  - Collisions
- One step back …
  - 2 if no collisions
  - 3 if collisions occur
- … but always safe!
Paxos Recap

Proposers → Coordinator

Coordinator → Acceptors
Paxos Recap: A Single Round

Proposers

Coordinator

Acceptors

Learner

Values

Pick a value

Majority Set

Phase 1a

Phase 1b

Phase 2a

Phase 2b

(“Prepare”)

(“Accept”)

Majority Set

“Prepare”

“Accept”
Paxos Recap

- Nontriviality (*Integrity*)
- Consistency (*Agreement*)
- Progress (*Termination*)
- Validity is guaranteed potentially.
Paxos Recap

• Improvements
  • Reduce number of messages used (not message delay!)

• Cost Analysis
  • Number of messages used: Multicast \( \lceil N/2 \rceil + 3 \)
    Unicast \( N(\lceil N/2 \rceil + 1) \)

• Latency is always 3 message delays.
Fast Paxos

- We first make some changes:
  - Classic Paxos → Fast Paxos
  - Majority Set → Quorums
  - $i$-quorums
Rounds In Fast Paxos

Two kinds of rounds

- **Classic round**: just classic Paxos

- **Fast round**:
  - If no acceptors have voted.
  - Coordinator can be bypassed
Fast Round in Fast Paxos

Proposers

Coordinator

Acceptors

Phase 1a

Phase 1b

Phase 2a

Phase 2b

"Any"!

Values

"i-quorum"

Learner
Fast Paxos

• “Any” message in fast rounds
  • Acceptors can vote for different values
  • Coordinator’s value picking rule no longer provides consistency if quorums are just majority sets …
  • … even in a classic round!
Picking Rule Fails

Number of acceptors: 5
Quorum size: 3

Round 1

Coordinator

- Acceptor 1: Not voted
- Acceptor 2: Not voted
- Acceptor 3: Not voted
- Acceptor 4: Not voted
- Acceptor 5: Not voted
Picking Rule Fails

Number of acceptors: 5
Quorum size: 3

Round 1

Coordinator

Any!

Acceptors:
- Acceptor 1
- Acceptor 2
- Acceptor 3
- Acceptor 4
- Acceptor 5
Number of acceptors: 5
Quorum size: 3

Round 1

Proposer 1

Proposer 2

I have X!

I have Y!

Acceptor 1

Vote for X!

Acceptor 2

Vote for X!

Acceptor 3

Vote for X!

Acceptor 4

Vote for Y!

Acceptor 5

Vote for Y!

Learner

OK I'll go for X. X is chosen!
Picking Rule Fails

Number of acceptors: 5
Quorum size: 3

Round 2

Coordinator

Accepter 1
Accepter 2
Accepter 3
Accepter 4
Accepter 5

Learner

I voted for X in round 1
I voted for Y in round 1
I voted for Y in round 1
I voted for Y in round 1

Wait, what?
Picking Rule Fails

Number of acceptors: 5
Quorum size: 3

Round 2

Coordinator
How about Y?

Wait, what?

Acceptor 1
Acceptor 2
Acceptor 3
Acceptor 4
Acceptor 5
Quorum Picking

• To preserve consistency, we need to add more limitations

• **Quorum Requirement:** For any round number $i$ and $j$,
  
  **A.** Any $i$-quorum and any $j$-quorum have non-empty intersection.
  
  **B.** If $j$ is a fast round number, then any $i$-quorum and any two $j$-quorums have non-empty intersection.
Coordinator Picking Rule

- In a **classic round** $i$, coordinator receives **Phase 1b** message from an $i$-quorum $Q$, and the largest round number $Q$ voted is **round** $k$. All values $Q$ voted in round $k$ forms a set $V$.
  - If $V$ is empty, choose any value from proposers.
  - If $V$ only contains one value, choose that value.
  - If $V$ contains more than one value, coordinator tries to intersect each possible $k$-quorums $R$ with $Q$.
    - In $R \cap Q$, if some acceptor’s most recent vote is in round $k$, choose that value.
    - Otherwise, choose any value from proposers.

*Same as classic Paxos*
Fast Paxos

- If $V$ contains more than one value, coordinator tries to intersect each possible $k$-quorums $R$ with $Q$.
  - In $R \cap Q$, if some acceptor’s most recent vote is in round $k$, choose that value.

- More than one value? $v$ & $w$?
- By quorum requirement(b), $R_w \cap R_v \cap Q \neq \emptyset$!
- Which means?
Fast Paxos

- $N$ processes, $N-F$ acceptors for classic quorum, $N-E$ for fast quorum

**Quorum Selection:**

- $N > 2F$
- $N > 2E + F$
- Maximizing $E$: $E = F = \lceil N/3 \rceil - 1$
- Maximizing $F$: $F = \lceil N/2 \rceil - 1$, $E = \lfloor N/4 \rfloor$
Fast Paxos

- “Any” message in a *fast* round causes collision
- No value can be chosen
- Recovery is required
  - Coordinated recovery
  - Uncoordinated recovery
Coordinated Recovery

- Phase 2b message in round $i$:
  
  - Acceptors
  
  I voted for round $i$, so I won’t vote for rounds less than $i$!

- Phase 1b message in round $i+1$:
  
  - Acceptors
  
  Here is my latest vote, in round $i$, and I won’t follow an earlier coordinator!

  - Coordinator

  Learner

  Here is my vote for this round!

  Same information
Coordinated Recovery

Phase 1a (round $i+1$)

Proposers

Coordinator

Acceptors

Phase 2b (round $i$)

Learner

Phase 1b (round $i+1$)

Pick a value

Phase 2b (round $i+1$)

Learner

Phase 2a (round $i+1$)
Uncoordinated Recovery

Proposers

Coordinator

Acceptors

Acceptors

Acceptors

Function as

Phase 2b
(round i)

Phase 2a
(round i+1)

Learner

Learner

Once receive from (i+1)-quorum acceptors

Pick value just like they receive these values from proposers

Can Still Collide
Uncoordinated Recovery Improved

Proposers

Coordinator

I’ll tell them in round i!

Acceptor

Phase 2b (round i)

function as Proposers

Phase 2a (round i+1)

Acceptor

Acceptors

Learner

No Collision

Acceptors

Phase 2b (round i+1)

Learner

Once receive from (i+1)-quorum acceptors

Pick value just like they receive these values from proposers
Fast Paxos Cost

- Latency decreased
- Number of messages used
  - For $E=F$, $\lceil 2N/3 \rceil + 2$ for multicast, and $N(\lceil 2N/3 \rceil + 1)$ for unicast
- Additional cost for recovery
  - Coordinated: additional phase 2
  - Uncoordinated: additional phase 2b

<table>
<thead>
<tr>
<th></th>
<th>Multicast</th>
<th>Unicast</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coordinated: additional phase 2</td>
<td>$\lceil 2N/3 \rceil + 2$</td>
<td>$N(\lceil 2N/3 \rceil + 1)$</td>
</tr>
<tr>
<td>Uncoordinated: additional phase 2b</td>
<td>$\lceil 2N/3 \rceil + 1$</td>
<td>$(N-1)(\lceil 2N/3 \rceil + 1)$</td>
</tr>
</tbody>
</table>
Thank You.