Paxos Made Moderately Complex

Arvind Krishnamurthy

*University of Washington*
Paxos

Phase 1
- Send prepare messages
- Pick value to accept

Phase 2
- Send accept messages
**Can we do better?**

Phase 1: “leader election”
- Deciding whose value we will use

Phase 2: “commit”
- Leader makes sure it’s still leader, commits value

What if we split these phases?
- Lets us do operations with one round-trip
Roles in PMMC

Replicas (like learners)
- Keep log of operations, state machine, configs

Leaders (like proposers)
- Get elected, drive the consensus protocol

Acceptors (*simpler* than in Paxos Made Simple!)
- “Vote” on leaders
Ballots (or proposal #s) in PMMC

(leader, seqnum) pairs

0  0.0, 1.0, 2.0, 3.0, 4.0, ...

1  0.1, 1.1, 2.1, 3.1, 4.1, ...

2  0.2, 1.2, 2.2, 3.2, 4.2, ...

3  0.3, 1.3, 2.3, 3.3, 4.3, ...
Paxos Made Moderately Complex
Paxos Made Moderately Complex
Acceptors

Acceptor

```
ballet_num: 0
accepted:[]
```
Acceptors

p1a(0.1) → Acceptor

ballot_num: _
accepted:[]
Acceptors

p1a(0.1)  →  Acceptor

ballot_num: 0.1
accepted:[]
Acceptors

p1a(0.1)
p1b([])

Acceptor

ballot_num: 0.1
accepted:[]
Acceptors

Acceptor

\[
\text{ballot\_num: 0.1}
\]

\[
\text{accepted: []}
\]
Acceptors

p1a(0.0) -> Acceptor

ballot_num: 0.1
accepted: []
Acceptors

p1a(0.0) -> Acceptor

Nope!

ballot_num: 0.1
accepted: []
Acceptors

Acceptor

```
ballet_num: 0.1
accepted:[]
```
Acceptors

\[ p2a(<0.1, 0, A>) \]

Acceptor

ballot_num: 0.1
accepted: []
Acceptors

$p2a(<0.1, 0, A>)$

Acceptors

ballot_num: 0.1
accepted:[<0.1, 0, A>]
Acceptors

\[ p2a(<0.1, 0, A>) \]

Acceptors

\[ \text{ballot\_num: 0.1 accepted:} [<0.1, 0, A>] \]

OK!
Acceptors

Acceptors

ballot_num: 0.1
accepted:[<0.1, 0, A>]
Acceptors

\[ p2a(<0.0, 0, B>) \]

Acceptors

| ballot_num: 0.1 |
| accepted: [(<0.1, 0, A>)] |
Acceptors

\[ \text{p2a(<0.0, 0, B>)} \]

Nope!

\[ \text{ballot_num: 0.1} \]
\[ \text{accepted: [<0.1, 0, A>]} \]
Acceptors

Acceptor

ballot_num: 0.1
accepted:[<0.1, 0, A>]
Acceptors

Ballot numbers increase

Only accept values from current ballot

Never remove ballots

If a value \( v \) is chosen by a majority on ballot \( b \), then any value accepted by any acceptor in the same slot on ballot \( b' > b \) has the same value.
Leader: Getting Elected

Leader

active: false
ballot_num: 0.0
proposals: []

Acceptor

p1a(0.0)

Acceptor

Acceptor
Leader: Getting Elected

Leader

active: false
ballot_num: 0.0
proposals: []

Acceptor

Nope!

Acceptor

Nope!

Acceptor

Nope!
Leader: Getting Elected

Leader

active: false
ballot_num: 1.0
proposals: []

Acceptor

Acceptor

Acceptor
Leader: Getting Elected

Leader

active: false
ballot_num: 1.0
proposals: []

Or...

Acceptor
Acceptor
Acceptor
Leader: Getting Elected

Leader

active: false
ballot_num: 0.0
proposals: []

Acceptor

OK([])!

Acceptor

OK([])!

Acceptor
When to run for office

When should a leader try to get elected?
- At the beginning of time
- When the current leader seems to have failed

Paper describes an algorithm, based on pinging the leader and timing out

If you get preempted, don’t immediately try for election again!
**Leader: Handling proposals**

Active: true

Ballot_num: 0.0

Proposals: []

Op1 should be A (A = “Put k1 v1”)

---

Acceptor

Replica

Op1 should be A (A = “Put k1 v1”)

---

Acceptor

Acceptor

Acceptor
Leader: Handling proposals

Leader

active: true
ballot_num: 0.0
proposals: [<1, A>]

Acceptor

Replica

Acceptor
Leader: Handling proposals

Leader:
- active: true
- ballot_num: 0.0
- proposals: [<1, A>]

Acceptors:
- p2a(<0.0, 1, A>)

Replica:
Leader: Handling proposals

active: true
ballot_num: 0.0
proposals: [<1, A>]

Leader

Acceptor

Acceptor

Acceptor

Replica

Nope!

Nope!

Nope!
Leader: Handling proposals

Leader

- active: false
- ballot_num: 0.0
- proposals: [<1, A>]

Acceptor

Replica
Leader: Handling proposals

- **Leader**
  - active: false
  - ballot_num: 0.0
  - proposals: [<1, A>]

- **Acceptors**
  - Or...

- **Replica**
Leader: Handling proposals

active: true
ballot_num: 0.0
proposals: [<1, A>]

Leader

OK!

Acceptor

OK!

Acceptor

OK!

Acceptor

Replica
Leader: Handling proposals

active: true
ballot_num: 0.0
proposals: [<1, A>]

Op1 is A
Questions

What should be in stable storage?
What are the costs to using Paxos? Is it practical enough?