Flexible Paxos: Quorum Intersection Revisited

Heidi Howard, Dahlia Malkhi, and Alexander Spiegelman

Presented by Andrew Loveless
Two phases of Paxos

Phase 1:
- Select a proposer as the leader

Phase 2:
- Propose a value; if enough accept it, it is learned
Both phases require quorums of $f + 1$ acceptors

I.e. *majority quorums*

Any two majority quorums intersect at $\geq 1$ acceptor
Quorums (cont.)

- **Safety**
  - Phase 1 and Phase 2 quorums intersect
  - If a value was learned in Phase 2, a new proposer will find out about that value
  - If f acceptors from Phase 2 quorum die, the other acceptor will be part of any future quorum

- **Liveness**
  - If f acceptors crash, there is still a quorum
Flexible Paxos

- **Key Idea:**
  - Majority quorums are *sufficient*, but not *necessary*

- **What’s Needed:**
  - Phase 1 and Phase 2 quorums must intersect

- **Not Needed:**
  - No need for quorums in same phase to intersect

Any quorum system with this property can be used. Not just majority quorums!
Small Quorums = Benefits

- Goal is to **reduce** quorum sizes
- **Reduce Latency**
  - P1: Proposers wait for fewer messages to become leader
  - P2: Learners wait for fewer messages to learn value
- **Increase Throughput**
  - P2: Disjoint sets of acceptors can accept proposals
Small Quorums = Benefits

- **Goal is to reduce quorum sizes**
- **Reduce Latency**
  - P1: Proposers wait for fewer messages to become leader
  - P2: Learners wait for fewer messages to learn value
- **Increase Throughput**
  - P2: Disjoint sets of acceptors can accept proposals
Immediate benefit

- $2f + 1$ acceptors is always odd
- What if there is an even number of acceptors?
  - E.g. Want higher availability
- Classic Paxos: $N = 10$
  - majority quorum size $= \lfloor N/2 \rfloor + 1 = 6$

More intersection than necessary
Immediate benefit (cont.)

- Can reduce size of Phase 2 quorum to $N/2$
  - Still has required intersection

Phase 1 quorum

Phase 2 quorum

No downside. Works for all even $N$. 
Immediate benefit (cont.)

- Say $N/2 = 5$ acceptors crash
  - Neither can select a new leader
  - Paxos cannot learn new values
  - But FPaxos can – still makes progress
Different Quorum Systems

- Majority Quorums (Classic Paxos) ✔
- Simple Quorums
- Grid Quorums

Different quorum systems have different tradeoffs
Simple Quorums

- Generalization of majority quorums
- **Main Idea:**
  - Decrease size of Phase 2 quorums (Q2)
  - Increase size of Phase 1 quorums (Q1)
  - |Q1| + |Q2| > N (# Acceptors)
Simple Quorums (cont.)

- When is this a good tradeoff? SMR and Multi-Paxos
  - Phase 1 more expensive, but less frequent
  - Phase 2 cheaper, more frequent

![Diagram showing the phases of a quorum system with leader election and data propagation.]
Simple Quorums (cont.)

- Smaller P2 Quorums:
  - Improved latency and throughput

- Larger P1 Quorums:
  - Increased latency of leader selection
  - Lower availability – may be two few acceptors to recover from leader failure
Simple Quorums (cont.)

- **Smaller P2 Quorums:**
  - Improved latency and throughput

- **Larger P1 Quorums:**
  - Increased latency of leader selection
  - Lower availability – may be too few acceptors to recover from leader failure

- **Optimization from Cheap Paxos:**
  - Send proposal to $|Q2|$ acceptors only
  - If they don’t reply, send to more acceptors
  - Reduces messages, but increases latency
Different Quorum Systems

- Majority Quorums (Classic Paxos)
- Simple Quorums
- Grid Quorums

**Limitation:** Still need to trade small Q2 for large Q1. Can we do better?
Grid Quorums

- Create grid, where $N_1 \times N_2 = \# \text{ acceptors}$, E.g. 20
- Each square is an acceptor
Grid Quorums (cont.)

- Create grid, where $N_1 \times N_2 = \#$ acceptors, E.g. 20
- Each square is an acceptor
- Each row is a P1 quorum
- Each column is a P2 quorum

Note: Quorums from same phase never intersect
Grid Quorums (cont.)

- **Advantage:**
  - Size of both P1 and P2 quorums is reduced!

**Paxos:**
|Q1| and |Q2| = ⌊N/2⌋ + 1 = 11

**FPaxos:**
|Q1| = 5 and |Q2| = 4
Grid Quorums (cont.)

- **Improved resilience?** E.g. $N = 10$

- **Simple quorums**
  - $|Q_1| + |Q_2| = N + 1$
  - $|Q_1| = 9, |Q_2| = 2$

Any set of $|Q_2|$ failures means you cannot recover from leader failure
Grid Quorums (cont.)

- Improved resilience? E.g. $N = 10$

- Grid quorums
  - $|Q_1| = 5$, $|Q_2| = 2$

Two acceptors crashed. But can still form a P1 quorum!
Resilience depends on which acceptors crash

- 4 failures
  - 1 possible $Q_1$
  - 1 possible $Q_2$

- 2 failures
  - 0 possible $Q_1$
  - 4 possible $Q_2$

Cannot select a new leader!
Evaluation

- Modified LibPaxos3 to use simple quorums
- $|Q1|$ and $|Q2|$ chosen at random
- 64 byte requests
  - 10 concurrent requests at a time
Evaluation

Decreasing $|Q2|$ decreases latency and increases throughput

(a) Performance of FPaxos and LibPaxos3 with 5 replicas.

(b) Performance of FPaxos and LibPaxos3 with 8 replicas.
Evaluation

Not a straightforward comparison.
Fpaxos sends to quorum of acceptors

(a) Performance of FPaxos and LibPaxos3 with 5 replicas.

(b) Performance of FPaxos and LibPaxos3 with 8 replicas.
Related Work

- Fast Paxos (2005)
- Ring Paxos (2010)
- Generalized Paxos (2005)
- Egalitarian Paxos (2013)
- many others …
Strengths/Weaknesses

- Strengths
  - Novel insight regarding 20 year old paper
  - Easy to modify Paxos (e.g. simple quorums)
  - Can trade off latency/throughput and resilience

- Weaknesses – evaluation
  - Send to quorum vs. all acceptors
  - Only 64 byte requests
  - Only 10 concurrent requests
  - Simulated 10 Mbit/s network