Mencius:
Building Efficient Replicated State Machines for WANs

Paper by Yanhua Mao, Flavio P. Junqueira, and Keith Marzullo

Presentation by Allison Easton

Paper link
Motivation - Wide Area Network (WAN)

- Span large geographical distances
- Used to connect local area networks (LAN)
- Higher latency and more variance than LAN
- Lower bandwidth than LAN
Motivation - Problems with Paxos

- Non-leader based Paxos has high message complexity
- Leader based requires all requests to be sent to the leader
- Bandwidth limitation at leader
Coordinated Paxos & Simple Consensus

Coordinated Paxos

- Slots are pre-assigned to servers
- Leaders = Coordinators

Simple Consensus

- Everyone that isn’t coordinator proposes noop
- Suggest = non-noop propose
- Skip = noop propose
Uneven Load to Servers

- Cannot commit to higher sequence numbers
- No client requests to one coordinator stalls whole system

Solution

- Skip slots you haven’t used when you receive a suggest
Uneven Load to Servers

- Cannot commit to higher sequence numbers
- No client requests to one coordinator stalls whole system

Solution

- Skip slots you haven’t used when you receive a suggest
Server Crash

- Cannot commit to higher sequence numbers
- Cannot broadcast skip command

Solution

- Other server revokes the slot from the failed server
- Revoke = leader change in paxos
Server Crash

- Cannot commit to higher sequence numbers
- Cannot broadcast skip command

Solution

- Other server revokes the slot from the failed server
- Revoke = leader change in paxos
Optimizations

1. Append SKIP to Accept messages
2. Don’t skip turns until a higher proposal is seen
   a. Unless you reach a certain time ($\tau$) or message ($\alpha$) bound
3. If a server fails, revoke multiple slots ($\beta$)
Constant Choices

\( \tau \text{ & } \alpha \)

- Large enough to combine skip messages
- Small enough servers don’t fall behind
- Implementations used 50ms and 20 messages

\( \beta \)

- Higher makes overhead of dead server smaller
- False suspicion can inflate indices with large beta
- Implementation found lower bound of 7,500 and used 100,000
Optimizations - Out of Order Commit

- Server receives simultaneous Suggest messages
-Responds with OK
- Receives out of order Learns
- Can’t commit the later instruction

Solution

- Dependency checking
- Commit if instructions are commutable
Optimizations - Out of Order Commit

- Server receives simultaneous Suggest messages
- Responds with OK
- Receives out of order Learns
- Can’t commit the later instruction

Solution

- Dependency checking
- Commit if instructions are commutable
Results - Normal Operation

Figure 4: Throughput for 20 Mbps bandwidth

Figure 8: Throughput vs. number of sites
Results - Failure

Figure 7: Mencius and Paxos’s throughput under failure
Results - Out of Order Commit

Figure 11: Commit latency vs offered client load

(a) $\rho = 4,000$, no network variance
(b) $\rho = 0$, no network variance
(c) $\rho = 0$, with network variance
Summary

Correctness

- **Coordinated Paxos** = each server has pre-assigned slots it is a leader in
- **Simple Consensus** = nobody but the leader can propose a non noop value
- If a server doesn’t have a client value and a higher slot does, **skip** turn
- If a server dies, **revoke** its turn by running leader election

Efficiency

- Don’t broadcast skip messages
- Revoke multiple slots at a time
- Commit commutable requests out of order