MDCC: MULTI-DATA CENTER CONSISTENCY

Authors: Tim Kraska, Gene Pang, Michael J. Franklin, Samuel Madden, Alan Fekete
Presenter: Zizhen Lian
BACKGROUND

- Problem:
  Outage of a single data center may cause outage or data lost for services

- Solution:
  Geo-replicated data stores

- New Problem:
  - High network latency (hundreds of ms)
  - Consistency
  - Availability
EXISTING SYSTEMS

• 2-Phase Commit
  • 2 Phase: Prepare and Commit
  • All nodes need to respond

• Classic Paxos
  • 2 round-trip communication
  • 1 transaction at a time

• Other Paxos
  • Attacking only one problem
MDCC

• Based on Multi-Paxos, Fast Paxos and Generalized Paxos
• Read-Committed Isolation
• Atomic Durability for updates (no update or persist)
• No write-write conflict
MDCC - MULTI

- Multi-Paxos per record
- \([StartInstance, EndInstance, Ballot]\)
- Option
  - \(v_{\text{read}} \rightarrow v_{\text{write}}\)
    - Acceptor makes deterministic decision based on ballot
- Clients can’t abort proposed transaction
- Single round-trip if all record masters are local
MDCC - FAST

- All updates starts with fast ballot number
  \((1,3, v_0 \rightarrow v_1), (2,4, v_1 \rightarrow v_2), (3,4, v_1 \rightarrow v_3), (5,4, v_1 \rightarrow v_2)\)

- Clients send proposal directly to acceptors
  
- Accept if fast quorum agrees
  \([2,4, v_1 \rightarrow v_2), (3,4, v_1 \rightarrow v_3)]\)

- Collision recovery by master
  \([3,4, v_1 \rightarrow v_3), (5,4, v_1 \rightarrow v_2)]\)

- Commutative Updates

- Demarcation
MDCC

- Generalized Paxos on each record,
- *Fast* quorum to make decision
- Disallowing aborts for successfully prepared records
- Piggybacking notification of commit state on subsequent transaction
MDCC