

Hot-Standby

Prof. Hasso Plattner

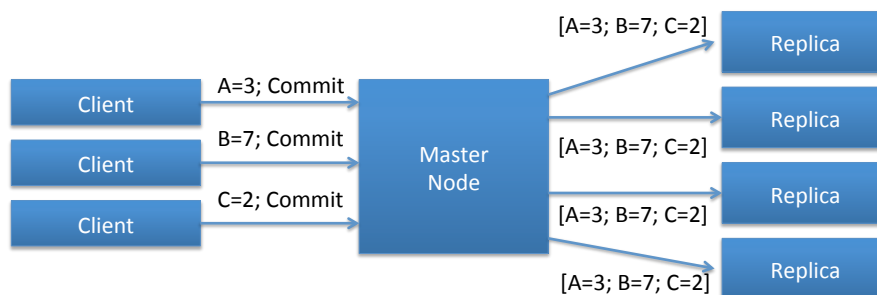
Hot Standby

- ❑ Problem: How to achieve High Availability?
 - ❑ Master could crash any time: Hardware failures, software crash, power failure, etc...
- ❑ Idea: Use replica to continue query processing
- ❑ **But:** Replica might not have latest transactions, how to make sure all committed transactions are present?
 - ❑ On commit, transactions must be replicated to at least 1 replica
 - ❑ Replica with the highest commit timestamp becomes new master
- ❑ k-safety
 - ❑ Wait for acknowledgement of *k nodes* instead of 1 node, survive the crash of master and k-1 replicas

Synchronous Replication

- During group commit phase, master sends log data to all replicas
- Each replica acknowledges that it received (but not yet processed) the data
- After master received k acknowledgments the commit is confirmed to the client

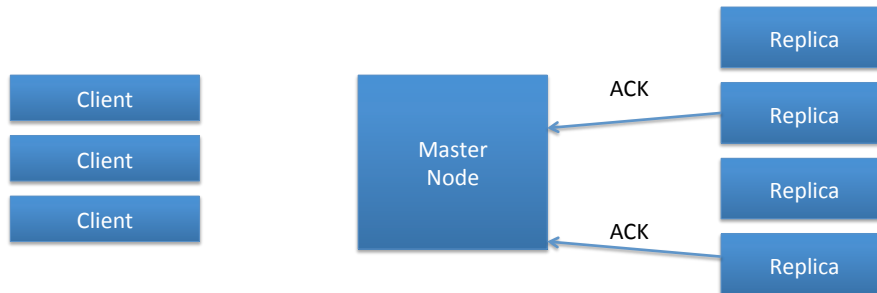
Replication Step 1: Commit Forwarding



Replication

Step 2: Acknowledgments

Shown for $k=2$

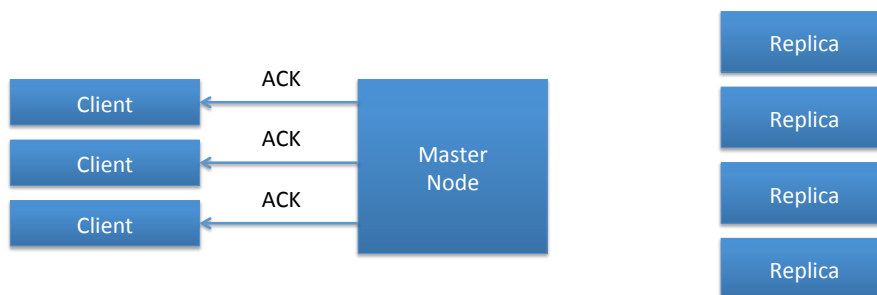


5

Replication

Step 3: Client Commit

Shown for $k=2$

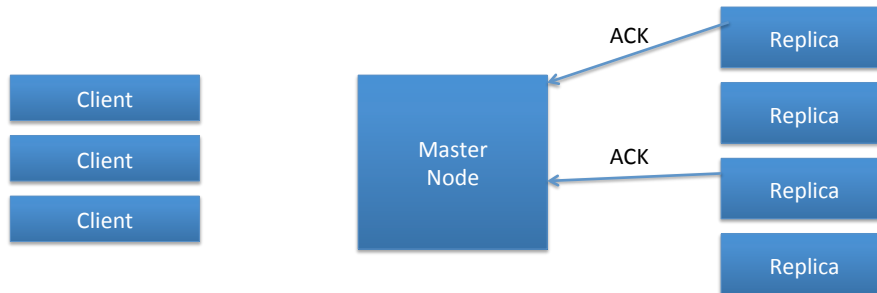


6

Replication

Step 4: Outstanding ACKs

Shown for $k=2$



7

Failover

- If master fails, a replica has to take over
 - How is failing master detected?
 - Which replica takes over?
 - Which data needs to be replayed?

- Master sends heartbeat message every 50 ms
 - Part of the group commit
 - Either with new transactional data or empty message
 - If all replicas miss 3 messages, failover is initiated

8

Failover II

- k replicas are guaranteed to have received the latest data
- The replicas vote for one of them (e.g. visible by most other replicas)
- The replica is promoted to be the new master
- All other replicas inform the new master about the current version of the data that they have
- New master replays all missing messages