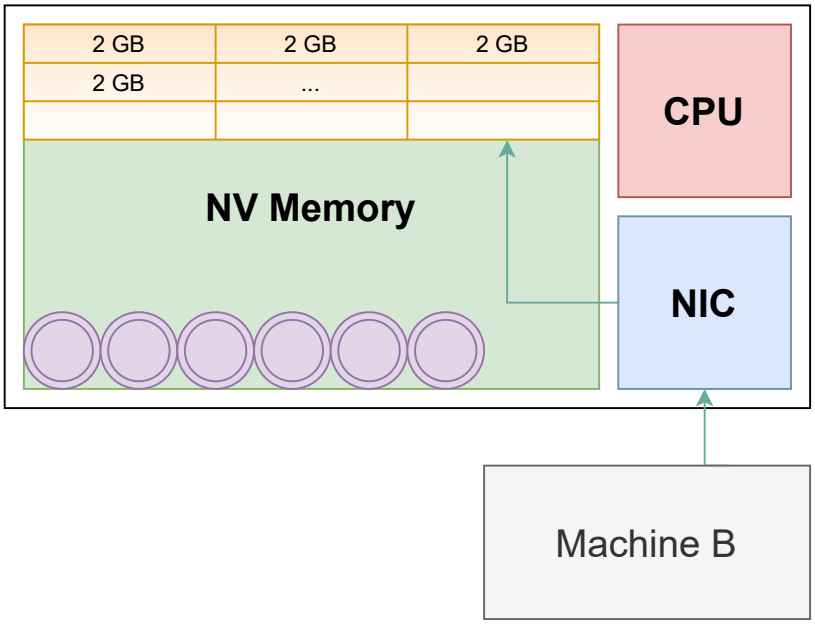
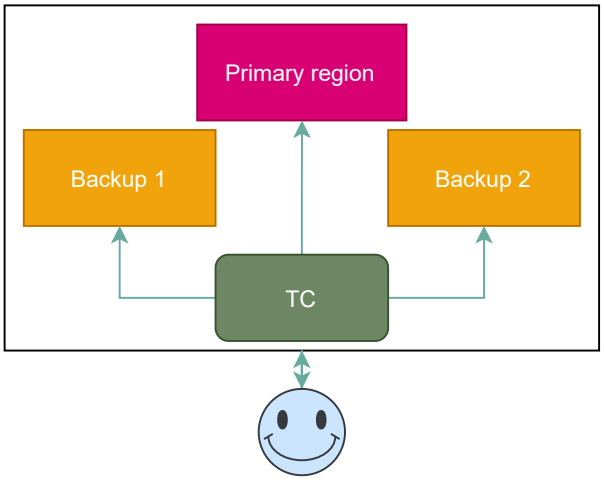
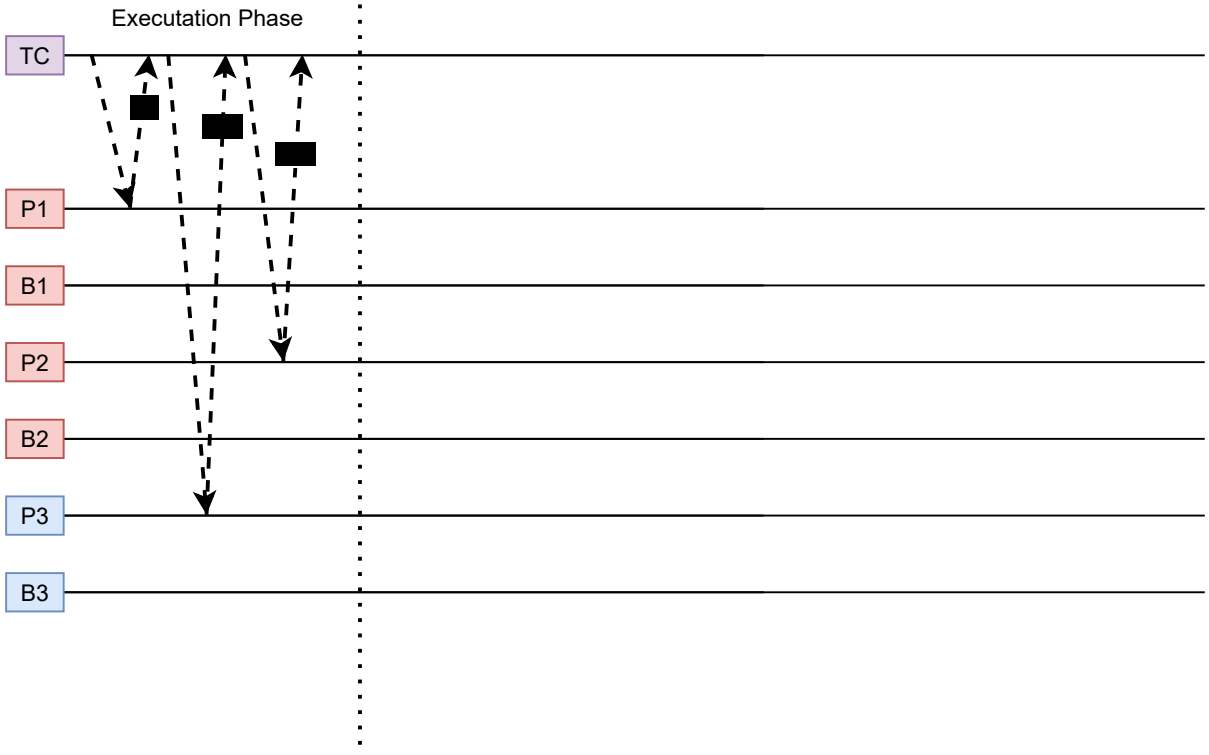




TC reads the objects it needs to from servers including records that it will write using RDMA without locking,

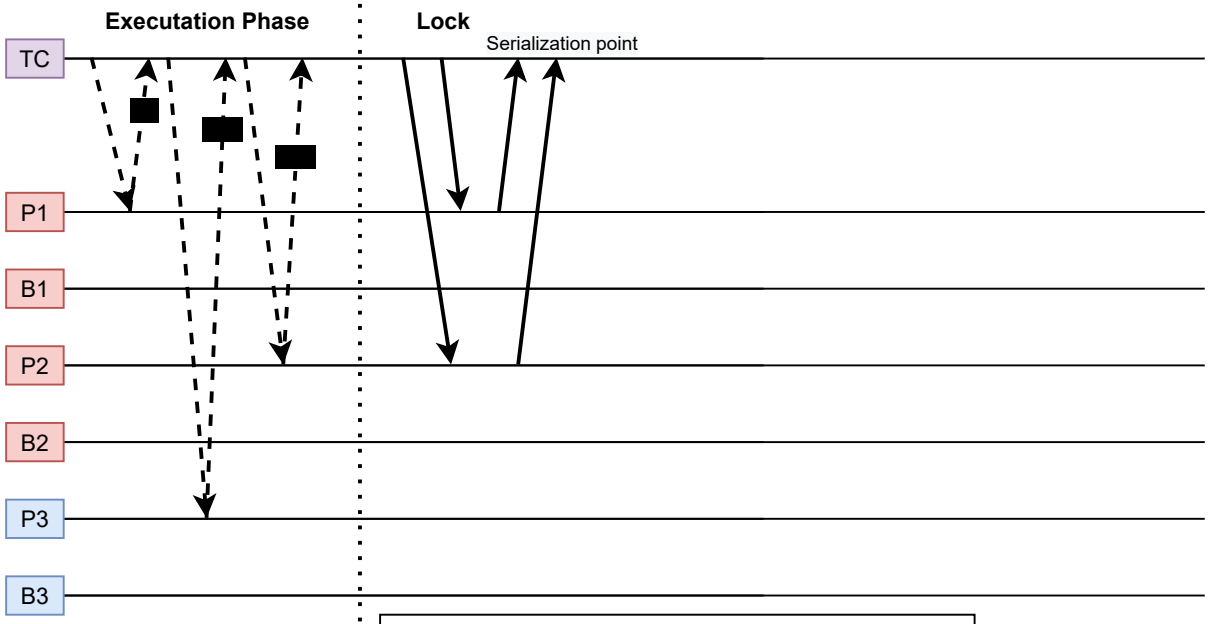
TC remembers the version numbers and buffers writers.



TC sends LOCK record to the log of each primary using RDMA

LOCK record contains oid, version #, new value,

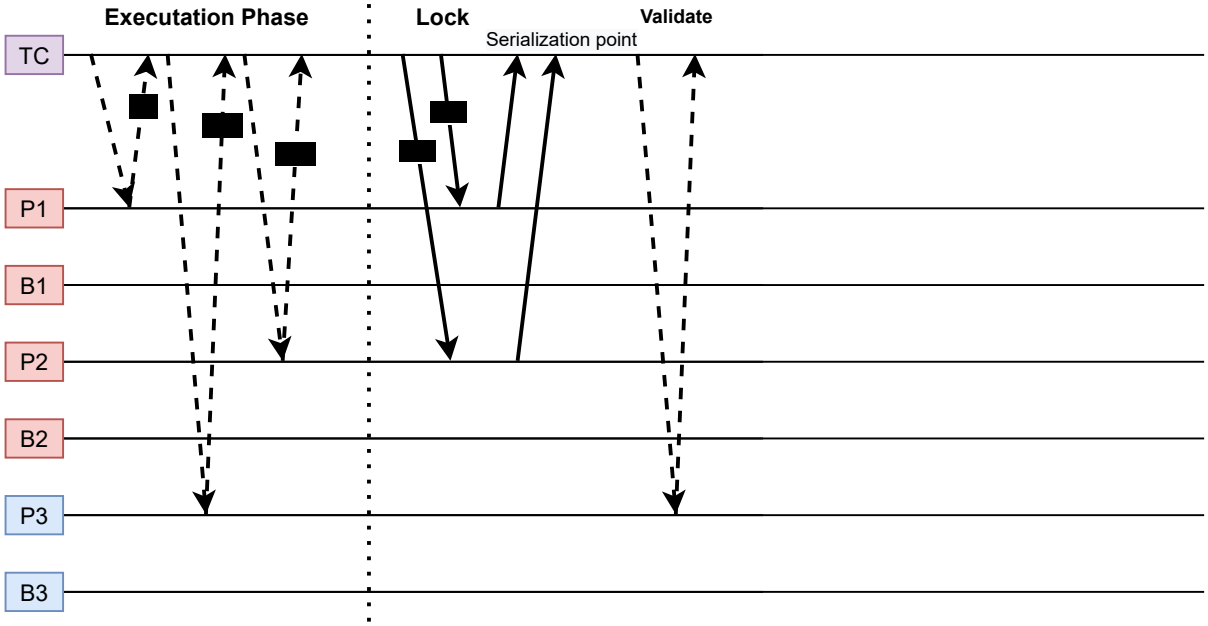
Primaries process records by attempting to lock the objects using compare-and-swap.

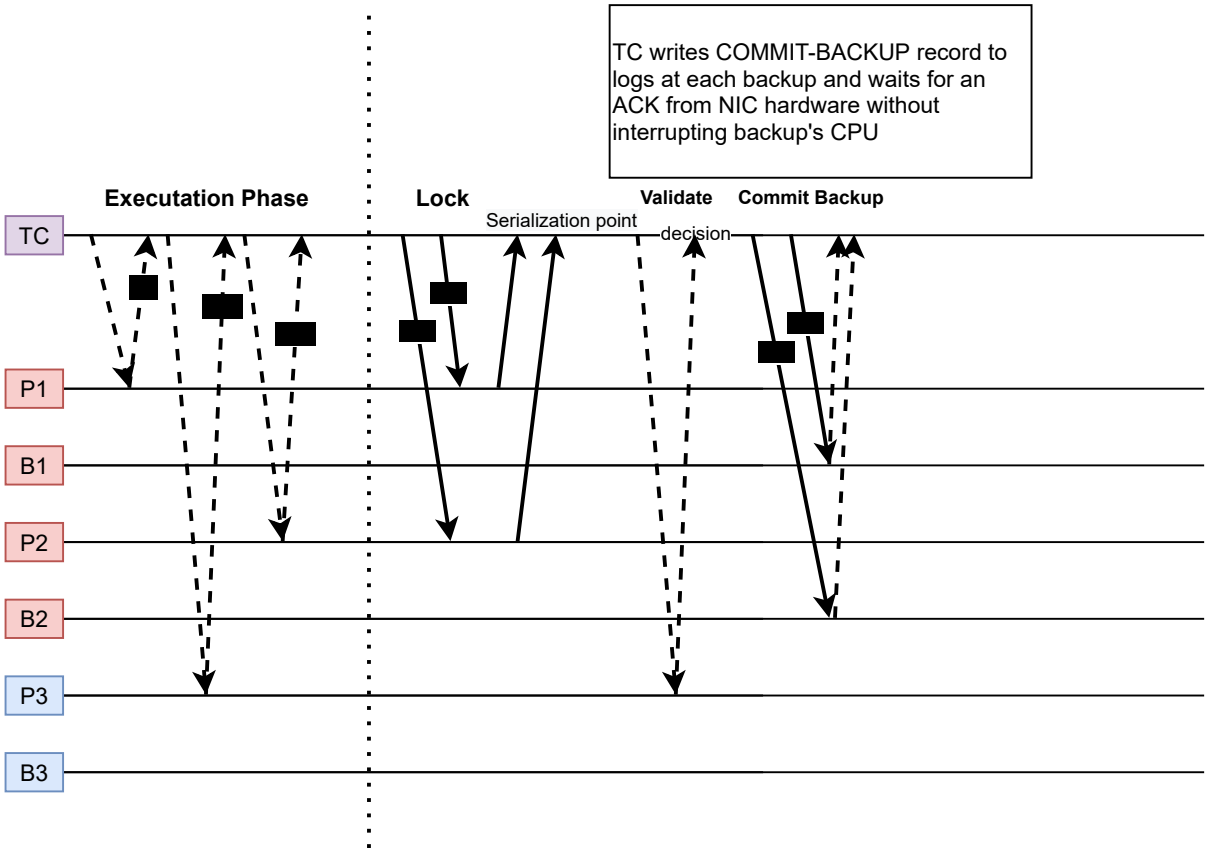


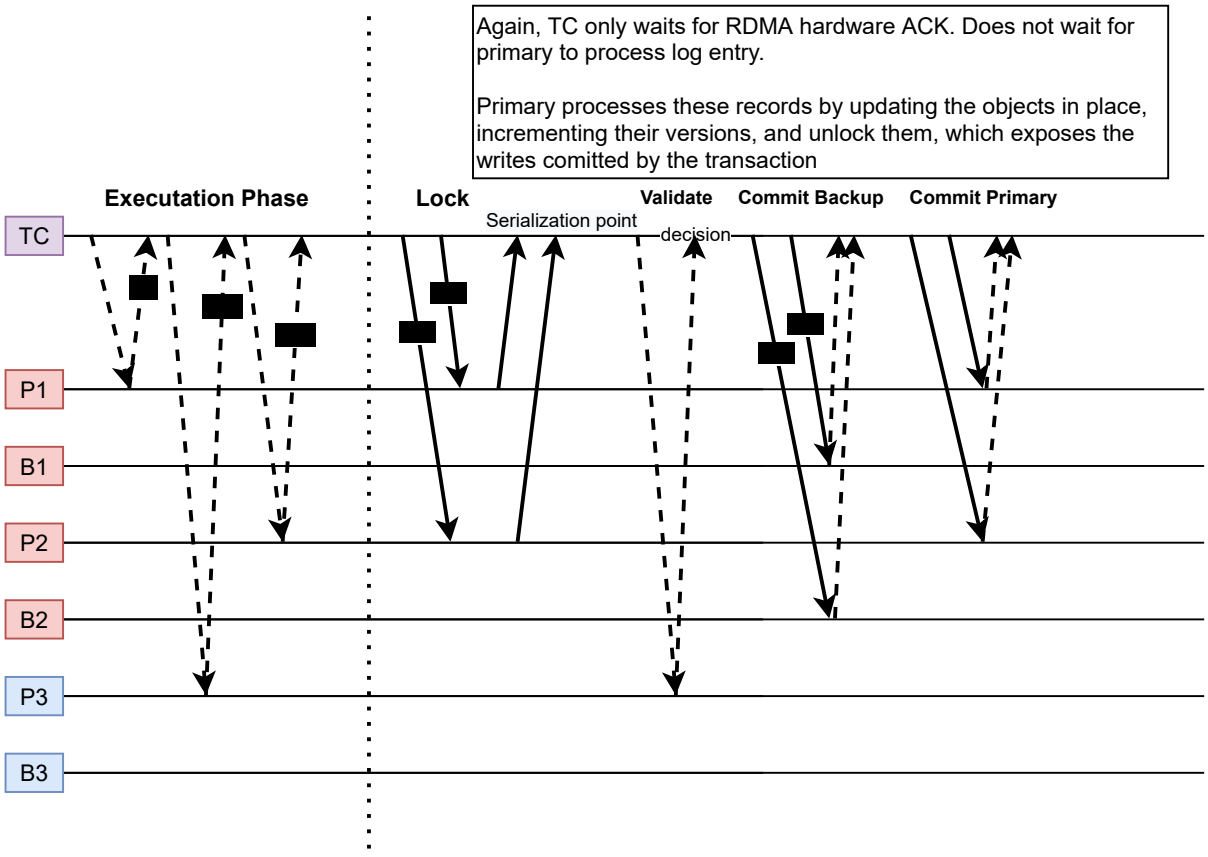
If version not matching, or object already locked, primary replies "no"

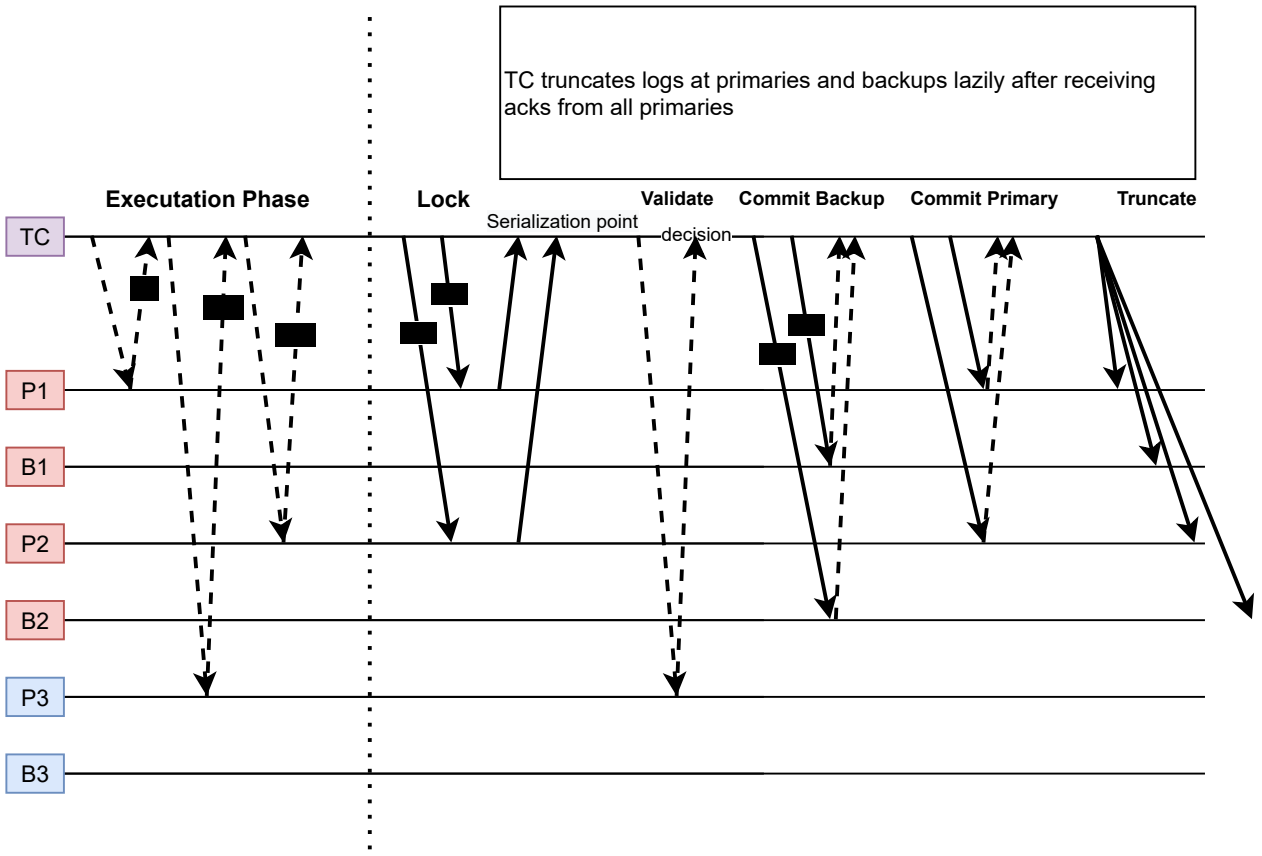
If TC sees "no", abort the transaction and returns "no" to Commit

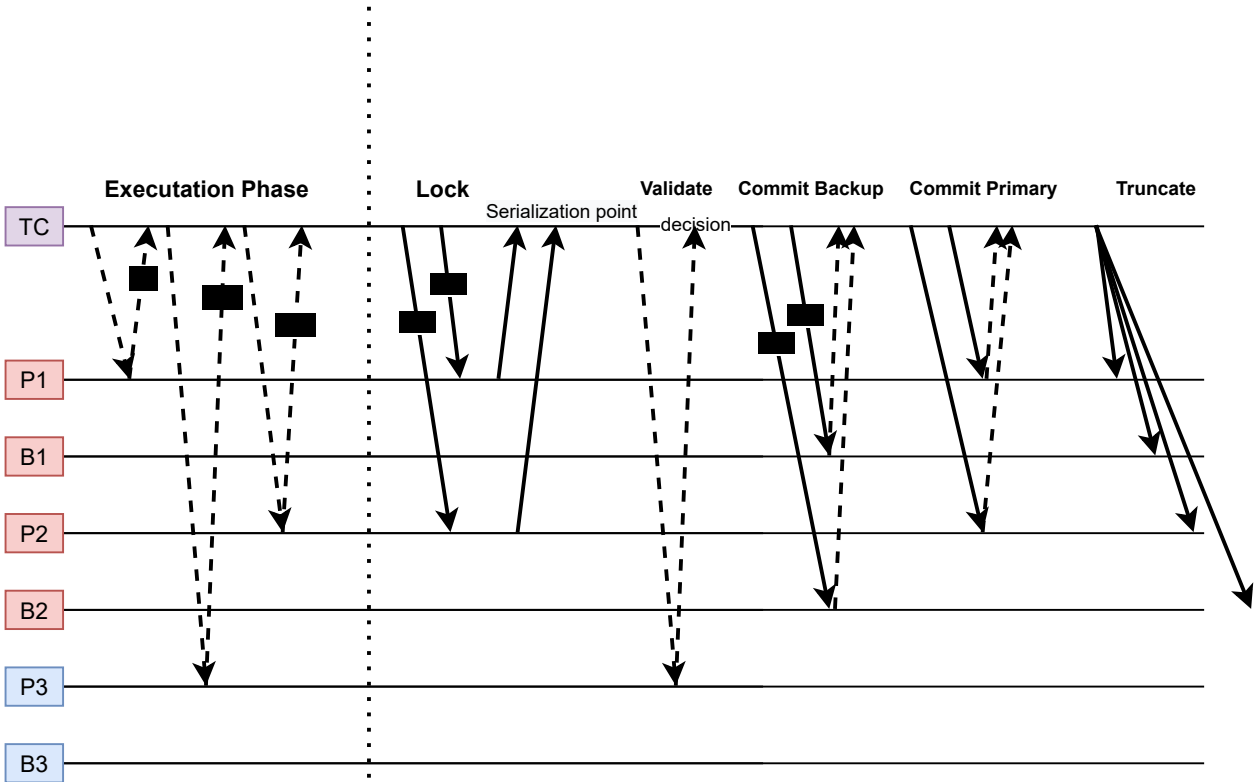
Re-fetch object's version # and lock flag. Does not set the Lock so it's very fast!!

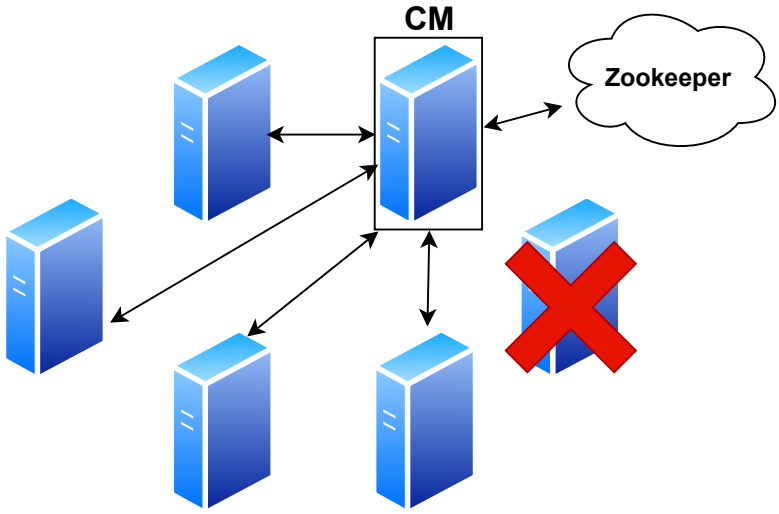



TC writes COMMIT-BACKUP record to logs at each backup and waits for an ACK from NIC hardware without interrupting backup's CPU
Executation Phase
Lock
Validate
Commit Backup
Serialization point
decision
TC
P1
B1
P2
B2
P3
B3


Again, TC only waits for RDMA hardware ACK. Does not wait for primary to process log entry.
Primary processes these records by updating the objects in place, incrementing their versions, and unlock them, which exposes the writes comitted by the transaction
Executation Phase
Lock
Validate
Commit Backup
Commit Primary
Serialization point
decision
TC
P1
B1
P2
B2
P3
B3


TC truncates logs at primaries and backups lazily after receiving acks from all primaries
Executation Phase
Lock
Validate
Commit Backup
Commit Primary
Truncate
Serialization point
decision
TC
P1
B1
P2
B2
P3
B3


Executation Phase
Lock
Validate
Commit Backup
Commit Primary
Truncate
Serialization point
decision
TC
P1
B1
P2
B2
P3
B3


CM
Zookeeper