LEADERLESS REPLICATION

69

69

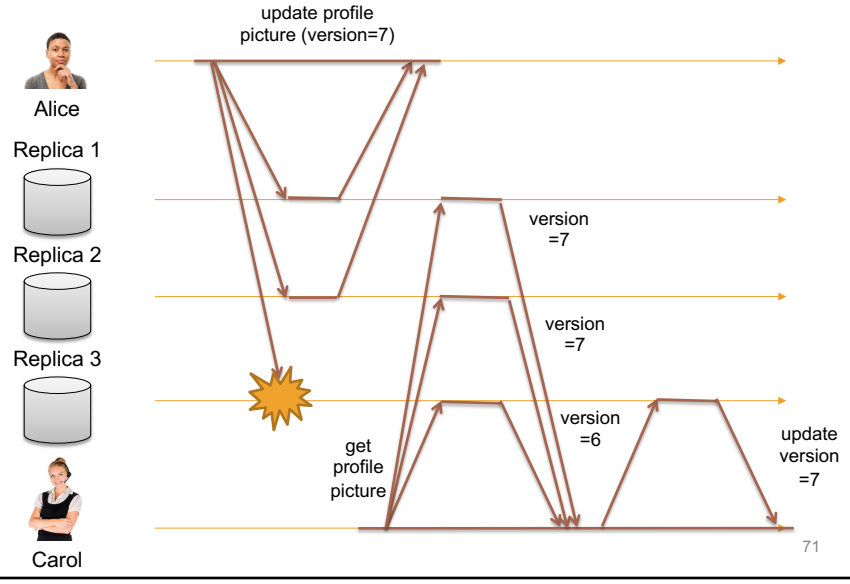
Leaderless Replication

- No leader to enforce order of writes
- Amazon Dynamo
- How to prevent stale reads?

70

70

Leaderless Replication



71

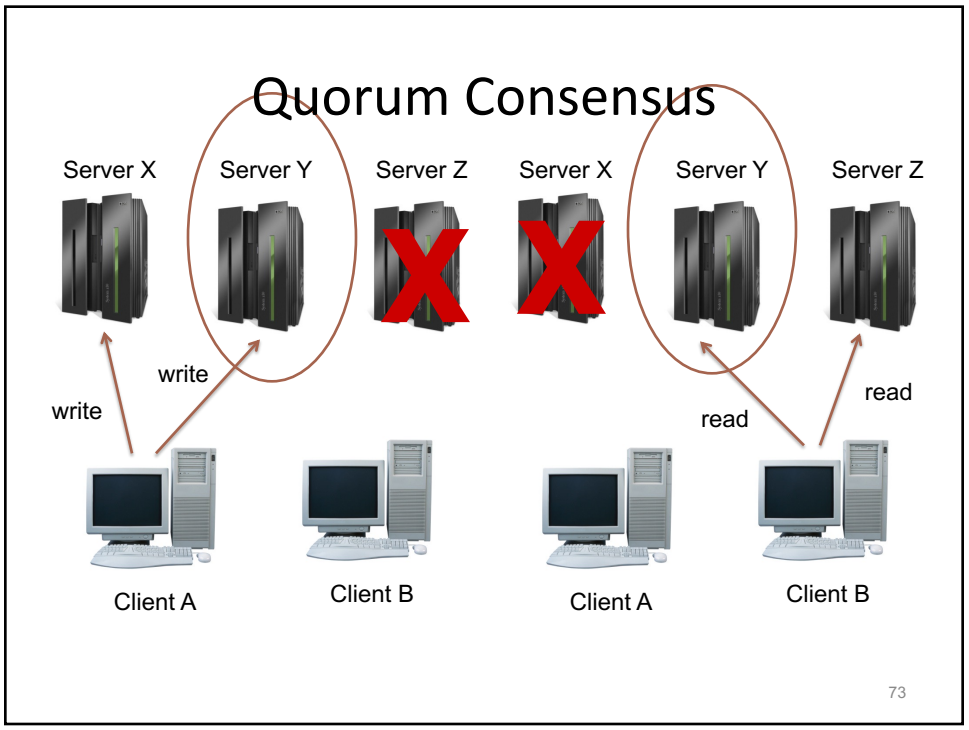
71

Fixing Stale Data

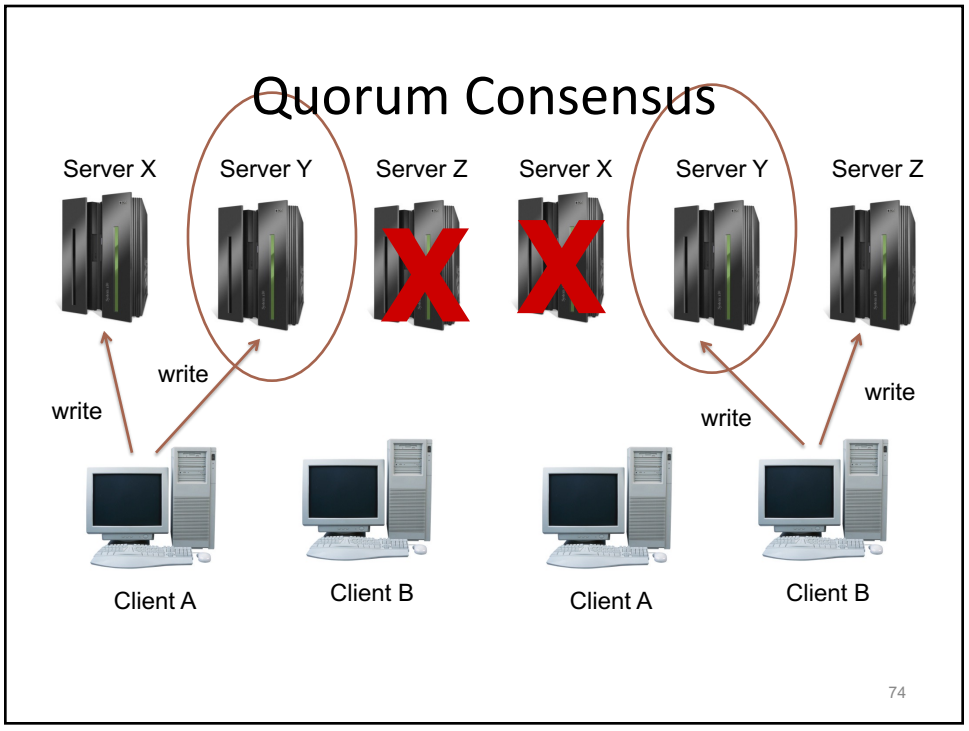
- Read Repair
- Anti-Entropy

72

72


Quorum Consensus
Server X
Server Y
Server Z
Server X
Server Y
Server Z
write
write
read
read
Client A
Client B
Client A
Client B
73

73


Quorum Consensus
Server X
Server Y
Server Z
Server X
Server Y
Server Z
write
write
write
write
Client A
Client B
Client A
Client B
74

74

Quorum Consensus

- Each replicated object has an update and a read quorum
- Rules
  - A quorum read should "intersect" any prior quorum write at ≥ 1 processes
  - A quorum write should also intersect any other quorum write
- So, in a group of size N:
  - $Q_r + Q_w > N$, and
  - $Q_w + Q_w > N$

75

75

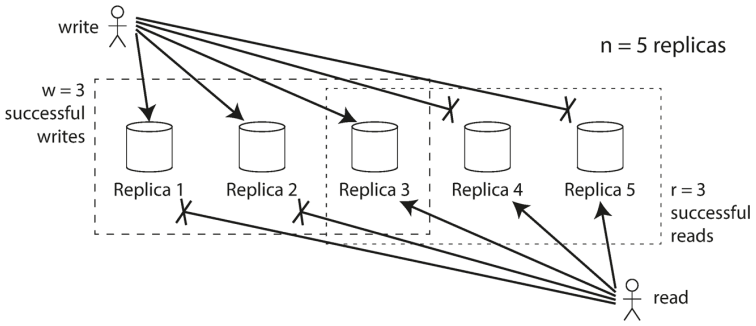
Quorum example

- X is replicated at {a,b,c,d,e}
- Possible values?
  - $Q_w = 1$, $Q_r = 5$ (violates $Q_w+Q_w > 5$)
  - $Q_w = 2$, $Q_r = 4$ (same issue)
  - $Q_w = 3$, $Q_r = 3$
  - $Q_w = 4$, $Q_r = 2$
  - $Q_w = 5$, $Q_r = 1$ (violates availability)
- Probably prefer $Q_w=4$, $Q_r=2$

76

76

Quorum Consensus

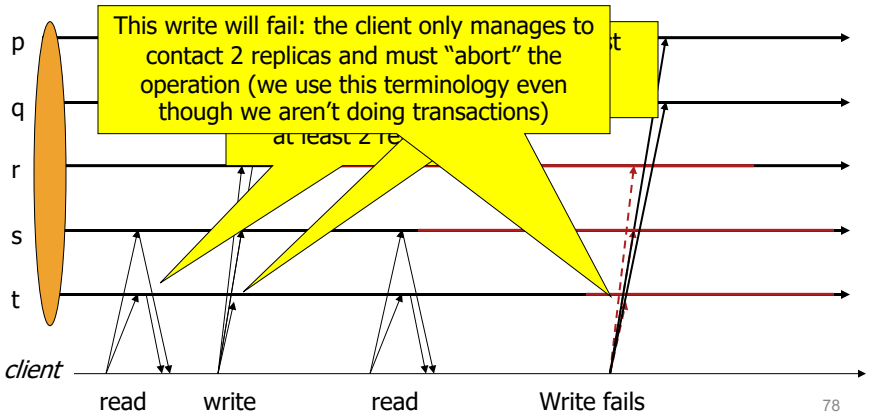


77

77

Static membership example

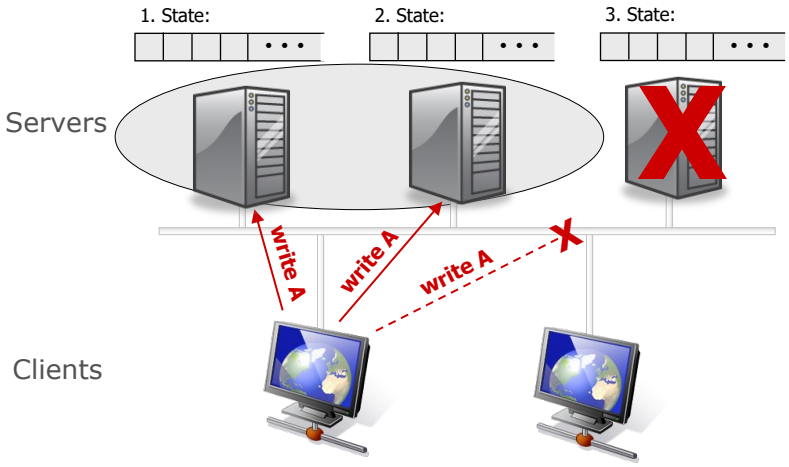
$Q_{read} = 2, Q_{write} = 4$



78

78

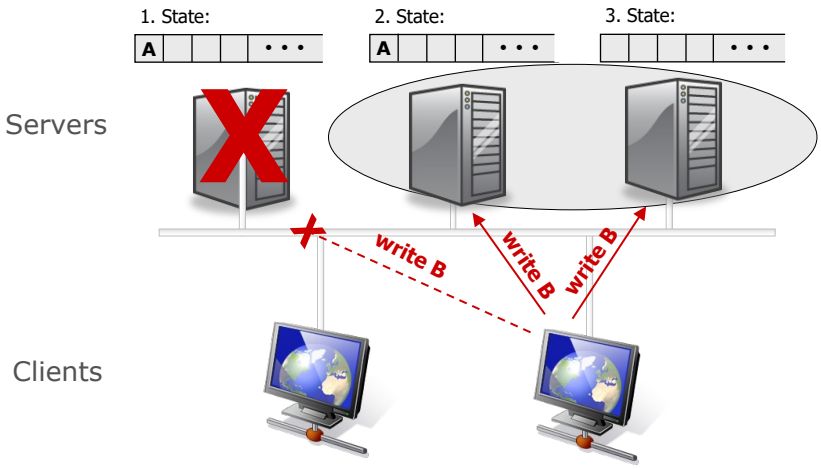
Quora



79

79

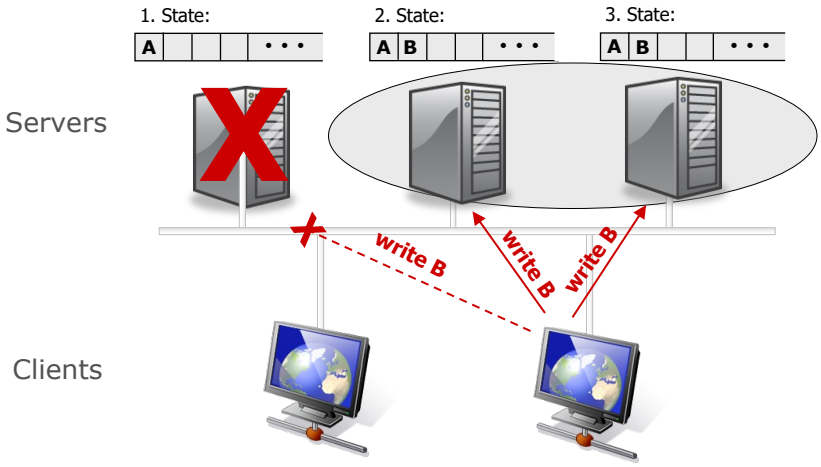
Quora



80

80

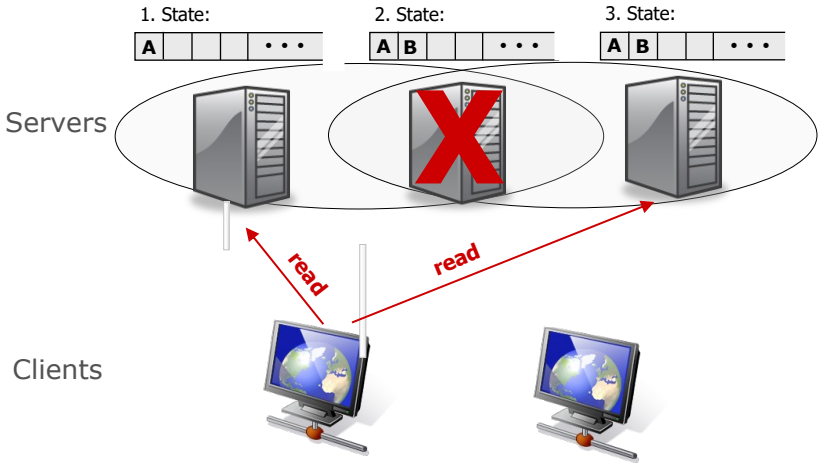
Quora



81

81

Quora



82

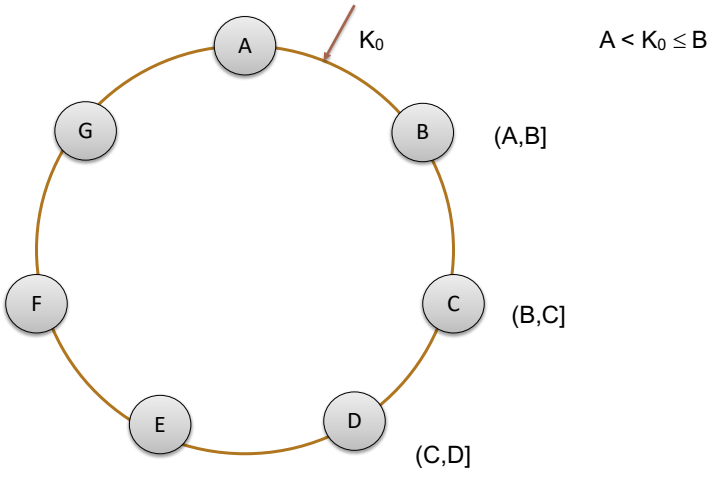
82

Leaderless Replication

- Client contacts available replicas
- Read quorum, write quorum
- Read operations repair missed updates

83

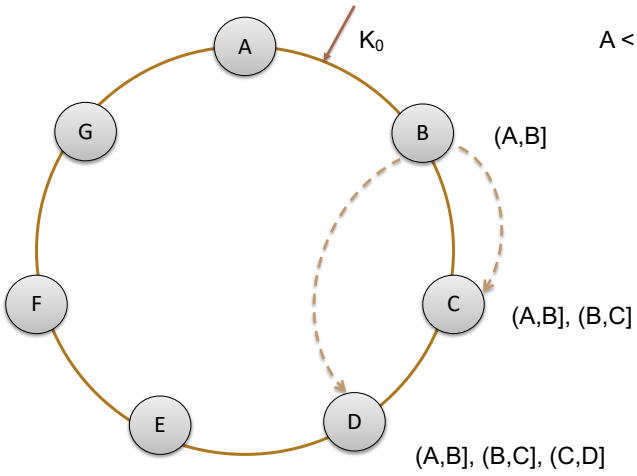
Example: Amazon Dynamo



84

84

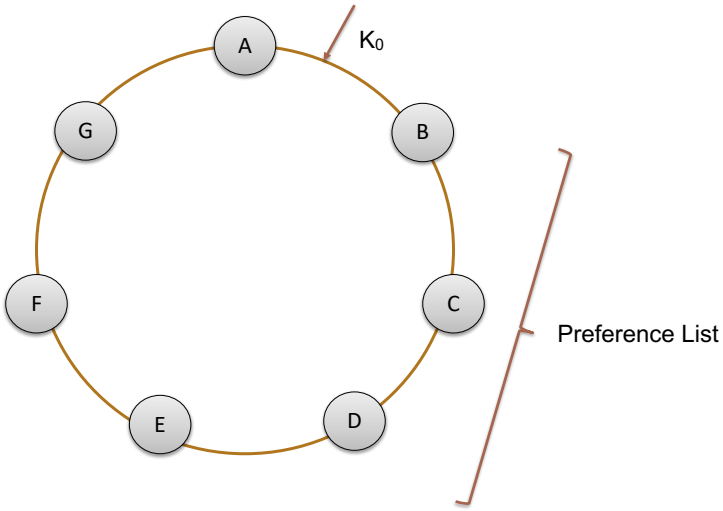
Example: Amazon Dynamo



85

85

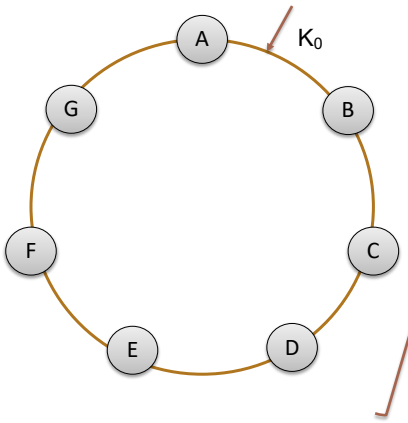
Example: Amazon Dynamo



86

86

Example: Amazon Dynamo

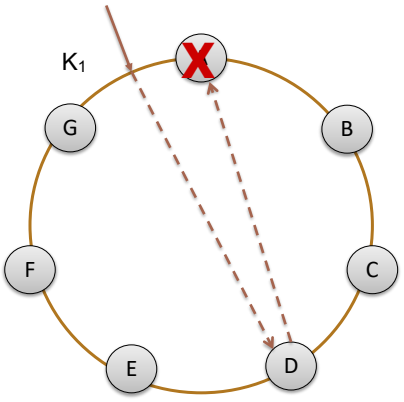


- Consistent hashing
- "Virtual Nodes"
  - Nodes back up other nodes
  - Load balancing
- "Preference list"
  - e.g. B, C, D for key K_0

87

87

Hinted handoff



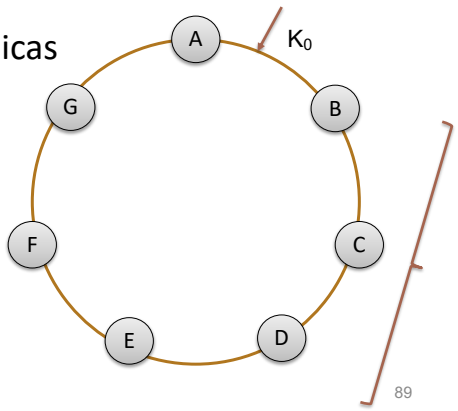
- Assume # replicas for K_1 N = 3
- A down ⇒ send replica to e.g. D
- D is hinted that the replica belongs to A
- D will deliver to A when A is recovered
- "Always writeable"

88

88

Sloppy Quorum

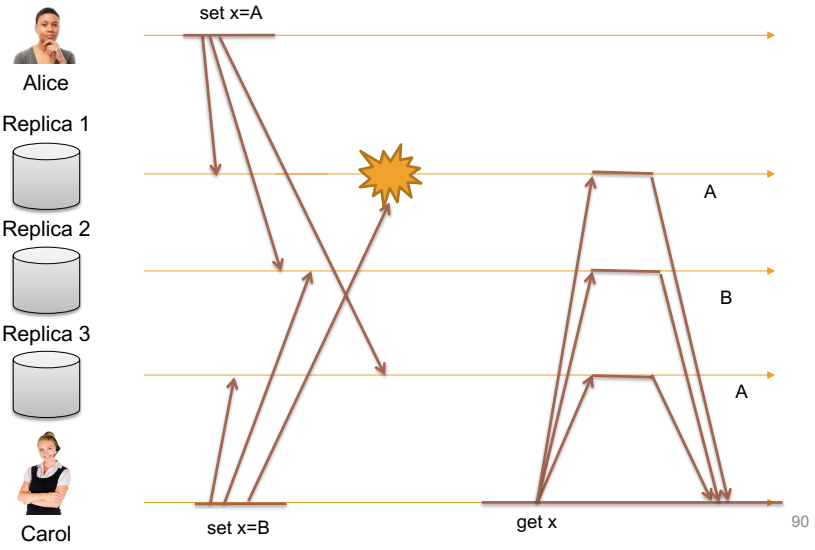
- R/W read and write quora
- R + W > N
  - Latency dictated by slowest of R (or W) replicas
  - …but which N?
  - Pref List (N=3)
    ⇒ consistent
  - Sloppy Quorum: **any** N (hinted handoff)
- R + W < N
  - better latency



89

89

Leaderless Replication: Ordering Writes



90

90

Leaderless Replication: Ordering Writes

- Last Write Wins (LWW)
  - writes may be lost (not durable)
  - Cassandra: keep version history
  - Concurrent writes?
- Detecting conflicts
  - "Happened-before"
  - Version numbers
  - Update must read and merge current values (conflict resolution)

91

Data Versioning

- Write: return before update applied at all replicas
- Read: return many versions of the same object

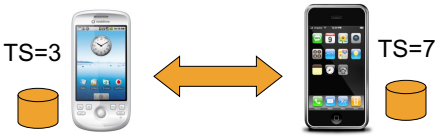
- *Challenge:* object has distinct version sub-histories
- *Solution:* vector clocks for reconciliation

92

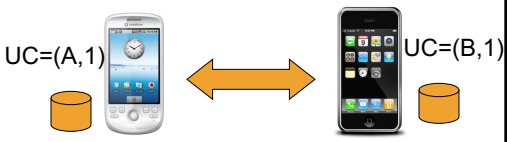
92

Data Versioning

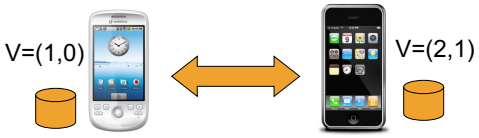
Update timestamps



Update counters
- Concurrent updates
- Resolve with device id



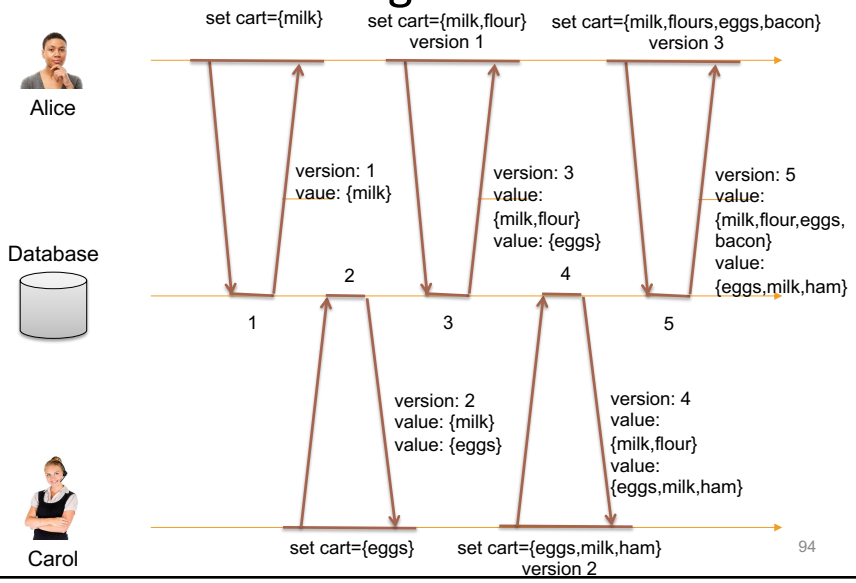
Version vectors
- Like vector clocks



93

93

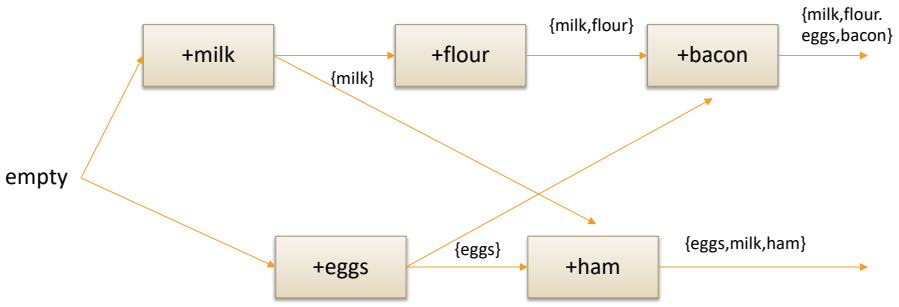
Leaderless Replication: Ordering Writes



94

94

Leaderless Replication: Merging Updates



95

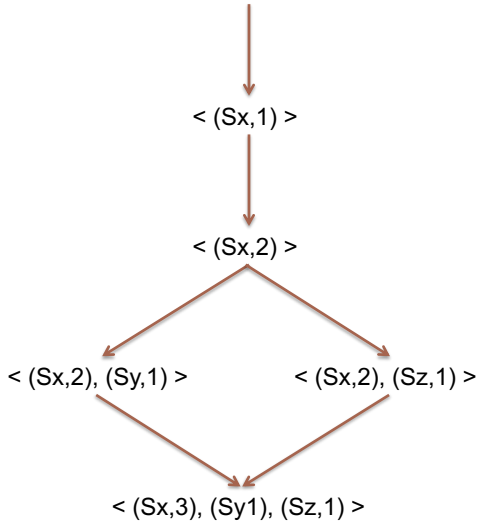
Merging Writes

- Server stores version number with each record
- Client read: All values not overwritten
- Client write (after read): merge together all versions in last read
- Server write: overwrite all earlier versions

96

96

Data Versioning



97

97

Limitations of Quorum Consistency

- Sloppy Quorum (Dynamo)
  - Qw and Qr on different nodes from "home" nodes
- Concurrent writes must be merged
- Concurrent write and read, read returns new or old value
- Failed write not rolled back, may be returned from read
- Recovering node may get stale value from another node
- Problems with linearizability
- Measuring staleness: Ordering of writes?

98