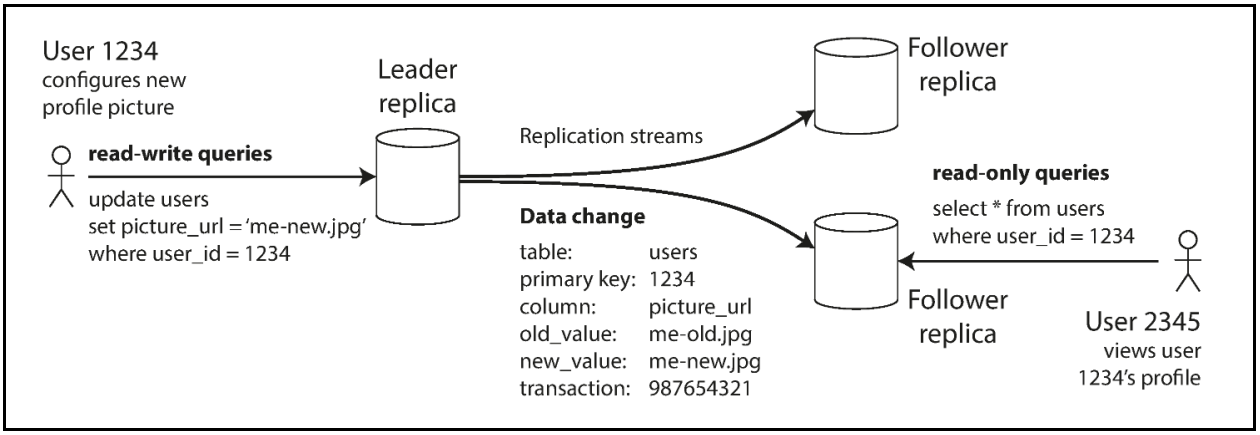
Designing Data-Intensive Applications (Kleppmann): Chapter 5, "Replication." **Figures**.



5-1. Leader-based replication

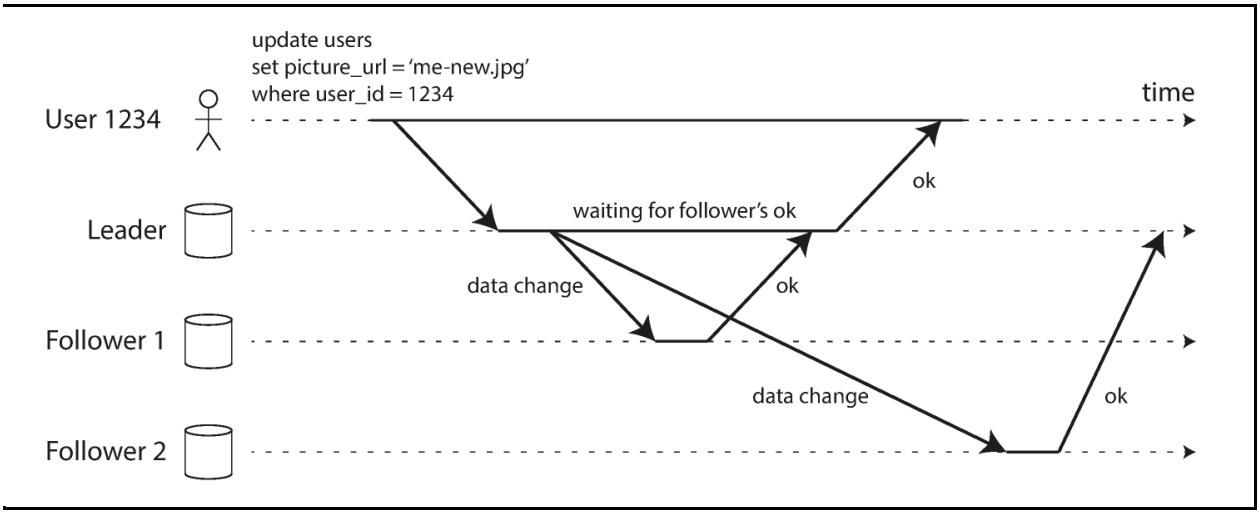


Figure 5-2. Leader-based replication with one synchronous follower and one asynchronous follower.

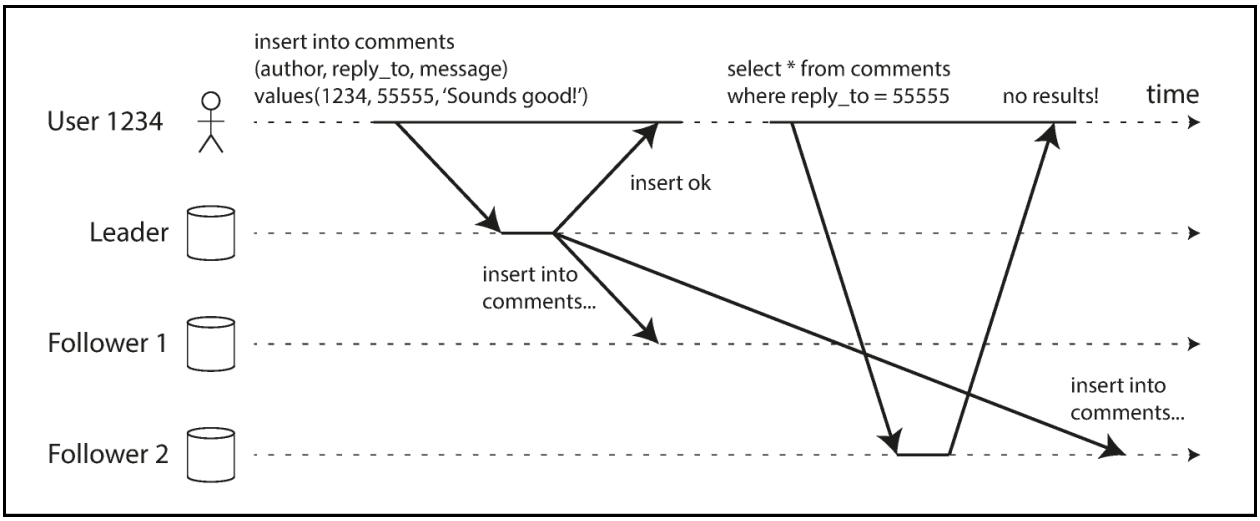


Figure 5-3. A user makes a write, followed by a read from a stale replica. To prevent this anomaly, we need read-after-write consistency.

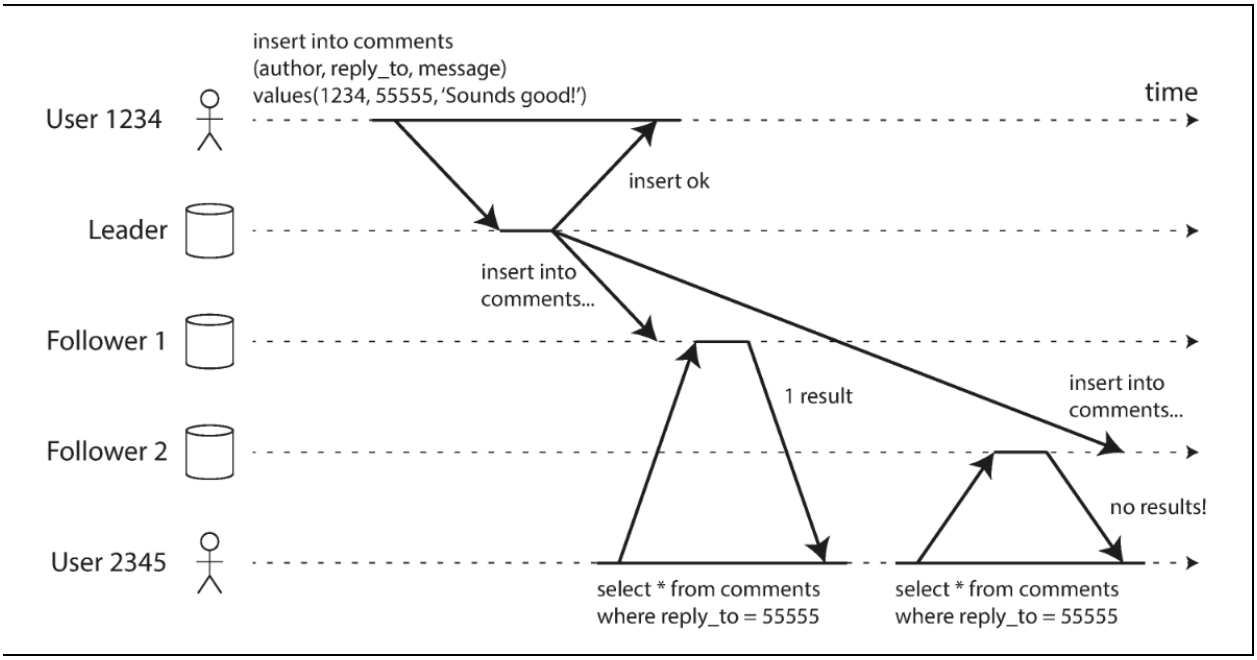


Figure 5-4. A user first reads from a fresh replica, then from a stale replica. Time appears to go backward. To prevent this anomaly, we need monotonic reads.

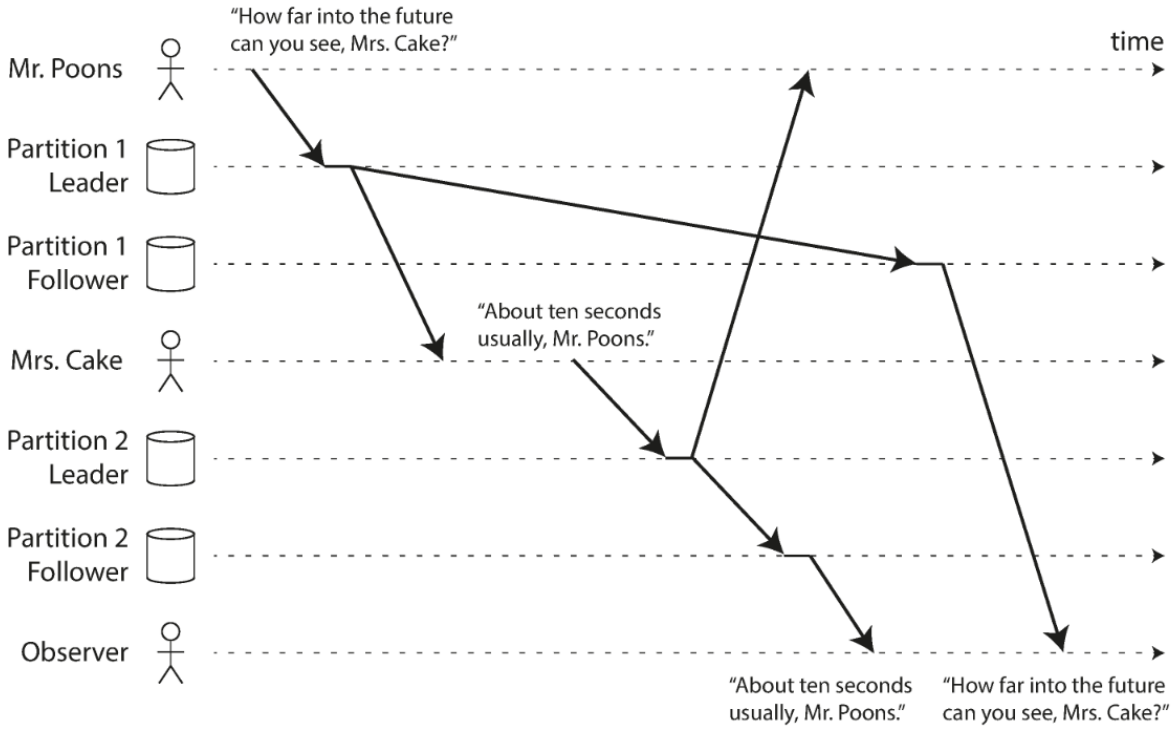


Figure 5-5. If some partitions are replicated slower than others, an observer may see the answer before they see the question.

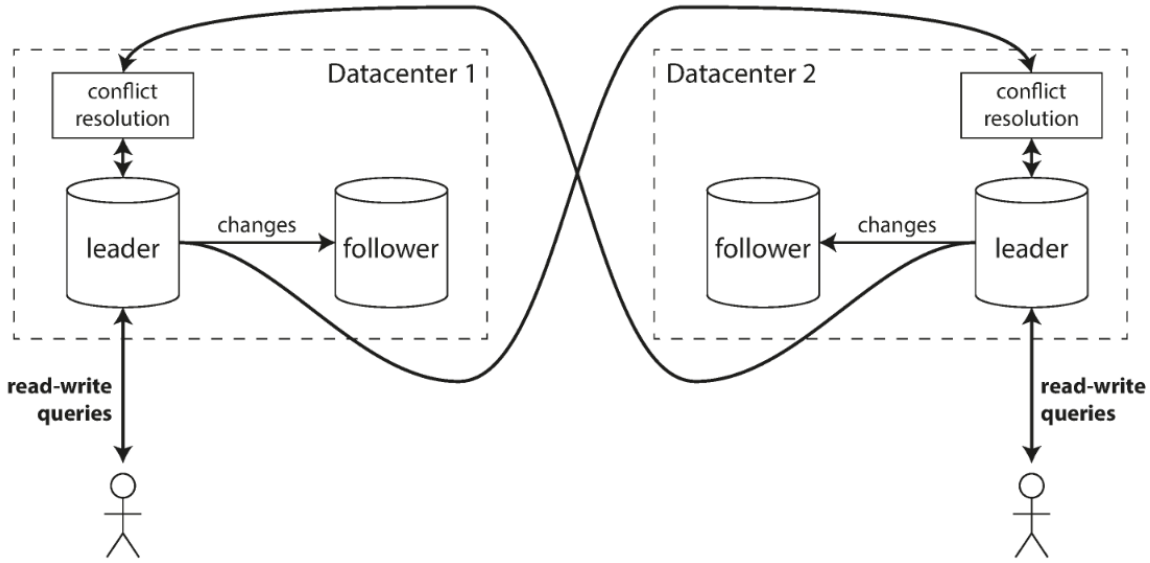


Figure 5-6. Multi-leader replication across multiple datacenters

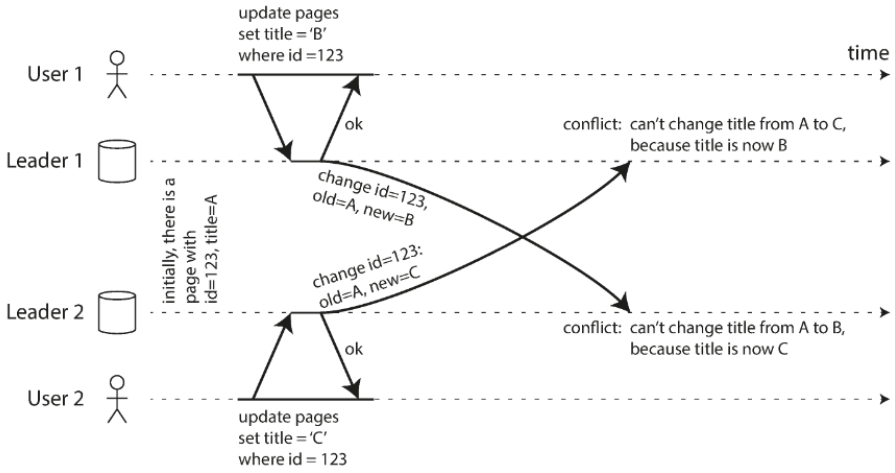


Figure 5-7. A write conflict caused by two leaders concurrently updating the same record.

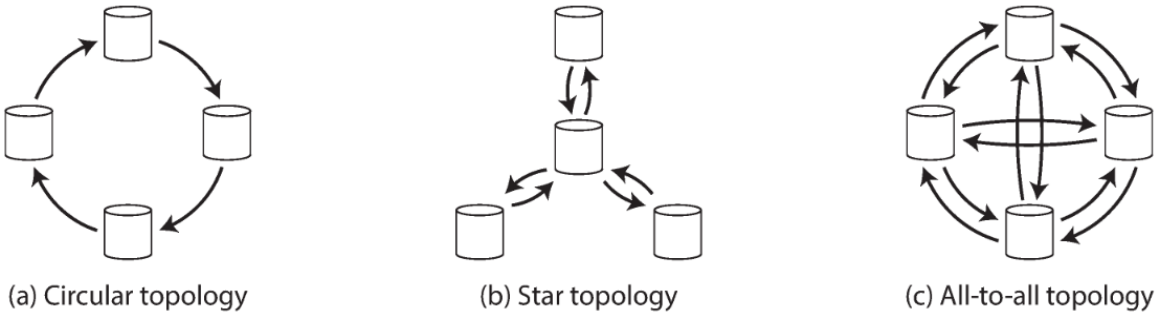


Figure 5-8. Three example topologies in which multi-leader replication can be set up..

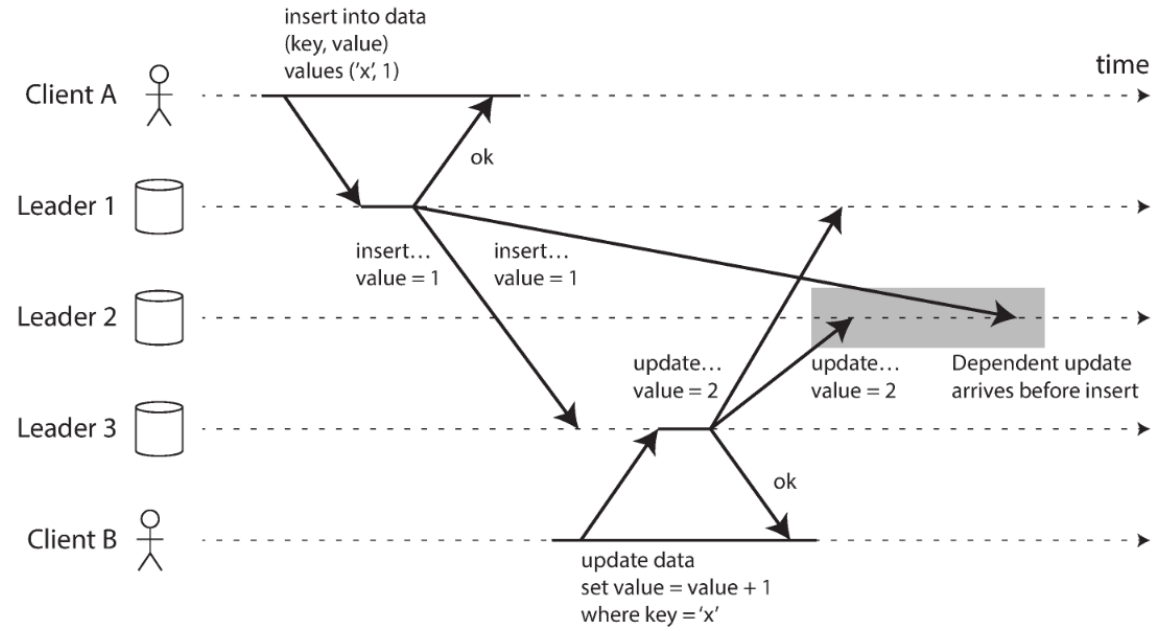


Figure 5-9. With multi-leader replication, writes may arrive in the wrong order at some replicas.

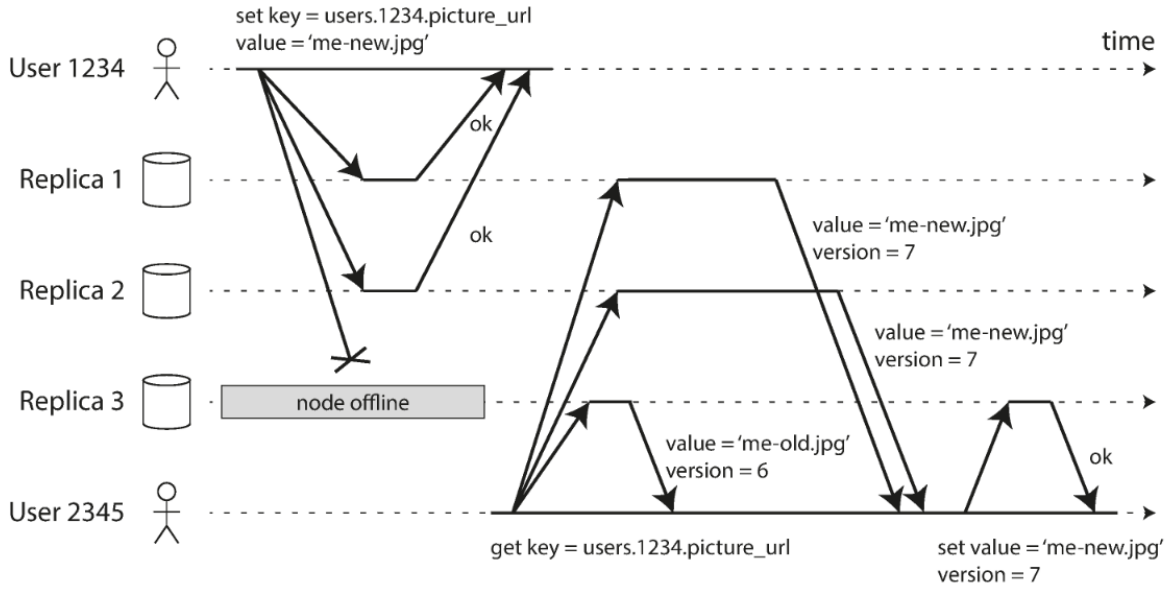


Figure 5-10. A quorum write, quorum read, and read repair after a node outage

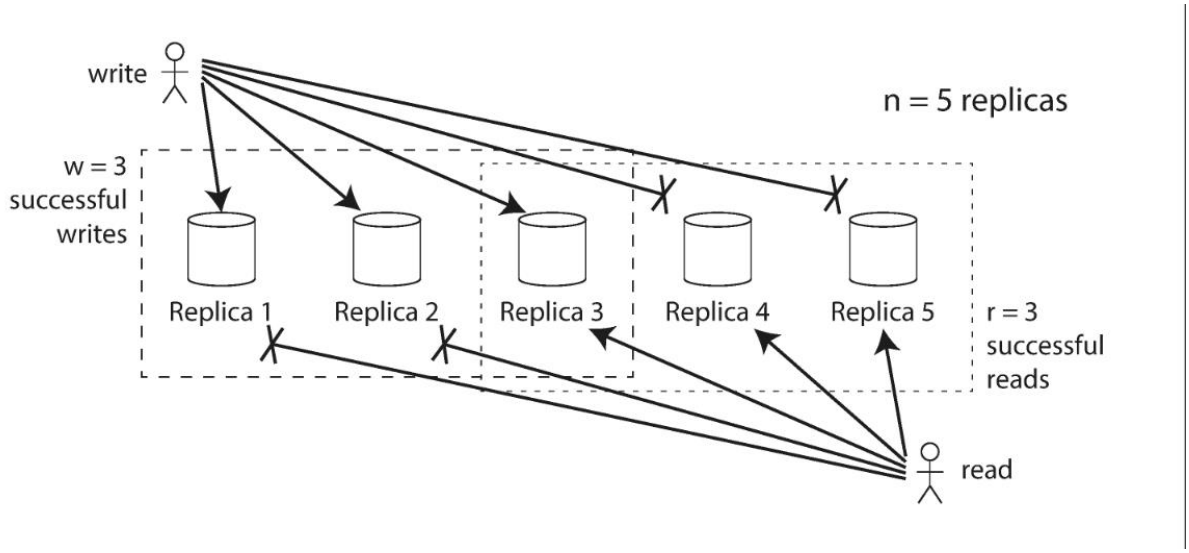


Figure 5-11. If w + r > n, at least one of the r replicas you read from must have seen the most recent successful write.

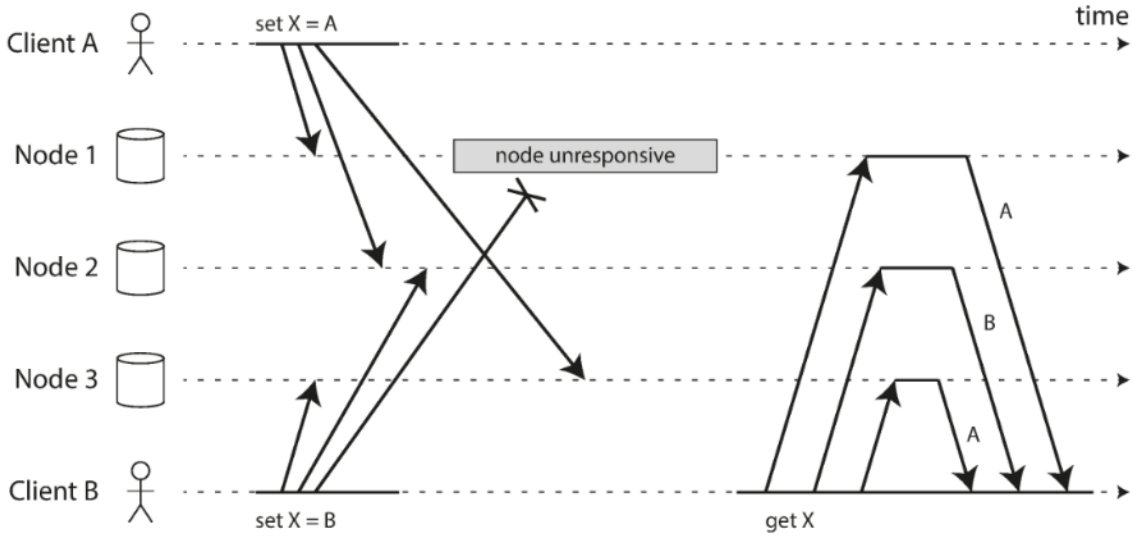


Figure 5-12. Concurrent writes in a Dynamo-style datastore: there is no well-defined ordering

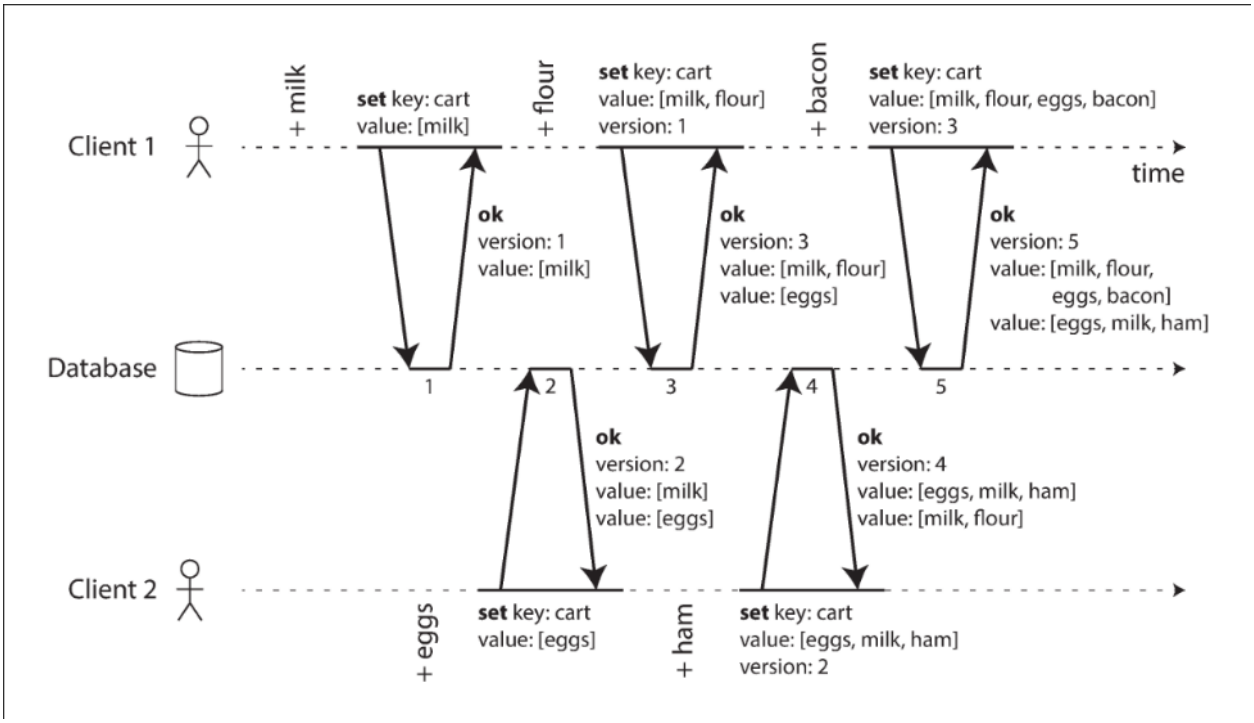


Figure 5-13. Capturing causal dependencies between two clients concurrently editing a shopping cart

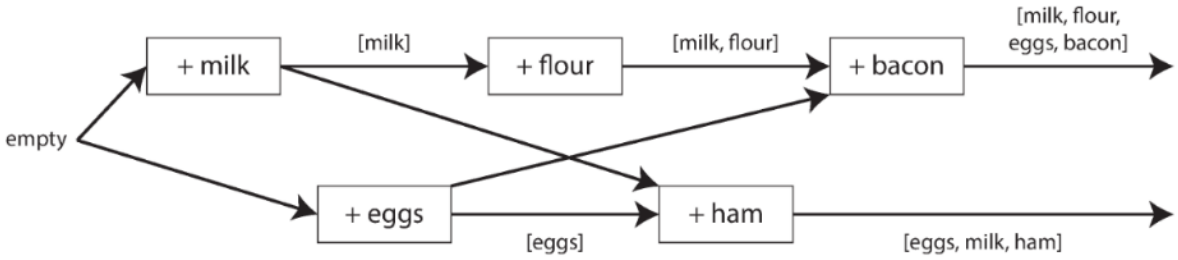


Figure 5-14. Graph of causal dependencies in Figure **5-13**.