NTNU
Kunnskap for en bedre verden

TDT4225 Very Large, Distributed Data Volumes

Exercise 4

Deadline 5 Nov 2021 16.00 (4pm)

1. **Kleppmann Chap 5**

a) When should you use multi-leader replication, and why should you use it in these cases? When is leader-based replication better to use?

b) Why should you use log shipping as a replication means instead of replicating the SQL statements?

c) How could a database management system support *read your writes consistency* when there are multiple replicas present of data?

2. **Kleppmann Chap 6**

a) Why should we support sharding / partitioning?

b) What is the best way of supporting re-partitioning? And why is this the best way? (According to Kleppmann).

c) Explain when you should use local indexing, and when you should use global indexing?

3. **Kleppmann Chap 7**

a) **Read committed vs snapshot isolation**. We want to compare *read committed* with *snapshot isolation*. We assume the traditional way of implementing read committed, where write locks are held to the end of the transaction, while read locks are set and released when doing the read itself. Show how the following schedule is executed using these two approaches:

r1(A); w2(A); w2(B); r1(B);c1;c2;

b) Also show how this is executed using serializable with 2PL (two-phase locking).

c) Explain by an example write skew, and show how SSI (serializable snapshot isolation) may solve this problem.

4. **Kleppmann Chap 8**

a) If you send a message in a network and you do not get a reply, what could have happened? List some alternatives.

b) Explain why and how using clocks for *last write wins* could be dangerous.

c) Given the example from the text book on “process pauses”, what is the problem with this solution to obtaining lock leases?

```
while (true) {
    request = getIncomingRequest();

    // Ensure that the lease always has at least 10 seconds remaining
    if (lease.expiryTimeMillis - System.currentTimeMillis() < 10000) {
        lease = lease.renew();
    }

    if (lease.isValid()) {
        process(request);
    }
}
```

5. Kleppmann Chap 9

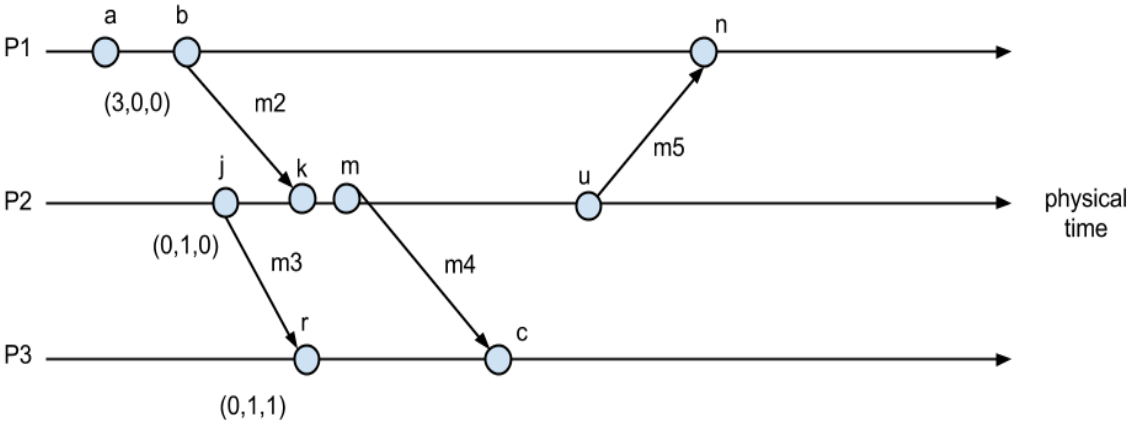
a) Explain the connection between ordering, linearizability and consensus.

b) Are there any distributed data systems which are usable even if they are not linearizable? Explain your answer.

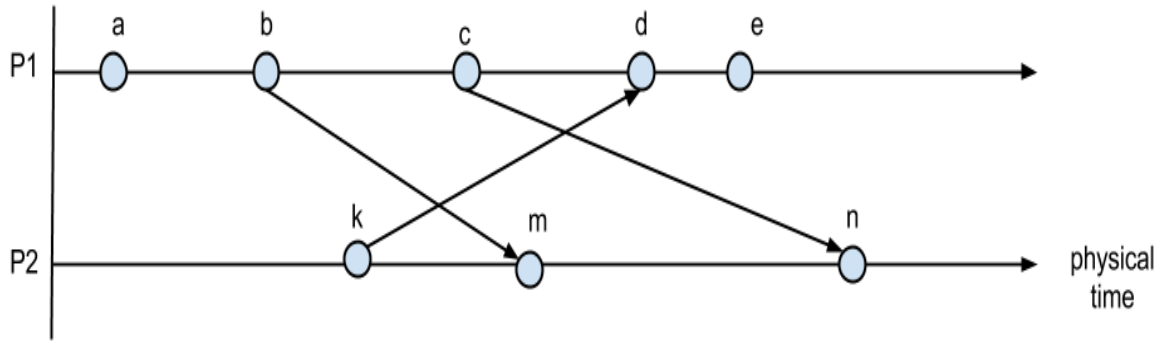
6. Coulouris Chap 14

a) Given two events e and f. Assume that the logical (Lamport) clock values L are such that L (e) < L (f). Can we then deduce that e "happened before" f? Why? What happens if one uses vector clocks instead? Explain.

b) The figure below shows three processes and several events. Vector clock values are given for some of the events. Give the vector clock values for the remaining events.



c) The figure below shows the events that occur at two processes P1 and P2. The arrows mean sending of messages. Show the alternative consistent states the system can have had. Start from state S_{00}. (S_{xy} where x is p1's state and y is p2's state)



7. **RAFT**
   RAFT has a concept where the log is replicated to all participants. How does RAFT ensure that the log is equal on all nodes in case of a crash and a new leader?

8. Dynamo

a) Explain the following concepts/techniques used in Dynamo:
   - consistent hashing
   - vector clocks
   - sloppy quorum and hinted handoff
   - merkle trees
   - gossip-based membership protocol