

CPS 512/590 first midterm exam, 10/6/2015

Your name please: _____

/40
/50
/50
/60
/200

Part 1. Managing replicated server groups

These questions pertain to managing server groups with replication, as in e.g., Chubby, Dynamo, and the classical Replicated State Machine (RSM) model introduced in the 1978 Lamport paper *Time, Clocks, and the Ordering of Events*.... Answer each question with a few phrases or maybe a sentence or two. [40 points]

- a) *Consistent Hashing* algorithms select server replicas to handle each object (e.g., each key/value pair). They use hash functions to assign each object and each server a value within a range referred to as a “ring” or “unit circle”. In what sense is the range a “ring” or “circle”, and why is it useful/necessary to view it this way?

- b) In practice, it is important for consistent hashing systems to assign each server multiple values for multiple points on the ring. Why?

- c) In the classical RSM model, it is important for all replicas to apply all operations in the same order. How do Chubby, Dynamo, and the Lamport RSM system achieve this safety property during failure-free operation?

- d) The Lamport RSM system is fully asynchronous and does not handle failures of any kind. Even so, it requires each peer to acknowledge every operation back to its sender. Why is this necessary, i.e., what could go wrong if acknowledgements are not received?

Part 2. Leases

These questions pertain to leases and leased locks, as used in Chubby, Lab #2, and distributed file systems like NFSv4. Answer each question with a few phrases or maybe a sentence or two. [50 points]

- a) In Chubby (and in Lab #2), applications acquire and release locks with explicit **acquire()** and **release()** primitives. Leases create the possibility that an application thread holding a lock (the lock holder) loses the lock before releasing it. Under what conditions can this occur?

- b) In this circumstance (a), how does the lock client determine that the lease has been lost? How does the application determine that the lock has been lost? Be sure to answer for both Chubby and your Lab#2 code.

- c) Chubby maintains a single lease for each client's session with Chubby, rather than a lease for each individual lock that a client might hold. What advantages/benefits does this choice offer for Chubby and its clients?

- d) Network file systems including NFSv4 use leased locks to maintain consistency of file caches on the clients. In these systems, what events cause a client to request to acquire a lock from the server?

- e) In network file systems the server can recall (callback) a lock from a client, to force the client to release the lock for use by another client. What actions must a client take on its file cache before releasing the lock?

Part 3. Handling writes

These questions pertain to committing/completing writes in distributed storage systems, including Chubby and Dynamo. Answer each question with a few phrases or maybe a sentence or two. [50 points]

- a) Classical quorum replication systems like Chubby can commit a write only after agreement from at least a majority of servers. Why is it necessary for a majority of servers to agree to the write?

- b) Dynamo allows writes on a given key to commit after agreement from less than a majority of servers (i.e., replicas for the key). What advantages/benefits does this choice offer for Dynamo and its clients?

- c) Given that Chubby and Dynamo can complete a write after agreement from a subset of replicas, how do the other replicas learn of the write under normal failure-free operation?

- d) What happens in these systems if a read operation retrieves a value from a server that has not yet learned of a recently completed write? Summarize the behavior for Chubby and Dynamo in one sentence each.

- e) Classical ACID transaction systems differ from Dynamo and Chubby in that they require agreement from *every* server involved in a transaction in order to commit (e.g., as in the two-phase commit protocol). Why?

