Name: AndrewID:

15-640/440 Homework #2
Fall 2013 (Kesden)

Replication and Quorums

1. Consider replication to multiple servers based upon a write-one, read-all static quorum with version numbering. What steps must be taken upon a write to ensure the correctness of the version number?

2. Consider Coda Version Vectors (CVVs). Give an example of two concurrent CVVs. What is indicated by concurrent CVVs? Please describe one situation in which this might occur.

3. Consider a situation with 5 replica servers employing a write-2/read-4 policy and version numbering. Please design and describe a locking scheme that ensures that version numbers remain correct, even in light of concurrent writes. You do not need to consider failure. But, you do need to describe all necessary communication and state, such as communication between or among servers and/or participants.

4. Please consider the special case of a mobile client using a read-one/write-all quorum. Please design an efficient locking protocol to ensure concurrency control. But, please ensure that your protocol permits the lock to be acquired at one replica and released by another.

5. Coda Version Vectors (CVVs) are a form of logical time stamp, specifically a vector time stamp, which are used to aid in the management of replication. What could cause two CVVs be *concurrent* (and non-identical)? Why? Illustrate your answer with an example involving 2 servers and 2 clients.

Agreement, and the Impossibility thereof

6. Consider content servers located at three distant points across the globe. Can we devise a system to ensure that, at any point in time, *exactly* one of them is visible to each client? Please assume that no more than two are unavailable at any point in time, and that it is possible for more than one to be available to provide server at any point in time. Why or why not? *Hint: Consider the network, all its pieces, and all its failure modes.*

7. Assume that we have two magical networks. The first, a *Worm Hole*, truly has zero latency. But, unfortunately, each time a message is sent, it is lost with some probability *p*. The second, *Perfect Fiber*, has a latency directly attributable to the speed of light, but never loses or corrupts a message. Reliable protocols, such as TCP, are available for use on either network. Which network has a faster *guaranteed* delivery time? Why?

Recovery: Checkpointing and Logging

8. Consider checkpointing as means of preparing for failure in a large scale distributed system. What is the most significant challenge that arises when attempting to recover the state of a distributed computation from checkpoints or log entries recorded only at individual participants? Why is preventing this problem, by its nature, very challenging?

9. Consider logging as a means of preparing for failure in a large scale distributed system. Why do recovery facilities rarely rely upon logging, without any checkpointing, for recovery?

10. Most of the time, for the purpose of recovery, logging of messages is performed at the recipient. Should it become necessary, this enables the host to replay the logged messages to recover its state. Imagine for a moment that the only log of messages exists only the senders, perhaps because the recipient is a thin client without significant storage. Can the thin client recovery its state simply by asking the senders to replay the messages they sent? If so, what property of the replay ensures that the recipient will be restored to the pre-crash state? If not, what is the problem and how can it be addressed?