15-440 Homework #2 (Spring 2010)
Due: Thursday, April 8th, 11:59 pm

Turnin Instructions: submit your homework to

```
/afs/andrew/course/15/440-sp10/handin/hw3/username/
```

You must create the folder with your username. If you mess up and need to submit again, create another folder, bobjones.1 or bobjones.2. Kesden's script will kill everything but the latest one and remember the dot between the username and the number. Also submit in PDF format!

---

1. In class we discussed two atomic commit protocols, 2PC and 3PC, three if you count the protocol for total order multicast. We said that each of these protocols had a “window of vulnerability", a.k.a. a “window of blocking”. What was meant by this?

2. If possible, please define an atomic commit protocol without this characteristic. If this is impossible, instead, please offer an intuitive explanation of why this is impossible.

3. Assume that an Applicaton Service Provider (ASP) has a redundant farm of servers distributed across several data centers. Please design a protocol that, absent the failure of each and every one of the servers, ensures that exactly one of the servers is world visible. If you believe this to be impossible, please explain the reason.

4. Consider the quorum-based protocols that we discussed in class. Each of these protocols requires some type of read-write locking to ensure that the client is playing with the same version of the object at each of the participating replicas. Without this, two writes could interleave or a read and a write could interleave.

   Please describe an efficient locking protocol to address these concerns. Assume that the client can interact with all servers, except in the event of failure.

5. Reconsider your answer to question #4. This time, 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 at another.

6. What are the advantages of using ECC over EDC?

7. What property of an encoding makes possible the detection of erroneous transmissions and possibly the correction of the errors?

8. Consider a file system such as HDFS. In particular, consider a file system that manages truly huge files by scattering multiple copies of their constituent blocks across a few to several data stores scattered across various switches, possibly across various locations.

9. Take two steps back and consider some of the unstated assumptions implicit in the present design and implementation of HDFS. Is it, in its present form, capable of serving as a truly global file system? Please provide and explain three specific examples of design or implementation decisions that support your conjecture. Note: Please focus your answer on aspects of the design and/or implementation other than dependability

10. Does the Hadoop Map-Reduce framework, itself, recover and continue if a Master node dies? If so, how? If not, how much benefit would be conveyed by this feature? Why?

11. HDFS currently provides no mechanism for fail-safe, fail-soft, or self-recovery for the failure of a NameNode. Please define a mechanism that would enable continued operation or automatic recovery after the failure of a single NameNode. Your solution should not significantly impact throughput in the normal case. You should describe normal operation, any transition after the failure, operation prior to the replacement of the failed NameNode hardware, and the transition back to operation with a full set of resources.