Name:

**15-440 Exam 3**
**Spring 2010 (Kesden)**

**Due: By noon on the Monday after Carnival.**

1. Consider some quorum-based replication scheme. Is it necessary for the operation to occur in the same order on each replica? Why or why not?

2. Consider client-side caches on the various quorum-based replication schemes we discussed. Assuming a system does Coda-style whole-file caching, is it feasible to count client-side replicas existing on various hosts toward satisfying the read or write quorums? Why or why not?

3. The *Coda* file system assumes that server-to-server bandwidth is high and client-to-server bandwidth is normally high, but can be low, e.g. weakly connected mode. Please consider how we might handle replication differently if we assume the network demands on the server to be bursty, but allow for lazy replication. Please describe an appropriate replication policy and what about it guarantees that requesting clients will see the most up-to-date version of the file.

4. At a high-level *MogileFS* and *Lustre*, are both distributed, network-based RAIDs. But, they have different strategies with respect to the detection and recovery from errors in file blocks. How do their strategies differ? If one strategy is universally better than the other strategy, please explain why. If the best strategy depends on some aspect of the file system's intended use, please explain what about the intended use of each file system favors the policy it has selected.

5. At a high-level *MogileFS* and *Lustre*, are both distributed, network-based RAIDs. But, they have different strategies with respect to the storage of file data. *Lustre* breaks files into chunks and stores these chunks across various repositories, whereas *MogileFS* organizes its storage in temrs of whole files. If one strategy is universally better than the other strategy, please explain why. If the best strategy depends on some aspect of the file system's intended use, please explain what about the intended use of each file system favors the policy it has selected.

6. How might the block size used on the HDFS system affect the appropriate level of parallelism, e.g. total number of map operations, associated with the first phase of a Map-Reduce process.

7. Is it essential for the Hadoop file system to be location-aware? Why or why not?

8. Consider a revised version of HDFS, one which automatically decides how many replicas of a file should be stored within the system. What factors might this system consider when deciding if an additional replica of a file should be created or if an existing replica should be allowed to age away?

9. Students often ask if, in designing a map-reduce process to solve some problem, they should somehow balance the number of maps and the number of reduces or the number of map-reduce phases, e.g. go wide or go deep or go somewhere in between. Does the number of maps in a phase generally affect the number of reduces? Why or why not?

10. In designing a map-reduce process to solve a particular problem, what factors determine the number of maps used in a particular phase?

11. Consider the case of distributed transactions, transactions where the various resources might exist at multiple repositories. Describe a simple locking scheme that can be integrated into 2PC to ensure that the updates to all resources remains consistent until, at least, the transaction commits. In other words, describe how you can integrate a locking protocol into 2PC to ensure the isolation property holds true. A critical aspect of your answer is describing where within the 2PC protocol locks should be checked, and/or released and/or acquired.

12. Your instructor argues that processor allocation policies are dramatically more critical than process migration policies. What is his (or any other reasonable) basis for this?

13. Why are log-based recovery schemes very often supplemented with check-pointing? What about vice-versa?

14. Is more communication required to play back and restore a system using a sender-recorded or a receiver-recorded log? What about to record the log entries during normal operation? Why?