CSE 490h/CSE M552
Problem Set
Due: 8:30am, Thursday, March 10, 2011

1. A causally consistent distributed checkpoint is a set of checkpoints at the nodes of a distributed system, such that if an event affects the state of a checkpoint at node P, all events that "happen before" the event, on any processor, are contained in the checkpoints on those processors. Sketch a method for taking a causally consistent distributed checkpoint.

2. Suppose that Coda never encounters a reconciliation conflict when re-integrating a disconnected node. In that case, are Coda operations sequentially consistent? If so, explain why or if not, give a counterexample.

3. In Paxos, suppose that the acceptors are *A*, *B*, and *C*. *A* and *B* are also proposers, and there is a distinguished learner *L*. According to the Paxos paper, a value is chosen when a majority of acceptors accept it, and only a single value is chosen. How does Paxos ensure that the following sequence of events *cannot* happen? What actually happens, and which value is ultimately chosen?

   a) A proposes sequence number 1, and gets responses from *A*, *B*, and *C*.
   b) A sends accept(1, "foo") messages to *A* and *C* and gets responses from both. Because a majority accepted, *A* tells *L* that "foo" has been chosen. However, *A* crashes before sending an accept to *B*.
   c) B proposes sequence number 2, and gets responses from *B* and *C*.
   d) B sends accept(2, "bar") messages to *B* and *C* and gets responses from both, so *B* tells *L* that "bar" has been chosen.

4. Consider the following systems: Coda, MapReduce, GFS, BigTable, MegaStore, Dynamo, and PNUTS. For each of the following, find one example of that feature in one of the systems, and sketch its role in the system. Use each system as an example at most once. 1-3 sentences are sufficient for each example.

   a) Weakly consistent caching
   b) Sequentially consistent cache coherence
   c) Operation logging
   d) State machine replication
   e) Application-specific failure recovery

5. Given a table of URL pairs, where the second URL refers to the first URL, sketch a mapreduce program to compute PageRank, in pseudo-code similar to that found in the mapreduce paper.