Projects in distributed systems (Draft)

1. *Design a fault-tolerant server using server replication*.

Start with a single client-server system. Use active replication to create server replicas. Your system must tolerate both crash and byzantine failures, so you have to simulate faulty channels and faulty servers.

2. *Design a distributed transaction system that tolerates crash failures using checkpointing and rollback recovery.*

Consider a network of eight banks. The clients will transfer money from one bank to another. Start with an algorithm for recording checkpoints as coordinated snapshots running at regular intervals. Implement the transaction manager and the data manager for each bank. Induce crash failures and show that the database never becomes inconsistent.

3. *Self-stabilizing spanning tree*

Implement a self-stabilizing spanning tree protocol on a network of n processes. Inject transient failures and study the time complexity for large values of n. Make conjectures about the time complexity from these experimental runs. Try to design a more efficient protocol for self-stabilizing spanning tree construction.

4. *Use Amazon EC^2 Elastic MapReduce to solve a graph theoretic problem*

Learn by yourself how to use Amazon's elastic computing cloud (EC^2) and run MapReduce for solving real-life data-intensive problems. Download big data from appropriate sites, and define the problem that you are going to solve. Then formulate the problem using MapReduce, run it, present the data / results and draw appropriate conclusions.