15-440 HW 3: Distributed Systems Basics

Due: In class on Tuesday, 3/21/09

April 9, 2009

1 Shared Memory

Shared memory systems, such as Pople at the PSC, are normally located in one data centre. While it's possible to implement a shared memory system (including OpenMP) on a system spread across the internet, this is not often done. A message passing approach is typically used instead.

a) What trade-off is being made when we decide between a shared memory model and a message passing model?

b) Why does this make shared memory a bad match for a system distributed across the internet?

2 Map-Reduce

a) Can Map-Reduce recover if a *Mapper* node dies? If so, how? If not, why haven't we solved this problem?

b) Can Map-Reduce recover if a *Master* node dies? If so, how? If not, why haven't we solved this problem?

3 "Fair" Clocks

In class we discussed using a host id, such as IP address, to break ties and impose a total ordering on Lamport time stamps. This approach, on its face, is unfair. Some hosts are more likely to win than others. We could alleviate this unfairness by randomly choosing between between two concurrent events.

a) Consider the voting districts algorithm. What problem could arise if two nodes ordered two concurrent events differently? Explain how this could occur.

b) How could you ensure that every node in the system agrees on this randomly chosen ordering of concurrent events? (i.e. what would the clocks need to do)

c) What are the advantages of giving up fairness?

d) Which approach do you like better? Why?