EECS 578 - Project Outline

1. Project title

Efficient execution of MapReduce applications on irregular NoC topology

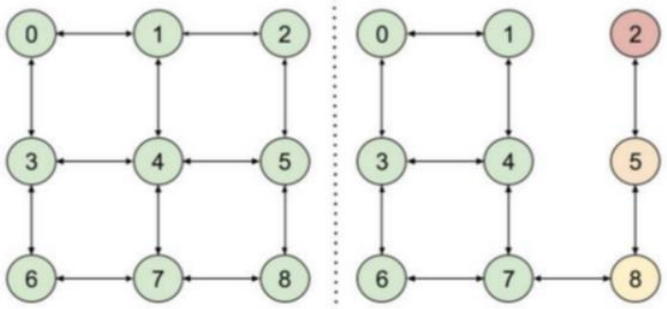
2. Team name

HAMM

3. Problem to be addressed

Hard faults in a network on chip (NoC) interconnect can cause links to break, creating an irregular network topology and potential bottlenecks. Since MapReduce applications create considerable network traffic they are sensitive to the network topology. Nodes communicating on congested links will experience communication delays causing them to finish later than well connected nodes.

For example, in figure 1, each node in the regular topology (left) completes the mapping phase of MapReduce at approximately the same time. Conversely, nodes 2, 5, and 8 in the irregular topology (right), might finish much later than other nodes when given the same number of jobs. The overall execution time depends on the node that finishes last. As a result, the total execution time will be prolonged by the slow node.



Regular Topology Irregular Topology

Figure 1. Regular versus Irregular topology

4. Progress so far

In order to gather network traces, we simulated Phoenix++ workloads on gem5. We have set up the infrastructure to use the Garnet NoC simulator in full system simulation mode. So far, we have generated network traces for k-means on a 4x4 mesh.

5. Issues

We found out that some of the Phoenix++ tests (word count, matrix multiply, string match, and pca) that we wanted to run could not compile. This requires us to debug and fix those. Since we couldn't run the MapReduce workloads in gem5's system call emulation mode, we opted for full system simulation. Consequently, our runs are taking significantly longer time than we had anticipated. Under the time constraints, we may need to simulate smaller workloads on a smaller network.