15-418 Project Proposal Checkpoint

Ashwin Adulla, aadulla

A Comparative Analysis of Various Network Topologies and Algorithms

Summary: I plan on performing a study that analyzes how well various interconnection network topologies (i.e. mesh, ring, fat-tree), flow control algorithms (i.e. worhmhole, virtual channel), and routing algorithms (i.e. deterministic, adaptive) perform with regards to specific metrics (i.e. throughput, delay, contention). Such an analysis can provide insight into how certain combinations of topologies and algorithms affect the overall performance of message transmission and how that performance scales with the number of nodes on a massively parallel system.

Implementation: I plan on implementing this in C++ and will be using either the ISPC task extension or the pthread library to simulate multiple nodes transmitting information to and from each other simultaneously. I will most likely be using the Latedays cluster to run my tests as I will be able to utilize 12 threads at once. I may also look into using OpenMP, though I think its best to have finer-grained controlled over the threads because I plan on having certain threads be specifically allocated for certain tasks. If there is time towards the end after I have finished the bulk of the implementation and have gathered enough data statistics to perform a meaningful analysis, I would like to extend this project with a python simulator where I can demo in real-time the communication across the network and even add in some visualizations to highlight where there is massive contention/traffic in the network. Such a simulation would be especially helpful for a hardware designer to help pinpoint where certain deadlock/livelock/starvation conditions are happening.

Goals: With this project, I hope to draw some conclusions regarding how certain interconnection network decisions can be made when targeting certain parallel architectures. For example, given the high level specs of parallel architecture such as the number of nodes, the expected amount of communication, and the expected message width, I would like to show how different network topologies and algorithms would perform and possibly even be able to propose one that best fits that parallel architecture. Moreover, I hope to achieve a deeper level understanding of hardware communication and how exactly it is implemented.

Challenges: The biggest challenges will be with making the simulation efficient enough and avoiding any deadlock/lievelock/starvation conditions when multiples threads are transmitting messages across the network. With regards to making the simulation efficient, I will need to cleverly implement fine-grain locking in the network so that multiple threads can be traversing it

at once without having to pass around locks between one another when they are not contending for the same network resource. In addition, depending on how the network data structure itself is laid out, I will have to look into cache locality to ensure that with every hop from one node to the next, there can be some reuse of data in the cache. With regards to avoiding any deadlock/livelock/starvation conditions, I will need to a) investigate various routing algorithms that may prevent these things from happening and b) design into the network sort of way to detect and break these conditions when they do occur. An immediate idea which comes into mind is maintaining an 'arbitrator thread' whose job is to supervise the network, detect any of these conditions, and take the proper actions to solve them. To do so, I would have to implement some sort of signaling or message passing between threads where the 'arbitrator thread' can signal to one of the threads executing a task that it is caught in some condition and needs to take certain steps to get out of it.

Research: I have been looking into current network on chip simulators, but most of these are very specific to hardware designers looking to implement these networks in their physical designs. I have found an existing simulator on github called Noxim that is built in SystemC and allows the user to specific various network parameters. It will then run a simulation over the user-specified network and return various metrics which the network performance has been evaluated on. However, Noxim was specifically designed for radio and wireless networks and so much of the topologies and routing algorithms which it supports are specific to that domain. I have also been reading up on literature that detail various interconnection networks and algorithms and the associated pros and cons with each. I have also been reading a few research papers that analyze these networks under different workloads and report their performance measures.