CS647 Distributed Software Systems

Project Pre-proposal

Omar Badran, Jordan Osecki, and Bill Shaya

Our CS647 group would like to explore the MapReduce distributed software system for our term project. We are proposing the development of a Java application that will simulate a MapReduce system on a P2P network that will count the number of words in a file. Upon running the application, our software framework will read a configuration file and will spawn a pre-configured number of peer nodes to simulate a P2P computational environment. The configuration file will also contain settings that the simulator will use to simulate various scenarios.

Our group plans to incorporate self adaptation through self healing and self configuration. Self healing will be accomplished by monitoring the worker nodes and the master node. If a worker node fails due to loss of connectivity to the network or some other fatal condition or runs very poorly, the failed node's computation will be redistributed to a healthy node. If the master node fails, one of the other peers will take over as the master and restart the map/reduce operation. Therefore, the overall computation can seamlessly complete despite the failures. Our application framework will include a module to induce random failures throughout the simulated network in order to exercise self healing. Self configuration will be accomplished by the peer nodes negotiating who will act as the master and the remaining nodes will be dynamically allocated as mappers or reducers depending on the size of the map/reduce operation. In order to evaluate the effects of self adaptation, we will ensure that the tasks completed correctly even in the midst of failures, inefficiencies, and re-configurations.

There are several notable MapReduce systems that exist, such as Skynet and Hadoop. Skynet is an open source Ruby implementation of Google's MapReduce framework, which is adaptive, fault tolerant, and has only worker nodes which can act as a master at any given time. Hadoop is a Java framework used to implement MapReduce functionality, which is currently used in Yahoo web searches. These systems are based on a network of computers connected via a local network versus a P2P network. The goal of this project will be more of a proof of concept to see if a map/reduce system can be implemented over a P2P network without a central command and control computer. It will also show novel ways to recover from inefficient or disabled nodes occurring at different parts of the process.

We feel that our project has adequate scope for a team of three. Work breakdown components will include the master functionality, worker functionality, self adaptation incorporation, fault detection and handling, performing experiments/trials with the simulation, and documenting our progress and conclusions. Each component can be completed independently by a group member and we do not anticipate any issues with completing the project by the end of the class term.