Noel Alexander

Cornell University

Modeling Big Data Networks Using Queueing Theory

nma44@cornell.edu

In a world of big data and cloud computing it is important to understand how data can be processed quickly with high accuracy. There are two widely used software packages that assist in this effort: Hadoop and MapReduce.

MapReduce is a software framework and an associated implementation for processing large data sets in a distributed fashion over several machines. The functional model behind the program allows us to parallelize large computations easily, use re-execution as a basis for fault tolerance, and perform data distribution and load balancing. MapReduce is useful in a variety of applications such as distributed search, distributed sort, and web-link graph traversal. The research currently being carried out focuses on how the presence of non-stationary data arrival streams and methods for job splitting affect software performance.

The goal will be to develop a mathematical queueing model to represent how the data is distributed and processed. A queueing model allows us to view the system from the operator and user perspective. The operator is interested in how many jobs there are, how to split those jobs, and scheduling various machines for processing. A number of policies for job routing and scheduling have been carefully examined in the past but have focused on striking the right balance between data-locality and load balancing to maximize throughput while simultaneously minimizing delay.

Improving locality can reduce both the processing time of map tasks and the network traffic load. Conversely, the user is more selfish than the operator, and is only interested in how long their individual jobs will take. The proposed queueing model will allow us to leverage the extensive amount of research that has been done on queues for our application, and derive distributions for job sojourn time and waiting time.