

Report for *Should Datacenters Converge and Supercomputers Converge?*

PI: Indranil Gupta

Students: Mainak Ghosh, Yosub Shin

In today's age of Big Data, large-scale distributed systems (also sometimes called *cloud computing systems* or *cloud software systems*) are largely run in datacenters, on top of loosely coupled network topologies, e.g., fat trees, CLOS, etc. However, we believe that supercomputer architectures offer significant opportunities to improve the performance of these applications. Yet today it is challenging to run software like distributed key-value stores and distributed graph processing systems atop a supercomputer. Besides the logistical challenges of actually getting the software to run (libraries for supercomputer setup, etc.), there is the open question of how to optimize the performance of these systems. A key idea we are exploring is to schedule these systems to run on the supercomputer so as to embed the communication pattern onto the 3D torus.

We worked on two different systems -- 1) a popular NoSQL database, Cassandra and 2) a graph processing system, Powergraph.

We deployed Cassandra on top of a supercomputer and compared the raw performance with that of a deployment in a datacenter (Emulab). The throughput provided by Bluewaters was 5x that of a datacenter even though the network bandwidth in the supercomputer was 2 orders of magnitude faster. We also compared parameters like read/write latency, consistency and the results were similar. This is because supercomputers are more tuned to handle high throughput applications than low latency ones. As part of future work, we wish to characterize the throughput improvement obtained from a supercomputer. We believe with careful optimization one can improve it to match the network bandwidth. For more details, please see our attached writeup [1].

PowerGraph is a distributed software to perform computations (e.g., PageRank, shortest path, etc.) on large graphs. PowerGraph partitions the graph across servers (or nodes) using a vertex cut strategy and replicates the cut vertices. The computation runs in iterations, synchronized across nodes. At the end of each iteration the replicated vertices exchange values calculated at the end of the iteration which is then aggregated at a master vertex and then disseminated back to the replicas. We wish to embed the communication pattern into the 3D torus topology by essentially embedding the graph itself into the 3D torus. This implies restricting communication to nodes that are a few hops away. We have modified PowerGraph's partitioning strategy and have got some encouraging results. Running approximate diameter, a problem requiring lot of data transfer at the end of each iteration, on a synthetic graph got us an improvement of almost 45% with the modified partitioning strategy as opposed to the naive one currently in use. We take advantage of the fact that many graphs in nature offer a power law distribution of degree. For more details please see our attached writeup [2].

Next Steps:

1. We wish to characterize the expected improvement. Current experiments show that it is dependent on the type of algorithm and power law nature of the graph.

2. We also wish to explore other partitioning algorithms (like Grid) that perform well with a 3D torus underlying topology. We have to compare our strategy to these existing ones.
3. Finally, similar to graph processing, stream processing applications also require high throughput and would benefit from a good embedding of its communication graph inside Bluewater's 3D torus network design. We also wish to explore this.

Finally, we wish to clarify that we have only used a small number of cycles from our allocation. This is because of the highly exploratory nature of our work--deploying the software itself on BlueWaters was a significant undertaking (with libraries, CCM, etc.), and we learnt a lot during the process. We believe our research has just started, but these baby steps portend good results to come.

Given the partially completed state of this work, and the relatively small number of cycles we have used, we will likely be requesting a small allocation in Fall 2016 to continue this work.

References:

- [1] Yosub Shin, Mainak Ghosh, Indranil Gupta, "Cassandra on Blue Waters," 2015.
http://web.engr.illinois.edu/~mghosh4/Cassandra_On_BW.pdf
- [2] Yosub Shin, Mainak Ghosh, Luke Leslie, Indranil Gupta, "Topology-Aware Graph Partitioning in Distributed Graph Processing Framework," 2015.
http://web.engr.illinois.edu/~mghosh4/Powergraph_On_BW.pdf