MPI Homework (Due 11:59EST, 3/20)

1. Quicksort using MPI (40 points)

1.1 Generate a large list of random numbers (either on head node, or on all nodes)

1.2 Assume a n-D hypercube and sort them.

1.3 Make sure your code has testing/debugging modes so I have an easy way to determine that your code works. Possible strategies could include a debug mode that only works with 100 numbers and checks the 100 numbers to confirm that the list is sorted. You need to have such strategies if you were doing this in a production environment

2 Parallelize the PageRank algorithm with MPI (60 points)

2.1 Use one of the datasets from the Stanford Network Repository

2.2 You can have the head node read and parse the file and initialize your matrices. Alternatively, you can write a separate program to read and transform your data, and write the transformed data to file. Then your parallelized version can work with this.

2.3 Compute the PageRank until they converge (let's assume that convergence is reached when the two-norm of the PageRank vector changes by $< 10^{-4}$)

2.4 Realize that your matrices are very sparse, so be smart about it. Also, do you think it makes sense to communicate the entire PageRank vector.

2.5 You will submit an evaluation strategy and submit a brief report on this. Discuss your parallelization strategy, your optimizations as well as your testing and evaluation plan.

2.6 For evaluation, I would suggest implementing a serial version (will also help you understand PageRank) and running a small dataset on it.

2.7 I will post a detailed scoring strategy soon. It will be based on speedup achieved by running the largest graph on say 64nodes. I will elaborate upon this in a message on Slack.

Useful Links

- http://www.ams.org/samplings/feature-column/fcarc-pagerank
- https://goo.gl/oNy5DG
- http://snap.stanford.edu/data/index.html