

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

1. Quicksort using MPI (40 points)

- 1.1 Generate a 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 the web-graphs from the Stanford Large Network Dataset Collection (<http://snap.stanford.edu/data/index.html#web>)
 - ***You can have the head node read and parse the file and initialize your matrices. Alternatively, you can write a separate program (in say python) to read and transform your data, create the various transition matrices, create the sparse matrix representations and write those to file. Then your parallelized version can work with this file.***
- 2.2 Compute the PageRank until they converge (let's assume that convergence is reached when the two-norm of the PageRank vector changes by $< 10^{-4}$)
- 2.3 Realize that your matrices are very sparse, so be smart about it. CSR maybe?? Also, do you think it makes sense to communicate the entire PageRank vector.
- 2.4 For evaluation, I would suggest implementing a serial version (will also help you understand PageRank) and running a small dataset on it.
- 2.5 Your submission will be your code (via git) and a **report*** that includes:
 - Name of the dataset that you worked on
 - Your parallelization strategy and optimizations
 - Your testing and evaluation strategy.
 - List your Top100 nodes in a file called **top100.txt**
 - You will also submit your speedup and efficiency, as measured against the largest dataset that you worked with.
 - Finally, while you can use the entire cluster, I would suggest no more than 64nodes. However, if the entire class can self-organize, then you can use more nodes. But wait until the last few days to do such large runs. By then you should have a good idea about the time your job will take, as well as it's efficiency.

Useful Links

- <http://www.ams.org/samplings/feature-column/fcarc-pagerank>
- <https://goo.gl/oNy5DG>

* Format: pdf; Length: 2 pages; Single-spaced; Font Arial; Size 11-14; 0.5" margins