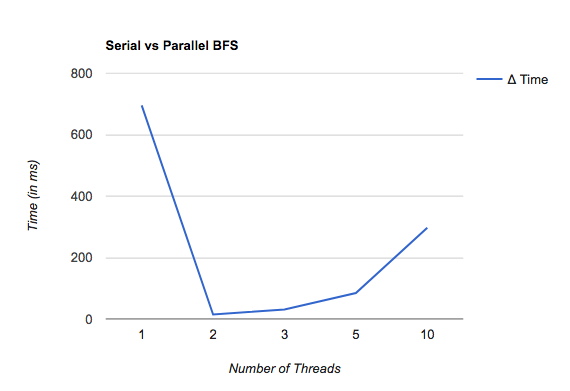
Daniel Acevedo (daacevedo) and Sashi Thapaliya (sbthapaliya)

CS 599-03

Parallel Breadth-First Search

For the following graph, we used large\_graph.txt, a file containing 1,161 vertices, and chose a common source node. The “1” label on the horizontal axis represents the serial run time, while the other labels represent a different amount of threads used for the parallelized version. The parallel BFS implementation ran better than the serial version in all tests. The implementation was optimized with a two threads.



We tried to implement parallel bfs various ways to produce better speed up but none were able to give us a huge jump we were looking for. Like in many of the cases the time will remain the same or will increase for more number of threads. Especially using an array of locks downgraded the performance of our parallel bfs significantly.

Also, we noticed that a parallel bfs is not suitable for long-chain structured connected graphs like A->b->c->D->E where each node is connected with only one node and no two nodes have direct edges to the same node. In this case, there is only one node at each level so our traditional level synchronous parallelization will not be beneficial.