Computer Science Tripos - Part II - Project Progress Report

Evaluating Betweenness Centrality Algorithms for Real World Datasets

Albert Kutsyy ak2149@cam.ac.uk Trinity College

February 3, 2021

Project Supervisor: Dr. Timothy Griffin

Directors of Studies: Dr. Sean Holden, Dr. Neel Krishnaswami, Prof. Frank Stajano

Project Overseers: Prof. Marcelo Fiore, Dr. Robert Mullins

Progress Report

Despite some unexpected difficulty, my Part II project is exactly on pace. My project requires implementing five algorithms, instrumenting them with a variety of metrics, and evaluating their performance on large graphs.

I quickly researched instrumentation and decided on a set of metrics for evaluating the performance of the algorithms (time spent per node, total time, memory usage, and the number of graph reads). By November 4th I also successfully logged onto the high power computing server I was granted access to, meeting my milestones.

By November 20th I created a framework for running and testing the centrality algorithms, and implemented my first algorithm, the `Brandes` algorithm. I verified its output using `JGraphT`, a Java graph analysis library.

I then optimized my graph representation and `Brandes` implementation, and ended up with an implementation that runs about 10 times faster than the `JGraphT` implementation.

I ran my implementation of `Brandes` on the high performance server to determine the maximum size of graphs I can use for my experimentation. I realized that my laptop actually has higher single threaded performance than the server, so will use it instead.

Following this, I began implementing the `Brandes++` algorithm. Difficulty arose because that algorithm requires implementing a specific (and very complicated) graph clustering algorithm to achieve the performance the paper describes. That graph algorithm itself requires implementing three more algorithms and a new data structure. I finished the graph partition algorithm by the 22nd of December, missing my milestone for finishing `Brandes++`. However, I'd set aside a buffer for unforseen difficulty and finished implementing and testing the `Brandes++` algorithm by January 12th.

By January 20th I finished the algorithm by Geisberger et al, meeting my milestone.

I additionally implemented multiple heap algorithms (Rank-Pair Heap, Binary Heap, Fibonacci Heap) to determine which allowed the `Brandes` algorithm to run fastest (It turned out to be the binary heap, which I'd already been using).

I have fully read through the remaining two algorithms and neither have complications like `Brandes++` does, so I expect to finish my project on time.