

Graph Algorithm Parallelization

Edmonds-Karp Algorithm Implementation in the Go Programming Language

Louis L. Fattet

Computer Science Capstone Project Spring 2013

1 Inspiration

For my first capstone, I wanted to attempt a project that was both feasible in scope and educationally interesting. After a fall semester of higher level computer science classes, I had become increasingly intrigued in algorithms, data structures, and software optimization.

2 Description and Scope

After speaking with my project advisor, I decided to create a challenge for myself that would be three-fold:

1. Learning a new programming language, Google's Go. Go is a very new language, created in 2007 and announced in 2009. It is a compiled and garbage-collected language, and allows for an object-oriented style of programming. The main reason Go is of interest to my project is its concurrency, or the ability to easily parallelize operations.
2. Continued studying of the implementation of both graph data structures and search algorithms. The algorithm in particular that will be implemented is the Edmonds-Karp, an algorithm used to find the maximum flow in a flow network.

3. Parallelization of the algorithm. Once the algorithm has been implemented, further study will be necessary in order to cut it up into separate jobs or threads that will be set up to run concurrently. Above all, this will be the most complicated step because it will require a fundamental understanding both of how the algorithm is executed and how Go handles its coroutines.

3 Schedule and Plan

I feel that there is a very clear and direct way to go about completing this project, and it all starts with spending time learning about the syntax and behavior of Go. For this reason, I only plan to have completed the initialization of the data structure by the semester's midpoint. From there, assuming that the structure was implemented correctly, writing a non-concurrent algorithm and then improving upon it should take the rest of the semester. If I can complete everything I have written here before the end of the semester, I'd like to test the algorithms on applicable real world problems or possibly even create a graphical interface.