Parallel Minimum Spanning Tree Algorithms

Mid-Project Report

Connor Clayton and Jacqui Fashimpaur

Changes and New Goals:

Since our original proposal, we have shifted our plan a considerable amount (we already spoke with Professor Railing about this). In particular, rather than writing and comparing the unrelated problems of Minimum Spanning Forest and Maximum Independent Set, we will be focusing on comparing multiple different algorithms for MSF. We are implementing parallel versions of Boruvka's algorithm and Prim's algorithm inspired by this Bader-Cong paper. We are using entirely our own code, though. We have an alternate implementations of Boruvka's (with a more complicated data structure) that we can explore if we have time. A more detailed schedule of this is below.

Our goals are similar to before, but adjusted for the new problems:

- Our plan is to create parallel versions of both algorithms, inspired by the above paper, and compare their effectiveness. We will do that via speedup graphs and a more detailed analysis--what parts are slow? Why?
- Our "hope to achieve" is analysis of a third algorithm like Kruskal's or the more complicated Boruvka's algorithm version we have.

Progress:

We have created a base for our project:

- Functions to parse text files and build graphs in two different data structures (adjacency list and flexible adjacency list)
- A main file including timing code
- A Makefile

We also have worked on our implementation of Boruvka's Algorithm:

- We originally planned on using a flexible adjacency list as our graph representation, and we have the sequential code for this mostly finished, but have set it aside for now
- Using a simpler but slightly slower adjacency list as our graph representation, we have finished a sequential implementation of Boruvka's Algorithm

Poster Session Plans:

At the poster session, we will describe our parallel algorithms and have speedup graphs for both of them.

Concern:

So far, we have been spending a lot of time getting the base programs to work and have not gotten to write any parallel code. Our concern then is that we are behind schedule and that we don't have a clear direction for parallelization. We believe that the MST problem has the right

level of natural parallelism to allow us to find significant, but non-trivial, methods of speeding up our programs, but we are relying on this for now.

Schedule:

Week	Dates	Plans
4	11/20-11/23	Jacqui - Write Code for Sequential Prim's Algorithm Connor - Debugging Boruvka's (Connor will be out of town 11/21-11/23)
	11/24-11/26	Jacqui - Start parallelizing Prim's Algorithm Connor - Start parallelizing Boruvka's Algorithm
5	11/27-11/30	Thanksgiving break; don't plan on getting too much work done during this time
	12/01-12/03	Jacqui - Optimize and Finish Prim's Algorithm Connor - Optimize and Finish Boruvka's Algorithm OR Jacqui and Connor, if done with above - Finish implementing and parallelizing Flexible Adjacency List Boruvka's
6	12/04-12/07	Code should be mostly complete Jacqui - Begin Poster (Jacqui will be super busy with play on December 6 and 7) Connor - Perform Analysis of Arograms; Begin Final Report
	12/08-12/09	Report due 12/09 at 11:59pm Jacqui and Connor - Get Poster Printed and Finish Final Report