Project 4 Writeup

Runtime Tables:

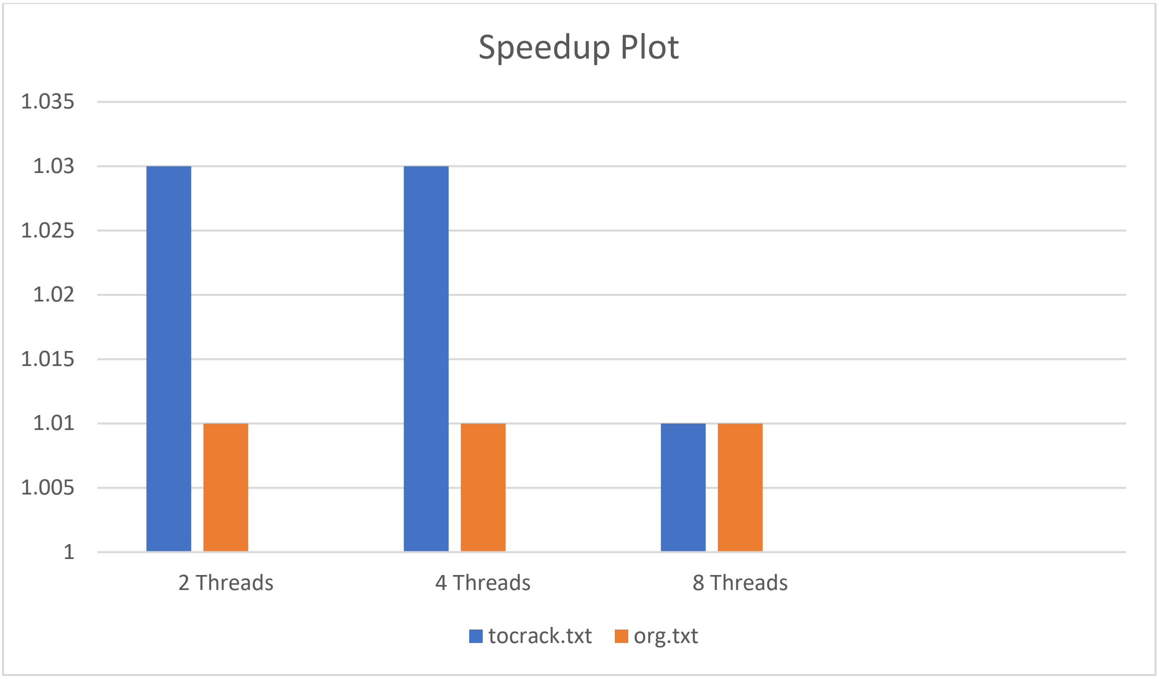
Tocrack.txt	Run 1 (s)	Run 2 (s)	Run 3 (s)	Average Speed (s)
1 thread	0.39	0.38	0.37	0.380
2 threads	0.37	0.37	0.37	0.370
4 threads	0.37	0.37	0.37	0.370
8 threads	0.38	0.38	0.37	0.377

org.txt	Run 1 (s)	Run 2 (s)	Run 3 (s)	Average Speed (s)
1 thread	0.38	0.38	0.38	0.380
2 threads	0.37	0.38	0.38	0.377
4 threads	0.38	0.38	0.37	0.377
8 threads	0.37	0.38	0.38	0.377

Speedup Table:

	2	4	8
Tocrack.txt	1.03	1.03	1.01
Org.txt	1.01	1.01	1.01

Speedup Plot:



Analysis:

General Strategy: The general strategy I used to implement my parallel solution was that I tried to improve the time that the program took to search for the correct password, so I did my best to parallelize that aspect of the program. The data structure I had to include in the project was the struct that came with the uthash header file, and I also chose to use an array that I iterated through with for loops and while loops. I tested my program using my Makefile, by writing various tests that allowed me to quickly test the different thread amounts and different .txt files, org.txt and tocrack.txt. I did not change the meat of my Part I solution, where I actually search through and find the passwords, because my Part I was able to complete the searching in a fairly timely manner. However, I did make significant changes to my other functions, including my main(), and added all of the pthreading functionality to it.

Benchmarking Results: My benchmarking results were not very good. As shown in my results tables, speedup table, and speedup plot, my speeds stayed relatively consistent throughout each of the tests, only showing a little improvement that I believe was, for the most part, negligent. Thus, I did not see consistent speedup as the number of threads increased. I believe this has to do with where I ended up implementing the parallel threading, and the actual increase in speed that I was able to get from the parallel threading.

Impact and Personal Reflection: Although the results I got told me that the parallel computing I did in this project did not have a huge impact on the password cracking, I know that is not the case. In actuality, I should have been able to see significant speedup with increasing threads. Some disadvantages to using Pthreads are that many library functions are not thread safe, particularly ones that return pointers which can cause issues, and if there is problem with one thread, the entire application will crash. The concept I struggled with the most was trying to figure out where to actually implement the pthreading in my program, which I don't think I was entirely successful at deciphering. The most valuable thing I learned in terms of programming was that C is a very useful language that already has a ton of tools built into it, so I should really use documentation more often. The most valuable thing I learned outside of programming was that I don't always have to do everything exactly right and that the world will not end if my project is not running perfectly.