Pat Lewis **CS 4234: Parallel Computation** **Project 2**

Due Friday, November 18, 2013

1. **Thread Creation and Destruction**

The object of this program is to examine the time it takes to create and destroy an individual thread based on how many other threads are being created. There timer starts right before the first thread is created, and ends after the call to `pthread_join` returns. Each of the threads does no work and returns immediately, so we can reasonably assume that each one takes no time to execute. The results of the timings versus thread size are shown below.

Number of Threads	Total Time (s)	Average Time(s)
1	0.000056000	0.000056000
2	0.000091000	0.000045500
3	0.000127000	0.000042333
4	0.000155000	0.000038750
5	0.000175000	0.000035000
6	0.000272000	0.000045333
7	0.000249000	0.000035571
8	0.000275000	0.000034375
9	0.000316000	0.000035111
10	0.000334000	0.000033400
11	0.000402000	0.000036545
12	0.000416000	0.000034667
13	0.000421000	0.000032385
14	0.000457000	0.000032643
15	0.000482000	0.000032133
16	0.000651000	0.000040687
17	0.003974000	0.000233765
18	0.000584000	0.000032444
19	0.000666000	0.000035053
20	0.000854000	0.000042700
25	0.000872000	0.000034880
30	0.000992000	0.000033067
35	0.001012000	0.000028914
40	0.001360000	0.000034000
45	0.001357000	0.000030156
50	0.001172000	0.000023440

Table 1: The results of the total and average creation time of 1 to 50 threads

While this data is great, there's a lot of it, and it isn't all that illuminating at first glance. Basically, the cost of starting and stopping a thread is high enough so that for a few threads, it is easily seen. However, as the number of threads increases, the creation/destruction time begins to converge to about 0.000035 s, or 35 μs. There are anomalies in the data: there are random spikes and occasional dips in the trend line. However, there is enough data overall to be able to see the trend.

The plot below shows how even though there is a high cost for small numbers of threads but the average cost converges suprisingly quickly.

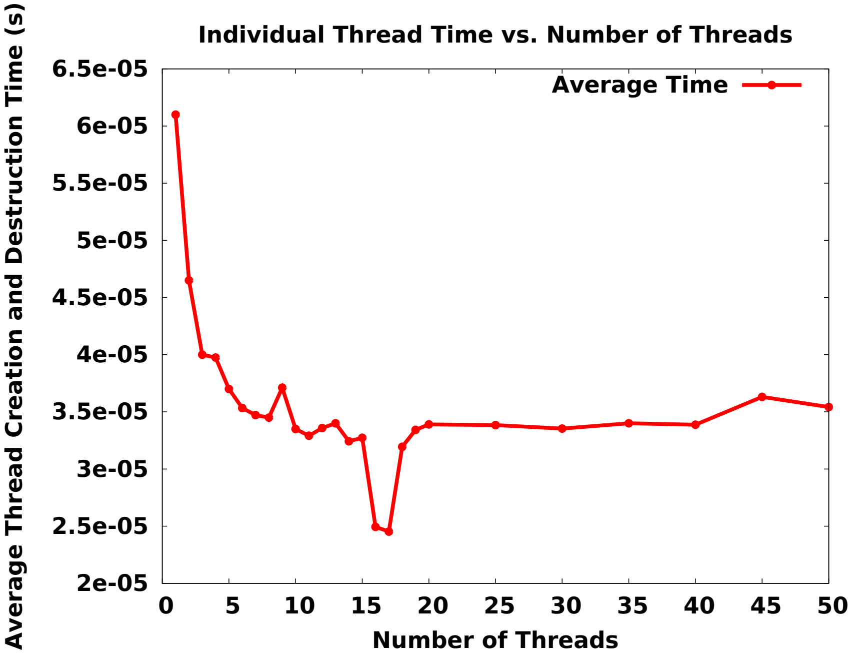

Figure 1: The plot showing the same data as Table 1.

The code that creates, destroys, and measures the time of each thread is included with this submission as `q1.c`. If you wish to run the program for yourself and collect your own data, I have a script that does so, outputs the results, and makes plots and tables for you. I will also include this with the submission as `proj2.sh`.