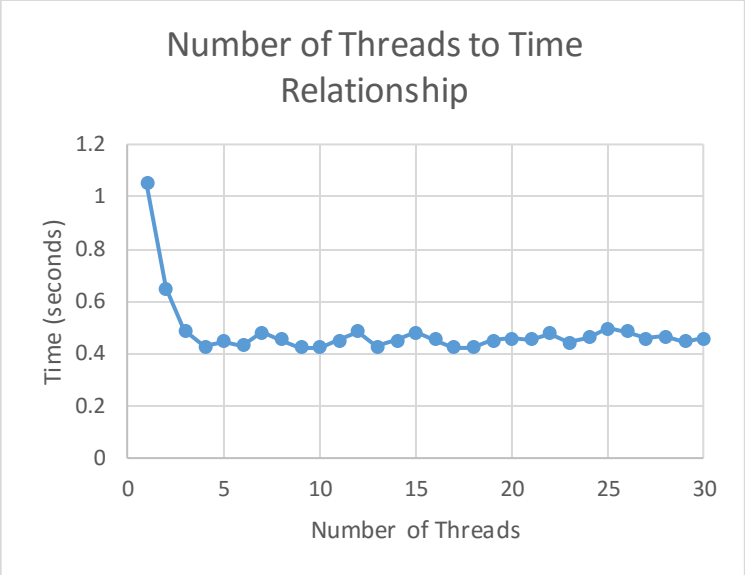
Luke Kledzik & Adam Mooers
Systems and Networks I - Project 2 - mtCollatz
Analysis

Parameters: We will be testing how the number of threads change the execution time of testing the collatz sequence to 1,000,000.

Number of Threads	Time
1	1.052047915
2	0.649505852
3	0.487532178
4	0.427830126
5	0.444811631
6	0.431158821
7	0.48216681
8	0.453640838
9	0.423212033
10	0.423471263
11	0.448775285
12	0.482644689
13	0.426613955
14	0.450972636
15	0.482476326
16	0.452321118
17	0.422428893
18	0.42189182
19	0.451082535
20	0.45819471
21	0.455191776
22	0.478461574
23	0.444156455
24	0.462598036
25	0.495364661
26	0.485933582
27	0.457453105
28	0.463567843
29	0.446187066
30	0.458419618



When analyzing our data that compares the number of threads to the time of program execution, we can clearly see that the worst case would be when you are only using one thread. At about 3-4 threads, we see the benefits of using multiple threads start to level out on the graph.

Several factors may contribute to the diminishing returns. The OS may limit the number of kernel threads. If multiple user threads have the same kernel thread, the throughput will remain constant. Also, the OS has access to a limited number of processing cores. Lastly, memory itself could be causing a bottleneck. Adding more threads will not benefit the process.