**Project 2 Experiment Analysis**

Mason Riley,

Cesar Santiago

October 20th, 2019

**Background**

This project focuses on the use of multithreading with the purpose of speeding up computation. For demonstration, we produce a list of Collatz sequences with the number of times the sequences happen being distributed amongst a number of threads, both of which are inputted by the user. The output of the program will be a time delta and a histogram that regards to the number of times the Collatz sequence took how many steps to complete.

**Experiment**

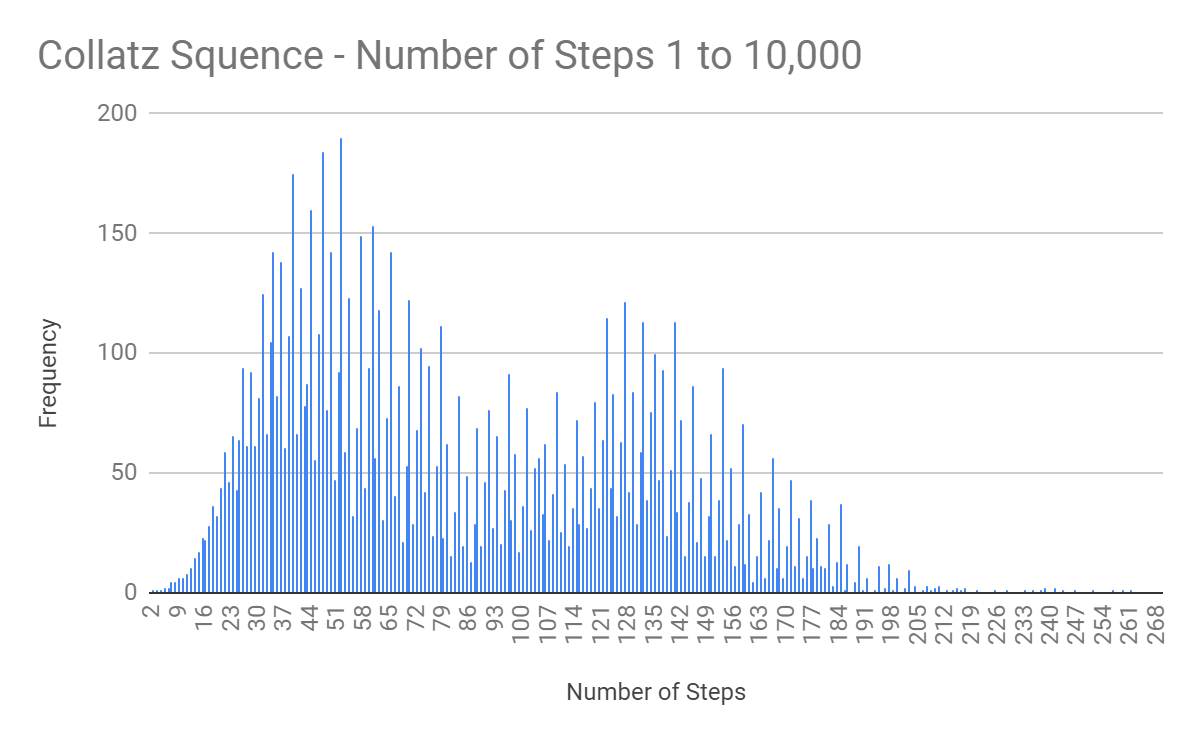
The experiment takes into account the amount of time taken for the computation of the Collatz sequence depending on the number of threads that were given. The experiment was conducted twice per number of threads from 1 thread to 29 threads, with the hypothesis being that every thread allowed would decrease computation time. From these experiments, a histogram will be noted with the purpose of observing the consistency of thread computation.

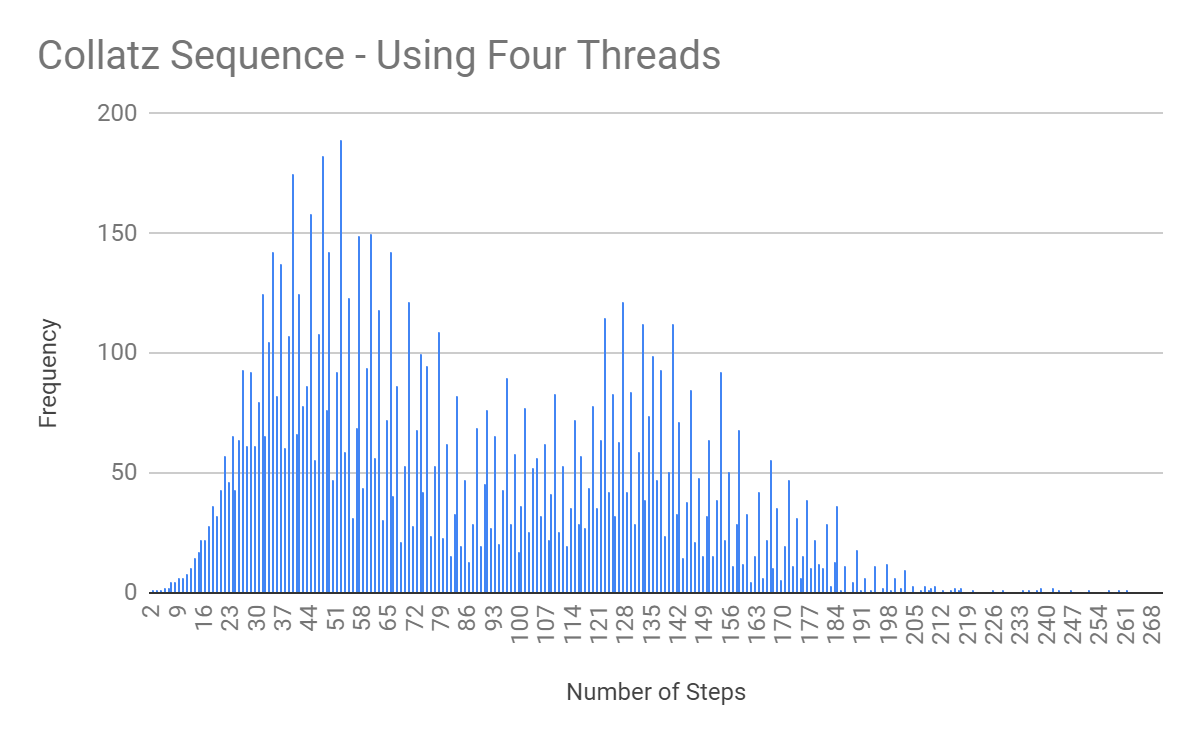
**Results**

These were the following results regarding the time and the thread count:

|  |  |  |
| --- | --- | --- |
| Limit | Number of Threads | Run Time (s) |
| 100000 | 1 | 0.174948 |
| 100000 | 1 | 0.119075 |
| 100000 | 2 | 0.117137 |
| 100000 | 2 | 0.139981 |
| 100000 | 3 | 0.107361 |
| 100000 | 3 | 0.093303 |
| 100000 | 4 | 0.126984 |
| 100000 | 4 | 0.116074 |
| 100000 | 5 | 0.112199 |
| 100000 | 5 | 0.108938 |
| 100000 | 6 | 0.106033 |
| 100000 | 6 | 0.111228 |
| 100000 | 7 | 0.091952 |
| 100000 | 7 | 0.09643 |
| 100000 | 8 | 0.105616 |
| 100000 | 8 | 0.104418 |
| 100000 | 9 | 0.105 |
| 100000 | 9 | 0.101972 |
| 100000 | 10 | 0.109331 |
| 100000 | 10 | 0.095406 |
| 100000 | 11 | 0.100254 |
| 100000 | 11 | 0.102117 |
| 100000 | 12 | 0.1021 |
| 100000 | 12 | 0.09881 |
| 100000 | 13 | 0.102987 |
| 100000 | 13 | 0.108488 |
| 100000 | 14 | 0.103725 |
| 100000 | 14 | 0.101213 |
| 100000 | 15 | 0.114408 |
| 100000 | 15 | 0.109228 |
| 100000 | 16 | 0.116701 |
| 100000 | 16 | 0.116556 |
| 100000 | 17 | 0.108234 |
| 100000 | 17 | 0.112271 |
| 100000 | 18 | 0.100443 |
| 100000 | 18 | 0.110677 |
| 100000 | 19 | 0.108387 |
| 100000 | 19 | 0.104964 |
| 100000 | 20 | 0.108577 |
| 100000 | 20 | 0.115431 |
| 100000 | 21 | 0.1047 |
| 100000 | 21 | 0.112904 |
| 100000 | 22 | 0.103314 |
| 100000 | 22 | 0.111625 |
| 100000 | 23 | 0.107081 |
| 100000 | 23 | 0.09695 |
| 100000 | 24 | 0.108711 |
| 100000 | 24 | 0.102588 |
| 100000 | 25 | 0.102934 |
| 100000 | 25 | 0.114888 |
| 100000 | 26 | 0.101731 |
| 100000 | 26 | 0.101436 |
| 100000 | 27 | 0.109057 |
| 100000 | 27 | 0.101259 |
| 100000 | 28 | 0.110412 |
| 100000 | 28 | 0.108353 |
| 100000 | 29 | 0.107556 |
| 100000 | 29 | 0.103676 |

The following are histograms that were outputted by each number of threads:





**Discussion**

From the results above we can determine that to compute the Collatz sequence 10,000 times has no change in effect between a high number of threads vs a low number of threads because of the overhead of working with threads outweighs the benefits of using them.

**Conclusions**

Threads are not a useful addition to this program regardless of the ability to compute in parallel since the outcomes show that the benefits do not necessarily outweigh the negatives of working with such overhead.