; **
; Name: Dylan Cozloff
; NSHE ID: 2001668380
; Section: 1003
; Assignment: 12
; Description: Parallel processing program to calculate the narcissistic number count.
****

Machine Description:

<machine type (desktop/laptop/mini), processor speed, disk type (SSD, hard drive, etc.), and memory>

For this assignment, I am working on a desktop, with a 3.6 GHz processor speed, an SSD, and 16 GB RAM.

Spin-Lock Results:

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 153, 370, 371, 407, 1634, 8208, 9474, 54748, 92727, 93084, 548834, 1741725, 4210818, 9800817, 9926315, 24678050, 24678051

- Perfect Numbers: 0, 6
- Abundant Numbers: 8208, 9474, 93084, 4210818
- Deficient Numbers: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 153, 370, 371, 407, 1634, 54748, 92727, 548834, 1741725, 9800817, 9926315, 24678050, 24678051

Speed-Up Timing:

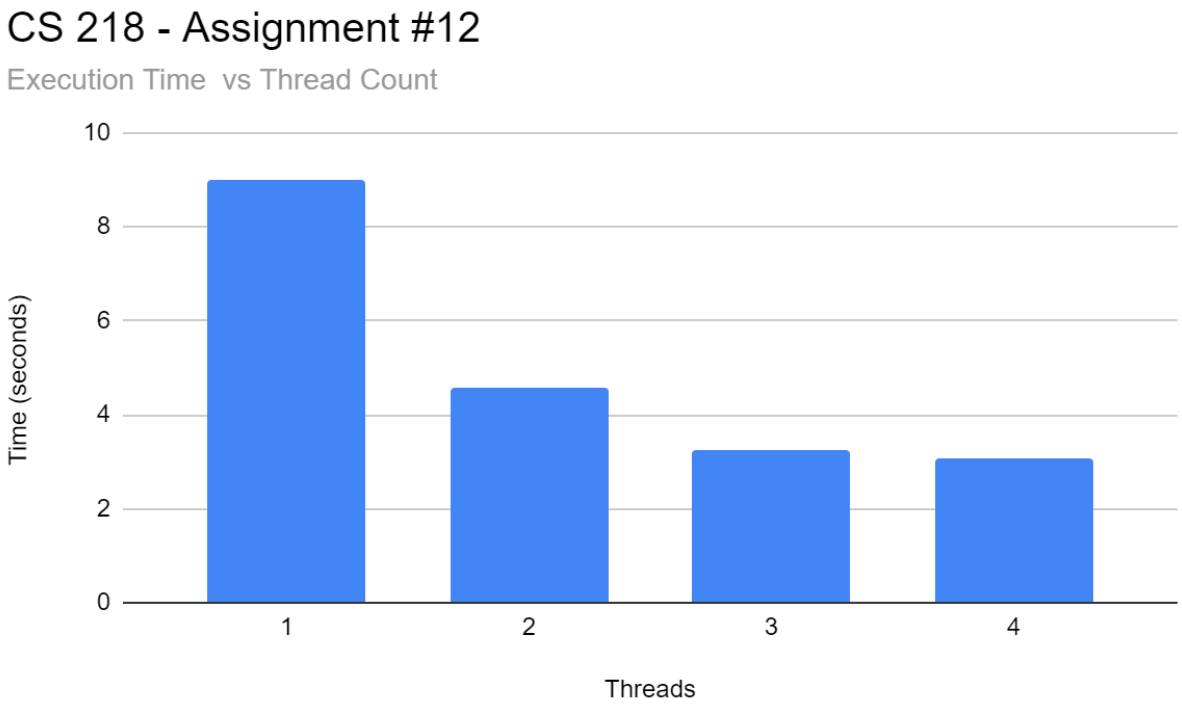
---

1 Thread: 9.002 seconds
- Speed-Up: 9.002 / 9.002 = 1

2 Thread: 4.555 seconds
- Speed-Up: 9.002 / 4.555 = 1.98

3 Thread: 3.257 seconds
- Speed-Up: 9.002 / 3.257 = 2.76

4 Thread: 3.071 seconds
- Speed-Up: 9.002 / 3.071 = 2.93



No Spin-Lock Results:
0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 153, 370, 371, 407, 1634, 8208, 9474, 54748, 92727, 93084, 548834, 1741725

- Perfect Numbers: 0, 6
- Abundant Numbers: 8208, 9474, 93084
- Deficient Numbers: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 153, 370, 371, 407, 1634, 54748, 92727, 548834, 1741725

Speed-Up Timing (no locks):
---
1 Thread: 9.577 seconds
- Speed-Up: 9.577 / 9.577 = 1

2 Thread: 4.753 seconds
- Speed-Up: 9.577 / 4.753 = 2.02

3 Thread: 3.364 seconds
- Speed-Up: 9.577 / 3.364 = 2.85

4 Thread: 3.143 seconds
- Speed-Up: 9.577 / 3.143 = 3.05

Explanation:

<not to exceed 300 words>

The execution time is larger for the program without the locking calls. This occurs because, the thread-context switch is a very expensive operation, costing between 2000 and 10,000 CPU cycles. Context switching is an operating system rescheduling process, and it is much slower than the short duration of the thread awaiting an unlock. The spinlocks are only held for short durations when writing to global variables so they don't interfere with the operating system scheduler. The OS scheduler will be stopping and activating the running processes, and every time the scheduler changes the process assigned to a core it will probably also need to cache the context of this process, adding a lot of cache-misses and consequently more time. Also, there is a high chance of race conditions, where two threads write to the same global variable in which there will be unpredictable errors. Spin locks prevent this from occuring, thus reducing the hard to debug erroneous values we saw in the output.