Homework 02: Report

Machine used: oxford

Java Command Used: java MainClass <gridNum> <numThreads>

Or: java MainClass <gridNum> <numThreads> <left> <top> <right> <bottom> <epsilon>

C Command Used: jacobi <gridNum> <numThreads>

Or: jacobi <gridNum> <numThreads> <left> <top> <right> <bottom> <epsilon>

Default values for missing args:

Left: 10

Top: 10

Right: 800

Bottom: 800

Epsilon: 0.1

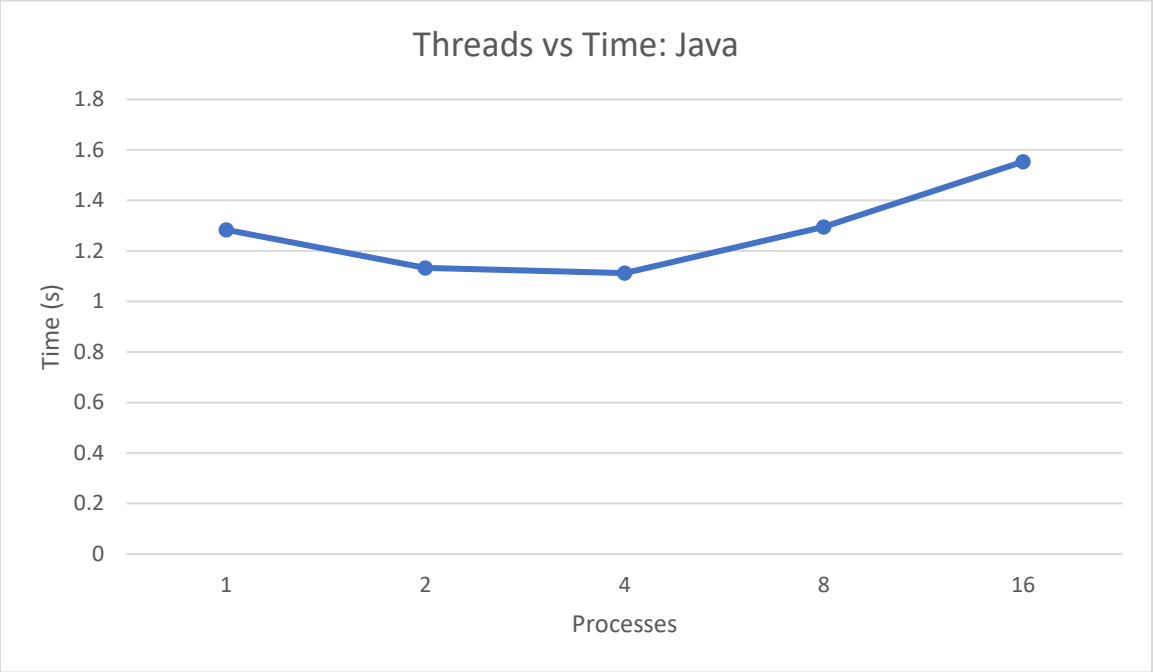
NOTES: C is supposed to be faster, no idea why it isn't. I couldn't find the operation that was taking the longest in those threads. Swapping the grids in C was giving me errors for some reason, so I had to copy every element individually. However, the C program times are similar to patrickJacobi. Analysis below.

Max iterations: 7000

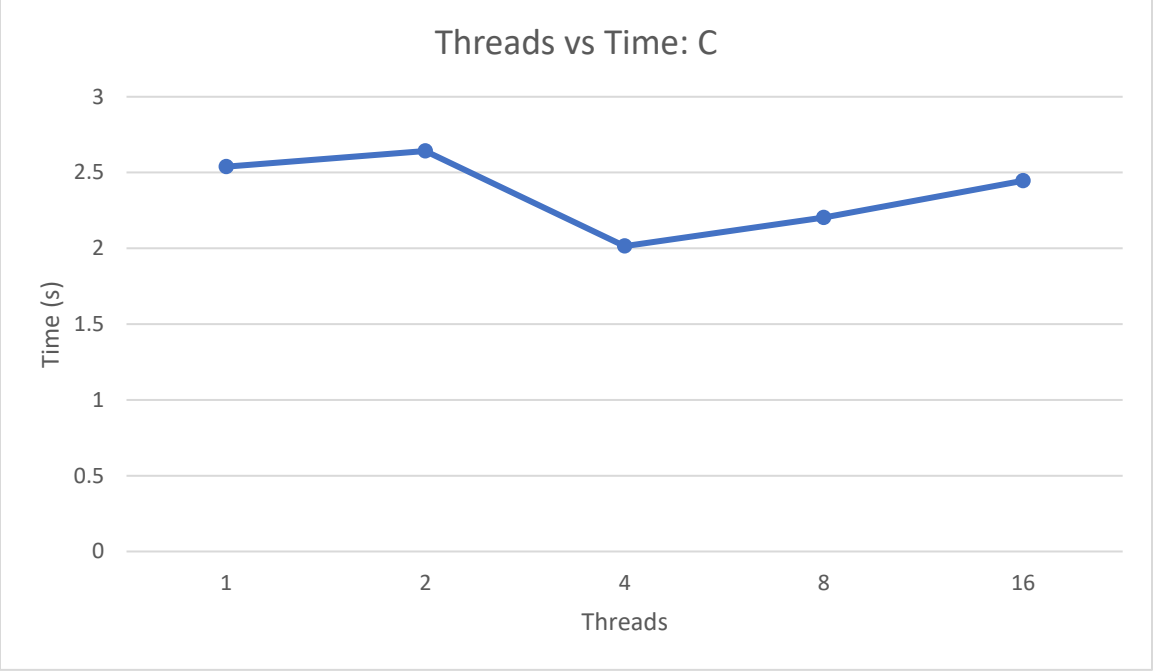
Program tested with gridNum = 200

Jacobi in Java:	**Jacobi in C:**
1 Thread: 1.284160 seconds 2 Threads: 1.132947 seconds 4 Threads: 1.112273 seconds 8 Threads: 1.294534 seconds 16 Threads: 1.553394 seconds	1 Thread: 2.539120 microseconds 2 Threads: 2.642235 seconds 4 Threads: 2.014519 seconds 8 Threads: 2.203529 seconds 16 Threads: 2.446596 seconds

Threads vs. Time: Java



Threads vs. Time: C



Analysis: It looks like the peak number of workers for the program is 4. In Java, two and four threads were very similar, but it went up from there. Interestingly enough, 16 threads was slower than one thread in Java.. maybe the thread overhead time was too significant? In C, two threads took longer than one thread (both tested about 20 times), which was interesting to me. Maybe the barrier wasn't quite as efficient as it could have been? 4 threads was still the peak, but both 8 and 16 threads were faster than 1 or two threads. Overall, Java was about 1 second faster on average than C.