

Bryce Graves

Professor Mike Bailey

CS 475

2020 5 13 (ISO 8601)

Project: 4

Bryce Graves

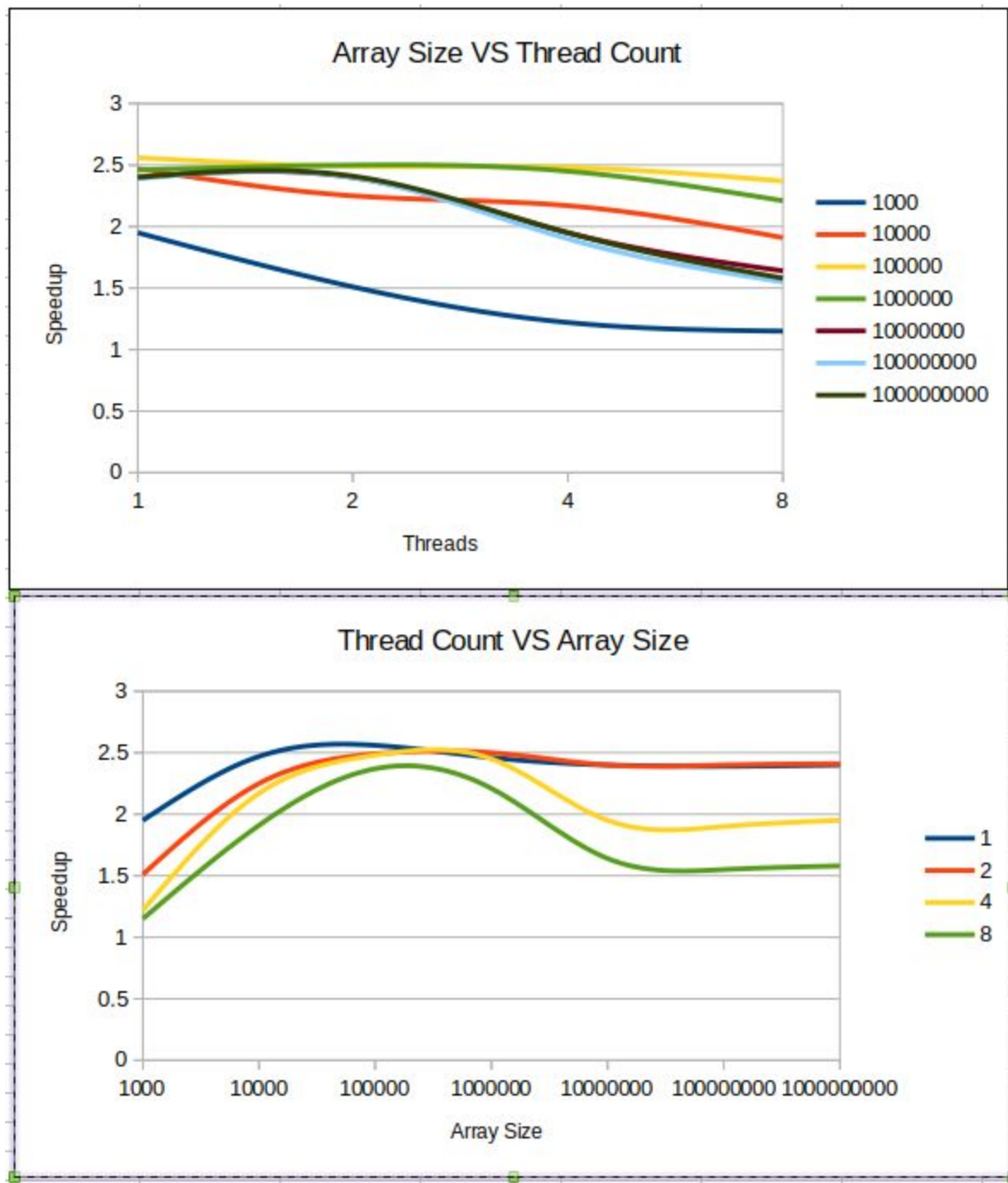
gravebry@oregonstate.edu

Hardware: still on the desktop ヽ_(ツ)_/

- **OS:** Linux Mint 19.3 Cinnamon
- **Kernel:** 5.3
- **CPU:** Intel® Core™ i7-6700K CPU
 - **Cores:** 4
 - **Threads:** 8
 - **Core Clock:** 4 GHz
 - **Boost Clock:** 4.2 GHz
 - **L1 Cache:**
 - 4 x 32 kB Instruction
 - 4 x 32 kB Data
 - **L2 Cache:** 4 x 256 kB
 - **L3 Cache:** 1 x 8 MB
 - **Simultaneous Multithreading:** yes - Hyper-Threading
- **GPU:** MSI Radeon RX 480 8 GB GAMING X Video Card
 - **Memory:** 8 GB GDDR5
 - **Core Clock:** 1120 MHz
 - **Boost Clock:** 1316 MHz
 - **Effective Memory Clock:** 8100 MHz
- **Memory:** 32 GB DDR4 overclocked to 3200 MHz from base 2133 MHz

Process thoughts: This project was quite the experience of why the heck am I not seeing speedup. The issue for quite some while working on this project was why am I not seeing speed up even though the multi threading seems to be working correctly. Turns out I do in fact have to perform a reduction while not dumping the results to the ether of my circuit board. The addition of **reduction(+ : total)** per each loop magically caused my resulting speedup to go from below 1 to around 2-3.

Data & Graphs:



Conclusion: Even after running through all of these tests I am not quite sure why the speed increase doesn't match the graphs seen during lecture. I am assuming it has something to do with

my implementation. At the end of the day I am glad that the graphs at least look right for what we should see when increasing array size.