

Alston Godbolt

CS475 Parallel Programming

Assignment 7B

June 9, 2017

1. What machines you ran this on

I completed the OpenMP and OpenCL on a Lenovo laptop. Below are the specifications:

Processor: Intel Core i3-4030U CPU @ 1.90 GHz

Installed memory: 4 GB

System type: 64 bit Operating System, x64 Based processor

I completed the SIMD portion of the project using FLIP and an iMac. Below are the specifications of the iMac:

MacOS Sierra

Version 10.12.4

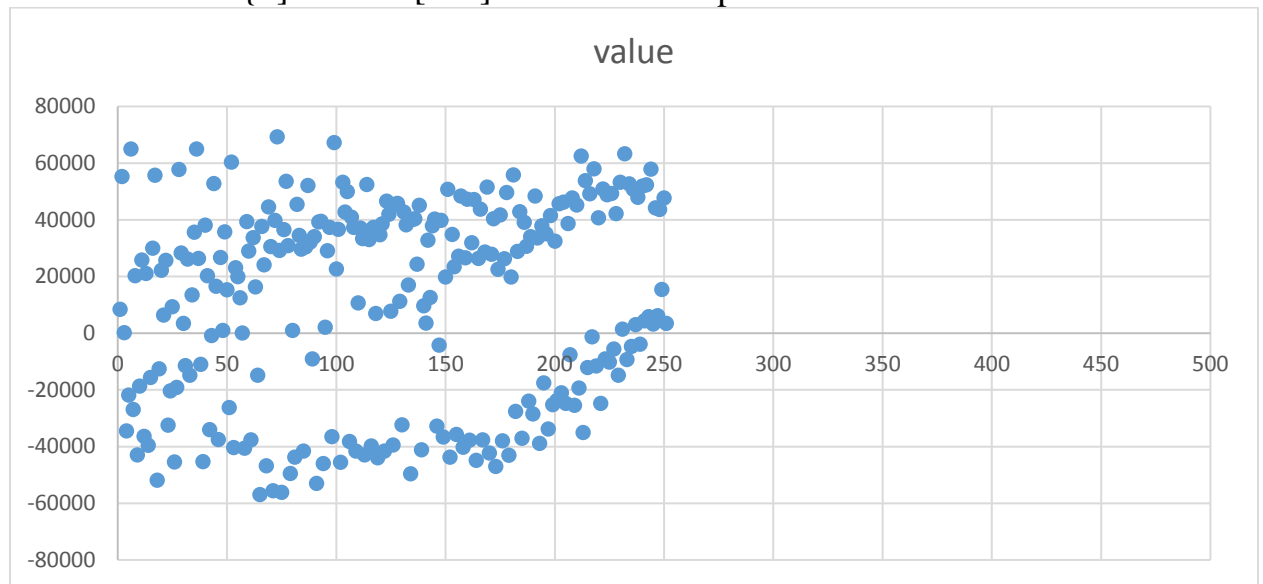
iMac Retina 5K, 27-inch Mid-2015

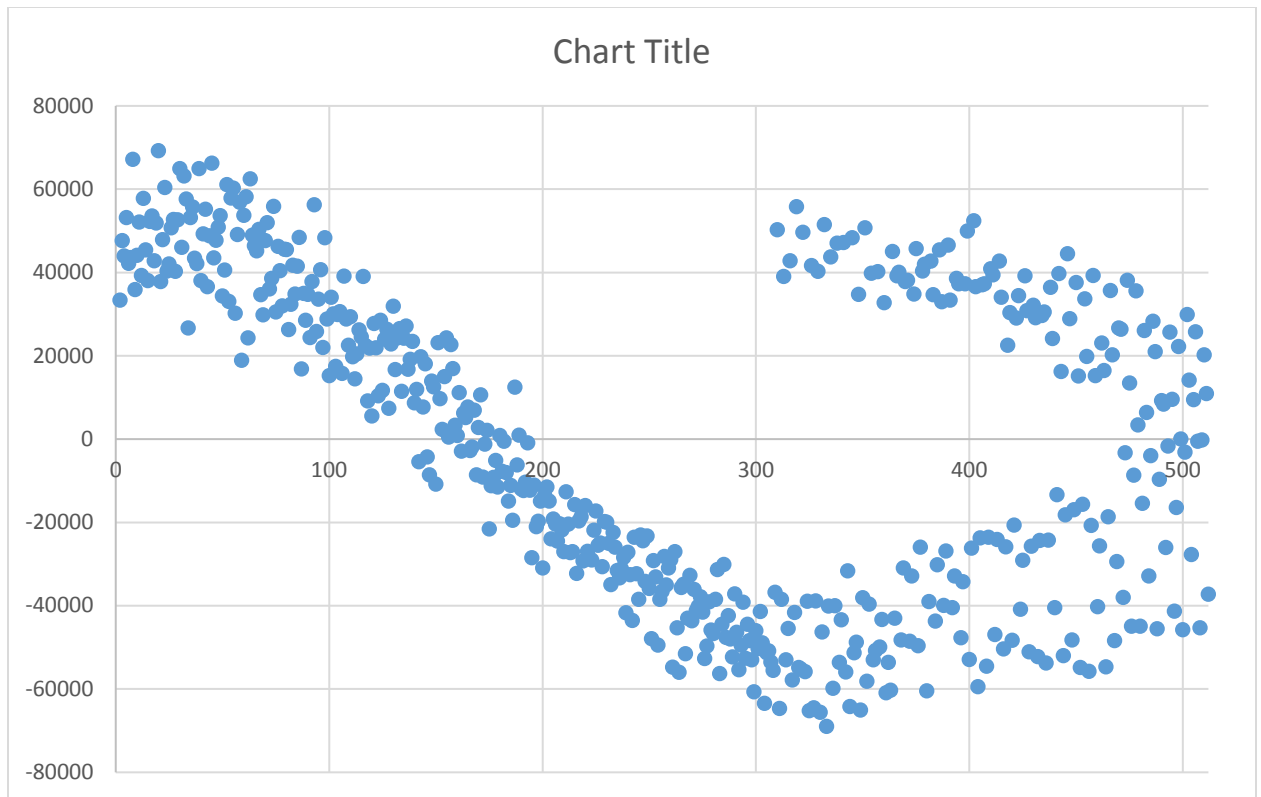
Processor 3.3 GHz intel Core i5

Memory 16 GB 1600 MHz DDR3

Graphics AMD Radeon R9 M290 2048 MB

2. Show the Sums{1} ... Sums[512] vs. shift scatterplot



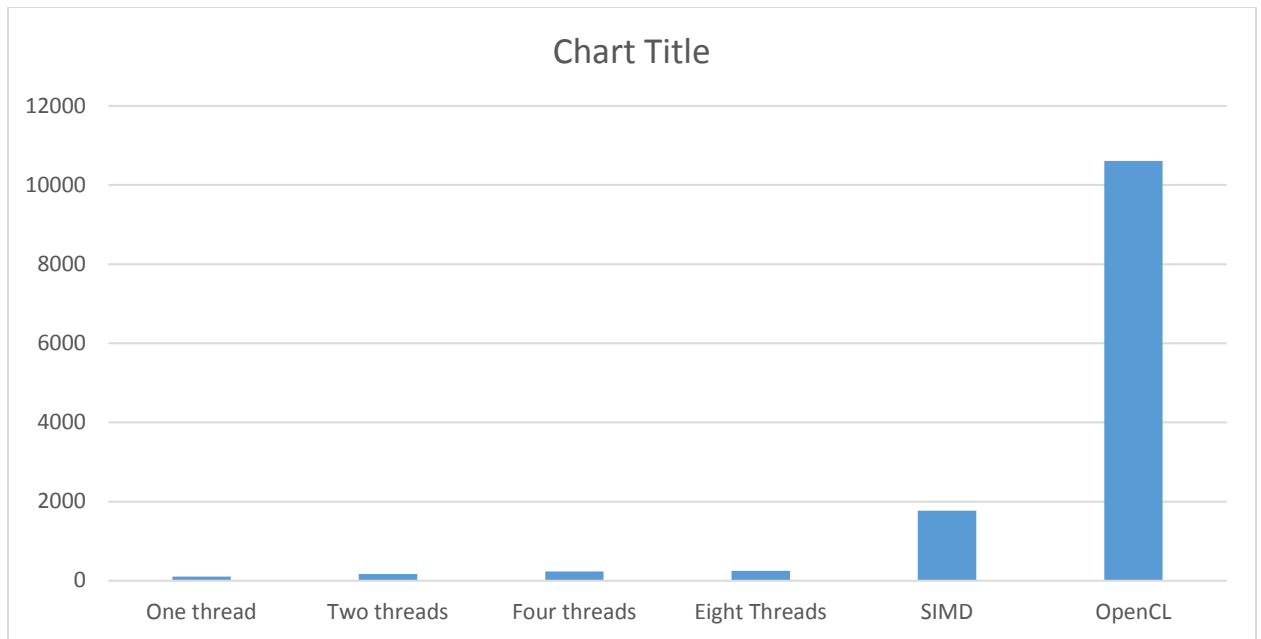


Oddly enough I had two completely different scatter plots. The first graph is from printing the Sums[shift] to the console. I was manually entering in the results when I accidentally hit enter, and losing the console window. The second graph is from printing the Sums[shift] results to a file named "file.txt." I ran this test a few times and received the same results. It's unique that a small change, which has no direct impact on the program output.

3. State what the hidden sine-wave period is, i.e., at what multiples of *shift* are you seeing maxima in the graph?

For both graphs my sine wave ranged between -1 and 1.

4. What patterns are you seeing in the performance bar chart? Which of the four tests runs fastest, next fastest, etc.? By a little, or by a lot?



As you can see above OpenCL easily outperforms OpenMP and SIMD. This bar graph is a short summary of what has been discussed in this class: no parallelism will be outperformed by parallelism, SIMD will outperform parallelism, and OpenCL will easily outperform SIMD, parallelism, and no parallelism. In addition, when OpenCL is removed there is a big difference between one thread and two threads. Also, there is a big difference between OpenMP and SIMD

5. Why do you think the performances work this way?

I think what we are seeing is the efficiencies of the different forms of parallel computing. OpenMP using multiple threads to process data, which is much faster than a single thread. SIMD uses multiple processing elements that perform the same operation on multiple data points simultaneously (Wikimedia Foundation, Inc, 2017). Finally, OpenCL uses CPUs and GPUs to accelerate the processing of information

Bibliography

Wikimedia Foundation, Inc. (2017, June 9). *SIMD*. Retrieved from Wikipedia.org:
<https://en.wikipedia.org/wiki/SIMD>