Project 7B – Autocorrelation using CPU OpenMP, CPU SIMD, GPU OpenCL

Armand Parajon (ONID: Parajona)

6/11/2018

Getting Started

I conducted project 7B on both the flip and rabbit machines. The trials for OpenMP and SIMD were run on flip3. The trials for OpenCL were run on rabbit.

The following files are included:

- openMP_run.cpp – used to run the trials for OpenMP
- simd_run.cpp, simd.p5.cpp, simd.p5.h – used to run the trials for SIMD
- /CL folder, OpenCL_run.cpp, autoCorrelate.cl – used to run trials for OpenCL
- p7start – file used to compile each of the three above programs. They must be executed on their own (instructions to follow)
- openCL_results.csv, openMP_8threads.csv, openMP_onethread.csv, simd_results.csv open – raw output data of last run
- p7_data.xlsx – Compiled data of above raw output, formatted with graphs

To run the code, first run “chmod +x p7start”. Then run “p7start”. This will compile all 3 of the programs. Now they must be run on the correct environment:

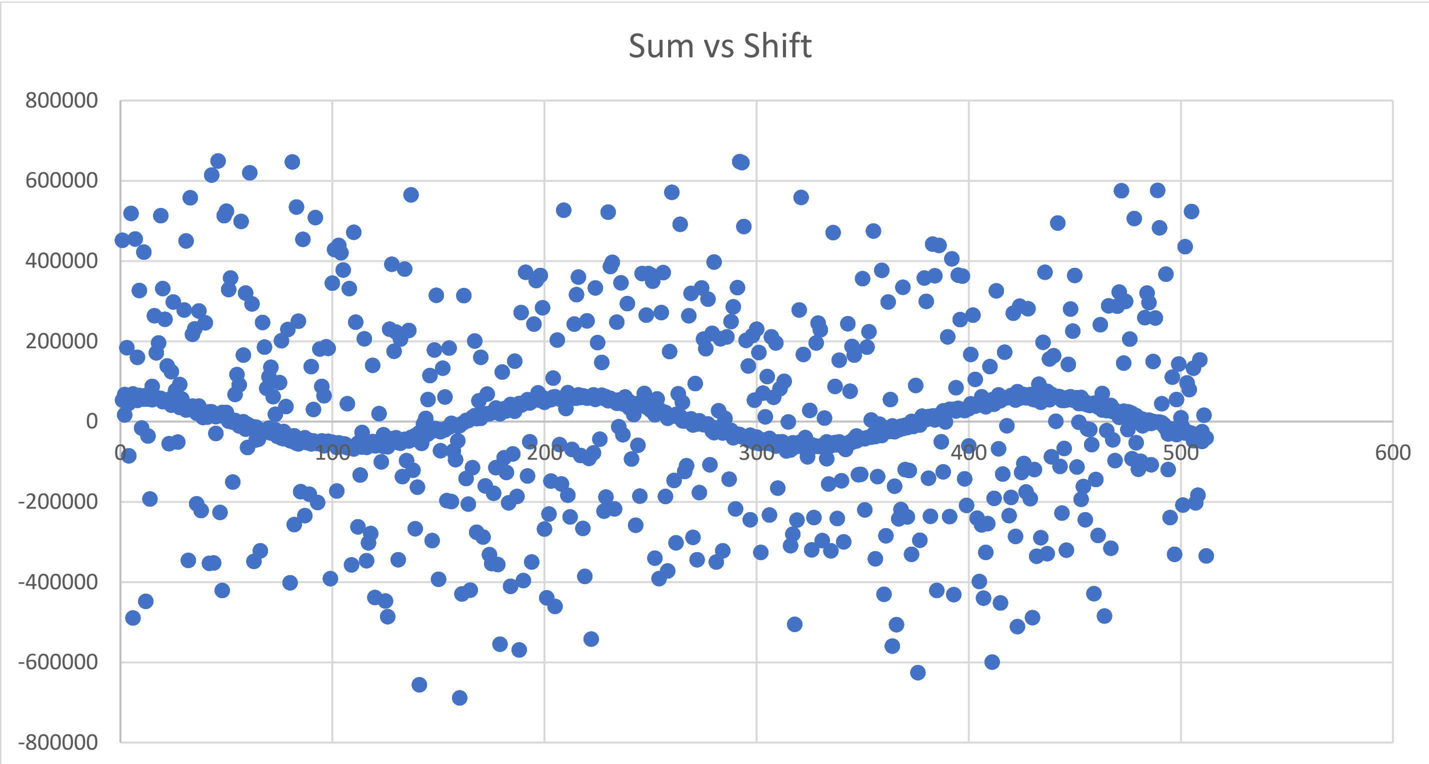
1) run ./openMP_run on any FLIP server
2) run ./simd_run pn any FLIP server
3) run ./openCL_run on rabbit

Running each program will yield a csv file that contains performance and sum/shift data:

Threads	Peak Performance (Mega AutoCors Per Second)	Average Performance (Mega AutoCors Per Second)
1	345.366786	332.093
Shift	Sum	
0	1074569344	
1	451743.625	
2	16377.09863	
3	183895.1875	
4	-85338.27344	
5	518317.3125	
6	-489273.4375	
7	455001.8125	
8	160257.4531	
9	326495.9375	
10	-15737.81348	

Figure 1:Snippet of raw csv output

<u>Scatterplot for Sums vs. Shifts</u>



From looking at the graph above (discounting the noise), it appears that the hidden **sine-wave period** is approximately **200** shift units. The maxima appear to be close to 0, 200, and 400.

Note about data and graph:

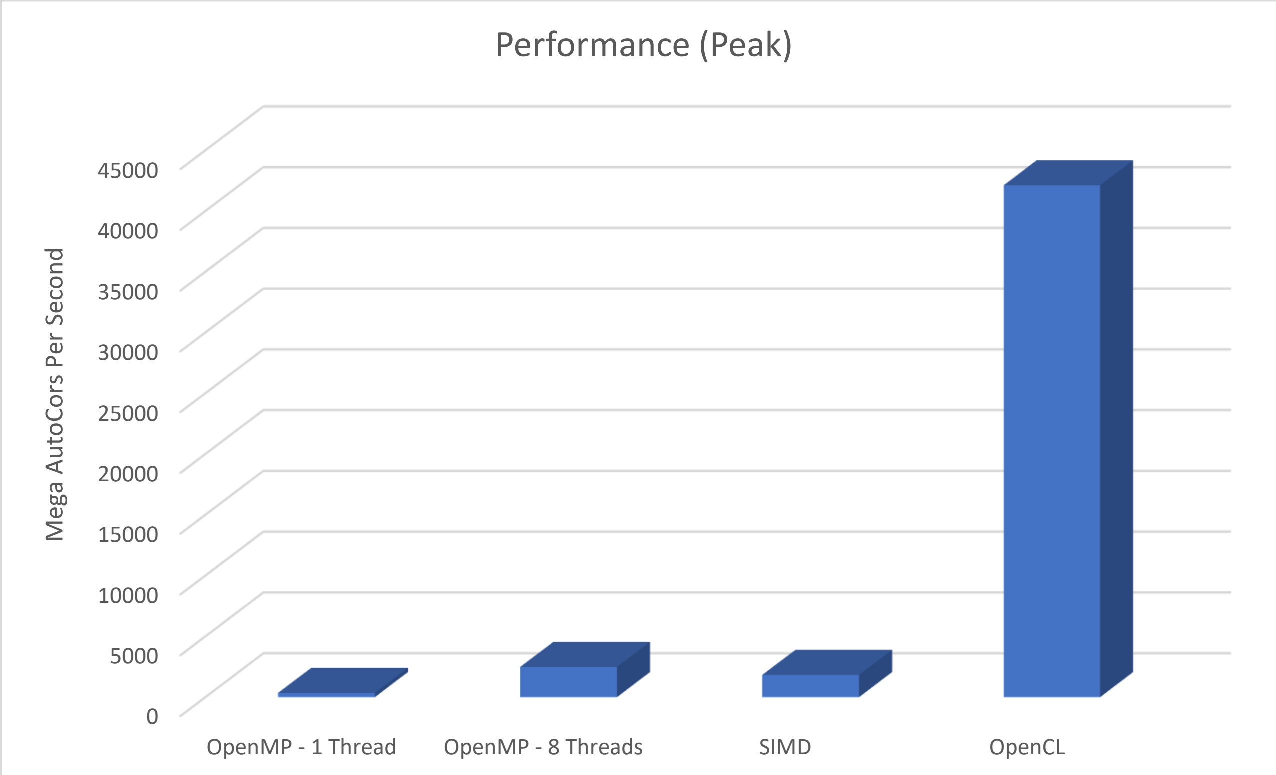
This data was created by taking the sum data for shifts 1 through 512 for OpenMP (run for 1 and 8 threads), SIMD and OpenCL, with a total of 4 compiled data sets. This can be found in the tab "Scatterplot" of p7_data.xlsx.

There is noise around the sine curve due to inconsistent values when running OpenCL:

451743.625	451743.6	451743.3	53185.1
16377.09863	16377.1	16378.56	67181.57
183895.1875	183895.2	183895.5	52135.95
-85338.27344	-85338.3	-85339.9	45437.37
518317.3125	518317.3	518317.9	53552.27

From the data snippet above, the columns represent OpenMP (1 thread), OpenMP (8 threads), SIMD, and OpenCL. The first three are relatively close, while OpenCL appears to be off. A discussion post was raised about this issue, and it may be related to rabbit.

<u>Performance</u>



Raw Data:

Type	Peak Performance
OpenMP - 1 Thread	345.366786
OpenMP - 8 Threads	2476.644029
SIMD	1825.134928
OpenCL	42119.08626

Patterns:

1. OpenCL's performance is significantly faster than the other three
2. Using multi-threads in OpenMP is faster than using a single thread
3. OpenMP with 8 threads is faster than SIMD

Order of performance speeds (fastest to slowest):

OpenCL >> OpenMP – 8 Threads > SIMD > OpenMP – 1 Thread

Pattern Explanation:

1. It's clear that GPU computing is best suited for this task. The large amount of available CUDA cores allows for parallelism on a massive scale, and since the mathematical operations and code logic involved

in auto correlation is simple and short, GPU computing with OpenCL significantly outperforms all the other methods.

2. This is expected behavior, since adding more threads tends to result in faster performance. This has been a common theme throughout the projects in this class, and this particular dataset is no exception because the data set and overall task size is large enough to make the performance benefits of using multi-threading outweigh the cost of using multi-threading at all.
3. SIMD by itself carries out vectorized operations using 4 floats at a time, which should in theory result in a 4.0 speedup over single instruction processing. But this is SIMD without leveraging the parallelism of multi-threads via OpenMP. With 8 threads in OpenMP, a theoretical 8.0 speedup should be faster than a theoretical 4.0 speedup of SIMD. In reality these speedup limits would be lower, but there is in fact evidence from this data set that OpenMP with 8 threads is faster than SIMD by itself, which thus supports the theoretical speedup disparity.