

Tanya Khemani

932988007

CS 575

Project #7B: Autocorrelation using CPU OpenMP, CPU SIMD, and GPU OpenCL

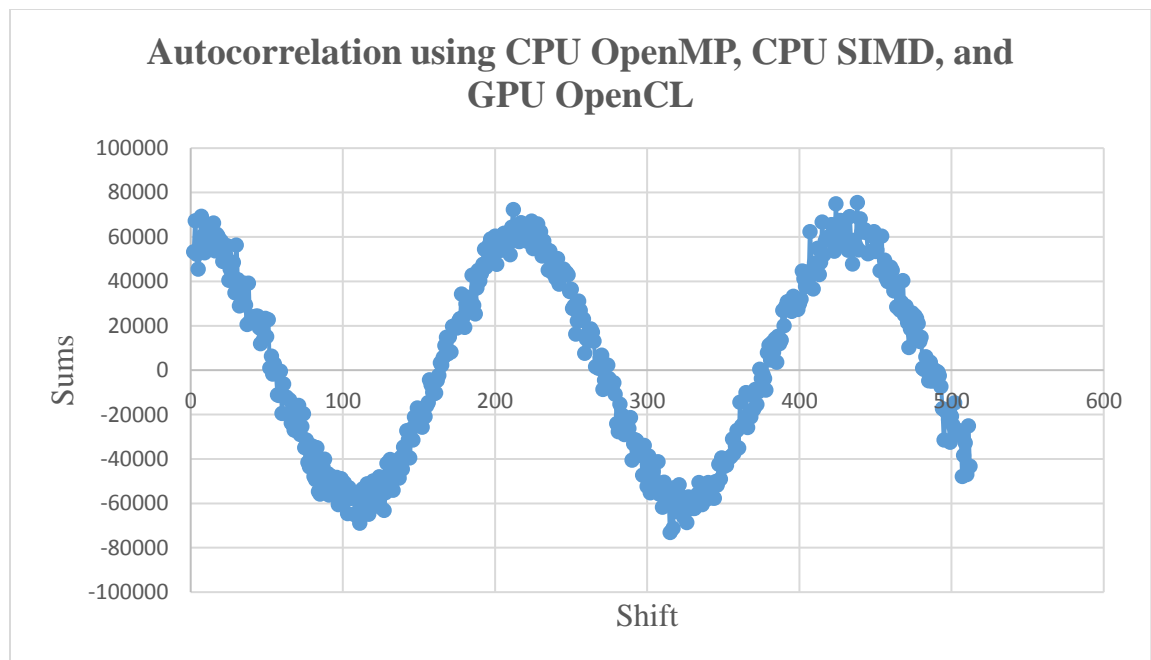
1. What machines you ran this on?

OpenMP with 1 and 4 threads on Rabbit machine

SIMD on MacBook Pro (on my system)

OpenCL on Rabbit machine

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



Tanya Khemani

932988007

CS 575

Project #7B: Autocorrelation using CPU OpenMP, CPU SIMD, and GPU OpenCL

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

It seems as if the hidden sine-wave period is 211 which means that we are getting maxima for 211 multiples of shift.

Talking about autocorrelation, also known as **serial correlation**, is the correlation of a signal with a delayed copy of itself as a function of delay. Informally, it is the similarity between observations as a function of the time lag between them. The analysis of autocorrelation is a mathematical tool for finding repeating patterns, such as the presence of a periodic signal obscured by noise, or identifying the missing fundamental frequency in a signal implied by its harmonic frequencies. It is often used in signal processing for analyzing functions or series of values, such as time domain signals. Also, most importantly, the technique of autocorrelation is used to show hidden sine waves in signals.

This implies that if there is a secret harmonic frequency or a hidden sine-wave period, the point where the period of the harmonic signal is, signifies the maxima on the graph whereas the point where the period of the harmonic signal is half, signifies the minima.

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?

The performance was tested for OpenMP with 1 thread, OpenMP with 4 threads, SIMD and OpenCL. According to the bar chart, OpenMP with 1 thread results in the slowest performance with 267.172657 Megamultiples/Sec, OpenMP with 4 threads having a performance of 709.175705 Megamultiples/Sec results in a faster performance than OpenMP with 1 thread, SIMD giving a performance of 1452.160418 Megamultiples /Sec results in a much faster performance than OpenMP with 1 thread and OpenMP with 4 threads whereas OpenCL results in the fastest performance with 42078.473570 Magamultiples/second.

Tanya Khemani

932988007

CS 575

Project #7B: Autocorrelation using CPU OpenMP, CPU SIMD, and GPU OpenCL

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

OpenMP with 1 thread is the slowest since it is sequential with only one value calculated at a time giving performance of 267.172657 Megamultiples/Sec.

OpenMP with 4 threads is faster than OpenMP with 1 thread giving performance of 709.175705 Megamultiples/Sec after OpenMP with 1 thread which is 267.172657 Megamultiples/Sec. The reason for this is we are using 4 threads and hence it is 4 times faster than with 1 thread with 4 calculations happening concurrently.

SIMD results in a much faster performance than OpenMP with 1 thread and OpenMP with 4 threads giving the output as 1452.160418 Megamultiples /Sec. This happens because in SIMD, 4 floating point operations happen at the same time and also all are loaded at same time which gives greater performance than OpenMP.

Finally, OpenCL results in the fastest performance 42078.473570 Megamultiples/Sec. The reason for this is that in OpenCL, 1000's of threads process data at a time which gives the highest performance.

