Project #7 – Autocorrelation using MPI

Hyun-Taek Oh
ohhyun@oregonstate.edu
11 June 2024

1. What machine you ran this on?

The program was conducted on a Predator HELIOS 300 (2022).

CPU:	12th Gen Intel(R) Core™ i9-12900H
Motherboard:	Mainboard PH315-55 Intel Ci912900H GN20-E6
Memory:	16GB DDR5 (8GB * 2)
Server:	rabbit.engr.oregonstate.edu

2. Show the Sums [1] … Sums [399] vs. shift scatterplot

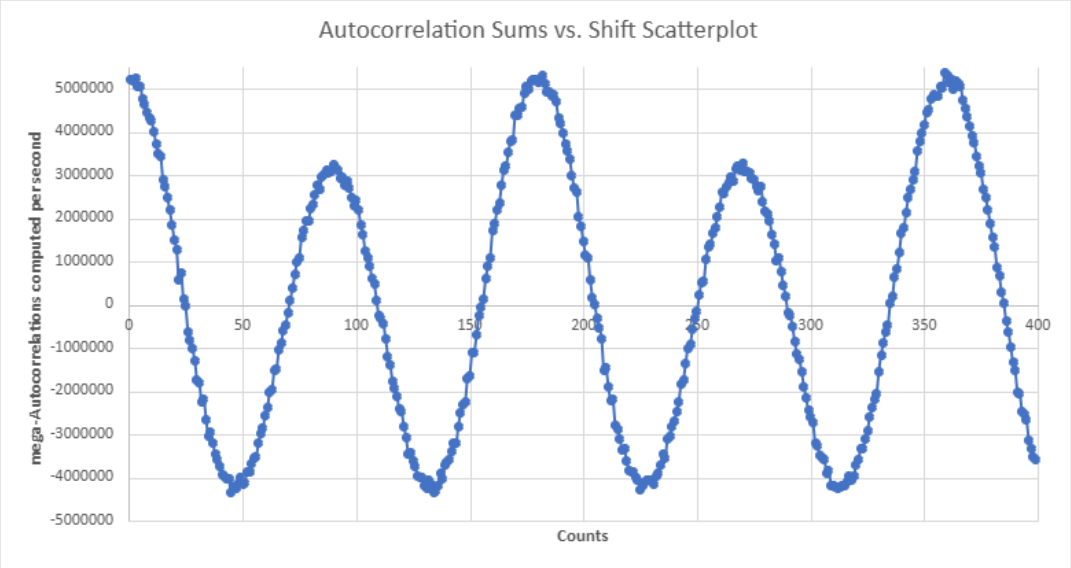


Figure 1. Autocorrelation Sums [*] versus Shift Scatterplot

As shown in Figure 1, there is a sine graph that consist of 399 sums versus shift. The highest amplitude in the sine graph above is 5,366,480 mega-Autocorrelations computed per second (about 5M). On the other hand, the lowest amplitude is -4,359,096 (about -4.3M).

3. **State what the two secret sine-wave periods are.**

As can be seen, there are two secret sine-waves in the signal, one is main and another with twice the period (a harmonic). Three big sine wave humps and two smaller ones. Three large humps are from Count 1 to 50, from Count 130 to 230, and from Count 310 to 399. Two smaller humps are from Count 50 to 130 and from Count 220 to 310. These large and smaller humps are consecutively connected to each other in turn like large->small->large->small->large.

The distances between two successive large humps are from Count 1 to 180 and from Count 180 to 360, which are one period. On the other hand, the distance between a small hump and the next larger hump is from Count 90 to 180 and from Count 270 to 360, which are the other period.

4. **Show your graph of Performance vs. Number of Processors used.**

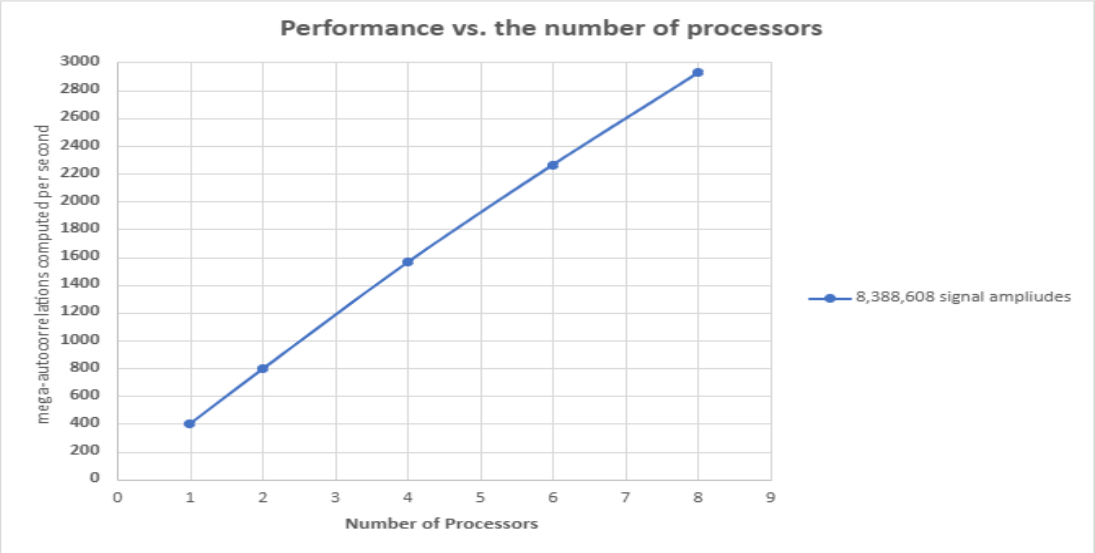


Figure 2. The performance graph versus the number of processors used

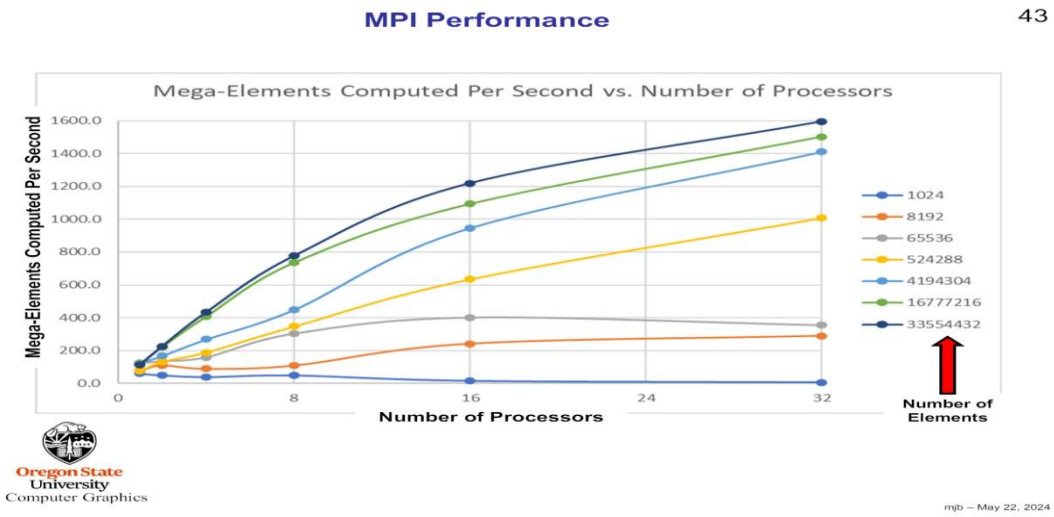
Figure 2 shows the performance graph of each number of processors when signal amplitudes are equal to 8,388,608 (8M). A single processor displays the worst

performance, which is about 400 mega-Autocorrelations computed per second. On the other hand, 8 processors outperform all other processors in the graph, which is about 2,900 mega-Autocorrelations computed per second.

5. What patterns are you seeing in the performance graph?

In general, the performance increases linearly with the number of processors, showing an upward-sloping graph like stock market, and the gap between each processors' performance is quite large. In my guess, if the number of processors increases, the performance also grows up rapidly since the tendency of the graph is guaranteed when the data size is large enough. However, if data size is small enough, the inclination can be changed into the graph with different shapes because of GPUs architecture.

In the Class Notes "MPI", there is a slide showing the MPI performance when the number of elements is different. The graph is below.



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

The reason why the performances work in this way is because of the communication method in MPI. In the MPI communication, each processors handle the data that is divided into smaller part and scattered. Then, the boss processor gathers the yielded results from all the other processors. During the processing data, each processor share values across the boundaries, and this can be used as the compute and communicate ratio for processing data. Due to the efficient work between each processor, the performance displays linear upward-sloping graph.