Project 7: Fourier Analysis Using MPI

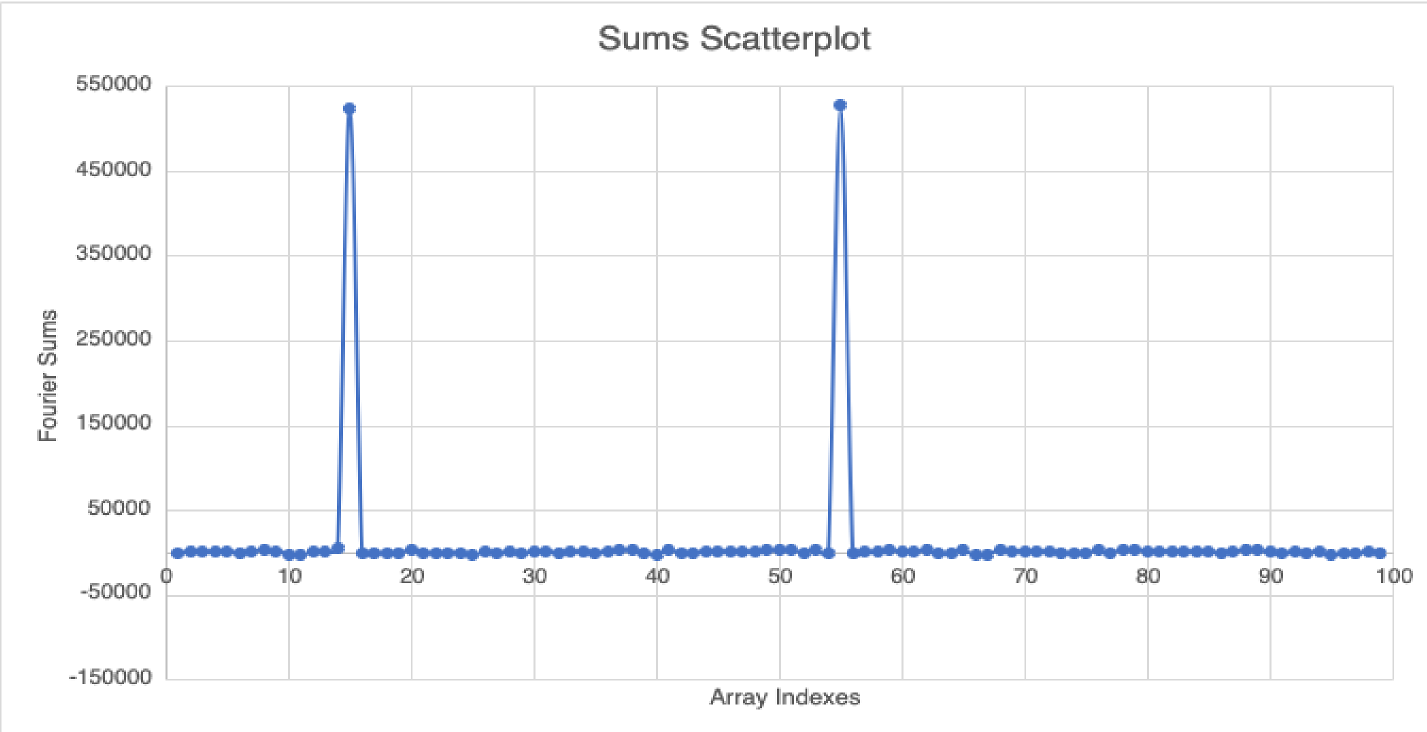
By Caleb Knight

knightca@oregonstate.edu

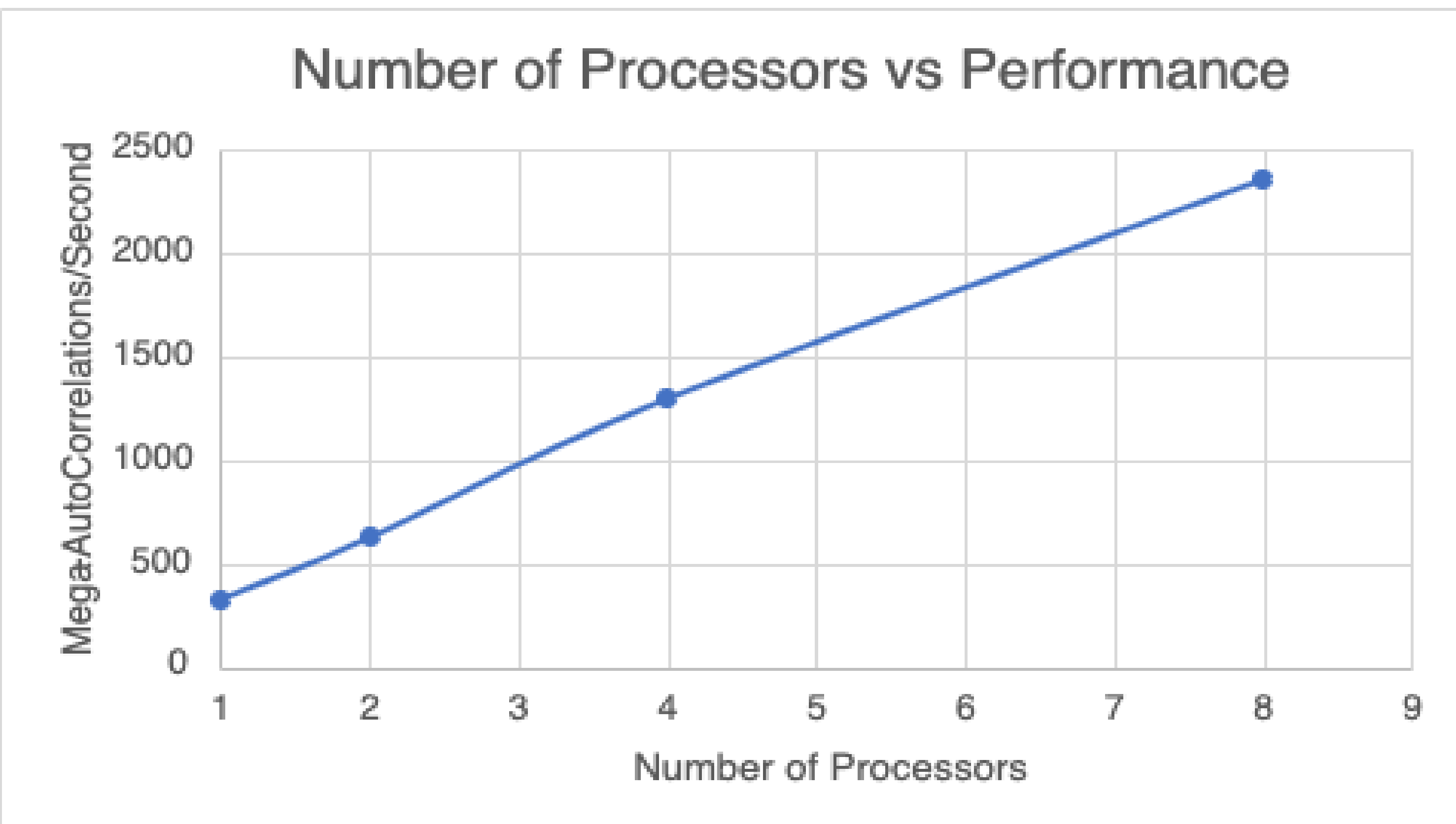
Introduction

This is a technical write-up for practicing a Fourier Analysis using MPI. I ran these tests on OSU's Nvidia DGX-2 system. Each of these DGX servers has 16 NVidia Tesla v100 GPUs, 28TB of SSD disk, two 24-core Intel Xeon 8168 Platinum 2.7GHz CPUs, 1.5TB of DDR4-2666 system memory, all running on the CentOS 7 Linux OS.

Graphs



Sums Scatterplot
550000
450000
350000
250000
150000
50000
-50000
-150000
Fourier Sums
0
10
20
30
40
50
60
70
80
90
100
Array Indexes



Number of Processors vs Performance
2500
2000
1500
1000
500
0
MegaAutoCorrelations/Second
1
2
3
4
5
6
7
8
9
Number of Processors

Observations

The first thing we are going to look at is the “secret sine wave periods”. There are two of these. If we look at our first graph, it is pretty obvious where there are two outliers that draw our attention. Specifically, according to our data, these are at index 15, and 55. These indexes have sums of 523753.7 and 527281.62, respectively.

Next, we will take a look at graph number two. Here, we are given the performance of any given number of processors (1, 2, 4, or 8). This graph looks very linear to me. As we can see, it is still increasing even through the end of the graph. This tells us that we have not reached peak performance yet as we haven’t begun to plateau.

This steady increase is due to our compute-to-communicate ratio. In this small range of processors, the ratios are adequate for efficient parallelism. We are sharing our data at the appropriate rate so that we can compute our values and not leave extra room for computation power while ensuring we don’t share too much data and overload our processors.