Project Assignment 04 Report

CS415

4-11-2017

By: Jervyn Suguitan

Table of Contents:

Description of Assignment

Project 04's main purpose was to teach students about matrix multiplication and how to parallelize the code.. This report will only talk about running matrix multiplication sequentially and how to do it and discussion based on several variables. The timing function that was used to time calculation for this project was MPI_Wtime.

Part 1: Sequential

The first part of the project tasks students in figuring out how to code matrix multiplication sequentially. The first thing to do is to create three 2-D arrays which act as the 3 matrices needed for the multiplication. Two of the matrices will be filled with numbers and act as the matrices that will be multiplied while the third matrix is used to hold the result. The three matrices are put into a function in which the actual matrix multiplication occurs. All matrices are of the same size, as in the width and length of the matrices are equal. The sizes of the matrices are all multiples of 120 due to the fact that for the parallel code, also uses a perfect square amount of processors (4, 9, 16, 25). The matrices are multiples of 120 as 120 can be divided by those perfect square amounts. The runtime of matrix multiplication is O(n^3).

Figure 1, Figure 2 and Figure 3 showcase the runtime of matrix multiplication with multiple different matrix sizes. The only difference between Fig. 1, Fig. 2 and Fig. 3 is that Fig. 1 includes the trials with all matrix sizes while Fig. 2 and Fig. 3 showcase a close look to each set of tests. These matrix sizes were chosen to show times over a large variety of sizes. These graphs show that as more numbers that are needed to be sorted, the longer it takes to sort. To specify all of the trials in Fig. 1 and Fig.2 all use a number range of 0 to 10000. Thus the numbers that are sorted will only be between that range.

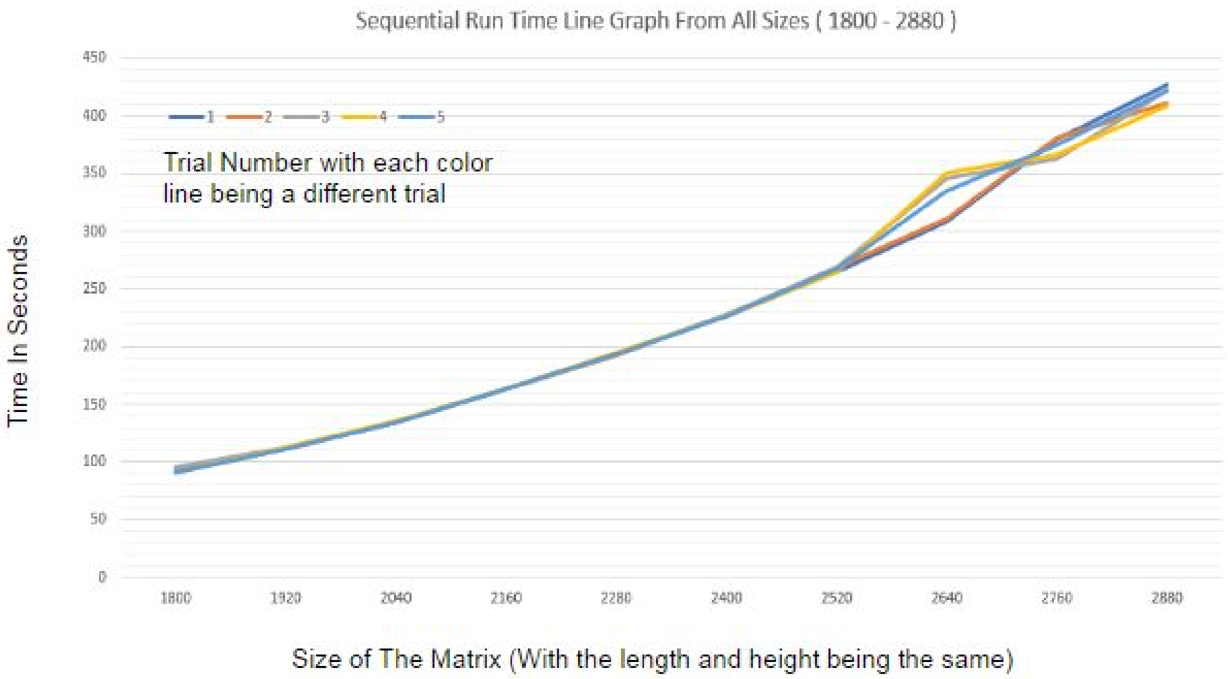

Fig. 1: Graphs that showcase the sequential times for different matrix sizes when doing matrix multiplication. The Y axis shows the time in seconds and the x-axis shows the matrix size. This graph includes times for 2880 by 2880 sized matrix. The times do not vary large amounts since everything is sequential thus there can not be a large speed up. Speed up or slow down may occur due to the processor speed which could be different for every trial. Some of the trials are hard to see as they are behind other lines.

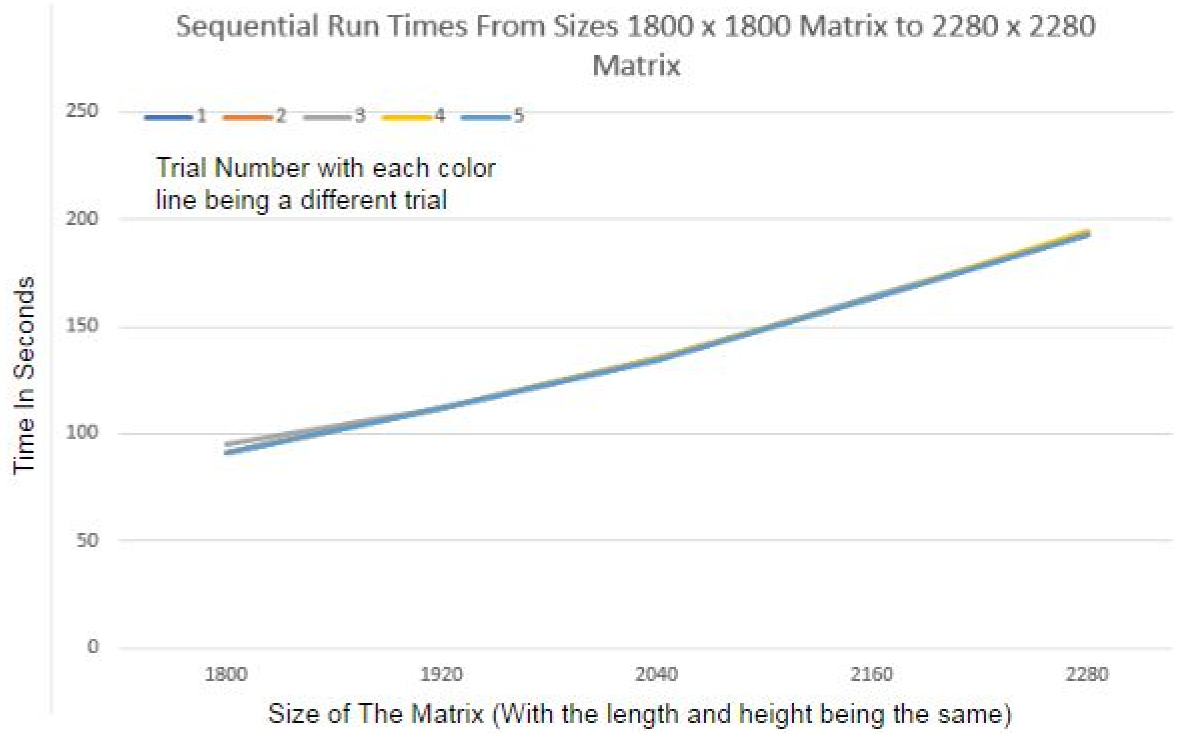

Fig. 2: Graphs that showcase the sequential times for different matrix sizes when doing matrix multiplication. The Y axis shows the time in seconds and the x-axis shows the matrix size. This graph includes times for 1800 to 2280 sized matrix. The times do not vary large amounts since everything is sequential thus there can not be a large speed up. Speed up or slow down may occur due to the processor speed which could be different for every trial. Some of the trials are hard to see as they are behind other lines.

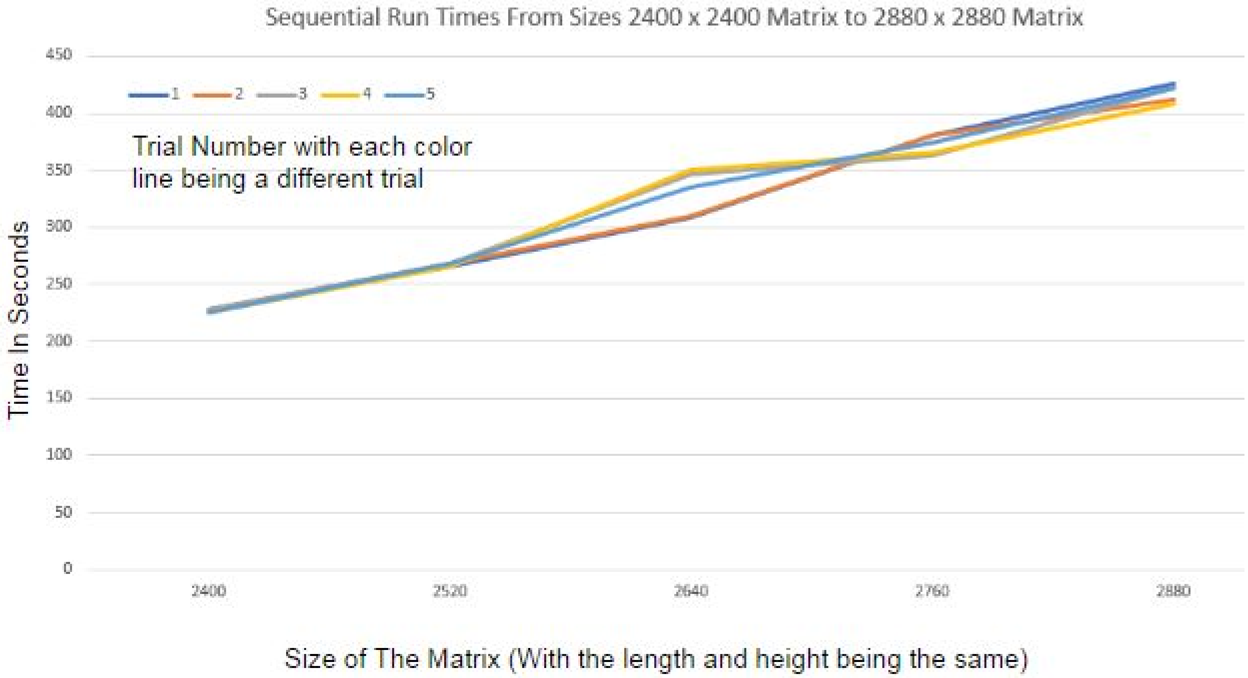

Fig. 3: Graphs that showcase the sequential times for different matrix sizes when doing matrix multiplication. The Y axis shows the time in seconds and the x-axis shows the matrix size. This graph includes times for 2400 to 2880 sized matrix. The times do not vary large amounts since everything is sequential thus there can not be a large speed up. Speed up or slow down may occur due to the processor speed which could be different for every trial. Some of the trials are hard to see as they are behind other lines.