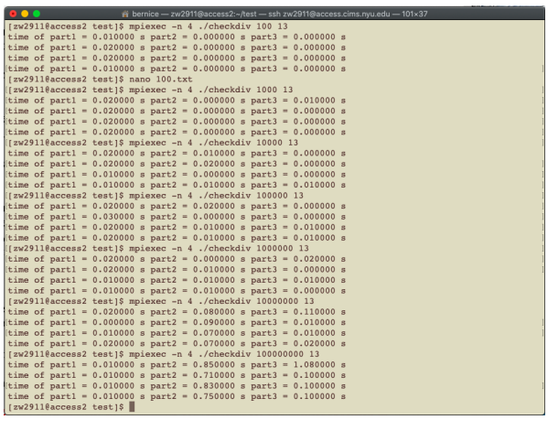
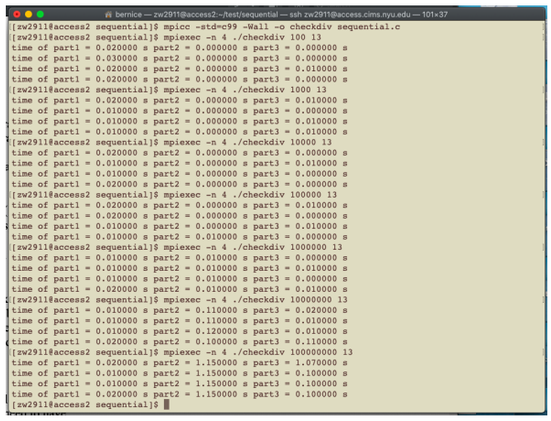
Lab 1 Report

Graph 1:

I tested the program by setting N with 100, 1000, 10000, 1 million, 10 millions and 100 millions. I got the execution time results from both parallel computing and sequential computing by using 4 processors:

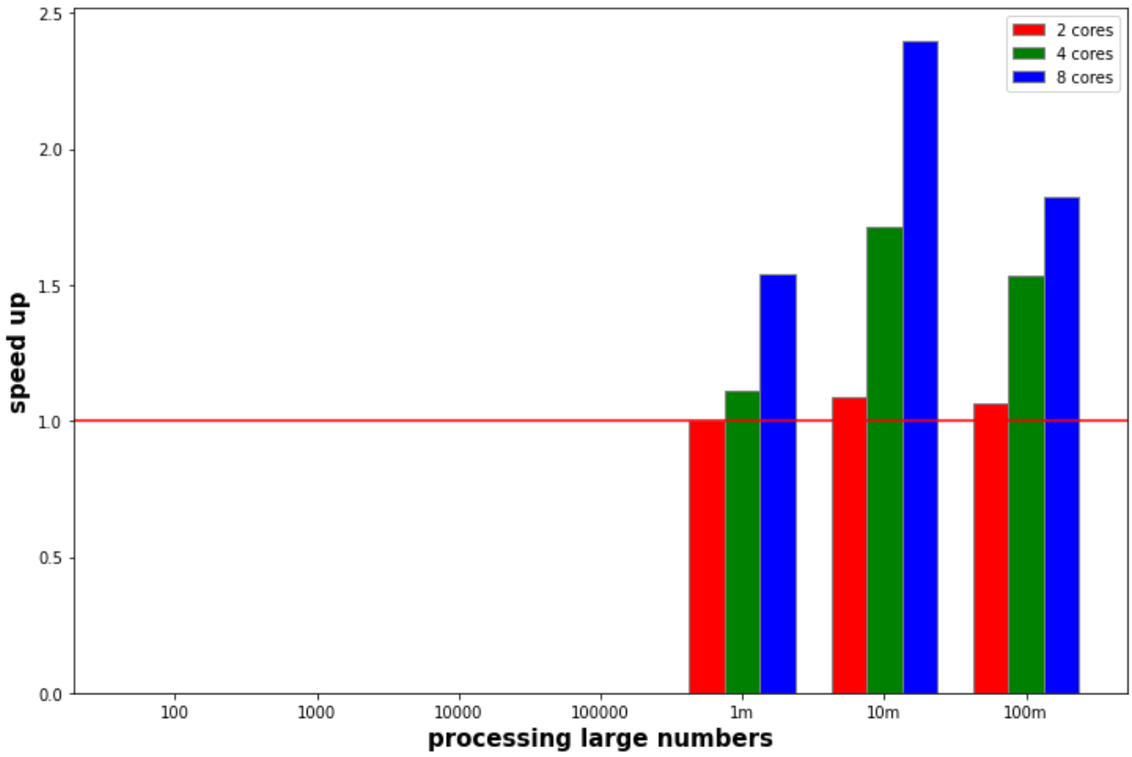




parallel results

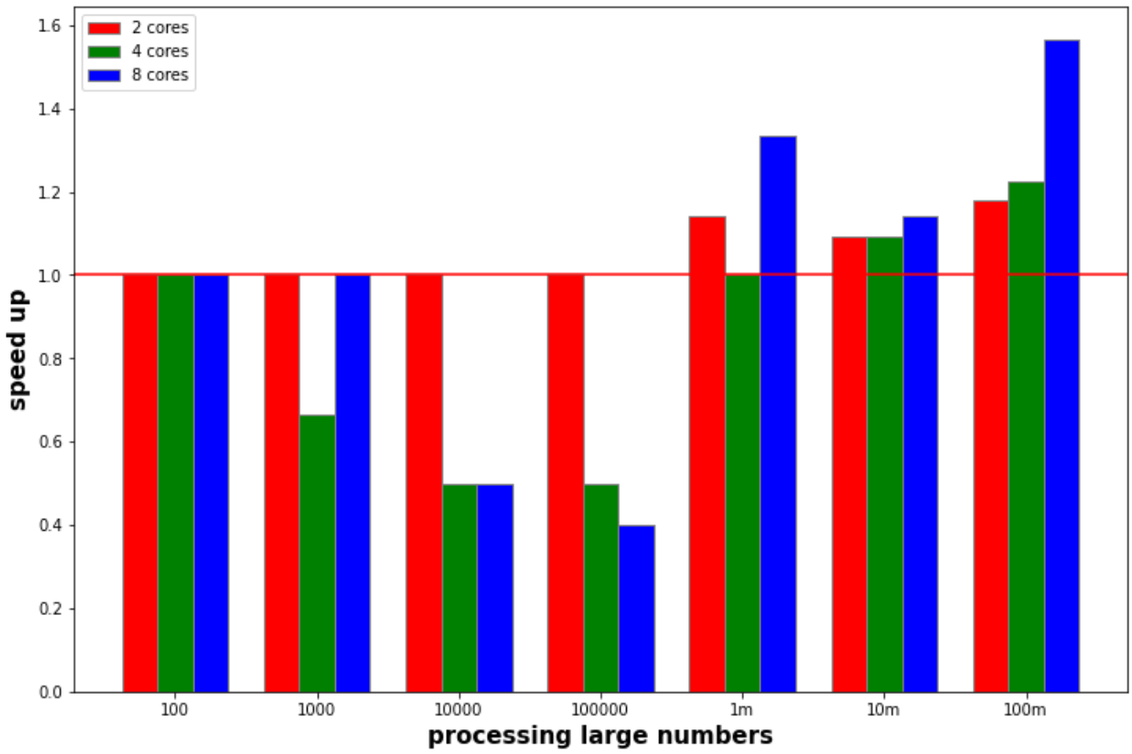
sequential results

In order to get a general idea about the speed up in parallel computing, I continued testing to get execution time by using 2 processors and 8 processors. In this case, I improved my code by applying MPI_reduce() so that get the actual execution time on the screen. Then I plot them in bar chart for comparison.



From the bar chart above, we could see the lack of data in 100, 1000 and 10000 cases, since the precision of measuring execution time under these cases is not enough. Therefore, we could just conclude that there is almost no difference between parallel computing and sequential computing in computing small data. As N keeps increasing to the big data level, the advantages of parallel computing become more and more obvious. Furthermore, with more computation in parallel, the speed up will be higher in general. Thus, parallel computing is more efficient in dealing with big data.

Graph 2:



By calculating the total time that each execution takes, we could see that, parallel computing will even slower the execution when processing small data. And the lag will be more prominent as parallel processors increase, which informs us that distributing and gathering data in parallel computing is very time consuming. While approaching the big data level, we could see that although there is a speed up in execution, the level of speed up decreases compared to the previous graph. Since now the distribution and collection time matter.

Drawbacks:

When doing this lab, the most difficult thing is about getting the most representative execution time. Since the time depends on the laptop, so I will get different output under different status. Therefore, there are multiple possibilities of speed up values. The second difficulty I encountered is about calculating the part2 time in graph1, since the precision is not big enough to detect the actual execution time, so there is a lack of data regarding speed up. I think there is some way to improve the precision and make the graph more informative.