Performance Analysis

Each graph uses 5000 samples and the final result is taken by average:

1. Execution time of 2, 4, 8, 16 and 32 rank run as function of their number of ranks.

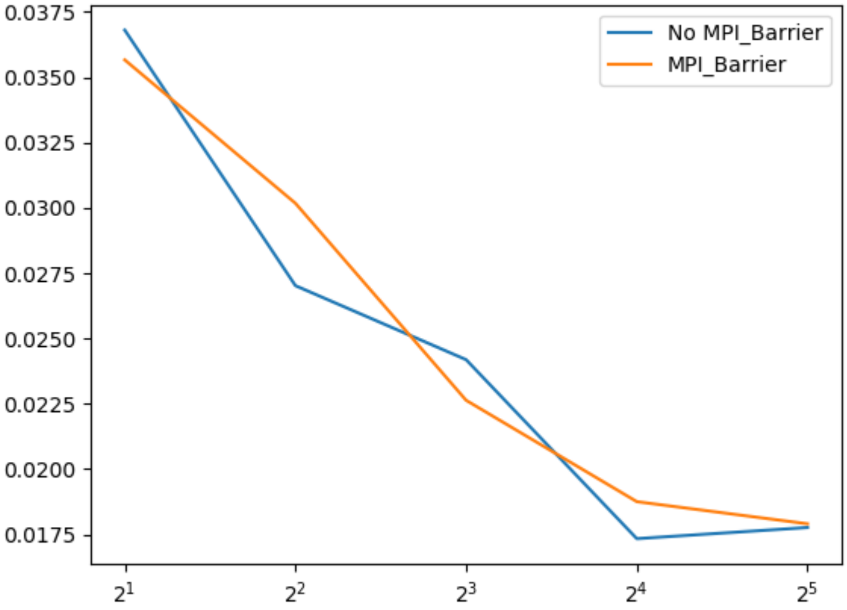

Figure 1: plot the execution time of 2, 4, 8, 16 and 32 rank run as function of their number of ranks (Together with graph 3)

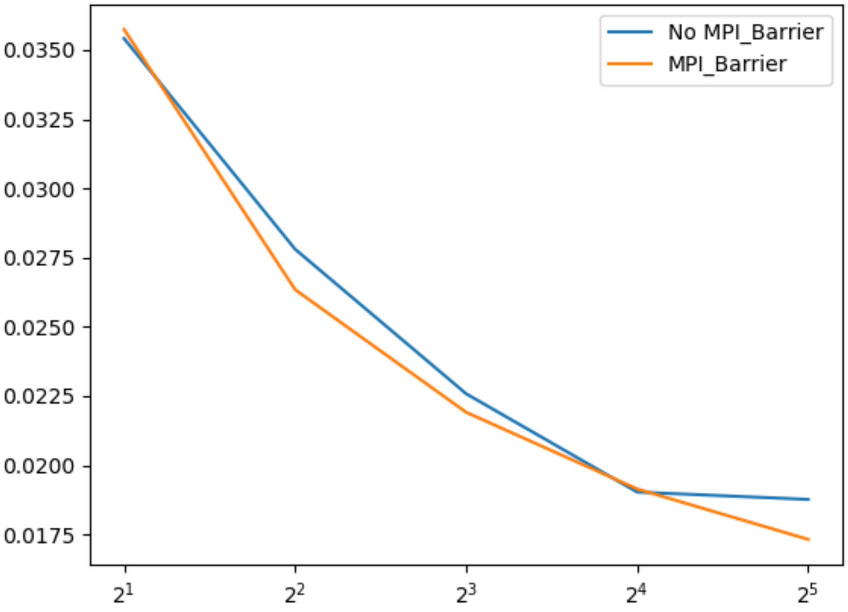

Figure 2 plot the execution time of 2, 4, 8, 16 and 32 rank run as function of their number of ranks (Together with graph 4)

As shown in graphs, the slope for mpi CLA using mpi_barrier is smoother than the slope that is not using mpi_barrier. Also, for performance of rank higher than 16, the program using mpi_barrier is generally better than the one that does not use mpi_barrier. This is because that synchronization makes each rank share relative average system resource, which makes less waste of system resource and the results more consistent when using the same size of ranks. Although in some rare condition, performance without MPI_Barrier is better than the one with MPI_Barrier, overall, the performance with MPI_Barrier is better, especially when in large size of rank.

2. Speedup relative to the execution time of the serial MPI CLA adder of the 2, 4, 8, 16 and 32 rank runs.

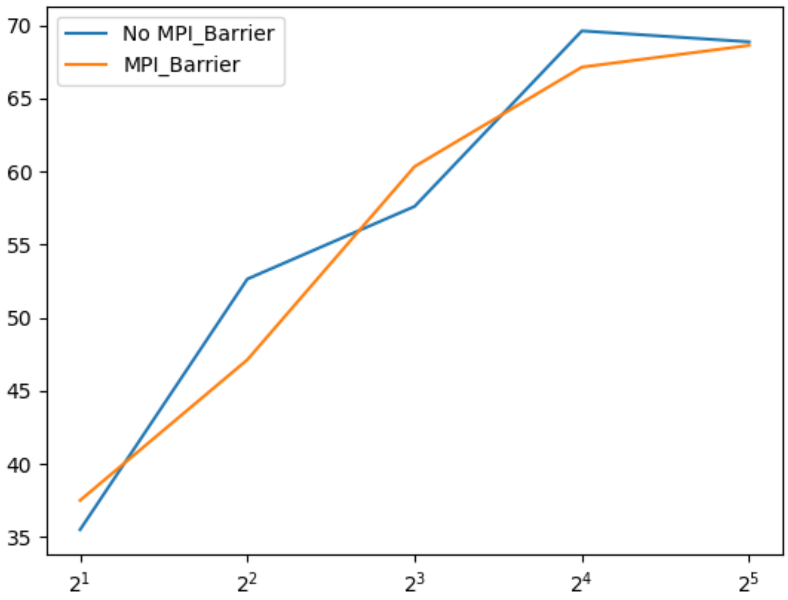

Figure 3: the speedup relative to the execution time of the serial MPI CLA adder of the 2, 4, 8, 16 and 32 rank runs

In general, performance is getting better with larger size of rank because many works can be done in parallel. And the curve with MPI_Barrier is smoother than the one without MPI_Barrier, which is consistent with Figure 1. In the range of size larger than 8, the speedup was relatively small compared with the speedup in smaller size. Because of the Amdahl's Law:

$$Speedup_{enhanced}(f,n) = \frac{1}{(1-f)+\frac{f}{n}}$$

In small size of rank, the slope is almost unchanged because at this point, the fraction of parallel still contribute most of the computation time. But when size grows larger, the contribution of speedup becomes smaller and smaller, and the curve become more flat.

3. Speedup of the relative to the execution time of the serial ripple carry adder to the MPI CLA adder running in parallel on 2 thru 32 ranks.

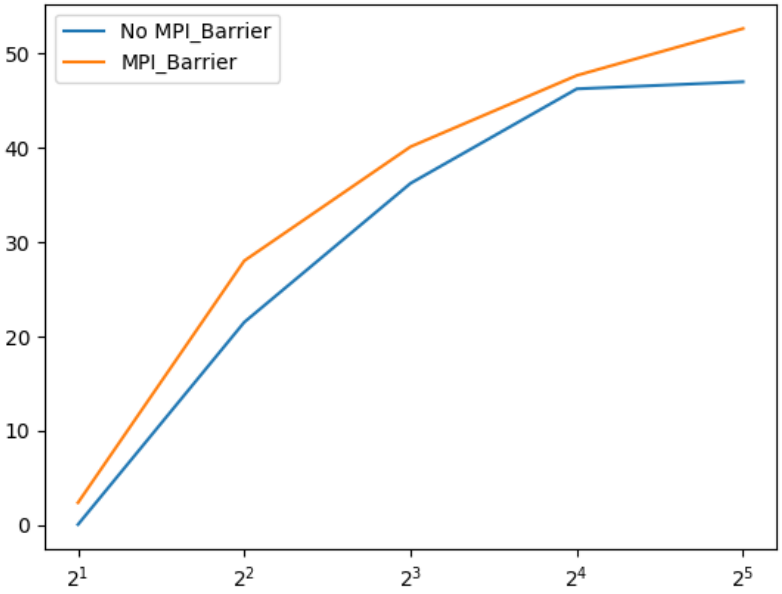

Figure 4: the speedup of the relative to the execution time of the serial ripple carry adder to the MPI CLA adder running in parallel on 2 thru 32 ranks

Figure 4 is almost the same as Figure 3. But the speedup starts at around 0. This is because that, in general, serial ripple adder save some operations, like calculate p, g, and s for groups, compared with single process CLA. Therefore, the performance of serial ripple adder is almost the same as the performance of Rank 2 serial ripple adder.