Results, observations and required experimental setup for assignment 2

March 29, 2021

1 Code explanation

Our code is trying to optimize the collectives namely `MPI_Bcast`, `MPI_Gather`, `MPI_Reduce` and `MPI_Alltoallv`. Code consist of default versions of the above specified collectives and an optimized version of the same. For the default version we have timed the collective part of the code and for optimized version both collective and sub communicator creation has been timed.Every function (default and optimized) are running 5 times and average of 5 execution time is taken. Further we are running our code for all the configurations mentioned in the assignment text and plotting bar charts with error bars for each collective both versions (default and optimized) and comparing them.

2 Optimizations

We have performed only one optimization i.e we tried to minimize the number of collective calls from a rank of a node to other ranks residing in different nodes. We did this by making a sub communicator out of all the ranks from a node and for all the nodes. Further we made another sub communicator comprising of all the 0^{th} rank of newly formed sub communicators to enable inter node communication.
We did this optimization because in `MPI_Bcast`, `MPI_Gather` and `MPI_Reduce` there is one rank(i.e. root) which is doing most of the work and is overloaded and hence performance can be improved by distributing the load of root process. So if we make the above specified optimization then root process's work is distributed into other processes from different nodes and thus it should improve the performance.

3 Generated Plots

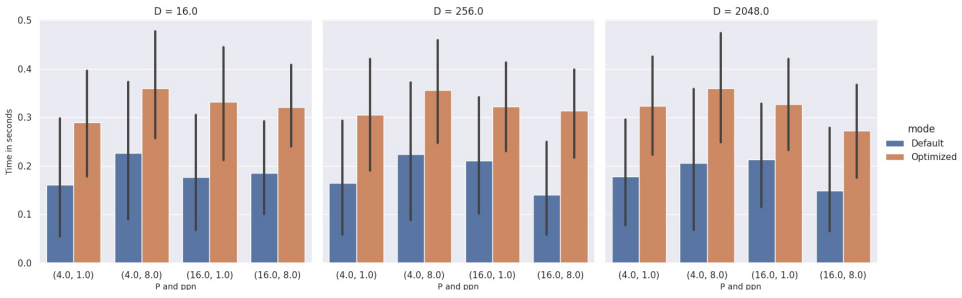

Figure 1: plot_Bcast including sub-communicator creation time

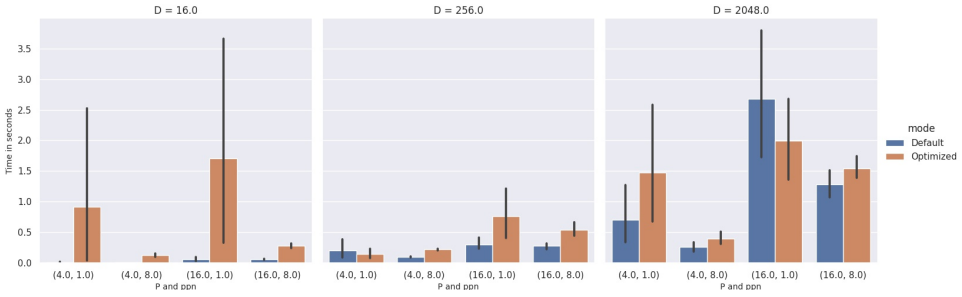

Figure 2: plot_Gather including sub-communicator creation time

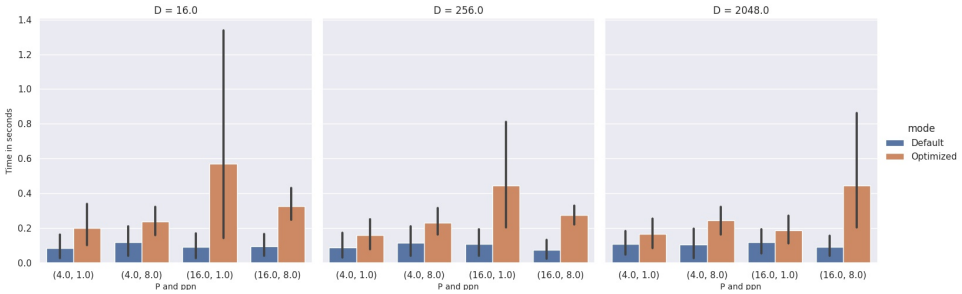

Figure 3: plot_Reduce including sub-communicator creation time

4 Observation

We can see from the plots that time taken for optimized version of the collective are greater than the default in many configurations. This may be because of overhead of sub communicator creation time. Therefore our optimized version of above collective does not perform as intended.

5 Additional testing

In order to understand how the sub-communicator creation time can hamper the overall execution time we tried to exclude sub communicator creation time. Following plots show the generated output.

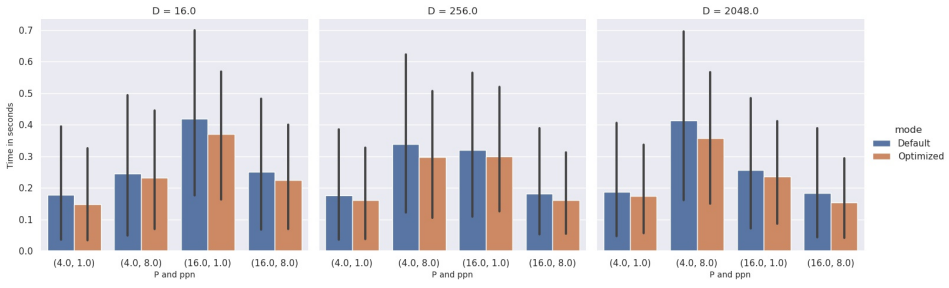

Figure 4: plot_Bcast excluding sub-communicator creation time

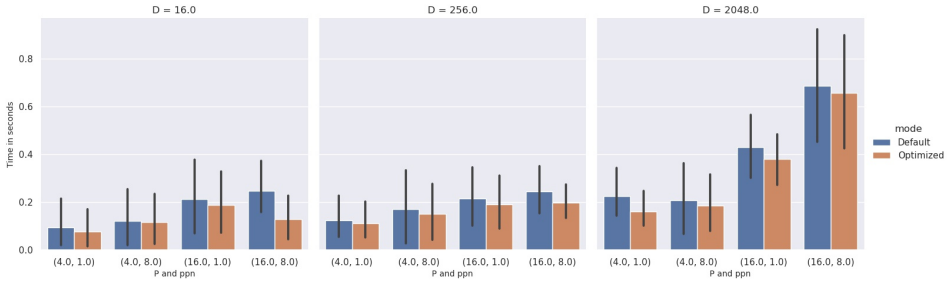

Figure 5: plot_Gather excluding sub-communicator creation time

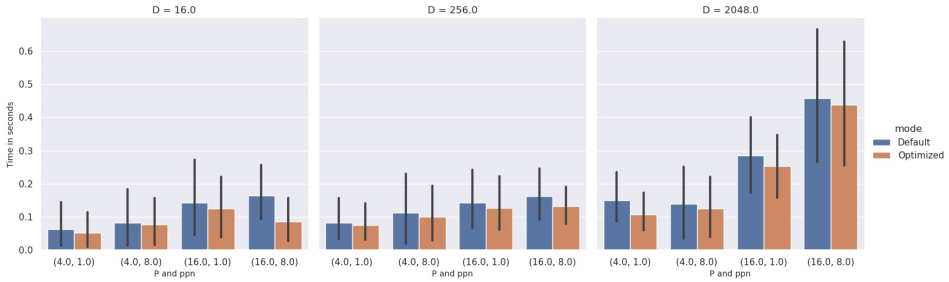

Figure 6: plot_Reduce excluding sub-communicator creation time

6 Running the code

To run the job script use the following command: `sh run.sh` This script compiles `src.c` code and generated the `plot_X.csv` files and `plot.py` uses these files to generate bar charts with error bars named as `plot_X.jpg` for X in `{Bcast, Reduce, Gather}`.

7 Experimental setup

Following are the dependencies:

1. `seaborn` - to plot the bar charts with error bars.
2. `pandas,numpy` - for data frame creation
3. We have used `script.py` to generate host file on-the-fly.