CS633A Parallel Computing - Assignment 3

April 21, 2021

Submitted By:

Akash Patel (20111007)

Prasoon Sahu (20111042)

1 How to Use:

First Create hostfile then run the program.

- bash NodeAllocator.sh
- python3 CreateHostFile.py $ppn
- make
- mpirun -np X -f hostfile ./code tdata.csv

2 Files:

- **Makefile:** Compiles all required files.
- **src.c:** Contains all Mpi code.
- **run.sh:** Script to run the complete program in one go.
- **plot.py:** Script to plot the graph, use python version 2 to run it.
- **NodeAllocator.sh:** Checks the available nodes and saves their information in Data_Temp.txt file.
- **CreateHostfile.py:** Takes Data_temp.txt file as input and produces hostfile. It arranges the nodes in the hostfile also takes care of load on the nodes.
- **hostfile:** Host file to be used by MPICH program.
- **output.txt:** Output File.

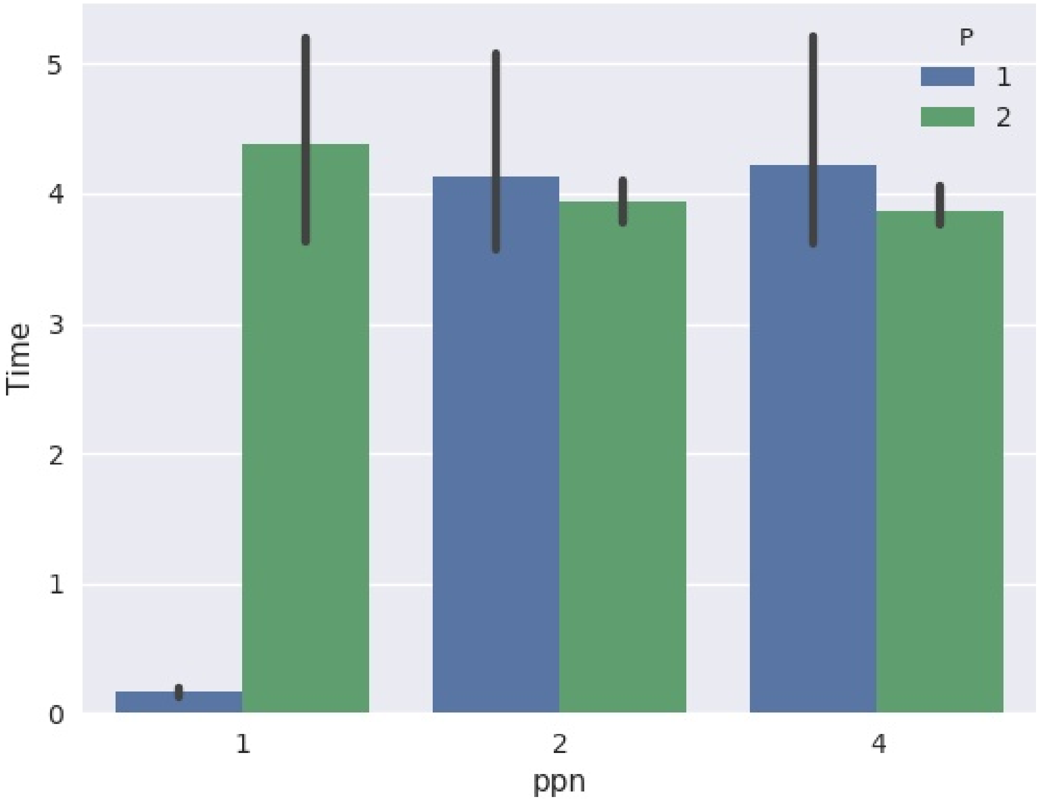
- **temp**$_o utput.txt : Storesdataforplottingpurpose.Manuallycleanitbeforeplottingthefigure.Deleterowsthaton$

3 Program Documentation:

3.1 WorkFlow

- **File Reading:** File is read by Rank-0 Process and stored it in linked list(Because we don't know the number of rows in the file). After reading all rows the linked list is converted into Matrix to easily transfer the data to other ranks.
- **Data Transfer:** Every process gets equal portion of data. Rows are divided equally to each process this reduces overhead of transfer after computation phase. Hence we divided data row wise.
- **Computation:** Every process calculate min value in every column and then stores the result in new array named result.
- **Reduce:** Minimum value in reduced to rank-0, and rank-0 stores array containing minimum of all the columns.
- **Computation:** Rank-0 does all other remaining computations and tasks, i.e. writing in file, calculating maximum time etc.

4 Observation



- **P = 1, ppn = 1 :** In this case there is no overhead due to data transfer hence the time was only due to computation.
- **P = 2, ppn = 1 :** Here two different nodes have one-one process hence computation overhead in each node in more than other also due to different nodes Transfer overhead is also more.

- **P = 1, ppn = 2 :** In this there are two processes both are running on same node. hence there in only computation overhead and less transfer overhead.
- **P = 2, ppn = 2 :** In this there are four processes. Here computation overhead is decreased but transfer overhead is still there.
- Computation overhead is degrading the speed of program more than Transfer overhead.
- We can speedup the program by increasing number of nodes.
- By increasing process per node will decrease the performance.

5 Problem Faced:

- It was bit tricky to read the csv file in C.
- Had no pre information about the numeber of rows and columns.

6 Experimental Setup

- MPICH should be installed in the system.