Assignment 3, CS633

Saketh Maddamsetty 170612,
Ayush Tharwani, 170201

Directory - Assignment 3
Files

- src.c - source code
- Makefile
- run.py - helper code for execution and plotting
- Readme.pdf - this file

Non standard Libraries : *matplotlib,searborn,numpy,pandas* from python3

Steps to run the code:

1. *python3 run.py*
   make clean, followed by make to compile source, then generates host file using Node Allocator, runs the executable for 10 times for P=1,2 nodes for ppn = 1,2,4 cores per node, outputs the for each (execution_number,P,ppn) type configuration in “outputP{P}ppn{ppn}epoch{execution_num}” and plots the barplots corresponding to each configuration in a single plot named ‘plot.png’.

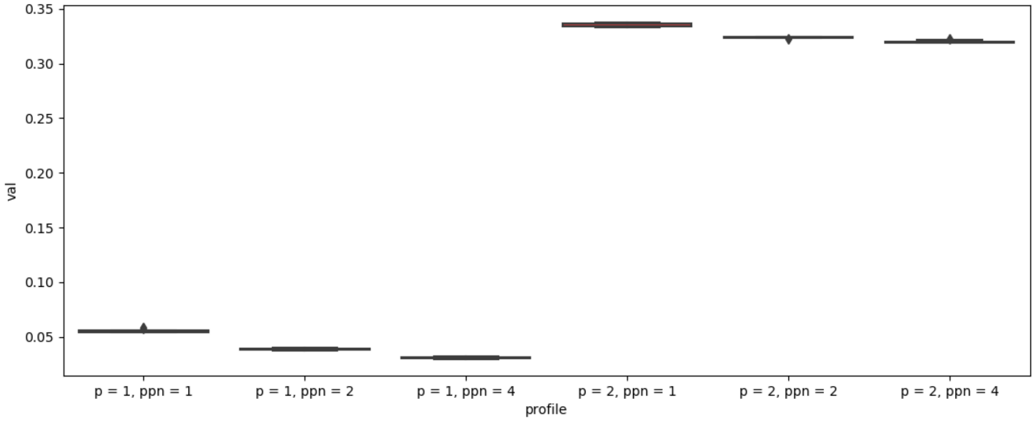
Code Explanation

src.c

- **Readfile :-** given filename of the csv file reads the contents using sequential io.
- **Find_yearwise_min** :- Finds yearwise minimum
- **Find_minimum** :- finds global minimum given yearwise minimum
- Main :-
  - First it reads the csv.
  - It broadcasts the integers “num_rows” and “num_cols”.
  - It scatters data by dividing it along rows
  - The processes individually calculate yearwise minimum.
  - They participate in a reduce call and global yearwise minimum is obtained at root.
  - Global minimum calculated and timing information is printed along with the minimums

The reason for distributing data along rows is to make more equal distribution of data as compared to when distribution is done along columns access ie when individual columns are sent scattered.

Plots



Trends observed in the plots

- The average time for running on 1 node is less as compared to 2 nodes indicating that for this data, communication time is dominating factor of the total time
- For one node we see improvement in performance when ppn is increased from 1 to 4. Speedup = 1.75729828032
- For 2 nodes, we see slight improvement in performance when ppn is increased from 1 to 4. Speedup = 1.05530653987. Reason for small speedup is communication time dominating the total time.
- Strong scaling(increasing number of nodes) is not useful since the communication time is dominating the total time taken.