

CS633A
PARALLEL COMPUTING

Assignment 3

Rohit Raj

20111051

Mani Kant Kumar

20111030

Instructor:

Dr. Preeti Malakar

1 Overview

In this assignment an input file is given which contains temperature readings of several years for several thousand stations (each station is identified by a latitude and a longitude). The aim of the assignment is to find

- year-wise minimum temperature across all the stations for each year
- global minimum across all stations and all years
- the maximum time across all processes

We performed the assignment for the following configurations:

- P (number of nodes) = 1, 2
- ppn (number of cores per node) = 1, 2, 4

2 Experimental Setup

We read the input file from command line argument. We first calculated the number of rows and columns given in the csv file. After getting the count of row and column we are using **MPI_BCAST** to broadcast the values of row and column to each process. Since first two columns contain latitude, longitude and first row contains field names so we reduce the column by 2 and row by 1. We dynamically allocated 2D-matrix with number of rows = col and number of columns = row. For instance, suppose csv file contains following data:

Lat	Lon	1960	1961	1962
44.6547	-88.9592	-5.67	-8.26	-10.7
51.8167	-111.4667	-14.23	-9.64	-12.35
41.4831	-83.711	-0.87	-0.77	-4.72
64.783	170.1166	-25.19	-21.96	-20.88
58.1667	18.4331	-0.94	0.32	1.57
43.18	126.55	-19.1	-19	-14.25

After taking the file as input from command line, we read the file and found row=7, col = 5. We reduce row by 1 and col by 2; we get row = 6 and col = 3. We dynamically allocate a matrix of dimensions 3*6 and assigned the values as shown below:

-5.67	-14.23	-0.87	-25.19	-0.94	-19.1
-8.26	-9.64	-0.77	-21.96	0.32	-19
-10.7	-12.35	-4.72	-20.88	1.57	-14.25

- To generate plot we used python and its libraries like matplotlib.pyplot etc.

3 Instructions to run the code

- ./run.sh
will run the entire code and will generate output in output.txt file.

4 Explanation of code

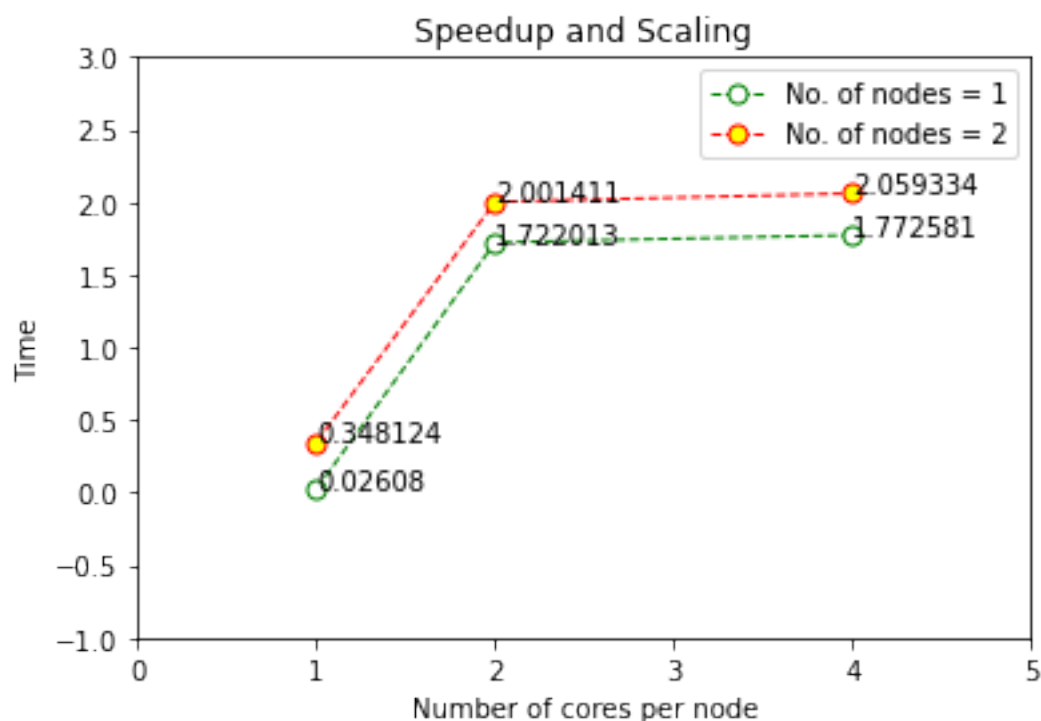
- After dynamically allocating the values in the matrix, we are equally dividing the number of rows to be sent to each process. We are trying to distribute the data to each process equally so that workload can be minimised.
- We are using **MPI_Type_vector** to send vector of rows to other processes using **MPI_Scatter**.
- In some cases when number of years is not exactly divisible by number of processes, we are doing computation for those data at root process since root already contains the whole matrix and we can save communication time for these data by doing computation at root process.
- Each process is computing its partial minima for each row and later these partial minima is collected using **MPI_Gather** at root process.
- Finally we computed the global minima of all partial minima received at root process.

- We started the timer after reading the file and stopped timer before `MPI_Finalize()` and reported the maximum time taken among all processes using **`MPI_Reduce`**.

The functionality of individual files submitted is described below:

- **`run.sh`** : This job script runs all the configurations mentioned in the assignment. It first compiles the code using the “make” command and then generates hostfile using node allocator based on the node status so that jobs never fail. After generation of hostfile, the script run for each configuration.
- **`Makefile`** : compile the code named ”src.c” using the “make” command.
- **`src.c`** : contains the code in C to perform the given experiment.

5 Plots



6 Observations

- We can observe from the plot that when cores per node increases the corresponding time also increases. So we can conclude that communication time is dominating over computation time.
- When number of nodes is 2, it takes slightly more time compared to single node. This is because when number of node is 2, computation time and communication time both adds up to the total time whereas when there is a single node, there is less communication time due to intra-node placement of processes. We also know that the communication time for inter-node communication is more than intra-node communication.
- The global minimum temperature that we got across all stations and all years for given "tdata.csv" file is -53.74 at Latitude = 63.55, Longitude = 143.05 in the year of 1967.
- We first tried sending vector using MPI_Send/Recv but MPI_Scatter gave the minimum time. So we implemented MPI_Scatter in our code.