Parallel Computing Lab
Nilay Ganvit - 200001053
8th September 2022

Lab 4

Minimum element in 2d Array

Code:

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include "mpi.h"

int main(int argc, char **argv)
{
  int np, pid, numworkers, source, dest, rows, offset, i, j, k,
N,answer=999;
  N = atoi(argv[1]);
  double a[N][N];
  MPI_Status status;

  MPI_Init(&argc, &argv);
  MPI_Comm_rank(MPI_COMM_WORLD, &pid);
  MPI_Comm_size(MPI_COMM_WORLD, &np);

  numworkers = np - 1;

  /*---------------------------- master ----------------------------*/
  if (pid == 0)
  {

    //printf("Taking input of matrices of size %dX%d from input files\n",
N, N);
    FILE *input1 = fopen("input_matrix1.txt", "r");


    //printf("Reading first matrix\n");
    for (i = 0; i < N; i++)
    {
      for (j = 0; j < N; j++)
      {
```

```
        fscanf(input1, "%lf", &a[i][j]);
      }
    }



    clock_t start = clock();

    /* send matrix data to the worker tasks */
    rows = N / numworkers;
    offset = 0;

    for (dest = 1; dest <= numworkers; dest++)
    {
      MPI_Send(&offset, 1, MPI_INT, dest, 1, MPI_COMM_WORLD);
      MPI_Send(&rows, 1, MPI_INT, dest, 1, MPI_COMM_WORLD);
      MPI_Send(&a[offset][0], rows * N, MPI_DOUBLE, dest, 1,
MPI_COMM_WORLD);

      offset = offset + rows;
    }

    /* wait for results from all worker tasks */
    for (i = 1; i <= numworkers; i++)
    {
      int curr_ans =  999;
      source = i;
      MPI_Recv(&offset, 1, MPI_INT, source, 2, MPI_COMM_WORLD, &status);
      MPI_Recv(&rows, 1, MPI_INT, source, 2, MPI_COMM_WORLD, &status);
      MPI_Recv(&curr_ans, 1, MPI_INT, source, 2, MPI_COMM_WORLD, &status);
      if(answer>curr_ans){
          answer=curr_ans;
        }
    }

    clock_t end = clock();

  printf("%d\n",answer);
  printf("Time taken in seconds is: %lf\n", ((double)(end - start) /
CLOCKS_PER_SEC));
```

```
  }
  /*---------------------------- worker-----------------------------*/
  if (pid > 0)
  {
    source = 0;
    MPI_Recv(&offset, 1, MPI_INT, source, 1, MPI_COMM_WORLD, &status);
    MPI_Recv(&rows, 1, MPI_INT, source, 1, MPI_COMM_WORLD, &status);
    MPI_Recv(&a, rows * N, MPI_DOUBLE, source, 1, MPI_COMM_WORLD,
&status);

    /* Matrix addition */
    int curr_ans=999;
    for (k = 0; k < N; k++)
      for (i = 0; i < rows; i++)
      {
        if(curr_ans>a[i][k]){
          curr_ans=a[i][k];
        }
      }

    MPI_Send(&offset, 1, MPI_INT, 0, 2, MPI_COMM_WORLD);
    MPI_Send(&rows, 1, MPI_INT, 0, 2, MPI_COMM_WORLD);
    MPI_Send(&curr_ans, 1, MPI_INT, 0, 2, MPI_COMM_WORLD);
  }

  MPI_Finalize();
}   }
```

Input Matrix:

```
1 483 443 118 382 96
2 24 122 493 290 371
3 56 339 136 63 174
4 277 154 189 496 198
5 147 474 477 185 358
```

Output And Time Taken to execute the code:

```
nilay@Nilay-PC:~$ mpicc -o mpi min2darr.c
nilay@Nilay-PC:~$ mpiexec -n 6 ./mpi 5
24
Time taken in seconds is: 0.000256
nilay@Nilay-PC:~$
```