Parallel Computing Lab
Nilay Ganvit - 200001053
18th August 2022

Lab 1
Multiplication
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;
  N = atoi(argv[1]);
  double a[N][N], b[N][N], c[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");
    FILE *input2 = fopen("input_matrix2.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]);
```

```
      }
    }

    printf("Reading second matrix\n");
    for (i = 0; i < N; i++)
    {
      for (j = 0; j < N; j++)
      {
        fscanf(input2, "%lf", &b[i][j]);
      }
    }

    printf("Adding matrices...\n");
    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);
      MPI_Send(&b[0][0], N * N, MPI_DOUBLE, dest, 1, MPI_COMM_WORLD);
      offset = offset + rows;
    }

    /* wait for results from all worker tasks */
    for (i = 1; i <= numworkers; i++)
    {
      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(&c[offset][0], rows * N, MPI_DOUBLE, source, 2,
MPI_COMM_WORLD, &status);
    }

    clock_t end = clock();
```

```
    FILE *output = fopen("resultantMatrix.txt", "w");

    printf("Writing the resultant matrix to the output file\n");
    for (i = 0; i < N; i++)
    {
      for (j = 0; j < N; j++)
        fprintf(output, "%.2f   ", c[i][j]);
      fprintf(output, "\n");
    }

    printf("Done\n");
    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);
    MPI_Recv(&b, N * N, MPI_DOUBLE, source, 1, MPI_COMM_WORLD, &status);

    /* Matrix addition */

    for (i = 0; i < rows; i++)
    {
        for(int j = 0; j < N; j++)
        {
            c[i][j] = 0.0;

            for (k = 0; k < N; k++)
            {
                c[i][j]+= a[i][k] * b[k][j];
            }
        }
```

```
    }


    MPI_Send(&offset, 1, MPI_INT, 0, 2, MPI_COMM_WORLD);
    MPI_Send(&rows, 1, MPI_INT, 0, 2, MPI_COMM_WORLD);
    MPI_Send(&c, rows * N, MPI_DOUBLE, 0, 2, MPI_COMM_WORLD);
  }

  MPI_Finalize();
}
```

Input:

Matrix1:

```
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
```

Matrix2:

```
1 11 83 161 307 99
2 405 36 41 22 418
3 137 293 286 376 82
4 156 432 167 38 496
5 341 315 396 276 56
```

Output:

```
1 293222.00   285875.00   231484.00   243407.00   437515.00
2 288966.00   392978.00   345210.00   308836.00   258414.00
3 225705.00   138726.00   141236.00   126204.00   199390.00
4 236204.00   360554.00   266205.00   232987.00   364397.00
5 409874.00   361716.00   352186.00   340747.00   363607.00
```

Time Taken to execute the code:

```
nilay@Nilay-PC:~$ mpicc -o mpi multiply.c
nilay@Nilay-PC:~$ mpiexec -n 6 ./mpi 5
Taking input of matrices of size 5X5 from input files
Reading first matrix
Reading second matrix
Adding matrices...
Writing the resultant matrix to the output file
Done
Time taken in seconds is: 0.000154
```