Class: Final Year (Computer Science and Engineering)

Year: 2022-23 **Semester:** 1

Course: High Performance Computing Lab

Practical No. 7

Exam Seat No: 2019BTECS00070

Name: Killedar Prathmesh

1. Implement Matrix-Vector Multiplication using MPI. Use different number of processes and analyze the performance.

Code:-

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

// size of matrix
#define N 1000

int main(int argc, char *argv[])
{
int np, rank, numworkers, rows, i, j, k;

// a*b = c
double a[N][N], b[N], c[N];
MPI_Status status;

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

numworkers = np - 1; // total process - 1 ie process with rank 0

// rank with 0 is a master process
int dest, source;
```

```
int tag;
int rows_per_process, extra, offset;

// master process, process with rank = 0
if (rank == 0)
{
printf("Running with %d tasks.\n", np);

// matrix a and b initialization
for (i = 0; i < N; i++)
for (j = 0; j < N; j++)
a[i][j] = 1;

for (i = 0; i < N; i++)
b[i] = 1;

// start time
double start = MPI_Wtime();

// Send matrix data to other worker processes
rows_per_process = N / numworkers;
extra = N % numworkers;

offset = 0;
tag = 1;

// send data to other nodes
for (dest = 1; dest <= numworkers; dest++)
{
rows = (dest <= extra) ? rows_per_process + 1 : rows_per_process;

MPI_Send(&offset, 1, MPI_INT, dest, tag, MPI_COMM_WORLD);
MPI_Send(&rows, 1, MPI_INT, dest, tag, MPI_COMM_WORLD);

MPI_Send(&a[offset][0], rows * N, MPI_DOUBLE, dest, tag,
MPI_COMM_WORLD);
MPI_Send(&b, N, MPI_DOUBLE, dest, tag, MPI_COMM_WORLD);

offset = offset + rows;
}
```

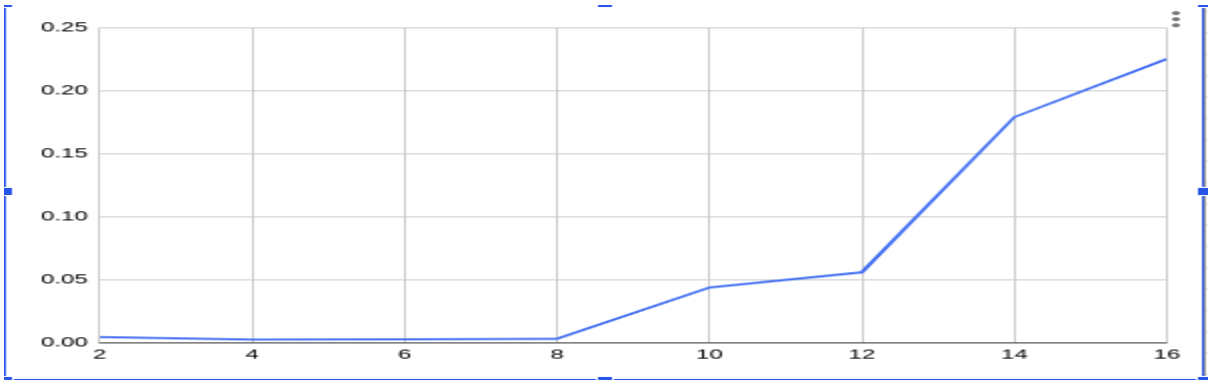
```
// receive data from other nodes and add it to the ans matrix c
tag = 2;
for (i = 1; i <= numworkers; i++)
{
source = i;
MPI_Recv(&offset, 1, MPI_INT, source, tag, MPI_COMM_WORLD, &status);
MPI_Recv(&rows, 1, MPI_INT, source, tag, MPI_COMM_WORLD, &status);
MPI_Recv(&c[offset], N, MPI_DOUBLE, source, tag, MPI_COMM_WORLD,
&status);
}

// print multiplication result
// printf("Result Matrix:\n");
// for (i = 0; i < N; i++)
// {
// printf("%6.2f ", c[i]);
// }

// printf("\n");

double finish = MPI_Wtime();
printf("Done in %f seconds.\n", finish - start); // total time spent
}

// all other process than process with rank = 0
if (rank > 0)
{
tag = 1;
// receive data from process with rank 0
MPI_Recv(&offset, 1, MPI_INT, 0, tag, MPI_COMM_WORLD, &status);
MPI_Recv(&rows, 1, MPI_INT, 0, tag, MPI_COMM_WORLD, &status);
MPI_Recv(&a, rows * N, MPI_DOUBLE, 0, tag, MPI_COMM_WORLD, &status);
MPI_Recv(&b, N, MPI_DOUBLE, 0, tag, MPI_COMM_WORLD, &status);

// calculate multiplication of given rows

for (i = 0; i < rows; i++)
{
```

```
c[i] = 0.0;
for (j = 0; j < N; j++)
c[i] = c[i] + a[i][j] * b[j];
}

// send result back to process with rank 0
tag = 2;
MPI_Send(&offset, 1, MPI_INT, 0, tag, MPI_COMM_WORLD);
MPI_Send(&rows, 1, MPI_INT, 0, tag, MPI_COMM_WORLD);
MPI_Send(&c, N, MPI_DOUBLE, 0, tag, MPI_COMM_WORLD);
}
MPI_Finalize();
}
```

Output:-

```
prathmesh@prathmesh-G3-3500:/media/prathmesh/DATA/sem7/HPC assignment/assignment7$ mpirun -n 2 ./mat_vec_mul.exe
Running with 2 tasks.
Done in 0.004572 seconds.
prathmesh@prathmesh-G3-3500:/media/prathmesh/DATA/sem7/HPC assignment/assignment7$ mpirun -n 4 ./mat_vec_mul.exe
Running with 4 tasks.
Done in 0.002626 seconds.
prathmesh@prathmesh-G3-3500:/media/prathmesh/DATA/sem7/HPC assignment/assignment7$ mpirun -n 6 ./mat_vec_mul.exe
Running with 6 tasks.
Done in 0.002798 seconds.
prathmesh@prathmesh-G3-3500:/media/prathmesh/DATA/sem7/HPC assignment/assignment7$ mpirun -n 8 ./mat_vec_mul.exe
Running with 8 tasks.
Done in 0.003234 seconds.
prathmesh@prathmesh-G3-3500:/media/prathmesh/DATA/sem7/HPC assignment/assignment7$ mpirun -n 10 ./mat_vec_mul.exe
Running with 10 tasks.
Done in 0.043911 seconds.
```

```
prathmesh@prathmesh-G3-3500:/media/prathmesh/DATA/sem7/HPC assignment/assignment7$ mpirun -n 12 ./mat_vec_mul.exe
Running with 12 tasks.
Done in 0.056010 seconds.
prathmesh@prathmesh-G3-3500:/media/prathmesh/DATA/sem7/HPC assignment/assignment7$ mpirun -n 14 ./mat_vec_mul.exe
Running with 14 tasks.
Done in 0.179218 seconds.
prathmesh@prathmesh-G3-3500:/media/prathmesh/DATA/sem7/HPC assignment/assignment7$ mpirun -n 16 ./mat_vec_mul.exe
Running with 16 tasks.
Done in 0.225369 seconds.
```



2. Implement Matrix-Matrix Multiplication using MPI. Use different number of processes and analyze the performance.

Code:

#include "mpi.h"

```
#include <stdio.h>
#include <stdlib.h>

#define MATSIZE 500
#define NRA MATSIZE /* number of rows in matrix A */
#define NCA MATSIZE /* number of columns in matrix A */
#define NCB MATSIZE /* number of columns in matrix B */
#define MASTER 0 /* taskid of first task */
#define FROM_MASTER 1 /* setting a message type */
#define FROM_WORKER 2 /* setting a message type */

int main(int argc, char *argv[])
{
int numtasks, /* number of tasks in partition */
taskid, /* a task identifier */
numworkers, /* number of worker tasks */
source, /* task id of message source */
dest, /* task id of message destination */
mtype, /* message type */
rows, /* rows of matrix A sent to each worker */
averow, extra, offset, /* used to determine rows sent to each worker */
i, j, k, rc; /* misc */
double a[NRA][NCA], /* matrix A to be multiplied */
b[NCA][NCB], /* matrix B to be multiplied */
c[NRA][NCB]; /* result matrix C */
MPI_Status status;

MPI_Init(&argc, &argv);
MPI_Comm_rank(MPI_COMM_WORLD, &taskid);
MPI_Comm_size(MPI_COMM_WORLD, &numtasks);
if (numtasks < 2)
{
```

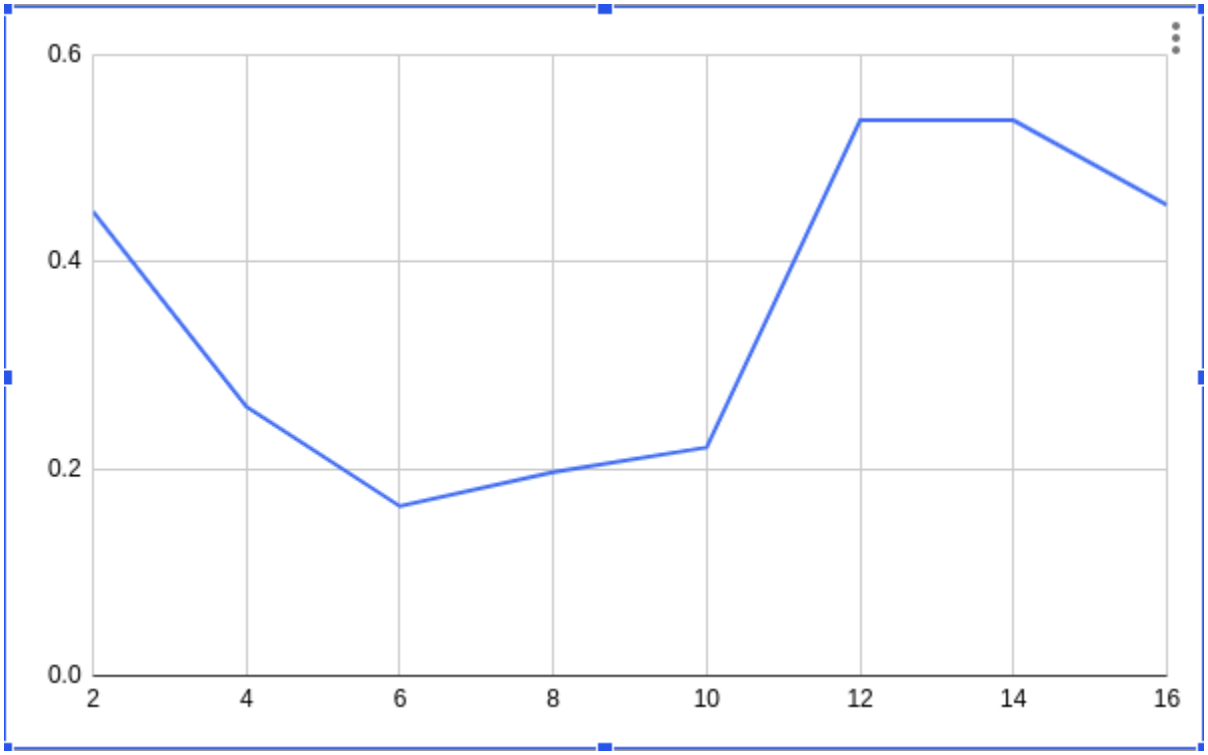
```
printf("Need at least two MPI tasks. Quitting...\n");
MPI_Abort(MPI_COMM_WORLD, rc);
exit(1);
}
numworkers = numtasks - 1;

/********** master task ************/
if (taskid == MASTER)
{
printf("mpi_mm has started with %d tasks.\n", numtasks);
// printf("Initializing arrays...\n");
for (i = 0; i < NRA; i++)
for (j = 0; j < NCA; j++)
a[i][j] = i + j;
for (i = 0; i < NCA; i++)
for (j = 0; j < NCB; j++)
b[i][j] = i * j;

/* Measure start time */
double start = MPI_Wtime();

/* Send matrix data to the worker tasks */
averow = NRA / numworkers;
extra = NRA % numworkers;
offset = 0;
mtype = FROM_MASTER;
for (dest = 1; dest <= numworkers; dest++)
{
rows = (dest <= extra) ? averow + 1 : averow;
// printf("Sending %d rows to task %d offset=%d\n",rows,dest,offset);
MPI_Send(&offset, 1, MPI_INT, dest, mtype, MPI_COMM_WORLD);
MPI_Send(&rows, 1, MPI_INT, dest, mtype, MPI_COMM_WORLD);
MPI_Send(&a[offset][0], rows * NCA, MPI_DOUBLE, dest, mtype,
MPI_COMM_WORLD);
MPI_Send(&b, NCA * NCB, MPI_DOUBLE, dest, mtype, MPI_COMM_WORLD);
offset = offset + rows;
}

/* Receive results from worker tasks */
```

```
mtype = FROM_WORKER;
for (i = 1; i <= numworkers; i++)
{
source = i;
MPI_Recv(&offset, 1, MPI_INT, source, mtype, MPI_COMM_WORLD, &status);
MPI_Recv(&rows, 1, MPI_INT, source, mtype, MPI_COMM_WORLD, &status);
MPI_Recv(&c[offset][0], rows * NCB, MPI_DOUBLE, source, mtype,
MPI_COMM_WORLD, &status);
// printf("Received results from task %d\n",source);
}

/* Print results */
/*
printf("******************\n");
printf("Result Matrix:\n");
for (i=0; i<NRA; i++)
{
printf("\n");
for (j=0; j<NCB; j++)
printf("%6.2f ", c[i][j]);
}
printf("\n********************\n");
*/

/* Measure finish time */
double finish = MPI_Wtime();
printf("Done in %f seconds.\n", finish - start);
}

/********** worker task ************/
if (taskid > MASTER)
{
mtype = FROM_MASTER;
MPI_Recv(&offset, 1, MPI_INT, MASTER, mtype, MPI_COMM_WORLD, &status);
MPI_Recv(&rows, 1, MPI_INT, MASTER, mtype, MPI_COMM_WORLD, &status);
MPI_Recv(&a, rows * NCA, MPI_DOUBLE, MASTER, mtype, MPI_COMM_WORLD,
&status);
MPI_Recv(&b, NCA * NCB, MPI_DOUBLE, MASTER, mtype, MPI_COMM_WORLD,
&status);
```

```
for (k = 0; k < NCB; k++)
for (i = 0; i < rows; i++)
{
c[i][k] = 0.0;
for (j = 0; j < NCA; j++)
c[i][k] = c[i][k] + a[i][j] * b[j][k];
}
mtype = FROM_WORKER;
MPI_Send(&offset, 1, MPI_INT, MASTER, mtype, MPI_COMM_WORLD);
MPI_Send(&rows, 1, MPI_INT, MASTER, mtype, MPI_COMM_WORLD);
MPI_Send(&c, rows * NCB, MPI_DOUBLE, MASTER, mtype, MPI_COMM_WORLD);
}
MPI_Finalize();
}
```

Output:

```
prathmesh@prathmesh-G3-3500:/media/prathmesh/DATA/sem7/HPC assignment/assignment7$ mpirun -n 2 ./mat_mat_mul.exe
mpi_mm has started with 2 tasks.
Done in 0.448954 seconds.
prathmesh@prathmesh-G3-3500:/media/prathmesh/DATA/sem7/HPC assignment/assignment7$ mpirun -n 4 ./mat_mat_mul.exe
mpi_mm has started with 4 tasks.
Done in 0.260298 seconds.
prathmesh@prathmesh-G3-3500:/media/prathmesh/DATA/sem7/HPC assignment/assignment7$ mpirun -n 6 ./mat_mat_mul.exe
mpi_mm has started with 6 tasks.
Done in 0.164012 seconds.
prathmesh@prathmesh-G3-3500:/media/prathmesh/DATA/sem7/HPC assignment/assignment7$ mpirun -n 8 ./mat_mat_mul.exe
mpi_mm has started with 8 tasks.
Done in 0.195552 seconds.
prathmesh@prathmesh-G3-3500:/media/prathmesh/DATA/sem7/HPC assignment/assignment7$ mpirun -n 10 ./mat_mat_mul.exe
mpi_mm has started with 10 tasks.
Done in 0.196801 seconds.
prathmesh@prathmesh-G3-3500:/media/prathmesh/DATA/sem7/HPC assignment/assignment7$ mpirun -n 12 ./mat_mat_mul.exe
mpi_mm has started with 12 tasks.
Done in 0.220816 seconds.
prathmesh@prathmesh-G3-3500:/media/prathmesh/DATA/sem7/HPC assignment/assignment7$ mpirun -n 14 ./mat_mat_mul.exe
mpi_mm has started with 14 tasks.
Done in 0.537073 seconds.
prathmesh@prathmesh-G3-3500:/media/prathmesh/DATA/sem7/HPC assignment/assignment7$ mpirun -n 16 ./mat_mat_mul.exe
mpi_mm has started with 16 tasks.
Done in 0.455361 seconds.
prathmesh@prathmesh-G3-3500:/media/prathmesh/DATA/sem7/HPC assignment/assignment7$
```



Github Link:

https://github.com/killedar27/HPC-assignments/tree/main/assignment7