

Problem

Write an MPI program that performs matrix multiplication. To keep things simple, multiply two square matrices $M \times M$, where M is evenly divisible by the number of processes. You can use simple point-to-point communication functions, or collective communication functions. Read about matrix multiplication [here](#).

```
#define N 2
#include <stdio.h>
#include <math.h>
#include <sys/time.h>
#include <stdlib.h>
#include <stddef.h>
#include "mpi.h"

void print_results(char *prompt, int a[N][N]);

int main(int argc, char *argv[])
{
    int i, j, k, rank, size, tag = 99, blksz, sum = 0;
    int a[N][N] = {{1,2},{3,4}};
    int b[N][N] = {{2,0},{1,2}};
    int c[N][N];
    int aa[N], cc[N];

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

    //scatter rows of first matrix to different processes
    MPI_Scatter(a, N*N/size, MPI_INT, aa, N*N/size, MPI_INT, 0, MPI_COMM_WORLD);

    //broadcast second matrix to all processes
    MPI_Bcast(b, N*N, MPI_INT, 0, MPI_COMM_WORLD);

    MPI_Barrier(MPI_COMM_WORLD);

    //perform vector multiplication by all processes
    for (i = 0; i < N; i++)
    {
```

```

for (j = 0; j < N; j++)
{
sum = sum + aa[j] * b[j][i];
}
cc[i] = sum;
sum = 0;
}

```

```

MPI_Gather(cc, N*N/size, MPI_INT, c, N*N/size, MPI_INT, 0, MPI_COMM_WORLD);

```

```

MPI_Barrier(MPI_COMM_WORLD);
MPI_Finalize();
if (rank == 0) //I_ADDED_THIS
print_results("C = ", c);
}

```

```

void print_results(char *prompt, int a[N][N])
{
int i, j;

printf ("\n\n%s\n", prompt);
for (i = 0; i < N; i++) {
for (j = 0; j < N; j++) {
printf(" %d", a[i][j]);
}
printf ("\n");
}
printf ("\n\n");
}

```

a.c - labwork4 - Visual Studio Code

64

EXPLORER

OPEN EDITORS

a

a.c

b.c ~/code/parralel/asigme...

LABWORK4

.vscode

a

a.c

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

1: bash

dmitriy@dmitriy-Extensa-2510:~/code/parralel/labwork4\$ mpxexec -np 2 ./a

C =
4 4
10 8

dmitriy@dmitriy-Extensa-2510:~/code/parralel/labwork4\$

(Global Scope) Ln 64, Col 1 Spaces: 4 UTF-8 LF C Linux