Name: Drew Masters
Date: 10/12/2016
Class: COSC 462

Homework 3

The problem is to implement matrix multiplication using openMPI. The mapping was defined in the homework by the following method:

$$row = rank \ \%N$$
$$col = rank \ / N$$

The distribution of the mapping occurs in process 0. This mapping meant that process zero got the first element, process one got the element that is in the second row first column, process two got the element that is in the third row first column, and so on. Process 0 sends elements to the corresponding processes and the processes receive the elements.

After mapping the first step of the multiplication is to gather the row from matrix A and the column from matrix B that corresponds with the process given. Process 0 gets the first row and first column, process 1 gets the second row and the first column, process 2 gets the third row and the first column, etc. The row and columns are constructed locally by creating communicators for each row and column than calling allgather to construct the local arrays that represent row and column.

The process then goes through these local rows and columns and calculates the dot product. After doing this the results are sent back to process zero. Process zero then reconstructs the results and writes the results to the specified file.

Files included

-makefile

-matrix_multiply.c: openMPI code for matrix multiplication. Compiled as matrix_multiply by makefile. To run the code “mpirun -np size matrix_multiply”. Results are written to C.dat

-matrix_reader.c: helper program that reads matrix file and outputs it in a human readable format. To run “./matrix_reader size file_name”. Size is the row or column size and the file name is the computer readable file.

-matrix_writer.c: helper program that generates matrix in specified files of specified size. To run the code “./matrix_writer size (A or B).dat human_readable_file_name”. The command will generate a matrix of size by size.

Steps to run code

-Run make. It will generate the executes listed above.
-The code will only run if the number of processes are a perfect square i.e. 4, 9, 16 etc.