

## Homework 6 Fall 2015 Parallel and Distributed Computing - MPI

### Problem

Implement a fast matrix multiplication program using MPI

### Directions

Implement a serial matrix multiplication program with the following requirements (you can use the previous implementation)

1. You should have a 2D matrix
2. The matrix should be read from a text file.
3. You should multiply the matrix to a second (different) matrix
4. The matrix multiplication result should be written to a file in the same format as in step #2
5. The code should be timed. Timing should be printed on the screen

Implement a distributed (and parallel) version of the above with the following requirements

1. You should use the same matrix input file as the one used in the serial program
2. You will use master slave model – master assigns/distributes work/data and slaves do the computation. Master can participate
3. You will measure and print timings to the screen
4. The matrix multiplication result should be printed to a file and should be the same result as above if using same input
5. You should store the matrix as a distributed data structure. **If you do not follow this requirement, you will lose almost all of your points for this homework**
  - a. In other words, you have to utilize the memory on all the mpi nodes and accomplish a bigger sized matrix multiplication than the serial implementation

### Measure speedup

Speedup will be measured as

Serial execution time / distributed execution time for a given input

To measure speedup, you will use an NxN matrix

N should be **fixed (ex. 200)**

Table 1: Vary the number of mpi processes

N	Number of processes (p)	Serial execution time	Distributed Execution time	Speedup
200				
200				
200				
200				

Table 2: Increase  $N$  (generate a bigger matrix) and *if required* vary the number of mpi processes. Serial implementation might fail at a higher value of  $N$  but the distributed implementation should be able to handle it (in such cases serial execution time is 0 and do not fill up speedup)

N	Number of processes (p)	Serial execution time	Distributed Execution time	Speedup
100				
200				
500				
1000				
2000				
5000				

## Submission

Your submission should include the following:

3. **Makefile** that builds all required binaries:
  - a. Part\_N\_type.out where N is the homework part and type can be serial/parallel
4. Serial code source file(s)
5. Parallel code source file(s)
6. Binary for both serial and parallel implementations
7. Document (word or excel) detailing the tabular results

All of the above should be zipped up into one file and uploaded on blackboard

If you are doing the bonus, add a comment in your code about it. (if applicable)

**Naming convention of your zipped file – If you do not follow this, you get zero points:**

FirstName\_LastName\_Homework\_<n>.zip

Where  $n$  is the homework number.

## Grading

Distributed Implementation	25%
Correct Distributed execution	40%
Correct serial execution	2%
Code clarity	3%
Speedup	5%
Timing	15%
Handling bigger matrices than the serial implementation (bigger than $N = 1000$ )	10%

## Speedup, timing and grading

Below are how speedup points will be calculated

- If you get a speedup ratio greater than 0.0 – you get the 5%
- For the timing 15% - you compete with your colleagues
  - This grade can vary from 0-15%

- If you are the top performer, you get 15
  - If you are the least performer you get 0
  - Others will be calculated accordingly
- **If you do any miscalculation intentionally or unintentionally on speedup, you get over all zero because that will be considered as cheating – no exceptions**
- **If you have a code base similar to your colleague, you BOTH will get a zero after being called to meet me**
- You are free to leverage pthreads/openmp to get higher speedup

#### Code clarity includes

- Use of variable names
- Function naming
- File names
- Formatting
- Comments where needed

Late submissions are subject to at least 20% grade deduction. 1 week after the deadline you get 50% deduction. 2 weeks after the deadline you get a 0.

**Reminder:** If one or any of your binaries don't execute on student.rit.albany.edu, you get 0 points for this and any other homework where that happens.