

Homework 5: Parallel and Distributed Computing

DUE: See exact time on blackboard

ICSI 520

In this homework you will implement a matrix multiplication using openmp:

- Use a data structure of your choice to represent the matrix. You have been given the code to generate input matrices (use from previous homework). The code generates matrices as csv files.
- You will be multiplying two square matrices. Use the code given to generate two different matrices of the same size (nxn) and multiply them.
- Examine the format of the input file. The output should be the same format as the input.
- Tip: Start with small matrix sizes (say 4x4 or even 2x2)

Serial and Parallel Implementations

You will do both serial (you should already have this) and parallel (using openmp) implementations of matrix multiplication

Use timing code to measure timings for this code.

Several Considerations

You will gain or lose points based on your decisions/considerations

1. Data structure – Which data structure will give you best performance
2. Code organization – Organize your implementation properly

Results

Matrix size (n)	Number of threads (p)	Serial execution time	Parallel Execution time	Speedup
10	<decide the best number>			
30				
50				
75				
100				
<add more ..>				

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

Serial Implementation	10%
Parallel Implementation	25%
Correct execution Parallel	30%
Correct serial execution	10%
Speedup (greater than 1)	10%
Code clarity	5%
Timing competition	10%

Timing competition

You are competing with your colleagues for these 10% of the points. Lowest timing gets 10 full points.

Highest timing gets 0.

Do not make up the speedup or the timing tables. You will get a 0 for all homework assignments (including previous).

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.

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.