Homework 2: Parallel and Distributed Computing

DUE: See exact time on blackboard

ICSI 520

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

- Use a data structure of your choice to represent the matrix. You have been given the code to generate input matrices. 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.
- Use the code you wrote in HW1 (optimized serial mat mul) – Convert that to parallel implementation
- Tip: Start with small matrix sizes (say 4x4 or even 2x2), Consider the use of condition vars, mutexes properly

Serial and Parallel Implementations

You will do both serial and parallel 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. Synchronization – Where and how many times you use mutexes
3. 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 ..>				

1. Homework2_Serial
    a. Include all of the source files for serial implementation
    b. Binaries
    c. Input and output files
2. Homework2_Parallel
    a. Include all of the source files for parallel implementation

    b. Binaries
    c. Input and output files
3. A document explaining the speedup and the tabular results
    a. Here you will explain for what value of p you are getting the most speedup
4. All of the above should be in one zip/tar following the naming convention noted below.

Name your zip/tar/rar/7z folder as FirstName_LastName_Team<number>_Homework2. You will lose points if you don’t follow this convention

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.

Submission:

Your submission should include the following:

1. **Makefile** that builds all required binaries:
    a. Part_N_type.out where N is the homework part and type can be serial/parallel
2. Serial code source file(s)
3. Parallel code source file(s)
4. Binary for both serial and parallel implementations
5. 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.