Lab - 3

There will be three sub-problems in this assignment. All the sub-problems have to be supplemented with a **brief write-up with the following details** (wherever necessary): bullet *points are common for all three sub-problems.*

1. Context:

   Brief description of the problem.

.

   Complexity of the algorithm (serial).

.

   Profiling information (e.g. gprof). Serial run-time.

.

   Compute to memory access ratio

.

   Memory bound vs compute-bound problem

.

   Optimization strategy.

2. hardware details: CPU model, memory information details, no of cores, compiler, optimization flags if used, precision used. Max. Theoretical performance.

3. Input parameters. Output. How accuracy is checked.

4. Problem Size vs Time (Serial) **curve**. Observations and comments about the results.

5. If more than one implementation, curves for all algorithms in the same plot.

6. Wherever necessary, use log scale and auxiliary units.

7. Measure the performance of the code in MFLOPS/sec. and plot problem size vs performance plot. How does it compare with the theoretical performance of the machine where the code is being executed?

Run time should be significant so that the run time can be measured properly using timer functions. Start with a small problem size and go to the maximum possible problem size as far as possible.

All runs need to be performed on Two machines → lab machine (207) and cluster.
At least 5 times for statistical analysis of fluctuations in run-time.

Problem A-1 → Conventional matrix multiplication (interchange the loops - 6 possibilities)
Problem B-1 → Matrix multiplication using transpose
Problem C-1 → Block matrix multiplication (divide and conquer strategy)

Goal - measure performance/runtime as a function of problem size (2^2 to 2^12).
Use integers (4 bytes) as well as floating points (8 bytes) for inputs.

For all problems:
Step-1 - Serial code development with proper comments
Step-2 - Run the codes on two machines: Lab 207 machine and HPC Cluster
Step 3 - Collect data for 5 runs which can be used for statistical analysis
Step 4 - Accurate run-time Data collection using scripts (algorithmic and end-to-end)
Step 5 - Upload data and commented code via a google form

Same input files will be provided to all the groups for comparison purposes.

Bonus marks for the top team (each sub-problem) - (20% of total assignment marks)
Run-time Performance comparison of all teams will be shown in the next - labs.

More details will be provided for assignment submission and run-time data submission before the deadline.