CS301-HPC Lab 2

Assignment 2: (3.5 marks)

Deadline: March 2, 2022

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): *point 2 will be common for all three problems*

1. Context:

    - Brief description of the problem.
    - Complexity of the algorithm (serial).
    - Possible speedup (theoretical). Speedup=Serial runtime/Parallel Runtime
    - Profiling information (e.g. gprof/ vtune).
    - Optimization strategy.
    - Problems faced in parallelization and possible solutions.
2. Hardware details: CPU model, memory information, no of cores, compiler, optimization flags if used, precision used.
3. Input parameters. Output. Make sure results from serial and parallel are the same.
4. Problem Size vs Time (Serial, parallel) **curve**. Speedup 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. Problem size vs. speedup curve.
8. No. of cores (or threads) vs. speedup curve for a couple of problem sizes.

In today's lab(22/02/2022) focus on Problem-1 and Problem-2.

Problem-1

1. Write a basic parallel code (OpenMP) for integration using trapezoidal rule. Serial/parallel code was discussed in the class.
2. Use the parallel code to calculate PI and verify the implementation against serial results.
3. Measure performance in MFLOPS/sec of the parallel code, compare it with serial performance.
4. Is it a compute bound or memory bound problem.
5. Start with a small N but eventually **take a very large value of "N"** for summation for generating the Problem size vs Speedup curve.

   Basic OpenMP Pragmas, how to implement/execute a parallel code on multi-threaded environment and Concept of Shared/Private variables to be learned during this assignment.

6. Implement the following versions involving different clauses and compare the performance of these different versions (**take large values of "N"**):

I. without padding

II. With padding to address false sharing problem

III. Critical clause

IV. Reduction clause

V. Two scheduling mechanisms : static and dynamic.

Problem-2

Write a **serial code** and **parallel code** (using openMP) for the following : **Take Large Vectors (for ex. 1e3, 1e5, 1e6, 1e7, 1e8,1e9)**

a. Summation of two vectors (serial and parallel comparison for accuracy of results).
b. Multiplication of Two vectors followed by summation.

In the report you must include important observations (point wise and brief), challenges faced while parallelization and how you addressed those issues. How you could improve performance by following different strategies for the same problem. Is it a compute bound or memory bound problem.

Make sure to go from small problem sizes (2^6) to big problem size at least (2^28).

Problem-3

Write a **serial code** and **parallel code** for the following (using openMP):
· **Calculation of pi using random numbers.**

First use the random function of the C library for random number generations and then understand the problem of parallel random number generations using OpenMP, and what are the main issues. How did you solve it. Report on it briefly.

Explore the following reference:

http://selkie.macalester.edu/csinparallel/modules/MonteCarloSimulationExemplar/build/html/SeedingThreads/SeedEachThread.html

Reference for serial code:

https://www.geeksforgeeks.org/estimating-value-pi-using-monte-carlo/