

CS 610 Semester 2020–2021-I: Assignment 4

4th November 2020

Due Your assignment is due by Nov 14 2020, 11:59 PM IST.

General Policies

- You should do this assignment ALONE.
- Do not plagiarize or turn in solutions from other sources. You will be PENALIZED if caught.
- We WILL check your submission(s) with plagiarism checkers.

Submission

- Submission will be through mooKIT.
- Write your programs in C++ .
- Feel free to use Intel DevCloud or a department workstation. Include the system description (for e.g., levels of private caches, L1/L2 cache size, processor frequency) and compiler versions in your report.
- Submit a compressed file with name “<roll-no>.zip”. The compressed file can have the following structure.

```
-- roll-no
-- -- report.pdf
-- -- <roll-no-p1.cpp>
-- -- <roll-no-p2.cpp>
```

Submit a PDF file with name “report.pdf”. We encourage you to use the L^AT_EX typesetting system for generating the PDF file. The file should include your name and roll number, and results and explanations for the following problems. Include the exact compilation instructions for each programming problem, for example, `g++ -O2 problem2.cpp`, and any other information and assumptions that you think will help the evaluation.

- You will get up to TWO LATE days to submit your assignment, with a 25% penalty for each day.

Evaluation

- Please write your code such that the EXACT output format (if any) is respected.
- Do NOT change code you are not supposed to touch (for e.g., correctness checks).
- We will evaluate your implementations on a Unix-like system, for example, a recent Debian-based distribution.
- We will evaluate the implementations with our OWN inputs (for e.g., different input array sizes) and test cases, so remember to test thoroughly.

Problem 1

[50 marks]

- (i) Implement Fibonacci computation using explicit OpenMP tasks (for e.g., `task` or `sections`).
- (ii) Try to optimize your task-based implementation to be competitive with the serial code.
- (iii) Implement Fibonacci with the blocking style parallelism in Intel TBB.
- (iv) Implement Fibonacci with continuation passing style parallelism in Intel TBB.

Compare performance of your implementations with the serial code and explain your observations.

Problem 2

[50 marks]

Parallelize the quick sort implementation with OpenMP. Compare and report the performance of the serial and parallel versions.

Problem 3

[50 marks]

Use (i) OpenMP and (ii) Intel TBB parallel algorithms to compute the value of π in parallel. Compare and report the performance of the serial version and the parallel versions.

Report the performance of your best parallel versions with OpenMP and TBB. Try to minimize the impact of false sharing for your parallel implementations.

Problem 4

[50 marks]

Implement a parallel version of “find max” for a given array of numbers using Intel TBB parallel algorithms. Compare and report the performance of the serial version and the parallel version. Assume the max size of the input array to be limited to 2^{26} .

Report the index of the first occurrence of the max value in case there are duplicates.