CS 7172 / ACS 7410 – Parallel and Distributed Computing

Fall 2019

Project: Parallel Program Development

Version 1

Design, analysis and implementation of algorithms for solving a Tree Search Problem (Comparative analysis of serial and parallel algorithms)

Total Points: 100%

In this term project, students (either a team of two; or individually) will need to study and review the algorithms and the implementations for solving a **tree search problem**, as described in Section 6.2 (textbook [1]). Then, each group or each student will solve the following specific problems, and then compare (i.e., analyze) the different approaches.

There are two fundamental parallel architectures: distributed-memory system and shared-memory system. Writing parallel programs is different for/on these two architectures. Students should at least learn the programming techniques for both systems. It is expected that either "MPI", or "OpenMP" ("Pthreads" can be also used if you prefer) be used in implementing the parallel algorithms.

Problem: There are three problems as below. The first problem A is to implement serial algorithm/programs with three different versions as described in textbook. The second problem B is to implement TSP using, either static MPI or static OpenMP (not both). These four implementations are required.

A. [30%] [**Required**][Implementations and **Performance of Serial Implementations**][Run-times of the Three Serial Implementations of Tree Search (unit of time = seconds)] [Reference: Table 6.7 in the textbook]
    a. Recursive
    b. First iterative
    c. Second iterative

Do either Problem B or Problem C (Not Both):

B. [70%] [**Required; Static MPI**] [**Implementation of tree search using MPI and *static* partitioning**] [Performance of MPI Program (time unit = seconds)] [Reference: Table 6.11 in the textbook], or

C. [70%][**Required; Static OpenMP**][**Implementation of tree search using *static* OpenMP & Performance Analysis**][Performance of OpenMP Implementation of Tree Search (unit of time = seconds)] [Reference: Table 6.9 in the textbook]

The following ("D") is NOT required, but this additional work is for extra points ONLY.

D. [NOT required; Additional points ONLY, up to 50 points; **Dynamic partitioning**] If you (or your team) solve the problem with dynamic partitioning (MPI or OpenMP) and demonstrate your program, up to 50 points will be given.

It is your full responsibility of making a team (if applicable) and finish this work until the end. So, the spirit of team-work is needed (no excuse.)

Point Breakdown: In terms of **quality** of the work – this is a rough break down of tasks. For evaluation, see "Dissemination" below.

- Task 1 [25%]: Correctness of parallel algorithms
  - Understanding of parallel algorithms
  - If the implementation was correctly and efficiently done, implementing wrong algorithms or implementing algorithms with misunderstanding must be avoid in the first place.

- Task 2 [25%]: Correct implementation
  - Two parallel architectures: Distributed-memory and Shared-memory systems
  - MPI for Distributed-memory system, or one of two APIs (Pthreads, OpenMP) for Shared-memory system

- Task 3 [30%]: Valid analysis
  - Performance measures and the comparisons between serial implementations and parallel implementations, and
  - among different parallel implementations, if applicable

- Task 4 [20%]: Readability / presentation (TBD – if applicable) of the submitted work
  - E.g.: source codes – with adequate comments

In terms of **dissemination** of the work:

- **Progress Report (Mid Report) [50 points]**
  - Due: Oct. 9/Wed
  - Turn-in: (a) Current progress - source codes, if done, (b) what was complete, what will be done – clear breakdown of the former and the latter, (c) next plans, (d) challenges & plans to resolve them.

- **Final Report and/or Presentation [Total: 350 points]**: MS Word format or PDF. No more than 20 pages, including the captured test runs. (If more pages are needed, please use Appendix).
  - Description of problem <less than 1 page>
  - Description of algorithms <less than 2 pages>
  - Source codes with comments
  - Test data: (a) **Data Set 1** (n = 4) from the textbook – Fig. 6.9, (b) **Data Set 2 (n = 15)**, starting city = 0 (for both data set). /* Data Set 2 – attached as a separate file */

- Execution output/results (including, the captured screen shots/output including “the solution of visiting cities with the minimum cost ” run by your program.)
- Comparative analysis of output/results (**Run-time analysis**; tables and/or graphs can be used for the comparisons).

Final Report is required. Demo/presentation – TBD.

Assessment 1 [250]: Final Report is to be submitted to D2L for grading.

Assessment 2 [100]: If class demo is conducted, either PPT slides and other reasonable forms of demo will be required. (NOTE: The rubrics for Class Demo, if applicable, should be distributed at least two weeks prior to the Project Demo Week.)

- The End -