Assignment 2, Fall 2024

CS4823/CS6643, Parallel Computing

PRAM Algorithm and OpenMP Programming

Collaborative

Due Date

This assignment is due next Thu, 09/18/2024, at 11:59pm. You may work with another student but each person submits, mentioning other person's name they worked with.

Materials to Review

1. Read Chapter 2 - Sections 2.3 and 2.4.1-2.4.5 - of textbook, related slide sets for this chapter, and OpenMP slides (part 1) posted.

Questions

1. (15 points) ***PRAM Algorithm***

   Design an algorithm for multiplying two square matrices of size n x n which uses $p <= n^3$ processors and achieves the fastest parallel execution time of O(log n). You may assume EREW PRAM model.

   (a) (10 points) Give major steps in high level description/pseudocode in enough detail to answer part (b) – that is, detailed pseudocode is not needed. You may employ diagrams to illustrate.

   (b) (5 points) For $p <= n^3$ processors, calculate expressions for Tp, Sp, Ep, cost and work of your algorithm as functions of n and p using O notation.

2. (20 points) ***OpenMP Programming***

   Write a shared memory OpenMP program on Fox server to find the sum of n integers in an array A[0..n-1] using p processors with 1<= p <=12. Fill up the array with some constant values so that it would be easier for you to verify the result for correctness. You need to employ the following steps.

   (a) (10 point) Phase 1 should find local sums in Sum[0..p-1] with each processor handling roughly n/p size subarray with P_i storing its local sum into Sum[i], 0 < = i< = (p-1). Each processor will need to figure out, using its own id, which subarray it needs to work on. Assume p divides n.

   (b) Phase 2 should find a global sum into Sum[0], in two ways (thus submit <u>two versions</u> of your program):

      i. (5 point) Processor 0 adds all local sums into Sum[0].

      ii. (5 point) Each processor adds their local sum to Sum[0].

   You will need to ensure phase 1 is completed before phase 2 starts, and prevent race condition in Phase 2.