

Assignment 1 – Parallel Computing CSci6702

Part 1: Theory

Read Chapter 2 of Introduction to Parallel Computing and answer the following questions:

Note: let t_c be the time for one person to add two numbers and t_w be the time for a person to pass a single number to a neighbour.

- 1.1 Consider the problem of adding n numbers. Assume that one person can add two numbers in time t_c . How long will a person take to add n numbers?
Now assume that eight people are available for adding these n numbers and that it is possible to divide the list into eight parts. The eight people have their own pencils and paper (on which to perform additions), are equally skilled, and can add two numbers in time t_c . Furthermore, a person can pass on the result of an addition (in the form of a single number) to the person sitting next to him or her in time t_w . How long will it take to add n numbers in the following scenarios:
 - (a) All eight people are sitting in a circle.
 - (b) The eight people are sitting in two rows of four people each.
- 1.2 Assume the scenario described in Problem 1.1. If you had the liberty of seating the people in any acceptable configuration, how would you seat them so as to minimize the time taken to add the list of numbers? (An acceptable configuration is defined as follows: Assume that the adders are vertices of a graph and that *neighborhood* is defined by the edges between the vertices. Any graph that can be drawn on a piece of paper and that has no intersecting edges represents an acceptable configuration.) What is the time taken by your configuration?
- 1.3 Consider again the problem of adding n numbers. Assume that one person takes time $t_c(n - 1)$ to add these numbers. Is it possible for p people to add this list in time less than $t_c(n - 1)/p$? Justify your answer.
- 1.4 Consider again the scenario of Problem 1.1, but assume that all eight people are adding the numbers standing at a blackboard. Each person can see results from the other person's calculations as they are completed (that is, instantaneously). How long would the eight people take to add the n numbers in this case?
- 1.5 Answer Problem 1.3 in the context of the scenario presented in Problem 1.4. Is your answer different for this scenario? If so, what specific change resulted in a different answer?

1.6 Consider a CRCW PRAM with p processors and m memory location which uses the Max rule, that is, if multiple values are written to a memory cell the value stored after the write operation is complete is the largest value that was written. We can emulate this model on a p -processor message passing parallel computer in which each processor has m/p memory locations. This emulation consists of three parts:

A) A mapping of the shared PRAM memory into the distributed memory of our p -processor message passing parallel computer

B) An algorithm for performing Concurrent Read

C) An algorithm for performing Concurrent Write with the Max rule.

Part A is straight-forward we simply divide our size m shared memory into size m/p blocks and store them consecutively on our distributed memory machine. Your task is to describe and analyze algorithms for Parts B & C on the following architectures:

- i. A p -processor ring
- ii. A p -processor mesh
- iii. A p -processor hypercube

If an algorithm takes time t on the CRCW PRAM how long does it take when emulated on each of these architectures?

Part 2: Sequential Performance and Benchmarking

The purpose of this part of assignment 1 is to get you to think about issues in sequential optimization and have you perform a systematic performance evaluation.

Create an efficient code for a stable sorting n records consisting of an integer key and a data payload of between 1 byte and 10KB. The input data should be read from disk into memory and the output data should be written to disk. There are many issues and possible optimizations to consider, including:

- What sorting algorithm should be used to achieve best performance? Should a single algorithm be used regardless of the problem size and data characteristics or is a hybrid method better?
- Can compiler optimizations improve the runtime?
- Can you use multiple cores to improve the performance?
- Does your method have better or worse cases based on the properties of the input data? For example, how well does it handle non-unique keys?
- If your method has internal parameters, for example it switches between methods when sorting less than n' items be sure to do experiments that explore how best to select this parameters.

Perform a careful benchmarking analysis to capture the performance characteristics of your sort. Compare your sort methods to the standard qsort. Discuss how theory and practice compare. Chart the results and write up a careful analysis of them.