CS415 Programming Assignment 2

Adam Lewis

January 26, 2016

Contents

1 Overview

Parallel programming is very different from the imperative and sequential model of programming you have learned to this point in your career. In this assignment you will use the C++ thread library to write a program that multiplies two square arrays and compare the difference between the imperative and parallel implementations of this algorithm.

2 Background

Matrix multiplication is a well-understood algorithm: if two matrices are multiplication compatible (i.e., one array is mXn dimensions and the other is nXp dimensions), then one computes each entry in the product as the dot-product of the matching row and column. This means that each element in the final array must be computed as the sum of the product of each corresponding row and column element.

This is not good from a performance standpoint as that algorithm is $O(n^2)$ (you do remember "Big-Oh" from CS372, don't you?). But each element in the product is computed independently of every other element. Thus, this problem is very amendable to being redone as a parallel program.

3 Problem Statement

Write a program that contains three functions:

1. A function that has an integer as a parameter and returns a pointer to square array of integers (i.e. both dimensions should be equal). The function should allocate storage from the heap and randomly fill that dynamically allocated array.

2. A function that uses the algorithm described above to compute the matrix square of an array. It should be passed a pointer to an array of integers and an integer defining the dimensions of the array. It should return a double pointer to the array containing the product. This is the standard matrix-multiply algorithm that can be found throughout the web.

3. A function that uses threads to compute the matrix square of an array. This function should have the same parameters and return values of the previous function

The main() function in your program needs to use these functions to compute the square of matrices with 100, 500, 1000, 5000, and 10000 integers.

3.1 A simplifying assumption

For the purposes of this assignment, you may assume that the values used for the size of these square arrays will always be even.

3.2 Hints

- You will have to think very carefully about how to partition the array when you implement your pthreads function. My suggestion is to think about dividing each array into smaller and smaller pieces until you reach some reasonably small size. At that point multiply the matrices using the iterative algorithm. (And yes... this means you have to use recursion!)

4 Submitting your program

Your submission document must be in PDF format; submission of documents in any other format will result in deduction of points from your grade. Attach your submission to the assignment entry in Blackboard.