ECE5640 High Performance Computing Spring 2014

Homework 2: Due Tuesday, January 28

Homework is due on Tuesday, January 28. Handwritten answers to book problems is fine. Code, scripts and output files should be uploaded to Blackboard. If you scan your homework, make sure it is legible!

Numbered problems are exercises from the book:

Introduction to High Performance Scientific Computing Victor Eijkhout

Please use revision 542 of the book, posted on Blackboard . The exercises are embedded in the chapters.

Each problem is worth 15 points unless otherwise noted.

1. (40 pts)

   (a) (5 pts) Write sequential C++ code that sums a vector of values `vect1[i]`. The result should be placed in a variable called `totsum`. Initialize your vector to integers such that `vect1[i] = i+1`.

   (b) (15 pts) Write a multithreaded version of your code to sum a vector of values using OpenMP. The vector should be split among the available threads and each thread should add its own components. Each thread should have its own partial sum and then add to a shared variable called `totsum`. Each thread should use an openmp `atomic` pragma when it updates totsum. Your code shoud work on any number of input threads. Turn in your code.

   (c) (10 pts) Use the openmp reduction sum to accomplish the same thing as your code in the previous part. Turn in your code.

   (d) (10 pts) Analyze the run time of your code on the discovery cluster. Compare sequential, parallel, and reduction sum code. Run on different numbers of threads. Give advice to someone running this code regarding how they should parallelize it. Hw many threads is optimal? Which version runs the fastest? Explain your answer. Turn in the script you use to run your file and the output file.

2. 2.1

3. 2.2

4. 2.3

5. 2.4