SC 290: High-Performance Computing
Spring 2015
Homework #5

Directions: Write your answers in a word document and email to will@accre.vanderbilt.edu before lecture (9:10am) on the due date (Wednesday, Feb 18). Please ensure that all the files/scripts you create remain in the appropriate location until the assignment has been graded!

Total Points Possible: 22

1. Devise formulas to calculate my_first_i and my_last_i in the global sum example discussed in class (see slide 15 in the Feb 6 lecture). Remember that each core should be assigned roughly the same number of elements of computations in the loop. (2 points)

2. Describe a research problem in your major that would benefit from the use of parallel computing. Provide a rough outline of how parallelism would be used. Would you use task- or data-paralellism? (2 points)

3. What are the major differences between shared memory and distributed memory computers? Also outline the advantages and disadvantages of the two. (2 points)

4. What are the differences between a process and a thread? (2 points)

5. If a problem of size W has a serial component W_s show that W/W_s is an upper bound on its speedup, no matter how many processing elements are used. (2 points)

6. Suppose the run-time of a serial program is given by T_{serial} = n^2. Suppose that a parallelization of this program has run-time T_{parallel} = n^2/p + \log_2(p). Write a C program that finds the speedups and efficiencies of this program for various values of n and p. Run your program with n = 10, 20, 40, ..., 320, and p = 1, 2, 4, ..., 128 and plot the results. What happens to the speedup and efficiency as p is increased and n is held fixed? What happens when p is fixed and n is increased? Point me to the location of the C program in your ACCRE home directory. Also include a Makefile for building the executable. (4 points)

7. Modify the matrix_multiply.c file at:

https://github.com/accre/SC290/tree/master/Cprogramming/hwk5

to compute the product of a n x n matrix with itself. Probably the easiest way to get this file on the cluster is by re-cloning the SC290 repo from scratch (if you’ve modified files in your local repo then “git pull” will complain). Just be sure you do not overwrite your current edited repo. You could create a new directory in your
home directory, cd into it, and then type “git clone https://github.com/accre/SC290” to be on the safe side.

Your program should read in a text file (the name of which should be passed from the command line) containing a n x n matrix (n integers per line, n lines) and copy the data into a dynamically allocated 1d array called matrix_in. The first row of the matrix should occupy the first n indices of the array, the second row of the matrix should occupy the next n indices, and so on. An example of a 50 x 50 matrix is included in the Github directory. You’ll be graded based on the output from processing this file. I would recommend building a file containing a simple 3 x 3 matrix, for which you know the solution. You should also pass the matrix length from the command line.

Follow the instructions included in the TODO line of the .c file. In particular, make sure you pass arrays by reference and write functions when you're told to do so.

Store the results of the matrix multiplication in another dynamically allocated 1d array called matrix_out. Your program should output the results to stdout (using printf) in the same format as the input file (n rows of n integers).

Point me to the file location. (8 points)