OpenMP Programming Assignment

Organization: CS312, Computer Science, SIUE
Author: Mark McKenney

Overview:

Write a multithreaded C++ program using open MP threads. I will provide a text file containing numbers that must be read into a two dimensional array. The first line of the file will have 2 integers, the number of rows in the array followed by the by the number of columns in the array. The rest of the file will contain integers that the array must be initialized to. Your program must also utilize a random number generator. I will provide source code.

Description:

Your program will indicate the cell address (the row and column) of the cell with the highest neighborhood average. If there is a tie, your program must report only a single cell that ties the maximum value. I will post a solution consisting of all cells that tie the maximum value so you can be sure your solution is correct. The neighborhood of a cell is all cells that immediately border the cell, including the cell itself. For each cell, you must compute the average of numbers in the neighborhood. For example, in the following array:

unsigned int M[10000][10000];

the neighborhood of cell M[2][8] consists of the following cells:

```
M[1][7] M[1][8] M[1][9]
M[2][7] M[2][8] M[2][9]
M[3][7] M[3][8] M[3][9]
```

Because the array will be large, you will need to use dynamic memory allocation (keyword new) on the heap.

Be careful not to go out of bounds on the array when computing neighborhoods

Your program should take a command line argument indicating the number of threads that will be used.

The group that has the fastest program when run with the number of threads equal to the number of available cores on a chosen computer will win a prize (of non-monetary value)! So don't give away your speedy secrets.

The openMP wiki has a nice intro to using openMP. http://en.wikipedia.org/wiki/OpenMP

Note that you must have a GCC compiler version 4.3 or later to use openMP. The departrment's home.cs.siue.edu server has a sufficient GCC version. You can download

the newest version of GCC for mac and linux. You may have to install cygwin to do this with windows. Microsoft's compiler has it too.

Sample input files are attached

Your program will be graded on home.cs.siue.edu, so make sure it compiles and runs correctly there. Your group must make exactly 1 submission, and your group member names should be in a comment in the first line of your file. Also, you MUST submit a MAKEFILE. A sample makefile is provided, you can use this one, or edit it for your needs. Your submission should be zipped and submitted as single file.

What to Turn In:

You must turn in your source code, a Makefile that will compile your source code on home.cs.siue.edu, and a README file that contains the names of your group members, and any additional information about your implementation that you think I may need to consider while grading.

Turn in a single zip or tar.gz file.

The README file should be a PLAIN TEXT FILE!

Example Input and Expected Output:

Here are some sample runs of the program so you can see what the output looks like. Your output should look EXACTLY like mine, except that you will only show a single cell address with the largest average. The input file small3.txt contains two cells tying for the largest average

The following shows the expected output with the timing output. You output should match these. If you can beat these times, you are doing very well!

Output that does match the format shown below will recieve a grade of 0.

```
[marmcke@home parallel1]$ make
g++ -fopenmp -ggdb  matAverager.cpp -o matavg
[marmcke@home parallel1]$ ./matavg small3.txt 4
largest average: 7.66667
found at cells: (0,1) (0,2)
elapsed time: 0.000877142
[marmcke@home parallel1]$ ./matavg rand 4 10000 2000
largest average: 9240.89
found at cells: (8524,739)
elapsed time: 0.474431
[marmcke@home parallel1]$ ./matavg rand 1 10000 2000
largest average: 9240.89
found at cells: (8524,739)
elapsed time: 1.88899
[marmcke@home parallel1]$ ./matavg rand 4 10000 3000
largest average: 9374.89
found at cells: (4211,2329)
elapsed time: 0.713477
[marmcke@home parallel1]$ ./matavg rand 1 10000 2000 3
largest average: 9370.22
found at cells: (874,617)
```

```
elapsed time: 1.88954
[marmcke@home parallel1]$ ./matavg rand 1 10000 2000 4
largest average: 9246.11
found at cells: (9795,1926)
elapsed time: 1.89039
[marmcke@home parallel1]$ ./matavg rand 4 10000 2000 3
largest average: 9370.22
found at cells: (874,617)
elapsed time: 0.475229
[marmcke@home parallel1]$ ./matavg rand 4 10000 2000 4
largest average: 9246.11
found at cells: (9795,1926)
elapsed time: 0.473736
[marmcke@home parallel1]$
```