Programming Assignment 3 - OpenMP

Due Date: Tue 10/25

Background

A major design goal for pthreads was to provide maximum flexibility and control to the programmer. OpenMP, on the other hand, was driven by the objectives of both making multi-processing easily available as well as specifically supporting retro-fitting MP into the existing code base without damage to the existing sequential applications.

The purpose of this assignment is to contrast a pthreads implementation with a similar OpenMP application, compare their performance, and consider the trade-offs within the scope of a sample application.

Assignment

Using your original simplified AMR application from lab1, parallelize that program using OpenMP.

- Use the same parallelization strategies from lab2 (disposable and persistent threads).
- Make the number of threads the third run-time parameter in your OpenMP program, and test your program requesting 2, 4, 8, 16, 24 and 36 threads. (note that your program will need to verify the actual number of threads created.)
- Write a short report (no more than 3 pages) which provides:
  - comparative run-times measuring the convergence time for your lab2 pthreads programs and your OpenMP programs using data file testgrid_400_12206.
    - select values for epsilon and affect rate which will result in your slowest of the four programs running between three and five minutes.
    - measure convergence time with the method you feel most appropriate from among those used in lab2.
  - wall-clock run times using the unix time command from the command line (not the time system call).
  - actual -vs- requested OpenMP threads created
  - answers to the following questions:
    - Which threading mechanism, pthreads or OpenMP, provided the best results in your case?
    - Which threading mechanism, pthreads or OpenMP, was the easiest to implement?
    - Which threading mechanism, pthreads or OpenMP, would you be most likely to select for a similar application?
    - Under what circumstances would you speculate that the other mechanism, pthreads or OpenMP, would be preferable?
    - Were there any surprises you encountered in this exercise?

Input Data Format

Use the same input data files as lab1 and lab2.

Testing and Submission

Follow the testing and submission guidelines for lab2, using directory cse5441_lab3.

- Ensure the program can be compiled with "make", before submitting.
- Compile your programs with optimizer level3 (-O3) and (-fopenmp) option.
- Create a directory "cse5441_lab3". Within this directory, place:
  - all program files (.c or .cc files);
  - makefile
  - report in .pdf format
- If you are in 12:45 session use command: "submit **c5441aa** lab3 cse5441_lab3" to submit
- If you are in 2:20 session use command: "submit **c5441ab** lab3 cse5441_lab3" to submit