Programming Assignment 3 - OpenMP

Early Due Date (+5%): Mon. 11/4, 11:59pm
Final Due Date: Fri. 11/8, 11:59pm

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 (both disposable and persistent threads).
- Make the number of threads the third run-time parameter in your OpenMP program, and test your program to determne an optimal number of OMP 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.
    - use the values for epsilon and affect rate which you developed in lab1 to obtain a serial run-time of between three and six minutes.
    - measure convergence time with the method you feel most appropriate from among those used in lab2.
  - actual -vs- requested OpenMP threads created
  - answers to the following questions:
    - How would you characterize the computational work-load of our sample program?
    - Which threading mechanism, pthreads or OpenMP, provided the best results in your case?
    - Would you say that pthread is more flexible, less flexible or the same as OpenMP?
    - How well did your OpenMP programs meet the design criterion of preserving your original serial program? What constructs did you use that would cause your program not to compile with OpenMP disabled?
    - Explain the results you observed.

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.

- Before submitting ensure both programs (*disposable* and *persistent*) can be created with a single invocation of "make".
- Compile your programs with optimizer level3 (-O3) and (-fopenmp) option.
- Create a directory "cse5441_lab3". Within this directory, place:
  - all program files (.c and .h files);
  - makefile
- Use the OSC “submit” command (/fs/project/PAS1603/submit/CSE5441/project) to turn in your program. This assignment is “lab3.”
- Please submit your .pdf report via Carmen.