Programming Assignment 3 - OpenMP

Due Date: Tues., 11/13

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

Based on your original serial application from lab 1, as well as your pthreads parallel implementation in lab 2, create an OpenMP parallel version of your producer-consumer application.

- You may implement either a disposable threads or a persistent threads model.
- Make the number of threads a run-time parameter in your OpenMP program, and test your program requesting 2, 8, 16 and 32 threads. (note that your program will need to verify the actual number of threads created.)
- instrument your OpenMP program in similar fashion to lab 1 and lab2, using time(1), time(2) and clock(3) to obtain producer and consumer total runtime as well as overall runtime.
- Write a short report (no more than 3 pages) which provides:
  - a comparative summary of run-time results for each of the the producer, consumer and overall run-times for each of lab 1 (serial), lab 2 (pthreads) and lab 3 (OpenMP). (note: you should have 4 sets of results for OpenMP, corresponding to the different numbers of threads.)
  - your observations and analysis regarding your results.

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 “lab3.”

- Create a directory "lab3". Within this directory, place:
  - all source code files (.c an .h files);
  - makefile
    - ∗ your program should build with the command “make” with no parameters.
    - ∗ name your executable lab3_omp.”
    - ∗ your program should use the first positional parameter to specify the number of threads (as an integer) and use no other command-line parameters.
- submit your report in .pdf format via Carmen.