CS1645/2045: Introduction to High Performance Computing (Spring 2021)

Department of Computer Science, University of Pittsburgh

Assignment #3: Parallel MMULT, NN-SOFTMAX, & TRAP using OpenMP

Release: Mar 4, 2021 Due: 8:00 PM, Mar 22, 2021

Goal

Gain familiarity with the parallel thinking using POSIX Threads.

Description

- You should download the provided code that was posted on the course website. You are given the code for three serial applications:
  - Application 1: Matrix Multiplication (MMULT).
  - Application 2: The NN SOFTMAX.
  - Application 3: Trapezoidal Rule for Integral Approximation

1 Study these applications from internet sources to understand the algorithm of the serial execution.

- Application 1:
  - https://en.wikipedia.org/wiki/Matrix_multiplication.
  - https://en.wikipedia.org/wiki/Multiply-accumulate_operation.
  - In one word it will perform $Y = AW + B$
- Application 2:
  - https://en.wikipedia.org/wiki/Perceptron.
  - https://en.wikipedia.org/wiki/Softmax_function.
  - In one word it will perform following steps,

$$Y = \delta(WA), Y = Softmax(Y), L = Y - G$$

- Application 3: https://en.wikipedia.org/wiki/Trapezoidal_rule

2 The purpose of this exercise is to parallelize the three applications using the OpenMP library as we discussed in the class.

3 For your convenience, pieces of code that can be parallelized are located between the runtime measurement functions. So, you need to parallelize the code that it is between those functions (gettimeofday (...)/* START TIME */ - gettimeofday (...)/* END TIME */).

Project report:

The project report should be a well-presented technical report discussing your implementation. You should explain your solution incrementally (step by step), provide experiments to support your findings and analyze the results. The analysis is an important component of your report and you should include and discuss the speedup and efficiency varying the number of threads and the problem size. The written report should be four to six pages in length.

The structure of the report should be as follows:

- **Introduction**: A brief introduction describing the problem and the serial solution.
- **Your Solution**: In this section you should explain how you created your solution using the shared memory systems (OpenMP) to achieve the best performance possible.
- **Experimental Evaluation**: Your need to compare your OpenMP solution with the serial version showing the speedup and the limitations of the parallelism. You need to prepare several experiments (varying the number of threads and varying the problem size) and plot and discuss your findings.
- **Conclusions**: Write your thoughts and the conclusions about your parallel solution and compare it with the serial.

To submit your assignment:

1. Create a single file named `hw3_report_<username>` in PDF (.pdf) or Microsoft Word (.doc) format, reporting your findings. In this file you should include a brief description of the parallel implementation that you have made for each application along with the time comparison between the serial and the parallel execution (preferably a plot). **Do not forget to include your name and username (account name) in the file**.
2. One file for each application, namely *mmult_par.c*, *nn_par.c* and *trap_par.c*. Your code for all applications should compile and run on the linux.cs.pitt.edu machines.
3. Submit your assignments through the Web_base submission interface you have used for homework #1 (at the class web page `db.cs.pitt.edu/courses/cs1645/current.term`, click the Submit button). **It is your responsibility to make sure the assignment was properly submitted.**
4. Submit your assignment by the due date (**Mar 22, 2021 at 8pm**). There is *no late submission*.

Academic Honesty

The work in this assignment is to be done *independently*. Discussions with other students on the assignment should be limited to understanding the statement of the problem. Cheating in any way, including giving your work to someone else will result in an **F** for the course and a report to the appropriate University authority.