CSCI596 Assignment 1—Computational Complexity and Flop/s Performance

Due: September 8 (Wed), 2021, 11:59 pm

As a warmup, you will perform small hands-on exercises to get a feel for computational complexity and flop/s (floating-point operations per second) performance of a computer.

0. Preparation: Reading Assignment

Read the following materials to prepare yourself for the assignments, and for the entire semester at large.

0.1. Asymptotic complexity analysis

Read a lecture note (https://aiichironakano.github.io/cs596/AsymptoticAnalysis.pdf) and "Appendix A.2—Order Analysis of Functions" of Introduction to Parallel Computing" by Grama *et al*. (page 581 of the PDF file, to which the link is found at the course homepage, https://aiichironakano.github.io/cs596.html, under "Textbooks" heading).

0.2. Theoretical peak performance of a computer

Read a lecture note (https://aiichironakano.github.io/cs596/PeakFlops.pdf) to learn how to compute the theoretical peak floating-point performance of a computer.

Now, here is the actual assignment: Submit your answers to the following two questions.

1. Measuring Computational Complexity

Use the data file, `MDtime.out`, in the assignment 1 package. In the two-column file, the left column is the number of atoms, N, simulated by the `md.c` program, whereas the right column is the corresponding running time, T, of the program in seconds. Make a log-log plot of T *vs*. N. Perform linear fit of $\log T$ *vs*. $\log N$, *i.e*., $\log T = \alpha \log N + \beta$, where α and β are fitting parameters. Note that the coefficient α signifies the power with which the runtime scales as a function of problem size N: $T \propto N^{\alpha}$. For detail, see slide 34 in https://aiichironakano.github.io/cs596/01MD-VG.pdf). ***Submit (i) the plot and (ii) your fitted value of α.***

For this and subsequent assignments, you need to use a scientific plotting software like Grace, Origin, Kaleidagraph, Gnuplot, Matlab, Mathematica and Excel. Please make sure that you are familiar with one such software. For this assignment, you also need to use the least-square fitting feature of your plotting software. In case you cannot find such feature, you can do it yourself following the lecture note on "least square fit of a line" (https://aiichironakano.github.io/cs596/LeastSquareFit.pdf).

2. Theoretical Flop/s Performance

Suppose that your computer has only one octa-core processor (a processor equipped with 8 processing cores) operating at a clock speed of 2.3 GHz (*i.e*., clock ticks 2.3×10^9 times per second), in which each core can operate 1 multiplication and 1 addition operations per clock cycle using a fused multiply-add (FMA) circuit. Assume that each multiply or add operation is performed on vector registers, each holding 4 double-precision (*i.e*., $4\times64 = 256$ bits) operands. ***What is the theoretical peak performance of your computer in terms of double-precision Gflop/s (gigaflop/s or 10^9 flop/s)? Submit the computed number.***

(Optional) A program named lmd_sqrt_flop.c is provided in the gzipped tar archive, `cs596-as01.tar.gz`, on Blackboard, along with instructions to compile and run the program in

RAEDME file.* This is a linked-list-cell molecular dynamics program, in which sqrt() function is implemented using a polynomial for counting the number of floating-point operations (see the lecture note on "Arithmetic implementation of sqrt() and floating-point performance"; see https://aiichironakano.github.io/cs596/Sqrt.pdf. ***Compile and run the lmd_sqrt_flop.c program on a computer of your choice, and report the flop/s performance you get. Better, answer how many % of the theoretical peak flop/s performance of the computer you achieved?***

* To extract the files from the archive, type `tar xvfz cs596-as01.tar.gz`.