ASTR415 Fall 2010 FINAL TERM PROJECT

Due Dec 9, 2010

The final term project is designed to familiarize you with parallel computing using OpenMP and benchmarking. You can choose to parallelize one of the two following codes that you have already coded: PS1 (benchmarking code) or PS4 question 2 (Monte Carlo integration). The second option is probably more rewarding but slightly more difficult. For your convenience, below I re-propose the two questions (choose one of these two):

a) Write a program to time the CPU expense of double-precision raise to the 0.5 power (`pow(x,0.5)` of $x = 1.1$. Repeat the operation n times, where $n = 10^7, 10^8, 10^9$. Your program should output the final value obtained after the three given values of n, and the CPU time required to complete them.

b) The total mass M of an object of density ρ is given by

$$M = \int_V \rho \, dx \, dy \, dz,$$

where V represents the volume of the object. Using simple Monte Carlo integration, write a program that computes M and its estimated error σ_M if $\rho = 1 + x^2 + 3(y + z)^2$, where the volume of the object V is defined by $x^2 + y^2 + z^2 \leq 9$, $x \geq 0$, and $y \geq -1$. Plot M with errorbars σ_M as a function of the number of points N used in the Monte Carlo integration, for $N = 10^5, 10^6, 10^7$.

Questions:

1. Plot the execution time as a function of the number of cores (as many as your computer has) for the three given values of n.
2. Does the speedup scale linearly with the number of cores? What is the loss of speed due to OpenMP overhead?
3. Estimate the number of floating-point operations your code carried out on average per second.

Hints: Use the links on the course web-page to see the solutions of PS1 and PS4 (with source codes) and look at some examples and tutorials on how to use OpenMP to parallelize a code, also available on the web-page. OpenMP allows you to add extra commands to the source code with directives to parallelize loops on a shared memory computers (for example a desktop with multi-processors and/or CPU with multi-cores). When you compile the code you need to use the option -openmp, or equivalent (look the manual page for the compiler you use). Without the openmp option, the OpenMP directives are treated as comments by the compiler. You need access to a desktop with $N \geq 2$ CPUs or cores. Your laptop may have a dual core processor, the workstation ursa.astro.umd.edu has 2 CPUs (single core), galileo.astro.umd.edu (my desktop computer) has 2 quad cores (2x4=8 cores). Try to use the computer that has the largest number of cores! On linux machines (most in the astro department) you can check how may cores are available on a host by viewing the file /proc/cpuinfo. Have fun, and start early: not late term project will be accepted.