Chad Bustard

Final Project Proposal

For my final project, I propose to use OpenMP to solve a set of partial differential equations on a grid. The method of finite differences is the most widely used method to solve partial differential equations sequentially on a grid; however, this can be quite computationally expensive. Therefore, I will hope to gain a reasonable speedup by implementing various parallelized algorithms using OpenMP on Euler, and I will test which algorithms give me the best speedup and why.

Why I've chosen this project: My research in the physics department pertains to solving steady-state (and in the future, time-dependent) galactic wind equations. Galactic winds are essentially galactic-scale versions of the more well-known solar wind. The hot wind particles are driven outward by supernovae or active galactic nuclei (AGN) in a galaxy and serve to expel mass and energy from a galaxy. This is a form of feedback for the galaxy in that, if the galactic outflow expels enough mass from a galaxy, there is no longer enough mass to make stars, and star-formation is quenched. Therefore, these processes are very important for modeling galaxy evolution. The most common "big" models of galaxy formation utilize OpenMP, though I believe some are now being converted to CUDA, as well, which was a large reason why I decided to take this course.

Galactic winds are generally described by fluid equations (many times with a complicated magnetic field structure affecting the motion of the wind particles), and these equations are partial differential equations with various boundary conditions. Therefore, I hope that this final project will teach me the basics of solving partial differential on a grid in a parallelized manner and that I can use this knowledge in the future in my own galaxy formation research.

The details: I will center my project about solving the Poisson equation with various Dirichlet boundary conditions. Sequentially, this equation can be solved using the Gauss-Seidel method, which updates approximations to the function in question at each grid point until the function converges to some value at each point. The condition for convergence is specified by a required accuracy, i.e. a tolerance, where if the difference in values between two iterations of the function are less than that tolerance, then we say the function has converged. The initial values at each grid point will be random

First, I will implement this sequential algorithm and compare it to the analytic solution, which I can retrieve from Mathematica. I will then implement a few parallel algorithms and compare the speedup and results of these algorithms to the sequential algorithm. The parallel algorithms I propose will all be variations of a parallel Gauss-Seidel method, one with the so-called wavefront computational scheme and one without the wavefront scheme.

I expect, from a preliminary reading of literature on the subject (see http://www.hpcc.unn.ru/mskurs/ENG/DOC/pp12.pdf), that the parallel Gauss-Seidel implementation will present a number of complications to gaining a desired positive speedup, including challenges with load balancing and excessive synchronization.

Proposed Goal: To implement a few variations of the parallel Gauss-Seidel method for solving the Poisson equation with various Dirichlet boundary conditions and to report a time analysis for each method compared to the sequential Gauss-Seidel method. The time analysis should contain plots of computation time vs number of grid points for each parallel algorithm and the sequential Gauss-Seidel method.

Project Management: I will be the only one working on this project.

Timeline:

11/19: Finish implementing sequential Gauss-Seidel method and test versus analytic solutions

11/26: Finish implementing parallel Gauss-Seidel method and a few variants that hopefully lead to greater speedup than the naïve implementation, which I expect to actually be slower than the sequential algorithm due to excessive synchronization

12/7: Finish implementing wave method and make plots/tables of time analysis comparing all methods used to solve the PDE.

Time-permitting: Now apply the Gauss-Seidel method, both sequential and parallel, to a steady-state galactic wind equation from my own research. I will likely do this not for a PDE but for an ODE.

12/21: Code and final report due

Reference: http://www.hpcc.unn.ru/mskurs/ENG/DOC/pp12.pdf