Members

Jasmine Brewer, Samuel Elliott, Matthew Gross, Aniq Shahid, & Li-yin Young

Optimizing Hydrodynamic Simulations of Quantum Fluids

CSCI 4576/5576 Project Proposal, Fall 2014

**Introduction**

The lattice Boltzmann method was first popularized for its success in mainstream computational fluid dynamics, but has recently also emerged as an important tool in the fields of physics, biology, and chemistry for understanding complex fluids. Unfortunately, the range of physical problems which can be solved with the algorithm is currently severely limited by its high memory and computational requirements. The capacity of the lattice Boltzmann method to solve scientific problems has therefore become an important problem in high performance scientific computing. The proposed project intends to tackle this problem directly by optimizing an existing scientific code that simulates the evolution of a quantum fluid for more efficient and scalable overall performance.

**Problem**

The naïve implementation of the lattice Boltzmann method which has been previously implemented is overly restrictive on the simulation sizes achievable with available resources, due at least in part to lack of parallelism and inefficient data and memory handling. We propose to remedy these problems by implementing following:

1. **Single-Core Optimization:** We will vectorize computations and adopt blocking wherever these result in improved performance and reevaluate data structures to improve memory-usage.
2. **OpenMP:** We will parallelize data-independent sections of the single-core optimized code using OpenMP.
3. **MPI:** We will implement MPI for running OpenMP-implemented functions in parallel and handling intercommunication on distributed-memory architectures.
4. **Data Handling:** We will port IO to HDF5 to improve data write times, implement data compression, and improve data processing.

We implement these techniques and methods with the goal of producing an output code which can solve scientific problems that the input code could not. With this in mind, we intend to demonstrate our achievement by producing the following:

1. **Benchmarks** indicating the performance enhancement due to solving each significant sub-problem
2. **In-depth profiling analyses** indicating that we understand precisely the reasons behind our computational and memory performance.
3. **Scientific Results.** (I AM SOLICITING OPINIONS AS TO WHETHER THIS IS A GOAL THAT WE AS A TEAM WOULD LIKE TO ENDORSE OR NOT!)We will finally demonstrate our code’s scalability and scientific capacity by simulating decaying turbulence, a simple-to-implement but computationally-intensive and interesting physical system.

**Simple explanation of the problem you want to solve, what techniques and methods you plan to implement. What input data you plan to use and what kind of output data you intend to produce.**

**Analysis**

Performance improvement will be gauged by running the code on INTEL Phi, and doing benchmarking as well as detailed profiling at the beginning of the project and after each of the above proposed implementations. Profiling will include MFLOPS, time taken per function and total execution time, and will be done using PerfSuite and Tau.

As mentioned, we will take a modular approach to improving the performance of existing code. We will begin by optimizing data structures and basic iteration techniques. This will be followed with an iteration that focuses on single core optimization using OpenMP. Another tier will then involve MPI implementation in order to spread threading to multiple cores/nodes. Each iteration will be compared to the original code performance so as to evaluate exactly how effective our changes are and provide us with insight as to what could still be improved. By adopting this incremental development methodology, we aim to be able to correct mistakes and gain foresight into future changes in order to maximize both efficiency and code development.