Status Report: December 4^{th} 2017

Benchmarking different hardware architectures using a parallel implementation of the N-Body Simulation algorithm

Jan Mrázek

CIS 750 - Advanced Computer Architecture Experiments

1 Gantt chart of estimated schedule

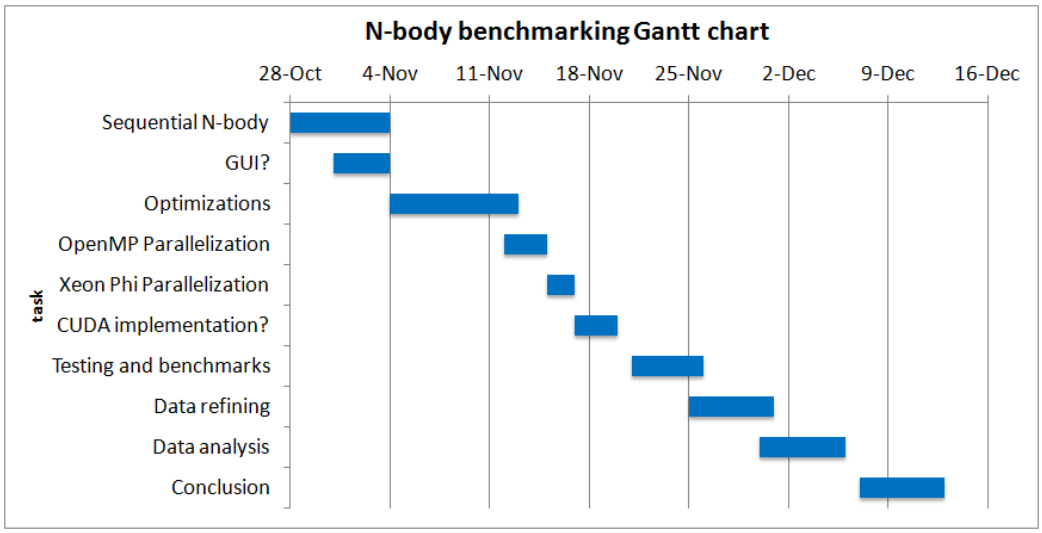

Figure 1: Gantt chart of an estimated project working schedule

2 Tasks completed to date

Since my last status report on November 13th, I managed to do a fair bit of work, though I did run into some problems.

After further tuning the overhauled `updatePositions()` method and integrating OpenMP functionality into my code, I took a closer look at redesigning the code once again to take an advantage of CPU caching.

I ended up changing my whole approach to storing data in memory. Instead of keeping on using an object-oriented "array of structures" kind of design, I rewrote the code to represent a "structure of arrays." In reality, I got rid of nearly the entire class structure of the code, and instead used dynamic arrays to store the attributes of all bodies.

After running quick tests, I found that this improved version of my simulation runs about 10 to 15 percent faster then the object-oriented one, depending on the data size (i.e. body count and number of iterations).

Next, I changed the code for use on the Intel Xeon Phi coprocessor. As the Phi largely uses OpenMP, I didn't have to change the structure of my

code very much. I mainly added some additional compiler directives that are needed to use the device. I also experimented with memory alignment to take advantage of automatic vectorization.

3 Tasks currently underway

Then I begun the benchmarking part of my final project. First, I found out that the OrangePi One microcomputer that I've got running at home can't be accessed via `ssh` for some reason. As I'm 5,000 miles away from the device and have no way of finding out what's wrong with it, I've decided to skip benchmarking my code on it for now.

I used K-State's Beocat supercomputer to run most of my benchmarks. I wrote several scripts that would take care of running my programs with varying numbers of threads and bodies. Since Beocat wasn't under very heavy load, I managed to run most of my benchmarks in a couple of days, though some are still stuck in the job queue.

For my Xeon Phi benchmarks, I used a Xeon Phi 7120X (Knight's Corner architecture) in one of my home university's computing nodes.

I gathered all the data from my benchmarks and have started to sift through it, sanitizing it by removing obvious outliers and other statistical anomalies. I haven't taken a very close look at the analysis of my results yet. This task is still currently underway, as I have not performed any benchmarks on the ARM-based device, which I will try to fix sometime in the near future.

4 Tasks soon to start

Soon, I would like to get around to diagnosing the problem with my OrangePi One, as I'd like to run the rest of my benchmarks on it.

In my Gantt chart at the beginning of this document, you can also spot a an entry for a CUDA implementation, albeit with a question mark. I might not be able to implement a CUDA version of this project due to certain time constraints. I will probably take a look at producing an implementation after I benchmark my OrangePi.

I will soon finish analyzing all the data I gathered so far during the (still ongoing) benchmarking phase. I would also like to start working on my final research paper that is supposed to be the output of this whole project.