Parallelization of Cross Beam Energy Transfer Program

Andrew Sexton
CSC 458
Term Project
March 20, 2020

Description:

The Cross Beam Energy Transfer Program (CBET) is a large scripted program written in the interpreted language Yorick developed by Dr. Adam Sefkow at the Lab of Laser Energetics (LLE). The script makes substantial use of multi-dimensional arrays, which are natively supported in Yorick. As such, there are many instances of nested for loops iterating over various parts of the arrays performing calculations.

As best I understand what the code does, it simulates two (or more) laser beams intersecting and computes the two dimensional ray traces and the interactions between them.

There are two parts to increasing performance of the supplied code. First is vectorization, which Yorick also supports natively. This is when instead of explicitly iterating over each element in an array, you can write the code to implicitly iterate. This allows Yorick to utilize optimizations it has for performing these computations. However, this is only useful for applying functions to the matrices. For example,

$$y = a*x^2 + b*x + c$$

where x is a vector, is easier to represent in this form than the equivalent loop form,

```
for i in range(length of x)
    for j in in range(length of x[i])
        // calculations on each element of x
```

When the elements of the matrix are used in calculating values for other matrices based on their value however, vectorization does not work as each element must be inspected at each step before computation can be done.

Instead, parallelization using message passing (MPI) is required (instead of other parallization strategies like threading by the request of Dr. Sefkow). Yorick does have an MPI package available, which makes this option possible without modifying the language source code.

Goals:

Upon running the software, the program outputs a text block with all the timing values of each section of the code. There are thirteen sections printed, but 20 represented in the code. This allows for an easy look at which sections are the most time consuming without any modifications.

On my home desktop (AMD 6 core processor) running Ubuntu 18.04 LTS, the software takes approximately 145-150 seconds to complete.

There are two main sections that make up the bulk of the time. First is RAY LOOPS SUM, which takes about 25 seconds to run (more than twice the next longest time). The second is the worst case, which is the CBET Gain Calculations, which takes ~120 seconds. This is roughly 80% of the run time of the entire program.

Therefore, my primary goal is to speed the CBET Gain calculations up first, and then move onto the other sections in order of run times.

Before any of that can be done though, I must learn how to use the Yorick language (as I have never heard of it before) and understand what is happening in the code.

It is also possible that instead of optimizing the Yorick code, I could instead convert it do C++. This would allow not only MPI support, but also CUDA support, which could achieve large speed ups on the matrix calculations with or without parallization (although CUDA support parallelization on the GPU).

I would need to discuss this more with Dr. Sefkow before pursuing though.

Currently I am the only person working on this project.

Related Materials:

Yorick Documentation: https://software.llnl.gov/yorick-doc/

Yorick Source code: https://github.com/LLNL/yorick/