

Transition into OpenCL

Justin McKennon

September 18, 2009

1 Introduction to OpenCL

OpenCL stands for Open Computing Language. The purpose of OpenCL is to provide a uniform programming environment that will allow software developers to write code that can run on a variety of architectures. Currently, if one were to program a GPU, and then want to run this same program on a CPU, two completely separate programs would have to be written. OpenCL is not hardware dependent and greatly increases the versatility of code as it is now able to run on virtually any type of architecture. This creates the ability for programmers to move between environments and execute code on any type of hardware supported by OpenCL.

2 History

This section is more of a piece of sarcasm. The true difficulty in using OpenCL is that it is literally brand new; not even 2 months old yet. It is in this, that I will face the most difficult of problems. In other environments, a user can simply browse the internet and/or ask an "expert" about a possible solution to a problem. With OpenCL, there really aren't more than a handful of "experts" in the world and their expertise is very limited, providing VERY few resources for developers who are having issues with their code.

3 My approach

With the recent breakthrough in the GPU code I have previously been working with, I am now able to obtain a 50x speed up with the Teukolsky equation running on the GPU as opposed to running on the CPU. Before this breakthrough, the GPU was only able to perform at half the speed of the CPU. The GPU has even double the computing power (in terms of speed) of even the PS3 cluster (50x speed up as opposed to a 24x speed up).

OpenCL has the potential to be the new standard environment for programming. By moving the GPU code into OpenCL (which, is no easy task) I will effectively be able to run my code on any GPU instead of being forced to work directly with the Nvidia CUDA code. I will then be able to experiment with many different GPU's and can compare the efficiency of different GPU's. Once I have successfully been able to move the GPU code into OpenCL, I can

then begin designing the quadruple precision library to really show the true power of GPU-Based super computing in the world of numerical relativity.

At this time, I am attempting to learn the programming practices associated with working in OpenCL. It is drastically different than the environments I am used to working with and there is much work to be done. I've isolated the source term code from my previous GPU program so that only the actual source term itself is being done on the GPU, eliminating the FORTRAN and other baggage that has been weighing the program down. It is from here, that I begin my journey into OpenCL.