Reflections and Experiences with OpenCL

KTH DD2360 HT20

Pontus Asp

December 13, 2020

1 Thoughts on OpenCL compared to CUDA

When getting started with OpenCL it is easy to be overwhelmed by the amount of boilerplate code that needs to be written, and what function calls are needed for the setup etc. It is simply a steep learning curve to learn about for instance what contexts and command queues are, or about what the differences between devices and platforms, programs and kernels are. However the fact that OpenCL is so general and works on so many platforms is a very strong argument to using it. CUDA on the other hand, only works on Nvidia GPUs, and this could be both a good, and a bad thing. In my opinion, CUDA is much easier to get into, with less to remember than OpenCL, and with less boilerplate code. CUDA is also made specifically for GPUs, which is what makes it able to be a bit more simple than OpenCL since it has to take less platforms into account. With all this in mind, I do prefer using CUDA, but will probably use OpenCL since it works on more platforms.

Another thing to also consider is the performance differences between CUDA and OpenCL. Of course, CUDA should be faster on a Nvidia GPU since it is Nvidia's own API to their drivers but the question is - in reality, does it make a difference? I tested this by running the two particle simulations we created since I made them both equivalent on the CUDA and OpenCL implementation. My results actually surprised me. The OpenCL performed better than my CUDA variant when using more iterations and particles. But on fewer elements and iterations my CUDA implementation took the lead. I think the CUDA variant however has more room for optimization and if done properly could beat OpenCL at the higher amounts of particles and iterations.