Personal Inquiry - Introduction to GPGPU (General-Purpose computing on Graphics Processing Units) Programming: Annotated References

Steven Twist
MScCAVE
stevetwist@gmail.com
f9097438@bournemouth.ac.uk

May 12, 2010

Please note: The target platform I have been developing for is an NVIDIA GPU. Therefore many of the following resources are going to be biased towards NVIDIA GPU programming.

Also, I should point out that whilst the concepts presented in my lecture don't require any substantial programming knowledge to understand, GPU programming is an advanced topic, and I would recommend becoming very comfortable with C/C++ programming for the CPU before diving into GPU programming with the following resources.

KHRONOS, 2010. *OpenCL Overview* [online].
Available from: http://www.khronos.org/opencl/ [Accessed 5th May 2010].

If you're interested in getting involved with parallel computing on the GPU, you're going to need to learn an API capable of targeting programs for the GPU. There are a few out there (CUDA is well known for the NVIDIA architecture), but I personally like OpenCL because it targets multiple platforms - multi-core CPUs, NVIDIA GPUs, ATI GPUs etc. The official OpenCL website should be your starting point for anything OpenCL related. It contains an OpenCL overview presentation, developer resources, and most importantly, it contains the OpenCL Specification:

KHRONOS, 2009. *OpenCL 1.0 Specification* [online].
Available from: http://www.khronos.org/registry/cl [Accessed 5th May 2010].

The official specification for OpenCL 1.0. A complete reference of the OpenCL API, regardless of which platform you're running it on (NVIDIA GPU, ATI/AMD GPU, multi-core CPU etc.). It contains full documentation of all OpenCL API calls, and the OpenCL C Programming Language for writing kernels (programs that execute on the GPU). Given it's depth, it's actually surprising easy to read in a linear manner. This was my number one point of reference once I'd picked up the basics.

However, the specification could be overwhelming for those entirely new to OpenCL. Therefore, I'd recommend the following video tutorials as a first step into OpenCL, with the Specification as a close companion along the way.

NVIDIA, 2009. *Introduction to GPU Computing and OpenCL* [online]. Available from: http://developer.download.nvidia.com/CUDA/training/ Intro_to_GPU_Computing_with_OpenCL.wmv [Accessed 5th May 2010].

This 1 hour webinar by NVIDIA gives an introduction to OpenCL for GPU programming. Whilst it is NVIDIA biased, I believe the vast majority of the content presented is generic enough to be applicable to most mainstream GPU architectures (NVIDIA or ATI/AMD). I found it to be a very interesting lecture, which whilst a little dry, was fairly easy to follow for an advanced C/C++ programmer. It's a good first step into OpenCL and GPU programming, however it doesn't present quite enough information to begin writing your own OpenCL programs.

MacResearch, 2009. *OpenCL Tutorials* [online]. Available from: http://www.macresearch.org/opencl [Accessed 5th May 2010]

A series of video tutorials covering OpenCL for NVIDIA GPU programming. They start with the very basics, and work up to some quite advanced topics, such as memory management. The basics are generic enough to be applicable to any platform. The more advanced topics focus on memory management, and that is quite heavily linked to NVIDIA hardware.

These videos are very well presented, and deliver a lot of useful information. However, don't take everything presented as gospel, as there are a few times where the presenter's explanations aren't quite accurate, either through a lack of understanding on his part, or more commonly though hardware developments that, on newer NVIDIA cards, relax some of the constraints presented in these videos. Notably, his explanation of "blocking" memory reads/writes is entirely wrong - refer to the official OpenCL 1.0 Specification for an accurate explanation.

Due to the occasional inaccuracies, my advice would be to watch these videos to get a general understanding of the concepts at hand, but then refer to the more complete written documentation (the official OpenCL 1.0 Specification above, and the documentation included with the NVIDIA OpenCL SDK below) before starting development on your own projects. Covering the information multiple times really helped me get an understanding of OpenCL and NVIDIA architecture. The MacResearch tutorials provided enough of an explanation to understand and benefit from the written documentation (which otherwise can be quite overwhelming), and the written documentation helped bring to my attention and clear up any inaccuracies in the MacResearch tutorials.

It is important to note that in these videos, the presenter is running Mac OS X Snow Leopard, which comes with OpenCL as part of the standard installation. As such, it still doesn't present quite enough information to begin writing your own OpenCL programs on operating systems where the necessary header files, libraries and drivers aren't installed as standard. For NVIDIA hardware, those necessary header files, libraries and drivers are available as part of the NVIDIA OpenCL software development kit (SDK):

NVIDIA, 2010. *OpenCL* [online]. Available from: http://www.nvidia.com/object/cuda_opencl_new.html [Accessed 5th May 2010]

This is NVIDIA's central page for their OpenCL developer resources. Near the top of the page (as of May 5th 2010) are a series of links to different resources. The most useful of those is the NVIDIA OpenCL SDK, which unfortunately is not clearly labelled, and is actually somewhat buried as part of the "GPU Computing SDK".

The current version of the SDK can be found on this page:

http://developer.nvidia.com/object/cuda_3_0_downloads.html

Find your target OS, and then download either the 32-bit or 64-bit version of the "GPU Computing SDK code samples". It actually contains a lot more than just code samples - it contains a wealth of documentation, plus all the libraries and header files you'll need to write programs with OpenCL.

You'll also need the latest display drivers for your graphics card, available from NVIDIA's main drivers page:

http://www.nvidia.co.uk/Download/index.aspx?lang=en-uk

The SDK documentation includes how to setup OpenCL on your machine, along with introductory guides, and advanced optimisation guides. It is extremely important to note that whilst OpenCL is cross-platform, optimising OpenCL code is very much platform dependent. Therefore, the knowledge on optimisation presented in the NVIDIA SDK documentation is only applicable to NVIDIA hardware, and even then, is likely to continually change as new generations of technology are released.

The included sample projects are initially useful to test your installation of OpenCL - if the project compiles, you're all set up and ready to write OpenCL programs! If it doesn't, you know that something is wrong with your setup, because the code has been tested by NVIDIA. Once you're up and running, the example projects contain a huge amount of knowledge. I found the OpenCL/OpenGL interoperability examples extremely useful for my work.

The amount of information provided in the above resources is huge, and I'd advise anyone looking into GPU programming with OpenCL for the first time to not get overwhelmed. Start simple with the video tutorials mentioned, and then progress onto the more advanced written documentation. I found it only took about a week of full-time work to go from no knowledge of GPU programming and OpenCL, to having programs that I'd written from scratch, running on my NVIDIA GPU using OpenCL.

I wish anyone getting into GPU programming with OpenCL the best of luck!