CUDA Programming on NVIDIA GPUs
Mike Giles

Practical 1: Getting Started

This practical gives a gentle introduction to CUDA programming using a very simple code. The main objectives in this practical are to learn about:

- the way in which an application consists of a host code to be executed on the CPU, plus kernel code to be executed on the GPU
- how to create different kinds of executable using the Makefile
- how to copy data between the graphics card (device) and the CPU (host)
- how to include error-checking, and perform simple debugging using emulation

The practicals are to be carried out on the `skynet` cluster. Before starting, please read the notes at
http://www.maths.ox.ac.uk/~gilesm/cuda/skynet_notes.pdf
(If you are reading this PDF document online, the link above should appear in blue and you can click on it to go to the notes.)

What you are to do (working in pairs if you wish) is as follows:

1. Use the command

   ```
   cp -r ~mgiles/cuda_course/prac1 ~/prac1
   ```

   to copy the directory `prac1` from my account to yours.

2. Following the `skynet` notes, produce the four different versions of the `prac1a` executable, and run each one.

3. Read through the `prac1a.cu` source file and compare it to the `prac1b.cu` source file which adds in error-checking.

4. Look at `Makefile` to understand how it works, and then modify it to produce executables for `prac1b`.

5. Try introducing errors into `prac1b.cu`, such as setting `nblocks=0`, and see what happens.

6. Add in `printf` statements in the kernel code in `prac1b.cu`. to print out the values of `threadIdx.x` and `blockIdx.x`.

   What happens when you try to compile it without emulation?

   What happens when you compile and run it with emulation?

7. Copy `prac1b.cu` to `prac1c.cu` and modify it to add together two vectors which you initialise on the host and then copy to the device. This will require additional memory allocation and two `memcpy` operations to transfer the vector data from the host to the device.

   Make sure you get the correct results, and use debugging in emulation mode if necessary to figure out what is going on.

8. If you have spare time, look at the NVIDIA SDK examples in
   `/opt/cuda/2.3/sdk/C/src/`

   You can use a web browser on `skynet` to access the HTML file
   `file:/opt/cuda/2.3/sdk/C/src/ReleaseNotes.html`
   or look at online info at
   `http://www.nvidia.com/object/cuda_sdks.html`