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 copy data between the graphics card (device) and the CPU (host)
- how to include error-checking, and printing from a kernel

The practicals are to be carried out on Google Colab which provides access to T4 GPUs through the execution of commands within a notebook.

What you are to do is as follows:

1. Click on the link in the course webpage to the Google Colab notebook.
2. Carefully follow the instructions in the notebook.
3. Look at the difference between the `prac1a.cu` and `prac1b.cu` source files to see the way in which error-checking is added in `prac1b.cu`
4. Try introducing errors into both `prac1a.cu` and `prac1b.cu`, such as trying to allocate too much memory (e.g. by specifying an enormous value like `(long long) 500000000000`), or setting `nblocks=0` or `nthreads=10000`, and see what happens.
5. Add a `printf` statement to the kernel routine `my_first_kernel`, for example to print out the value of `tid`. Note that the new output may be written to the screen after the existing output from the main code, because it gets put into a write buffer which is flushed only intermittently.
6. Modify `prac1b.cu` 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.