CUDA Programming on NVIDIA GPUs
Mike Giles

Practical 9: pattern-matching

This application is non-numerical, pattern-matching looking for certain keywords in a long stream of text. This could have lots of applications in the humanities and social sciences, and is also possibly relevant to some bioinformatics applications.

Build with Nsight, or copy the directory `prac9` from my account to yours, and then make and run the application. In this case, I have written the Gold code for CPU execution, and your task is to write the corresponding GPU code.

The "words" we are looking for have 4 characters which are held in a single 32-bit unsigned integer so that a single integer comparison checks for a match with a corresponding integer holding 4 characters of the input text. Because the beginning of each word in the text is not necessarily aligned with the beginning of an integer, the Gold code has to also consider 1, 2 and 3-byte offsets.

In outline, my suggested GPU implementation approach is as follows:

- copy text and words to device memory
- each thread in a block responsible for one test word
- load a chunk of text into shared memory, with each thread doing one text word
- compute 3 offsets, and store them also in shared memory, with each thread doing 1 word in each offset
- loop through this chunk of text looking for matches
- when necessary load in the next chunk of text and repeat
- at end of block, store number of matches in a global array

Initially, develop the implementation for a single thread block. In a real application, different thread blocks could handle completely different sections of the input stream.

Looking at the likely performance, each word of input text gets moved once from the host to the device, and then once from the device memory to the SM in the GPU which then processes it. Thus, this minimises the bandwidth requirements.

If the number of words being matched against is of the order of 64 or more, I suspect the application will be dominated by the computational cost. Below that, it may be dominated by the host-to-device bandwidth.