CUDA Programming on NVIDIA GPUs
Mike Giles

Practical 10: auto-tuning

The idea of an auto-tuner is that an application has a number of parameters, each of which can take a small number of values, and the auto-tuner determines the best combination to either minimise the execution time or maximise some user-defined FoM (Figure of Merit).

An Oxford student, Ben Spencer, has developed an auto-tuner, Flamingo, which is novel in two respects:

- It is very flexible, in that it can control the compilation of an application as well as providing run-time arguments to the application. It can be used with applications written in any language; current examples include C/C++, MATLAB and CUDA.
- It uses brute-force optimisation but reduces the cost by exploiting user-supplied information about which parameters (or subsets of parameters) can be optimised independently. A good example of this is the optimisation of a CUDA application in which the blocksize for each kernel can be optimised independently.

This practical is not included in the Nsight projects. Copy the directory `prac10` from my account to yours and look in subdirectory `Autotuning/examples/laplace3d`. This has a testcase which is the same as the code in Practical 3 except that the "Gold" CPU code and the `#define` declarations of `BLOCK_X` and `BLOCK_Y` have been removed. The latter two are now set by the `Makefile` under the control of the auto-tuner.

Look at the configuration file `laplace3d.conf`. This specifies the parameters to be optimised (and their possible values) and how to compile and run the application. Note how the values of `BLOCK_X` and `BLOCK_Y` are passed to the `Makefile`, and look in `Makefile` to see how they are then passed to the compiler.

Run the auto-tuner using the supplied job script.

It will run for a couple of minutes and generate lots of output in the file `tune_results`. Right at the end, it will print out the optimal parameter values. Look also at the output file `results/laplace3d.csv` which show the 9 different combination of parameter values tested. For each combination, the test is run 3 times and the minimum time is used – this is specified in the configuration file.

Once you understand this example, you might like to develop your own based on any of the other CUDA applications you have developed in this course.