High-performance scientific code for GPUs
Andrew Haigh, Computer Systems Group, andrew.haigh@anu.edu.au

1 Graphics Processing Unit (GPU)
- Thousands of threads executing the same instructions
- Multi-level memory hierarchy
- Context switch-free hardware instruction scheduling
- But very simple cores: no branch prediction, no out-of-order scheduling
- Limited on-chip resources to be shared by threads

![GPU Architecture Diagram]

2 Research objective
Is it possible to use high-level representation to generate high-performance scientific code for GPUs?

Our intention is use this work to generate high-performance code for performing ultrasound simulation on GPUs.

3 Auto-tuning
- Auto-tuning refers to software that is able to modify itself to maximise performance
- There are many approaches based on empirical, model-driven and machine-learning based techniques
- GPU auto-tuning is very difficult due to the sensitivity of the performance on many interrelated factors

4 Current work
- A framework that modifies codes to retain a given portion of the data in the on-chip shared memory, for a restricted subset of CUDA C codes.
- In addition, the mapping between work and threads can be controlled.
- For all possible configurations, performance is predicted using a very simple model of the GPU architecture, and the best approach is selected.

![Runtime Comparison Diagram]

5 Future work
- Translation from high-level abstract representation
- Incorporation of a more sophisticated model of GPU warp scheduling
- Focus on which loops to parallelise, not just how to parallelise

References