Parallel Computing and Graphical Processing Units

Daniel Playne
Graphics Cards

NVIDIA GTX690
NZ - $1300
Graphics Cards

These cards were designed to render graphics for computer games.
Graphical Processing Units

Graphical Processing Units or GPUs are the processors inside these cards.
Graphical Processing Units

In the early 2000s GPU design was moving towards a more general computing architecture and had overtaken CPUs in performance.

This gave rise to GPGPU

General Purpose computing on GPUs.
General-Purpose computing on Graphical Processing Units

2002 – OpenVIDIA developed to run computer vision algorithms on GPUs.

2007 – NVIDIA releases CUDA 1.0

2010 – Tianhe-1A with NVIDIA GPUs (M2050s) ranked 1st in the TOP500 (2.57 petaflop/s).

2012 – Titan with NVIDIA GPUs (K20X) ranked 1st in the TOP500 (17.59 petaflop/s).
GPGPU Applications

Lattice Simulations
- Field Equations
  - Cahn-Hilliard
  - Ginzburg-Landau
  - Lotka-Volterra
  - Classical Heisenberg
  - Shallow Water
- Memory Optimisation
- Optimised 2D & 3D lattices
- Real-time visualisation
- Lattice Geometries
- Complex Data Types
- Multi-GPU Simulations
- GPU-Cluster Simulations
GPGPU Applications

**Cahn-Hilliard**
Phase Separation
50x – 120x faster

**Time-Dependent Ginzburg-Landau**
Superconductivity
50x – 100x faster

Leist, A., Playne, D.P. & Hawick, K.A.
"Exploiting Graphical Processing Units for Data-Parallel Scientific Applications",

Playne, D.P. & Hawick, K.A.
"Comparison of GPU Architectures for Asynchronous Communication with Finite-Differencing Applications"

Hawick, K.A. & Playne, D.P.
"Numerical Simulation of the Complex Ginzburg-Landau Equation on GPUs with CUDA"
Proc. International Conference on Parallel and Distributed Computing and Networks, February 14-16, Innsbruck, Austria.
GPGPU Applications

Lotka-Volterra
Predator-Prey
50x – 100x faster

Shallow Water

50x – 80x faster

Hawick, K.A., Playne, D.P. & Scogings, C. J.,
“Simulating the Generalised Lotka Volterra Equations with Multiple Species on GPUs with Automatic Code Generation.”
Proceedings 12th IASTED International Conference on Parallel and Distributed Computing and Networks.
pp 560-567, 11-13 Feb, 2013, Innsbruck, Austria.
GPGPU Applications

Lattice Simulations

- Computational Models
  - Ising
  - Potts
  - Sznajd
  - Lattice Gas

- Memory Optimisation
- Bit-packed data storage
- Real-time visualisation
- Rewired lattices
- Hybrid CPU-GPU Simulations
- Multi-GPU Simulations
GPGPU Applications

**Sznajd**
- Opinion Model
- 10x – 20x faster

**Lattice Gas**
- Fluid Flow automata
- 20x– 40x faster

Hawick, K. A. & Playne, D.P.
Halo Gathering Scalability for Large Scale Multi-dimensional Sznajd Opinion Models Using Data Parallelism with GPUs.

Johnson, M.G.B., Playne, D.P. & Hawick, K.A.
“Data-Parallelism and GPUs for Lattice Gas Fluid Simulations”
GPGPU Applications

N-body Simulations
- Benchmarking
- Update Methods
- Collision Models
- High-Order Integration Methods
- Multi-GPU Simulations
GPGPU Applications

N-body Simulation

50x – 80x faster

Hawick, K.A., Playne, D.P. & Johnson, M.G.B.
“Numerical Precision and Benchmarking Very-High-Order Integration of Particle Dynamics on GPU Accelerators”
GPU Applications

Other:

- Connected Component Labelling
  - 10-30x faster
- Random Number Generation
  - 30-60x faster
- Hypercubic Data Storage

Hawick, K.A., Leist, A. & Playne, D.P.
“Parallel Graph Component Labelling with GPUs and CUDA”

A. Leist, Playne, D.P. & Hawick, K.A.
“Interactive Visualization of Spins and Clusters in Regular and Small-World Ising Models with CUDA on GPUs”

Hawick, K.A. & Playne, D.P.
“Hypercubic Storage Layout and Transforms in Arbitrary Dimensions using GPUs and CUDA”
Graphical Processing Units

How can GPUs be so much faster?
Central Processing Units

A typical CPU will have a design something like:
Central Processing Units

Most processor architectures are designed with various levels of cache to keep the cores supplied with data.
Central Processing Units

This causes major problems for multi-core processor design as the different cache areas have to be kept consistent.
Central Processing Units

The cache-coherency problem is one of the reasons that most CPUs still only have 4-8 cores.

In contrast the latest NVIDIA Graphics Card (GTX Titan) contains 2688 cores.
Graphical Processing Units

1 GPU core ≠ 1 CPU core

CPUs contain few powerful cores supported by several levels of cache.

GPUs contain many simple processing cores, with small amounts of cache.
GPUs contain a number of multi-processors that each contain their own memory and a number of cores.
Multiprocessors

GPU architectures differ mainly on the design of the multi-processors.
Graphical Processing Units

GPUs take a unique approach to solving the cache-coherency problem:

They don’t
Graphical Processing Units

GPU programs are split into kernels that are called from the host (CPU).
Graphical Processing Units

When a kernel is executed on a GPU, the same kernel code will be run by thousands or millions of threads on all the available cores.
Graphical Processing Units

A thread in one kernel, cannot depend on a value written by a different thread in the same kernel.

If calculation in one thread does depend on a value calculated by another thread, it must be split into multiple kernels.
Graphical Processing Units

The challenge of GPU programming is how to accomplish this.

Easy for highly parallel problems
   (50x-100x faster).

Hard for problems with dependencies
   (10-20x faster)
Multi-GPU

A GPU program is limited by the speed of the fastest GPU. Solution – more GPUs.
Multi-GPU

Most motherboards can support a limit of around 4 graphics cards.
PCle chassis can support up to 16 GPUs.
Multi-GPU

GPU-GPU Communication:

• Transfer through Host
• Direct Transfer
• Direct Access
To go beyond the limit of 16 GPUs we need to distribute the GPUs across different nodes.
Summary

Graphical Processing Units are a highly parallel and powerful computing architecture.

They can provide a lot of computational throughput if programmed correctly.

Optimising a GPU program requires understanding of the GPU programming model and architecture.

The details matter
Questions?