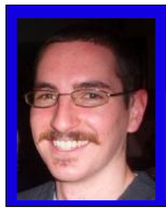
Advanced Architectures for Astrophysical Supercomputing



SWIN BUR NE — CENTRE FOR ASTROPHYSICS AND SUPERCOMPUTING

Benjamin Barsdell (bbarsdel@astro.swin.edu.au), David Barnes, Christopher Fluke

Swinburne Centre for Astrophysics & Supercomputing

Melbourne, Australia

This research is supported under the Australian Research Council's Discovery funding scheme (DP0665574).

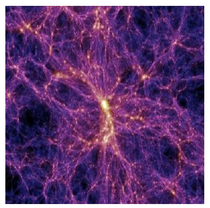
CCT — From the First Objects to the Cosmic Web

Abstract

Astronomers have come to rely on the increasing performance of computers to reduce, analyse, simulate and visualise their data. The recent transition from increasing clock rate to increasing core count in CPUs has seen Moore's Law maintained, but at the cost of a paradigm shift from serial to parallel processing. Current generation graphics processing units (GPUs) reflect the situation well: their massively parallel architecture provides orders of magnitude more raw processing power than current CPUs, but present a new and foreign programming model. While efforts to port a number of astronomy algorithms to GPUs have been successful in obtaining significant speed-ups (frequently 100x over CPUs), the approaches so far have been somewhat "ad-hoc" in nature.

Here we motivate an "algorithm analysis" approach to the use of GPUs and other advanced architectures in astronomy. Such an approach will identify the expected performance and scaling of astronomy algorithms on new hardware architectures prior to implementation. The current direction of computer architecture development suggests that an understanding of our algorithms will be of great importance to the future of computational astronomy.

Motivation: From Video Games to Science



Modern astronomy has come to rely heavily on high-performance computing (HPC). However, all research areas are facing significant challenges as data volumes approach petabyte levels. The Australian Square Kilometre Array Pathfinder project for instance will produce data at a rate that makes storage in raw form impractical, necessitating on-the-fly reduction and analysis to produce 4GB/s of products. On the modeling front, there is an ongoing desire for larger and more-detailed simulations, and particle counts have exceeded 10^{10} (e.g., the Millennium simulation by Springel et al. 2005).

The HPC scene has recently witnessed the bold introduction of the GPU as a viable and powerful general-purpose co-processor to CPUs. GPUs were developed to off-load the computations involved in 3D graphics rendering from the CPU, primarily to the benefit of video-games. Their continued development has been driven by the $60 billion/year video-games industry, the result of which is seen in Figure 1.

Along with rapidly-increasing performance, GPUs have undergone a shift from containing special-function processors to instead being composed of flexible general-purpose processors. This, combined with the availability of general-purpose GPU programming tools, has opened up GPU computation to a wide range of non-graphics-related tasks, notably in the area of HPC.

The power of many-core architectures like GPUs, if harnessed, could lead to significant speed-ups in computational astronomy and potentially to new science outcomes.

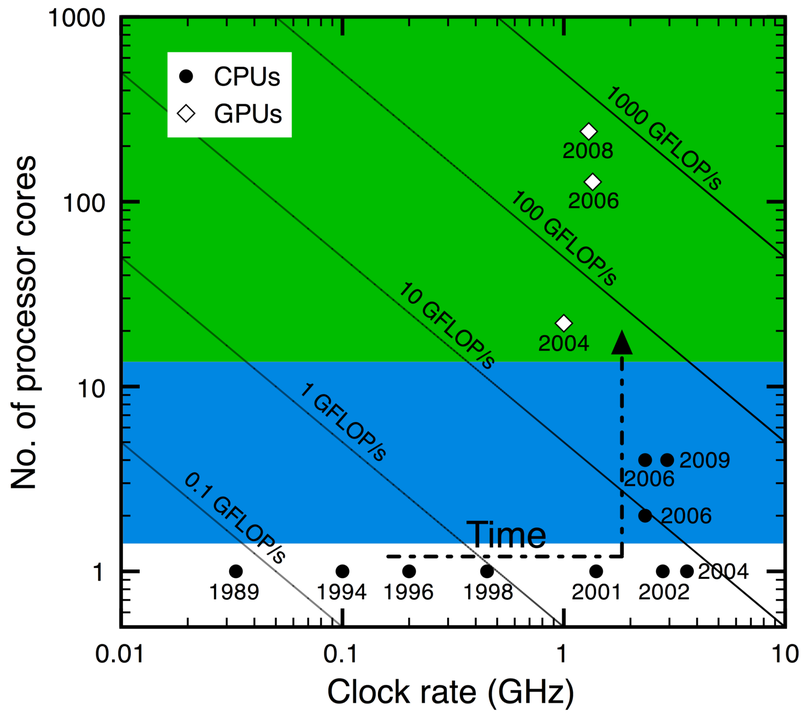


Figure1. Clock-rate versus core-count phase space of Moore's Law. The classical Moore's Law trend is seen in the approximately equi-spaced black dots at 1 on the vertical axis. The paradigm shift from increasing clock-rates to increasing core-count is seen as a sharp turn at a clock-rate of around 3Ghz. This "multi-core corner" presents significant challenges to the computational astrophysics community. Recent GPUs appear beyond the turn, and represent a likely direction for future CPUs.

General-purpose GPU (GPGPU) computing brings a number of significant benefits to HPC:

	CPU	GPU
Monetary cost	$\$10^5$/Tflop/s	$\$10^4$/Tflop/s
Environmental cost	0.5 Gflop/s/watt	4 Gflop/s/watt
Desktop supercomputing	10 Gflop/s	1000 Gflop/s

Note: flop/s = floating-point operations per second

The Future of Computing

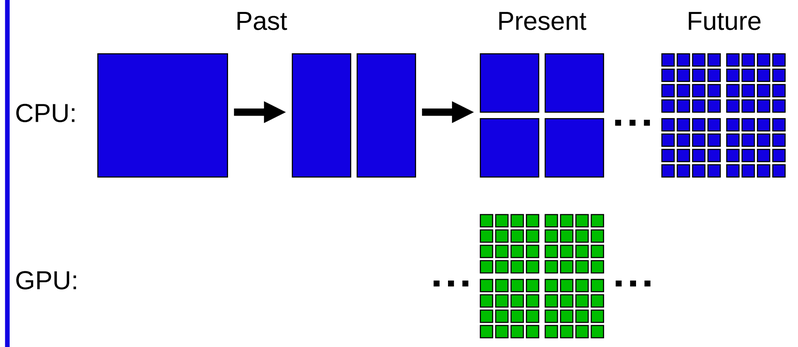


Figure 2. The direction of CPU architecture development and the current state of GPU architecture. Boxes represent processor cores.

Figure 2 depicts the recent evolution of CPU architecture and compares it to the current state of GPU architecture. Prior to 2005, developments in CPU performance came largely from increasing clock-speeds. Since then, hardware issues have forced manufacturers to turn to multi-core architectures. We have thus seen a shift from single-core processors to dual-core and then to quad-core CPUs (and 8-core CPUs have recently been announced). Looking at it this way, it becomes obvious that if CPU performance is to continue to increase, the number of cores must continue to rise, and **CPUs will almost inevitably move to a *many-core* architecture**.

The significance of the GPU is that **GPUs *already have* many-core architectures**. They thus present us with the motivation and opportunity to study how our astronomy codes will perform and scale on many-core architectures in the future.

Both the immediate performance boost provided by GPUs and the expected future of CPU computing provide strong motivation for a thorough analysis of the performance and scalability of our astrophysics algorithms in advanced parallel processing environments.

GPU Use in Astronomy to Date

A small number of astronomy algorithms have been implemented on GPUs to date, including:

- Direct N-body simulations (e.g., Hamada & Iitaka 2007)
- Radio-telescope signal correlation (e.g., Harris, Haines & Staveley-Smith 2008)
- The solution of Kepler's equation (Ford 2008)
- Gravitational lensing ray-tracing (Thompson et al. 2010)
- Phase-space study of post-Newtonian binary black hole inspirals (Herrmann et al. 2009)
- 3D Cartesian shapelets (Fluke et al., in preparation)
- Pulsar signal processing

All have reported **speed-ups of *O*(100)** over CPU implementations. However, these algorithms are for the most part "embarrassingly parallel" "low-hanging fruits", meaning that they can be run on a parallel processing architecture with little to no overhead. This makes them obvious candidates for efficient GPU implementation. The question that remains is: **Exactly which classes of astronomy algorithms are likely to obtain significant speed-ups by running on advanced architectures?**

Our Approach

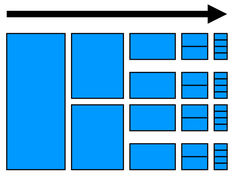
We propose a generalised approach based around two key ideas:

1. **Building and using a taxonomy of astronomy algorithms**
2. **Developing an algorithm analysis methodology relevant to new hardware architectures**

We believe that such an approach will minimise the effort required to turn the "multi-core corner" for computational astronomy and ensure that the solutions found will continue to scale with future advances in technology.

Rules of Thumb

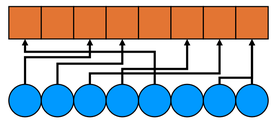
Here are some of the issues that are important when considering a GPU implementation of an algorithm (Barsdell et al., in prep):



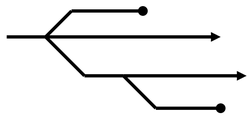
Massive parallelism

- Having enough parallel granularity to use all of the available hardware parallelism
- Current GPUs hit peak at $O(10^4)$ threads

Memory access locality and patterns



- Locality and alignment strongly impact bandwidth
- Read collisions are bad, write collisions are really bad



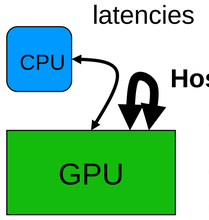
Branching

- Should be minimised
- At least aim for locality in branch paths

Computation / memory operation ratio

- Arithmetic instructions generally much faster than memory operations
- Increasing arithmetic intensity hides memory latencies

+ - x ÷ / I/O



Host ⟷ Device memory transfers

- Bandwidth is O(10x) less than within device memory
- Minimise transfers by implementing as much of the algorithm as possible on the GPU

Precision

3.14159265358979323...

- Single-precision FLOPs can be (significantly) more than 2x faster than double-precision
- Often worth the effort to assess whether and where double-precision is necessary

References

Springel, V. et al., 2005, Nature, 435, 629-636
Hamada, T., Iitaka, T. 2007, arXiv:astro-ph/0703100v1
Harris, C., Haines, K., Staveley-Smith, L. 2008, ExA, 22, 129
Ford, E.B. 2008, NewA, 14, 406
Thompson, A.C., Fluke, C.J., Barnes, D.G., Barsdell, B.R. 2010, NewA, 15, 1, 16-23
Herrmann, F., Silberholz, J., Bellone, M., Guerberoff, G., Tiglio, M., 2009, arXiv:0908.3889v2 [gr-qc]

The Algorithms of Astronomy

Here we present an initial classification of astronomy algorithms based on application of the "rules of thumb" and known GPU-efficient algorithms (Barsdell et al., in prep):

Efficiency, speed-up on GPUs
- High, O(100x)
- Moderate, O(10x)
- Untested

- **Simulation**
  - Direct N-body simulations
  - Tree-code N-body simulations / SPH
  - Halo finding
  - Fixed-resolution mesh simulations
  - Adaptive mesh refinement
  - Semi-analytic modelling
  - Gravitational lensing ray-shooting
  - Other Monte-Carlo methods
- **Data reduction**
  - Pulsar signal (coherent) dedispersion
  - Radio-telescope signal correlation
  - Image processing
    - Optical data reduction
      - Flat-fielding etc.
      - Stacking / mosaicing (e.g., DRIZZLE)
    - Source extraction
    - Convolution and de-convolution
    - CLEAN algorithm
    - Gridding of visibilities and single-dish data
- **Data analysis**
  - Data mining
    - Selection based on criteria matching
    - Machine learning
  - Fitting / optimisation
  - Numerical integration
  - Volume rendering

We conclude that the **data-rich nature of computational astronomy** combined with the **efficiency of data-parallel algorithms on current GPU hardware** make for a **very promising relationship** with current and future massively-parallel architectures.

Processors are likely to become **even more flexible** in the future, potentially improving the efficiency of many astronomy algorithms and opening up **new avenues to significant speed-ups.**

Conclusions

Modern astronomy relies heavily on HPC, and GPUs can provide both significant speed-ups over current CPUs and a glimpse of the probable future of commodity computing architectures. However, their more complex design means algorithms must be considered carefully if they are to run efficiently on these advanced architectures. There is therefore strong motivation to thoroughly analyse and categorise the algorithms of astronomy in order to take full advantage of current and future advanced computing architectures and maximise our science outcomes.

SWIN BUR NE
CENTRE FOR ASTROPHYSICS AND SUPERCOMPUTING