Heterogeneous Hardware Acceleration of Parallel Algorithms

Ra Inta, David J. Bowman and Susan M. Scott

Department of Quantum Science,
The Australian National University,
Canberra ACT 0200, Australia

000: Introduction

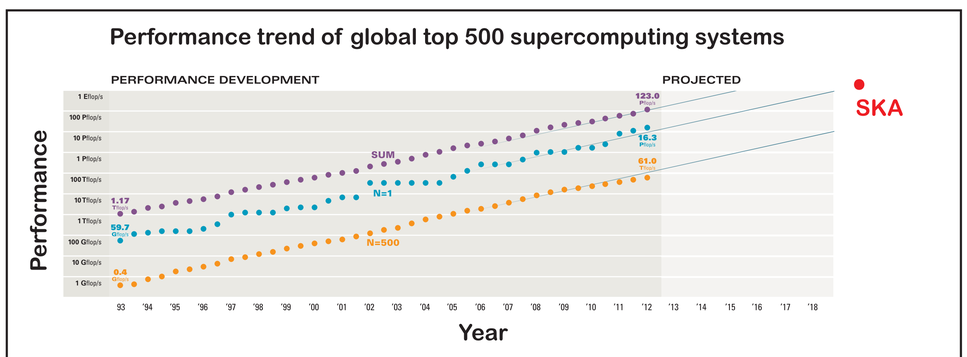
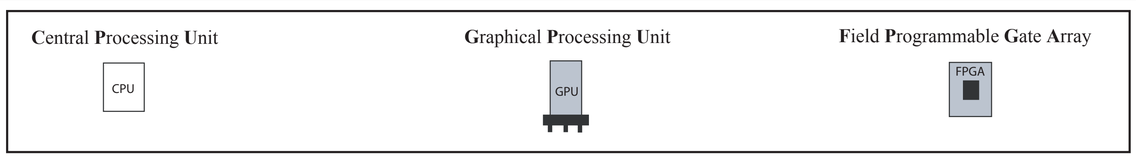


Figure 1: Current and projected performance of the top five hundred supercomputing systems. The computational demands of the SKA (Square Kilometre Array) alone will require the projected most powerful computing system (i.e. exaflop/s scale) by 2020.

We are experiencing an ever-increasing deluge of data from large scientific projects, requiring commensurately significant computational power. As a result, we are currently in a transition period for high performance computing (HPC) architectures, and the timing is right to explore alternative computing architectures. For example, the Square Kilometre Array alone [1] is expected to require the top projected supercomputing system [2] for its expected first-light date of 2020 (Figure 1). However, there are many other data-hungry systems about to come on-line, such as the Large Synoptic Survey Telescope [3].

001: Advantages of Hardware Acceleration



In light of the above pressures, many researchers have turned to hardware acceleration as a solution. The type of accelerator depends on the algorithm/pipeline in question (Table 1).

Platform	Pros	Cons
CPU	Analysis 'workhorse,' multi-tasking	Power hungry, limited processing cores
(GP)GPU	Highly parallel, fairly simple interface (e.g. C for CUDA)	Highly rigid instruction set (don't handle complex pipelines)
FPGA	Unrivalled flexibility and pipelining	Expensive outlay, specialised programming interface, prohibitive development time

Table 1: Comparison of the advantages and disadvantages of CPU based calculations to those of the GPU and FPGA. This list is neither comprehensive nor exhaustive.

By far the most widely adopted hardware acceleration platforms are the (General Purpose) Graphical Processor Unit (GPU) and the Field Programmable Gate Array (FPGA). The former emerged from the demands of computer gamers for ever-better graphical displays, while the latter derives from pressures from manufacturers demanding platforms more flexible than application-specific integrated circuits.

010: The 'Chimera' Computing System

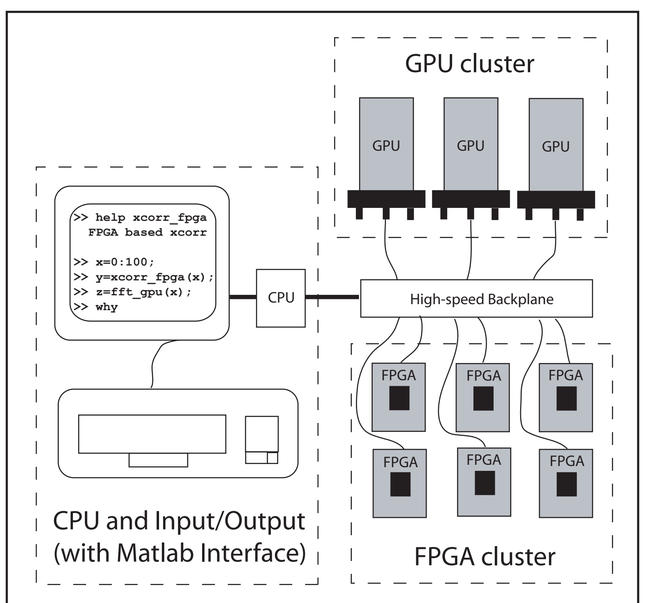


Figure 2: Schematic of our CPU/GPU/FPGA based computing platform at the Australian National University. Because of commercial-off-the-shelf constraints, the 'high-speed backplane' described here is the PCI Express bus resident on a standard computer motherboard.

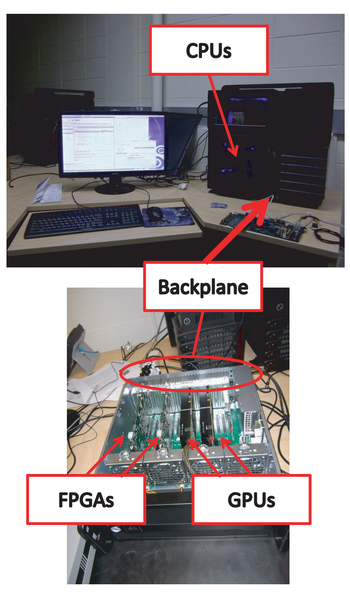


Figure 3: Photograph of an implementation of the system. Because of the three hardware classes, we named the system the 'Chimera,' after the mythical Greek beast with three different heads.

At the Australian National University, we constructed a computing system to exploit the advantages of both FPGAs and GPUs for certain algorithm classes [4, 5]. The initial concept was to mediate separate clusters of FPGAs and GPUs via a high-speed backplane. One of the design constraints is for the components to be commercial-off-the-shelf. This meant the backplane adopted is the PCI Express bus, resident on a standard CPU motherboard.

For many algorithm classes, the bottleneck in throughput is in transporting data (I/O). Here, communication between the hardware is facilitated by Linux kernel modules, with the intention of minimising the mediation role played by the CPU.

Image kindly produced by Elizabeth Koehn

011: Performance

Performance on this type of heterogeneous platform is highly algorithm dependent. One of the simplest algorithms that illustrates the concept of this platform is the pedagogical introduction to Monte Carlo integration (Figure 4A).

A
Monte Carlo calculation of π

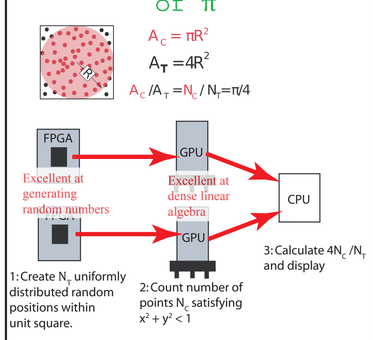


Figure 4A: Monte Carlo calculation of π. Pairs (x, y) of uniformly distributed random numbers are generated to lie within a square, circumscribed about a unit circle. In the limit of large numbers of pairs, the ratio of the points that lie within the circle to the total number converges to π/4.

Using 32-bit integer pairs, we tested the whole pipeline in the FPGAs, which gave the fastest throughput of 24 Gsamp/s, compared to the GPU, at 2.1 Gsamp/s . This compares to the roughly 10 Msamp/s for our CPU. This result is surprising, although considerable effort went into optimising the FPGA pipeline. We expect the GPU to fare a lot better with more complex integration problems.

B
Normalised cross-correlation

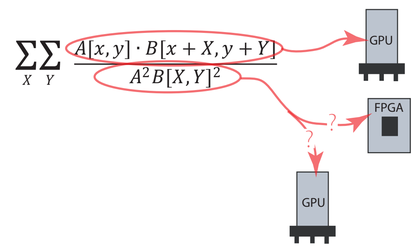


Figure 4B: Normalised cross-correlation in two dimensions. This is widely used e.g. for image processing, resolving images in synthetic aperture arrays such as the SKA or the VLBA, and in video compression. The numerator, being multiply-intensive, is well suited to GPU-based computation. It is not entirely clear what hardware is most suitable to calculate the denominator.

We exhaustively searched a 1024×768 pixel (16-bit greyscale) image for a 8×8 template. The numerator was calculated on the GPU at 158 frame/s. The denominator was tested separately on the GPU and the FPGA, giving 894 frame/s for the former and 12,500 frame/s for the latter.

Thus, in this implementation, ignoring the impact of the PCIe bottleneck, a hybrid system comprising a GPU working in tandem with an FPGA was found to achieve a better result than a system consisting of two GPUs.

C
Other algorithms

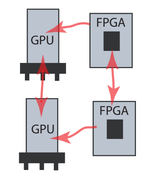


Figure 4C: Other algorithms.
Many promising data analysis applications that are considered computationally expensive may be implemented rather simply using this type of heterogeneous hardware acceleration. For example, digital filter application is efficiently implemented on FPGA devices. A number of promising analysis techniques based on compressed sensing are considered computationally expensive that may be efficiently implemented on an FPGA via Cholesky decomposition.

Finally, another promising pipeline capability of this system exists where instruments acquire and filter data using an FPGA, but require the more rapid development and floating point capability of a GPU based platform.

100: Analysis of Parallel Architectures via the 'Thirteen Dwarves of Berkeley'

It is interesting to consider the possibility that the entire landscape of parallel algorithms and pipelines may be represented by a handful of algorithm classes. Initially, Phillip Colella identified seven broad classes [6], which were quickly termed 'dwarves,' after the Snow White fairy tale. This concept of a dwarf, as an "algorithmic method encapsulating a pattern of computation and/or communication," was developed further by the Computer Science Department of UC Berkeley and extended to thirteen [7]. These dwarves, along with representative problems or algorithms, are listed in Table 2.

	Dwarf	Examples/Applications
1	Dense Matrix	Linear algebra (dense matrices)
2	Sparse Matrix	Linear algebra (sparse matrices)
3	Spectral	FFT-based methods
4	N-Body	Particle-particle interactions
5	Structured Grid	Fluid dynamics, meteorology
6	Unstructured Grid	Adaptive mesh FEM
7	MapReduce	Monte Carlo integration
8	Combinational Logic	Logic gates (e.g. Toffoli gates)
9	Graph traversal	Searching, selection
10	Dynamic Programming	'Tower of Hanoi' problem
11	Backtrack/Branch-and-Bound	Global optimization
12	Graphical Models	Probabilistic networks
13	Finite State Machine	TTL counter

Table 2: The "Thirteen Dwarves of Berkeley". This is a list of the main classes or processes spanning the present and projected parallel algorithm landscape. A representative problem from each class is also given.

We analysed the current implementation of the Chimera computing system, in terms of expected performance, on the Thirteen Dwarves (Figure 5). This is not intended to be comprehensive, as I/O constraints, choice of fixed/floating point support etc. are highly implementation dependent. However, as far as we are aware, we are the first group to have performed such an analysis.

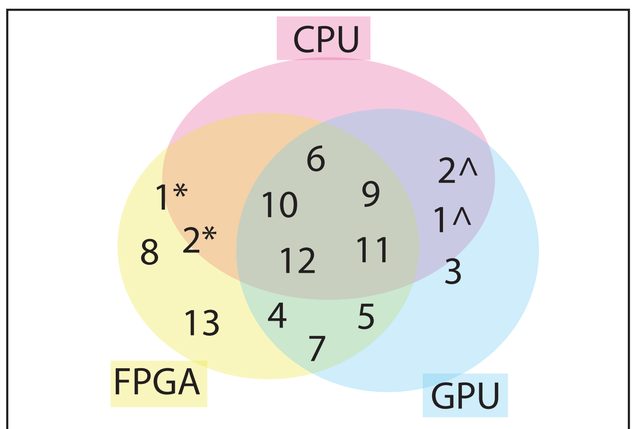


Figure 5: The most appropriate hardware acceleration subsystem combination for representative problems from the "Thirteen Dwarves" (Table 2). The * refers to fixed point, while ^ represents floating point calculations.

101: Conclusion

Because of the ever-increasing demand for high performance computing, we are in a period of phase transition towards new parallel computing architectures. A promising avenue is the exploitation of hardware accelerators, the most common being the GPU and the FPGA, for different reasons. Because the technology is still new for both these accelerators, they have performance growth easily exceeding that of Gordon Moore's famous law.

We have shown here that it is possible to exploit the advantages of heterogeneous hardware accelerators for certain algorithms [4], demonstrating this on a proof-of-concept platform using commercial-off-the-shelf components [5], a platform we call the 'Chimera'.

Finally, we have attempted to estimate the performance of the sub-system components of the Chimera system on all possible instances of parallel computational algorithms, via the "Thirteen Dwarves of Berkeley" [7].

This system is scalable and extremely energy-efficient, if optimised for the algorithm or pipeline. The initial capital outlay for the components is extremely competitive. However, largely because of the FPGAs, development time can be costly.

110: References

[1] T. Cornwell and B. Humphreys, "Data processing for ASKAP and SKA," http://www.atnf.csiro.au/people/tim.cornwell/presentations/nzpathwaysfeb2010.pdf (2010)

[2] H.Meuer, E. Strohmaier, J. Dongarra, and H. Simon, "Top 500 supercomputers," http://www.top500.org/ (June 2012)

[3] LSST Corporation, "Large synoptic survey telescope," http://www.lsst.org/lsst/ (Sept. 2012)

[4] R. Inta and D. J. Bowman, "An FPGA/GPU/CPU hybrid platform for solving hard computational problems," in Proc. eResearch Australasia, Gold Coast, Australia (Nov. 2010)

[5] R. Inta, D. J. Bowman, and S. M. Scott, "The 'Chimera': An Off-The-Shelf CPU/GPGPU/FPGA Hybrid Computing Platform," Int. J. Reconfigurable Comp., 2012(241439), 10 pp. (2012)

[6] P. Colella, "Defining Software Requirements for Scientific Computing," DARPA HPCS (2004)

[7] K. Asanovic and U C Berkeley Computer Science Department, "The landscape of parallel computing research: a view from Berkeley," Tech. Rep. UCB/EECS-2006-183, UC Berkeley (2005)

111: Contact

Ra Inta: ra.inta@anu.edu.au

More information:
www.anu.edu.au/physics/cgp/Research/chimera.html