An FPGA/GPU/CPU hybrid platform for solving hard computational problems

Ra Inta[1], David Bowman[2]
[1] Centre for Gravitational Physics, The Australian National University, Canberra, Australia
[2] School of Engineering and Information Technology, UNSW@ADFA, Canberra, Australia

1: Introduction

Most high performance computing (HPC) these days is performed on clusters of CPUs [1]. Yet many classes of algorithms and pipelines used in a range of complex and data intensive (i.e. 'hard') computational problems are better suited to more specialised, non-CPU, computing platforms. The two most prevalent non-CPU platforms currently are **Graphical Processing Units** (GPUs) and **Field-Programmable Gate Arrays** (FPGAs).

GPU

Multi-CPU system specially adapted to the high-speed vector processing requirements of modern computer graphics. Very cost-effective because of huge demand from the computer games market, offerring a single user access to 1 Tflop/s processing power for a few hundred dollars.

FPGA

Single chip containing a dense array of programmable logic gates, reconfigurable within a single clock cycle. Basically a powerful, programmable extension of the integrated circuit, offering unprecedented computing flexibility. The fastest platform for flexible digital signal processing applications.

This is an outline of a proposed FPGA/GPU/CPU hybrid cluster that makes use of hardware optimisation to exploit the strengths of each platform. The proposed interface is a set of Matlab libraries that allocate resources to separate platforms in a way to optimise the throughput of the pipeline being used.

2: Technology mapping

At the hardware level, blind comparison of flop/s can be a disingenuous measure, because not all operations are handled in the same way. There are superior methods of calculation which may usually thought of as a waste of resources on conventional platforms. Hence much of the problem here lies in 'technology mapping,' i.e. identifying which algorithm, or pipeline, has the best throughput on which system combination.

3: Hybrid cluster design

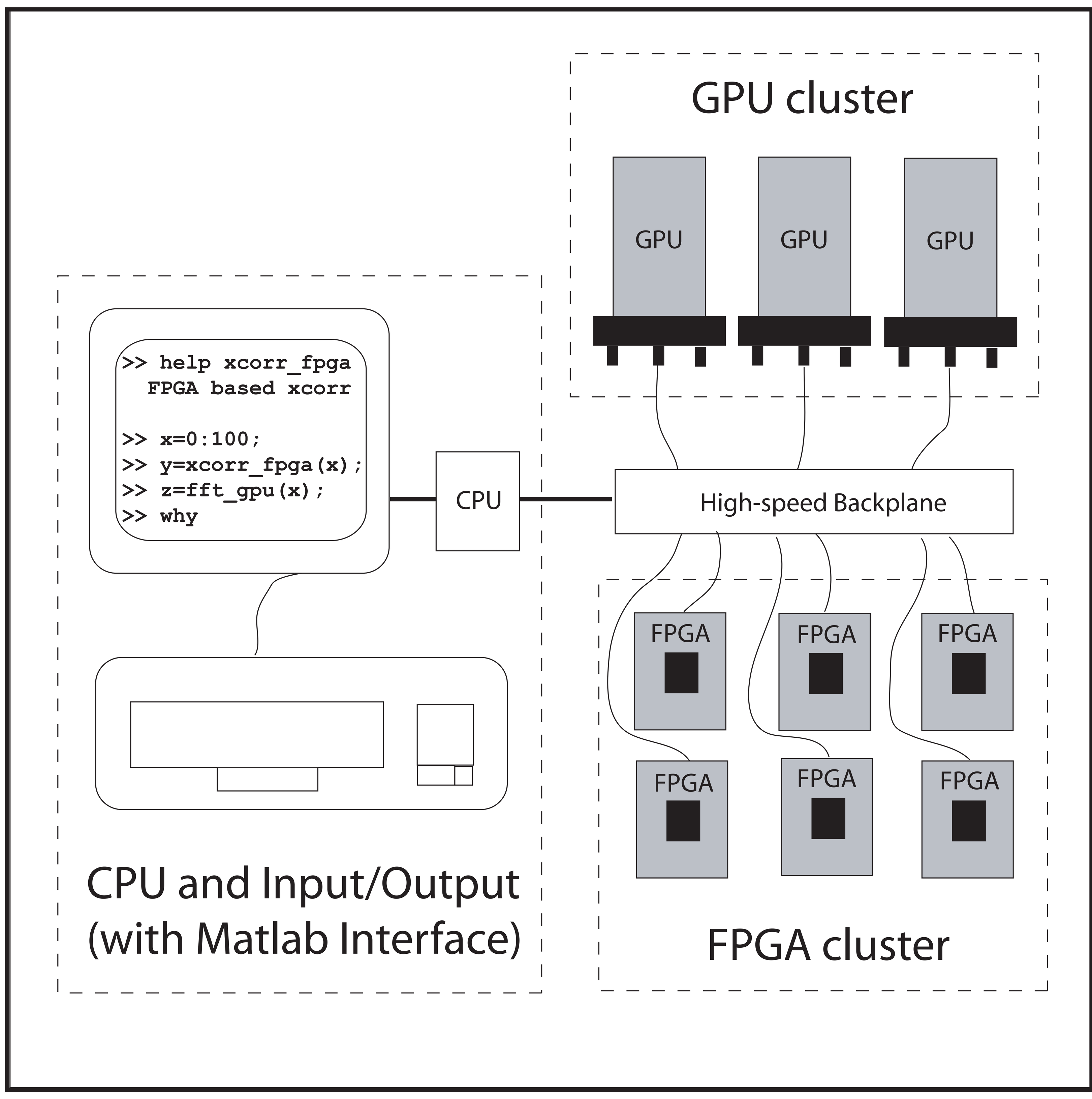
The hybrid system proposed here (Figure on right) consists of a CPU controlling user I/O devices as well as scheduling and task allocation to a cluster of FPGAs and/or a cluster of GPUs, depending on the algorithm. A Matlab interface is used to parse functions to the separate subsystems.

In the early development phase, a simple suffix would be used to denote hardware optimised functions. For example (after a hardware query), the user takes a hardware-optimised square root of a vector `x`:

```
>> sqrt_gpu(x)
```

Or an autocorrelation:

```
>> xcorr_fpga(x)
```



4: Potential applications

A number of fields and industries would immediately benefit from a cost-effective HPC system:

Astronomy: gravitational-wave data analysis use GPUs [2]; Australian Square Kilometre Array Pathfinder (SKAP) use FPGA processors [3, 4]
Finance: low latency arbitrage strategies for high-frequency trading [5]
Military: USAF's rapid Automatic Target Recognition systems [6]
Computer security: bitwise encryption algorithms more efficient with FPGAs [7]
Film: Weta Digital, with *six* computers in the most recent top500 list [1], have to compromise between effects quality and computing power [8]

5: Other advantages/future performance

Inexpensive initial outlay: both subsystems are extremely cost effective per unit
Very low power consumption: cooling infrastructure such as fans or cooling towers are unnecessary. This also adds value for money, considering half of the lifetime cost of a conventional supercomputer results from energy consumption [9].
Low form-factor: GPUs and FPGAs occupy a relatively tiny volume
Beats Moore's Law: Historically CPUs have roughly quadrupled performance every three years; in the same time period, GPUs have improved at five times, while FPGAs have improved performance *tenfold* [10].
This exponential improvement means that these platforms will become increasingly competitive in the future.

References

1. Top500 Supercomputers, http://www.top500.org/, accessed on June 29, 2010
2. S. K. Chung, L. Wen, D. Blair, K. Cannon and A. Datta, *Class. Quantum Grav.* **27** 135009 (2010)
3. ANZSKA, http://www.ska.gov.au/media/factsheets/Documents/anzSKA_factsheet_ASKAP_TU_v1_LR-0911.pdf , accessed June 29, 2010
4. CSIRO Australia Telescope National Facility, http://www.atnf.csiro.au/projects/askap/ASKAP_dst_v1.0_final.pdf , accessed June 29, 2010
5. C. Duhigg, *New York Times,* July 23 (2009), http://www.nytimes.com/2009/07/24/business/24trading.html , accessed June 30, 2010
6. Starbridge Hypercomputing, http://www.starbridgesystems.com/hypercomputing/hypercomputers/ accessed on June 29, 2010
7. Pico Computing, Inc., http://www.scribd.com/doc/26191199/Using-FPGA-Clusters-for-Fast-Password-Recovery , accessed June 29, 2010
8. J. Ericson: "Processing AVATAR," Information Management Newsletters, December 21 (2009), http://www.information-management.com/newsletters/avatar_data_processing-10016774-1.html , accessed June 30, 2010
9. P. E. Ross, *IEEE Spectrum: Inside Technology* (August 2008) http://spectrum.ieee.org/computing/hardware/a-computer-for-the-clouds , accessed June 30, 2010
10. Underwood, K., *Proc. 12th ACM. Int. Symp. on FPGAs*, pp. 171–180 (2004)