A Heterogeneous Platform with GPU and FPGA for Power Efficient High Performance Computing*

Qiang Wu[*^], Yajun Ha[#^], Akash Kumar[^], Shaobo Luo[#], Ang Li[^], Shihab Mohamed[^]
College of Information Science & Engineering, Hunan University, Changsha, China[*]
Institute for Infocomm Research, A*STAR, Singapore[#]
Department of Electrical & Computer Engineering, National University of Singapore, Singapore[^]
wuqiang@hnu.edu.cn, (ha-y@i2r.a-star.edu.sg, elehy@nus.edu.sg), akash@nus.edu.sg, luos@i2r.a-star.edu.sg, {liang, shihab}@nus.edu.sg

***Abstract*—*Heterogeneous computing is gaining attention from both industry and academia nowadays. One driving factor for heterogeneous computing is the power efficiency. GPU and FPGA have been reported to achieve much higher power efficiency over CPU on many applications. Comparisons between GPU and FPGA show different characteristics of GPU and FPGA in accelerated computing. Some tasks run better on GPU, some run better on FPGA. Combining GPU and FPGA in one heterogeneous computing platform may provide us the advantages from both sides. This paper presents a heterogeneous computing platform with GPU and FPGA that we have built for power efficient high performance computing. The experimental results of 4 application examples show that different applications have different favorite computing architectures, which suggests a matching of the characteristics between the computation task and the computing architecture is the key to the power efficient high performance computing on heterogeneous computing platforms.*

***Index Terms*—Heterogeneous Computing, Power Efficiency, High Performance Computing, GPU, FPGA.**

I. Introduction

Heterogeneous computing is an approach for supporting various computation demands by combining different processing elements together. These processing elements can be CPU, GPU, FPGA or other accelerators and co-processors. The idea behind the heterogeneous computing approach is that the matching of the computation task to the computing architecture will lead to better execution of the computation task in terms of performance and power consumption as well as other measurements[1].

Among the performance and power consumption, the latter has gained increasing concerns from the industry, especially from the data centers. In 2010, data centers accounted for about 1.2% global electricity use and was increasing by about 56% from 2005 to 2010[2]. More and more efforts have been invested into the improvement of the power efficiency in addition to the performance.

Heterogeneous computing systems attack the problem by providing suitable candidate computing architecture to the application running on the computer clusters. A matching computing architecture is believed to be able to execute the application in a more power efficient way hence reducing the computing power.

GPU and FPGA have been used to accelerate compute-intensive applications in high performance computing. The reported performance surpasses that of CPU by one to three orders of magnitude. The power efficiency of the GPU or FPGA accelerated computing is also reported much higher than that of the CPU. The success stories of GPU/FPGA accelerated computing inspire the industry and academia to put more efforts on the research and development of the heterogeneous computing platforms.

There are a few efforts to combine the GPU, FPGA and CPU altogether [3][4]. Combining the GPU and FPGA in one heterogeneous computing platform may help to integrate the advantages of the different computing architectures.

However, different computing architectures present many challenges to the designers. It takes a lot of time to port an application to the different architecture, and it is very hard to know the performance of the new code before the porting has been done. In this paper, we introduce our experience in developing a heterogeneous computing platform with GPU and FPGA, and compare the power efficiency and performance of some application examples on CPU, GPU and FPGA. For each of the architecture, the application is manually ported to get the best possible implementation.

Four application examples are selected for the comparison of performance and power efficiency between the GPU and FPGA. The application examples are Matrix Multiplication, N-body Simulation, Heston Pricing and Finite Difference Modeling. We believe that these application examples represent some typical work loads in the high performance computing area.

The main contributions of this paper are listed in below:

1) A system framework of the heterogeneous computing platform with CPU, GPU and FPGA;

2) Estimation and analysis of the performance and power efficiency of the application examples;

3) Performance and power efficiency comparison of 4 representative application examples.

The rest of the paper is organized as follows. Section II summarizes the related work on the heterogeneous computing with GPU and FPGA. Section III describes the heterogeneous computing platform. Section IV presents the test results of the 4 application examples running on the heterogeneous computing platform. Section V concludes the paper.

This paper is supported by “the Fundamental Research Funds for the Central Universities”.

978-1-4799-4833-8/14/$31.00 ©2014 IEEE

II. RELATED WORK

There are a few heterogeneous computing platforms which integrates GPU and FPGA together. In 2009, Michael Showerman et al. report in [3] a cluster that is built with compute nodes each containing 2 dual-core AMD Opteron CPUs, 4 NVIDIA Quadro FX 5600 GPU cards and 1 Nallatech H101 FPGA accelerator card. In 2010, Kuen Hung Tsoi and Wayne Luk introduce in [4] a heterogeneous cluster with 16 nodes each containing 1 AMD Phenom Quad-Core CPU, 1 NVIDIA Tesla C1060 card, and 1 ADM-XRC-5T2 FPGA card. Note that we focus on the heterogeneous platforms that have GPU and FPGA integrated into one node rather than separated into different nodes since the former has a better chance for high performance than the latter [0].

There are many efforts made by the industry and academia to compare the performance and power consumption of the GPU and FPGA.

Ben Cope et al. carried out an in-depth comparison and analysis of GPU (NVIDIA GeForce 6800 GT, 7900 GTX) and FPGA (Virtex II Pro, Virtex 4) with several test examples including the Primary Color Correction, 2D Convolution, Video Frame Resizing, Histogram Equalization, and 3-step non-full-search Motion Vector Estimation[5].

John Bodily et al. compared several optical flow and communication algorithms with GPU (NVIDIA GeForce 8800 GTX) and FPGA (Virtex-4 FX60)[6].

Xiang Tian et al. mapped the Quasi Monte Carlo Financial Simulation engine to the FPGA (Xilinx Virtex-4) and GPU (NVIDIA 8800 GTX) and found that the FPGA achieved about 3x speedup and 16x energy efficiency over the GPU[7].

Karl Pauwels et al. studied several algorithms for the computation of phase-based optical flow, stereo, and local image features (energy, orientation, and phase) using FPGA (Virtex-5 XC5VLX330T, Virtex- 4 XC4VFX100) and GPU (GeForce GTX 580, 280)[8].

Jeremy Fowers et al. studied the 1D convolution for image processing on the CPU, GPU and FPGA platforms [9].

In [10], four representative benchmarking examples including STREAM benchmark, Matrix Multiplication, FFT and Asian Option Pricing were selected to compare the GPU (Tesla C1060) and FPGA (totally 16 Xilinx V5LX330).

In [11], Shuichi Asano et al. compared CPU(Core 2 Extreme QX6850), GPU(GeForce 280 GTX) and FPGA(Xilinx XC4VLX160 @100MHz) with 3 algorithms used in image processing.

For all the comparison work mentioned in above, we note that there is no full winner between the GPU and FPGA. Each has its advantage over the other for some particular application cases, even the same application with different input data or algorithmic parameters.

Our heterogeneous computing platform differentiates from the previous work by focusing on power efficient high performance computing. We include power measurement facility and a run-time manager to take advantage of the hardware diversity.

III. HETEROGENEOUS COMPUTING PLATFORM

We build a prototype of the power efficient heterogeneous computing platform. Fig. 1 shows the system framework of the proposed heterogeneous computing platform.

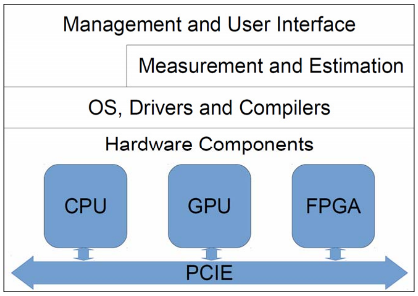

Management and User Interface
Measurement and Estimation
OS, Drivers and Compilers
Hardware Components
CPU
GPU
FPGA
PCIE

Fig. 1. *System framework of the heterogeneous computing platform*

A. Hardware Components and System Software

Hardware components are purchased off-the-shelf and then integrated to build the heterogeneous computing platform.

The CPU used in the prototype of the heterogeneous computing platform is a quad-core Core i7 870 operating at 2.93 GHz with 8 GB memory and a thermal design power of 95 Watts.

The GPU card used in the prototype heterogeneous computing platform is a Tesla C2075 from NVIDIA. The thermal design power of the Tesla C2075 is 225 Watts.

The FPGA board used in the prototype heterogeneous computing platform is a MAX3 board from the Maxeler[12], which has a Xilinx's Virtex 6 series XC6VSX475T chip and 24 GB on board memory. The power consumption MAX3 FPGA board varies from about 12W to 45W.

The prototype heterogeneous computing platform uses a specialized power supply unit with the electrical current voltage measurement facility. The model of power supply unit is ODIN GT 800W manufactured by the Gigabyte.

CentOS 6.4 is installed on the heterogeneous computing platform. Device drivers and development tools are obtained from the vendors of the GPU and FPGA.

B. Measurement and Estimation

Measurement is done by utilizing the available hardware and software facilities. Execution time can be recorded by using software function calls or the hardware performance counters built in the modern processors.

For power consumption measurement, fortunately, GPU and FPGA vendors provide their own tools to measure the power consumption of the GPU card and FPGA board respectively.

CPU is connected to one of the output channels of the power supply, so its power consumption can be calculated by multiplying the dynamic current and voltage values.

Estimation of the performance and power consumption is the key for management module in dispatching the application kernels to the appropriate computing architecture. We adopt an approach of profiling and classification to the estimation of the performance and power consumption of applications. Basically, it is assumed that the code for CPU execution is available. By utilizing some profiling tools, such as gprof and gcov, the profiling of the application is performed which generates some profiling data indicating the number of operations, size of data processed and other information. The available computing resources of the computing architectures are described as the resource information. Aligning the required operations from the profiling data to the available computing resources, the estimation of the resource utilization is obtained. Combining the resource utilization and upper bounds of the computing capability calculated by the roofline model[13], performance of the application kernels on the designated computing architecture is estimated. Once the performance estimate is available, power consumption can be estimated from the estimated performance and resource utilization. Fig. 2 shows the work flow of the estimation of the performance and power consumption.

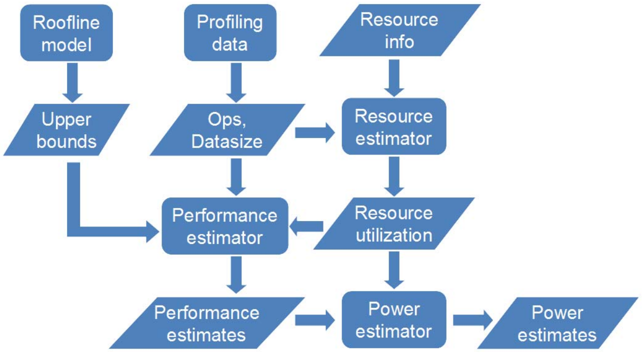
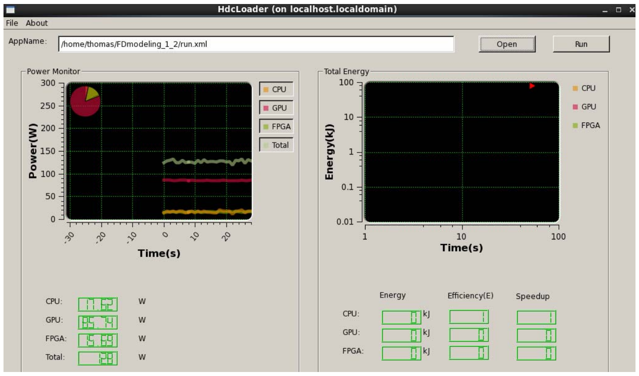


Fig. 2. *Work flow of the estimation of the performance and power consumption for the heterogeneous computing platform*

C. Management and User Interface

Fig. 3 shows the user interface of our preliminary prototype.



AppName: /home/thomas/FDmodeling_1_2/run.xml

CPU:	17.62	W
GPU:	85.74	W
FPGA:	15.69	W
Total:	128	W

	Energy	Efficiency(E)	Speedup
CPU:	0 kJ	1	1
GPU:	0 kJ	0	0
FPGA:	0 kJ	0	0

Fig. 3. *User interface of for the heterogeneous computing platform*

The management module takes input from the user through the user interface, which is an XML file describing the information of the application to be executed on the heterogeneous computing platform. The application information includes the executable path and name of the application, the arguments to pass into the application, as well as other parameters for the execution of the application. Based on the application information, the management module maps the kernels of the application to the appropriate architecture. The performance and power consumption of the application are measured and displayed periodically.

IV. Experimental Results

A series of experiments have been carried out on the prototype heterogeneous computing platform.

A. Verification of the Idea

We test 4 application examples, the N-body simulation, the matrix multiplication, the finite differential modeling for shock waves transmitting through the earth, and the Heston model for option pricing. Fig. 4 and Fig. 5 show the performance and power efficiency of these four application examples running on CPU, GPU and FPGA respectively.

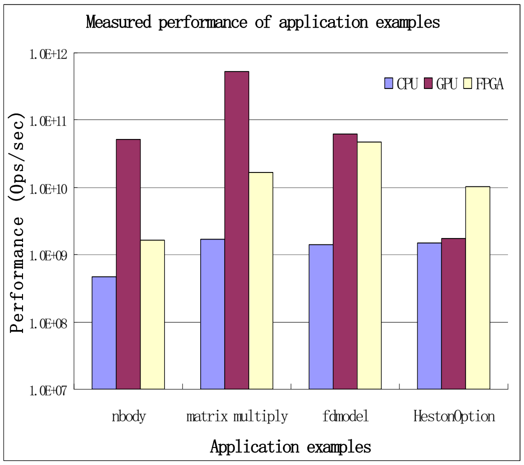


Fig. 4. *Performance of CPU, GPU and FPGA for the 4 application examples on the heterogeneous computing platform*

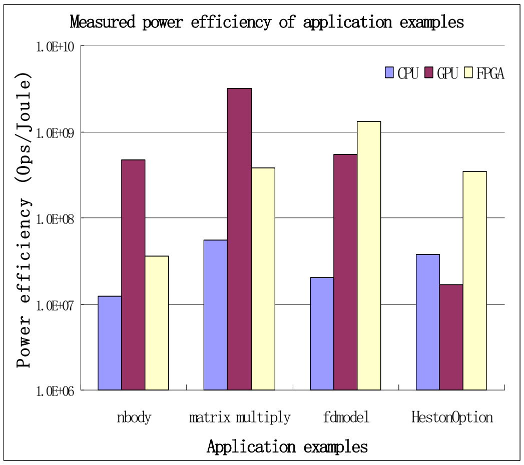


Fig. 5. *Power efficiency of CPU, GPU and FPGA for the 4 application examples on the heterogeneous computing platform*

It can be seen from Fig. 4 and Fig. 5 that among the 4 application examples, the N-body simulation, the matrix multiplication and the finite differential model get the highest performance on GPU, while the Heston Option pricing gets the highest performance on FPGA. For the power efficiency, the Nbody simulation and the matrix multiplication get the best score on GPU, while the finite differential model and the Heston Option pricing get the best score on FPGA. Apparently, there is no one best computing architecture for all applications.

B. Estimation and Real Execution

Fig. 6 and Fig. 7 show the estimated performance and power efficiency on the CPU, GPU and FPGA respectively for the 4 application examples. Note the error bars show the differences between the estimations and real measurements. It looks that the estimated performance and power efficiency are over 10 times different from the real ones. Fortunately, the estimation conforms to the real execution qualitatively.

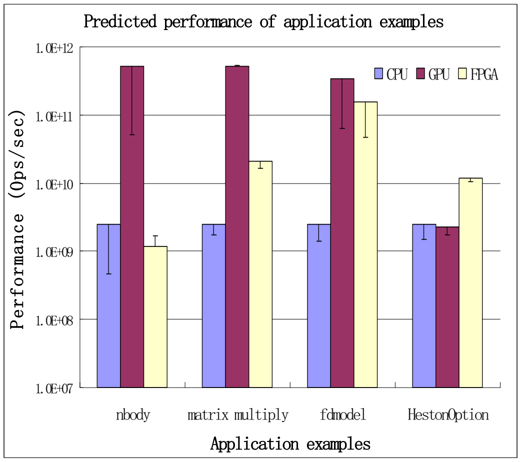


Fig. 6. *Performance estimation of CPU, GPU and FPGA for the 4 application examples. Note the error bars show the differences between the estimations and real measurements*

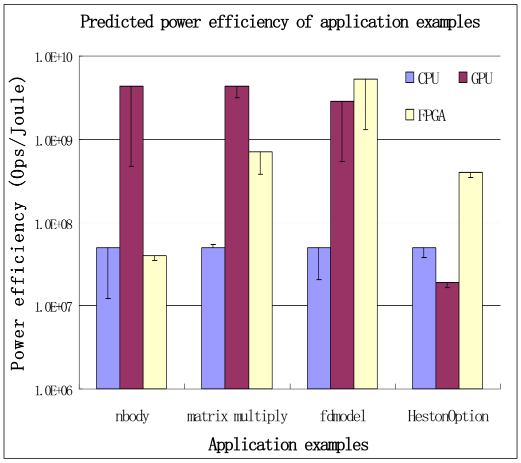


Fig. 7. *Power efficiency estimation of GPU and FPGA over CPU for the 4 application examples. Note the error bars show the differences between the estimations and real measurements*

V. Summary and Conclusions

In this paper, we compare the performance and power efficiency of 4 representative application examples on a heterogeneous platform with CPU, GPU and FPGA. The 4 application examples are the N-body simulation, the matrix multiplication, the finite differential modeling of shock waves in earth and the Heston Model for option pricing. The test results show that different applications really need different computing architectures to take advantage of the application's characteristics for the best performance and power efficiency.

One of the key problems in utilizing a heterogeneous computing platform is the estimation of the application performance and power efficiency on different computing architectures for appropriate architecture dispatching. In our experiment, we note that the estimations of the application examples conform to the real execution results qualitatively. We will continue to study on the estimation methodology to improve its accuracy.

References

[1] T. El-Ghazawi, *et al*. The promise of high-performance reconfigurable computing. Computer, pages 69-76, February 2008.

[2] Jonathan Koomey. Growth in Data center electricity use 2005 to 2010. Oakland, CA: Analytics Press. 2010.

[3] Michael Showerman , Jeremy Enos, *et al*. QP: A heterogeneous multi-accelerator cluster. In Proc. of 10th LCI Intl. Conf. on High-Performance Clustered Computing, 2009.

[4] Kuen Hung Tsoi, Wayne Luk. Axel: A Heterogeneous Cluster with FPGAs and GPUs. In Proc. of FPGA, 2010.

[5] Ben Cope, *et al*. Performance Comparison of Graphics Processors to Reconfigurable Logic: A Case Study. IEEE Trans. on Computers, vol. 59, no. 4 : 433-448, 2010.

[6] John Bodily, *et al*. A Comparison Study on Implementing Optical Flow and Digital Communications on FPGAs and GPUs. ACM Trans. on Reconfigurable Technology and Systems, vol. 3, no. 2, Article 6, May 2010.

[7] Xiang Tian, Khaled Benkrid. High-Performance Quasi-Monte Carlo Financial Simulation: FPGA vs. GPP vs. GPU. ACM Trans. on Reconfigurable Technology and Systems, vol. 3, no. 4, 2010.

[8] Karl Pauwels, *et al*. A Comparison of FPGA and GPU for Real-Time Phase-Based Optical Flow, Stereo, and Local Image Features. IEEE Trans. on Computers, vol. 61, no. 7, 2012.

[9] Jeremy Fowers, *et al*. A Performance and Energy Comparison of Convolution on GPUs, FPGAs, and Multicore Processors. ACM Trans. on Architecture and Code Optimization, vol. 9, no. 4, 2013.

[10] Brahim Betkaoui, *et al*. Comparing Performance and Energy Efficiency of FPGAs and GPUs for High Productivity Computing. Proc. of FPT, 94-101. 2010

[11] Shuichi Asano, *et al*. Performance Comparison of FPGA, GPU and CPU in Image Processing. Proc. of FPL, 126-131. 2009

[12] Maxeler Technologies. http://www.maxeler.com/, 2013.

[13] Samuel Williams, Andrew Waterman, David Patterson. Roofline: an insightful visual performance model for multicore architectures. Communications of the ACM, vol. 52, iss. 4, 2009.