An Overview of Current and Future Computing Accelerator Architectures

Scott J. Krieder*, Ioan Raicu*†
*Department of Computer Science, Illinois Institute of Technology
†MCS Division, Argonne National Laboratory

***Abstract*—Accelerator technologies are now quite common in Supercomputers, Clusters, Grids, and personal desktops. This work aims to provide an overview of the current technologies that are available today, and examine future accelerator technologies. This work examines the 3 major competitors in the Accelerator market including: NVIDIA, Intel, and AMD.**

***Keywords*-Accelerators, Coprocessors, GPGPGU, NVIDIA, AMD, Intel Xeon Phi.**

I. Introduction

Hardware accelerators allow the machine to offload work from the host CPU to the Accelerator. The Accelerator then completes the computation and returns the solution back to the host CPU which relieves the host of previous compute cycles. In this work we analyze three major competitors in the HPC Hardware Accelerator market including: NVIDIA, AMD, and Intel. The motivation for examining accelerators revolves around Many-Task Computing(MTC).[1] By enabling MTC on Accelerators we believe that can bridge the gap between clusters and GPU compute. MTC is an excellent candidate for the GPUs many power efficient cores. Some disadvantages that provide added motivation include: 1)Slow data transfers, 2)Lag induces communication, and 3)Programmability which requires additional languages. Our future work aims to address the shortcomings to provide efficient support for MTC on Hardware Accelerators.

II. Accelerators

A. NVIDIA GPGPUs

NVIDIA recently launched their newest GPU architecture called Kepler. [2] Kepler provides many added benefits, but the 3 most noteworthy include: 1)SMX, 2)Dynamic Parallelism, and 3)Hyper Q.

Under the new architecture SMX compute elements can contain up to 192 cores per Streaming Multiprocessor compared to 32 cores per SM in previous architectures. This accounts for a 3X performance/watt.

Dynamic Parallelism allows a thread on the GPU to spawn more threads. Under previous architectures a GPU thread had to be initiated from the host CPU resulting in increased communication back and forth across the PCI bus. Dynamic Parallelism allows for an extensive reduction in the number of communications across the PCI bus and therefore increased performance.

Finally, Hyper Q allows additional CPU cores to interact with the device. This eliminates host CPU idle time and adds to performance.

B. AMD GPGPUs

AMD GPUs provide an increase level of openness in regards to programmability. [3] OpenCL is an open standard and there are any array of related works to support Java, Fortran, C++, .NET and Python. However, due to the abstract nature of OpenCL it might not provide the highest level of hardware performance and adds challenges to code reusability and maintenance.

C. Intel Xeon Phi Coprocessor

The Intel Xeon Phi is Intel's answer to GPGPU computing.[4] Unfortunately, it is a new device and there is not much public information. Because Intel is in the market for producing CPUs we can expect this device to have a familiar x86 architecture. The device should have a common form factor to the competitions GPUs but will not require a host CPU to drive the device. Because of this, Intel is not marketing the device as an Accelerator but rather as a Coprocessor, something that complements a CPU but can work on it's own. The device should have on the order of 50 x86 cores and provide support for familiar languages including C, C++ and OpenCL.

III. Future Work

GPU Virtualization is one future work that we believe can provide increased GPU utilization and performance. For applications that cannot utilize an entire GPU, a portion of that GPU remains idle. A virtualized GPU would provide a small cluster for compute and enable separate applications and users to share hardware resources.

Workflow Systems have proven a highly successful solution for homogenous CPU based systems. Todays architectures are growing increasingly hybrid and by enabling Workflow Systems to support hardware accelerators such as NVIDIA GPUs we believe that we can increase application performance and system utilization.

IV. Conclusions

In conclusion this work evaluates current generation hardware accelerators including the NVIDIA GPUs, AMD GPUs, and the Intel Xeon Phi. Running CUDA on NVIDIA GPUs is

one of the most mature GPGPU solutions and provides high raw computational performance, however this does require code ported to the CUDA platform. The Intel Xeon Phi suffers from a lack of availability, but once this device is highly available it should bring large improvements in regards to accelerator programmability due to the familiar x86 environment. Finally, AMD GPUs provide a high level of openness in regards to programmability. AMD supports open standards such as OpenCL but may see difficulty in adoption within the HPC markets due to performance. The future of HPC will definitely involved Accelerators, but only time will tell which vendors win the war.

REFERENCES

[1] I. Raicu, Z. Zhang, M. Wilde, I. Foster, P. Beckman, K. Iskra, and B. Clifford, "Toward loosely coupled programming on petascale systems," in *Proceedings of the 2008 ACM/IEEE conference on Supercomputing*. IEEE Press, 2008, p. 22.

[2] NVIDIA. (2012, May) Nvidia kepler compute architecture — high performance computing — nvidia. [Online]. Available: NVIDIA Kepler - http://www.nvidia.com/object/nvidia-kepler.html

[3] AMD. (2012, May) Opencl zone — amd. [Online]. Available: http://developer.amd.com/resources/heterogeneous-computing/opencl-zone/

[4] Intel. (2012, May) Intel many integrated core architecture - advanced. [Online]. Available: http://www.intel.com/content/www/us/en/architecture-and-technology/many-integrated-core/intel-many-integrated-core-architecture.html