An Overview of Current and Future Computing Accelerator Architectures

Michael Wilde[†‡], Ioan Raicu[*], Justin M. Wozniak[†], Scott J. Krieder[*]
[*]Department of Computer Science, Illinois Institute of Technology
[†]MCS Division, Argonne National Laboratory
[‡]Computation Institute, University of Chicago & Argonne National Laboratory

***Abstract*—This project explores a programming model and runtime environment which addresses the urgent yet vexing problem of how to simplify the programming of complex hybrid systems architectures. As computing systems from personal tablets to the largest extreme-scale systems become increasingly parallel, and as performance gains are often made using hybrid architectures, the difficulty of programming such systems increases dramatically. The long-sought-after goal in programming model research is to make parallel programming automatic freeing the programmer from having to explicitly code the intricate data passing and data sharing operations that are needed in common parallel programming models such as distributed and shared memory models. As hybrid architectures become increasingly turned-to as the means for further cost/performance improvements, the complexity increases with difficult orthogonal factors such as passing data across the CPU-GPU boundary.**

***Keywords*-Keywords.**

I. Introduction

This is the intro. [1]

II. Updates

A. Swift/T

B. GeMTC

III. Future Work

A. Applications

References

[1] S. J. Krieder and I. Raicu, "Towards the support for many-task computing on many-core computing platforms," Doctoral Showcase, IEEE/ACM Supercomputing/SC, 2012.