

Towards Exascale System: An Automatic Hardware Software Co-design Framework for Current and Future Architectures in Aspen

Mariam Umar*, Jeremy S. Meredith (Advisor), Jeffrey S. Vetter (Advisor), Kirk W. Cameron*(Advisor)

Department of Computer Science, Virginia Tech*

Oak Ridge National Laboratory

mariam.umar@vt.edu, jsmeredith@ornl.gov, vetter@ornl.gov, cameron@vt.edu

I. INTRODUCTION

As we approach the exascale era, we face numerous design and codesign challenges of large-scale systems. These challenges span both hardware and software design. While the software design engineers are progressing towards a consensus in the context of exascale, doubts in hardware design still linger. This delay is due, in part, to a lack of an agreement of a generalized hardware technology. Moreover the integration between software and hardware is still a pending decision. Difficulties for exascale stem from post-Moores era sudden increase in challenges and complexities in hardware design (along with increase in complexity of applications). We still are at the cusp between fat and thin nodes and between no accelerators to multiple types of accelerators [1]. This problem is exacerbated with the way the benefit from architectural improvements have slowed, such as the lack of performance improvement with increase in clock speed or changing DVFS, and the ever increasing energy demands are further limiting our solution set. The situation is worse when we need to integrate hardware with software in order to design the entire system.

We believe that automated discovery of hardware specifications and software parameters can help modelers keep pace with architectural changes and to continue benefiting from the ever increasing demands of better performance and lesser energy consumption. The ultimate goal could be seen as the development of application specific hardware — i.e., the right parametric configurations for hardware that are adapted as per application requirements without much programmer effort and vice versa. We therefore propose the automated codesign with an emphasis on hardware and software for the Aspen domain specific language [2] for performance engineering. There have been some reports and initial studies in the field of hardware-software codesign and automation. For example, Shalf et al., [1] propose CoDEX: A hardware-software codesign for exascale systems, seems promising but suffers from long execution times that are characteristic of simulators. Another abstract machine model is proposed by Ang et al., [3], which suggests using of an abstract machine model for exascale systems, but suffers from programming and portability overheads. We believe that previous approaches

have failed because of tremendous programmer effort and lack of portability. We therefore propose automated hardware-software co-design for Aspen, which is fully automated and portable to a wide range of current and future systems.

Aspen is a domain specific language that helps in creating modular, sharable and composable performance models. It enables analysis that can be used to explore the design space of hardware and software. Aspen has the potential in exploring and improving existing architectures and applications. This automation solves numerous issues faced by the research community, such as designing current and exascale system with maximum performance and minimum energy consumption, fusing multiple memories together to achieve maximum bandwidth, implementing resilience methods on memory hierarchies, making the approach portable, simple, accurate, and robust, and changing application specific changes in hardware and vice versa.

Our approach consists of automatic generation of application and machine models for Aspen. For application models, an input program analyzer takes source code, analyzes application characteristics — such as flops, loads, stores, iterations, complexity, etc. — and generates Aspen IR (Intermediate Representation), which is adapted from Compass [4]. It is then consumed by an Aspen IR post processor and produces an Aspen application model. For abstract-machine models, a machine-specification extractor uses a combination of kernel and user level functions to wring out machine specifications — such as sockets, cores, nodes, memory type and hierarchies, network topology, PCIe latency, memory bandwidth, etc. Using the extracted machine parameters, we generate the Aspen IR, which is consumed by the Aspen machine IR post processor and yields an Aspen abstract-machine model.

The automated framework provides several advantages, such as no requirement for hardware while testing applications, portability and extensibility of current and future architectures, requires no programmers effort, has low overhead, and inherits accuracy from Aspen as previously established in [2].

Our framework has several applications, three of which are shown in this poster: 1) sensitivity analysis of six applications that demonstrate the % differences of various hardware parameters in relation to execution times, with respect to baselines (which are specifications of hardware as provided

by vendors). We tested these configurations with varying input sizes. 2) strong and weak scaling, as well as energy profiling of applications for current supercomputers. 3) an abstract machine model of a homogeneous exascale system, and performed scalability testing and energy profiling for the LULESH proxy application.

An appealing motivation with abstract models for exascale system is that it doesn't require full-scale tests that consumes 20 MW of energy and it provides useful feedback to the user particularly with respect to scaling. Our plans to explore other architectural features and their details such as non-volatile memory, memory hierarchies and network topologies

will appear in our upcoming publication.

REFERENCES

- [1] J. Shalf *et al.*, "Exascale computing technology challenges," in *High Performance Computing for Computational Science-VECPAR 2010*, 2010.
- [2] K. L. Spafford and J. S. Vetter, "Aspen: a domain specific language for performance modeling," in *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, 2012.
- [3] J. Ang and R. F. e. a. Barrett, "Abstract machine models and proxy architectures for exascale computing," in *Hardware-Software Co-Design for High Performance Computing (Co-HPC)*, 2014, 2014.
- [4] S. Lee *et al.*, "COMPASS: A framework for automated performance modeling and prediction," in *Proceedings of the 29th ACM on International Conference on Supercomputing*, 2015.