

# Hardware/Software Co-Design for High Performance Computing: Challenges and Opportunities

X. Sharon Hu  
Department of Computer  
Science and Engineering  
University of Notre Dame  
Notre Dame, IN, USA  
shu@nd.edu

Richard C. Murphy  
Scalable Computer  
Architecture Department  
Sandia National Laboratories  
Albuquerque, NM, USA  
rcmurph@sandia.gov

Sudip Dosanjh  
Computer and Software  
Systems Department  
Sandia National Laboratories  
Albuquerque, NM, USA  
ssdosan@sandia.gov

Kunle Olukotun  
Department of Electrical  
Engineering  
Stanford University  
Stanford, CA, USA  
kunle@stanford.edu

Stephen Poole  
Computational Sciences  
Division  
Oak Ridge National Lab  
Oak Ridge, TN, USA  
spoole@ornl.gov

## ABSTRACT

This special session aims to introduce to the hardware/software codesign community challenges and opportunities in designing high performance computing (HPC) systems. Though embedded system design and HPC system design have traditionally been considered as two separate areas of research, they in fact share quite some common features, especially as CMOS devices continue along their scaling trends and the HPC community hits hard power and energy limits. Understanding the similarities and differences between the design practices adopted in the two areas will help bridge the two communities and lead to design tool developments benefiting both communities.

## Categories and Subject Descriptors

C.0 [General]: Systems specification methodology; C.4 [Performance of Systems]: Design studies

## General Terms

Design, measurement, performance

## Keywords

High performance computing, hardware/software codesign

## 1. INTRODUCTION

High performance computing (HPC) has long been considered a different breed from embedded computing (EC). The two corresponding platforms, targeting at two separate markets, handle disparate workloads with different performance concerns. It seems natural for the two communities to have diverse design philosophies and design tools.

Hardware/software codesign, as a design paradigm introduced in the early nineties, has been studied extensively by the EC system design community. Tremendous progress

has been achieved in this area of research, which has led to the birth of new languages, new CAD tools, new companies, etc. The integrated design concept has brought new thinking to many aspects of computer system design including process scheduling, communication protocols, memory management, software development, as well as design of application-specific processors and reconfigurable architectures. However, the impact of hardware/software design has mostly be limited within the EC system domain.

As the CMOS scaling trends (in terms of density, performance and power) continue, the boundary between HPC and EC design practices is becoming blurry. The HPC community has recognized that it is no longer affordable to simply procure faster machines with bigger memories and that rigorous analysis and design tools can be indispensable in order to meet power/performance requirements of HPC systems. This recognition presents a unique opportunity for the hardware/software researchers to expand their influence and advance the design technologies for HPC, and for the HPC system designers to leverage the knowledge and experience gained in the last two-decades' hardware/software codesign research.

In the rest of this extended abstract, we comment on general practices that have been adopted for specifying and designing HPC systems as well as their limitations, and highlight key challenges that the HPC system design community is facing. We also briefly sample recent progress toward addressing some specific design problems in HPC systems. We end the paper by examining similarities and differences between HPC and EC and discussing what aspects of HW/SW co-design may be useful for designing HPC systems and what aspects need further study.

## 2. HPC SYSTEM DESIGN PRACTICES

As a general practice, the procurements of supercomputers for HPC have often been based on some combination of (1) benchmarks, (2) byte to flop ratios (e.g., the bandwidth divided by the floating point rate must be greater than some number) and (3) aggregate requirements. The Red Storm [6] and Ceilo [1] procurements further required the bidding com-

panies to guarantee that the supercomputer's performance on a range of applications would be a certain factor greater than the performance of these applications on a previous supercomputer.

It is not difficult to see from the above description that trying to define the current HPC design process can be a challenge. The challenge actually exists at most all levels in the process. This lack of a cohesive co-design process has been a gradual process of degradation over the past few decades. There are many symptoms we should have seen along the way, that seem to have gone either unnoticed or at least not acted upon. For example, overly constrained design requirements ("work on everything but the processor"), increasing system imbalance, decreased processor utilization have all contributed to increased energy cost. It is now estimated that many HPC systems will cost as much to power over their lifetime as to purchase.

The HPC community has by in large lost the capability to have a major impact on a few elements of the base elements of a computing system. As we moved away from HPC centric processors a few decades ago, we currently are forced to deal with COTS processors. These processors by in large are NOT designed with HPC in mind as their primary or even secondary target. With this in mind, the HPC community has had to "live with what we get". This philosophy/constraint is extended into the software arena as well. Various government elements have spent hundreds of millions of dollars trying to improve the software space. Unfortunately, it has not had a much better impact on our plight, and it seems that most of these expeditions are only expensive band-aids on a severe wound. We have gone from 25-50% utilization in some HPC applications to 3-7% utilization of these HPC systems. The lower utilization leads to increased energy waste.

The general practice and the methods adopted for HPC system design work well when the architectures are well understood and meaningful measurements can be made. However, new design methodologies are desperately needed now because (1) there is a significant architectural change underway; (2) both hardware and software R&D will be needed to overcome the challenges associated with Exascale computing; and (3) we are approaching hard power and energy constraints.

HPC researchers are keenly aware of the problem and have started to investigate ways to mitigate the major design challenges. By far the biggest design challenge is increasing the energy efficiency of HPC systems to enable Exascale computing. One approach to energy efficiency has been the move to simpler processors. Simpler processors reduce the number of logic transitions per instruction, while adding more of them in parallel overcomes the per core performance loss and increases aggregate performance. Unfortunately one cannot do this forever, since at some point reducing each processor's performance has only a small effect on energy efficiency but dramatically increases the level of application parallelism required.

The most effective method of achieving energy efficiency is reformulating a task so it needs fewer operations to complete. This sort of "customization" of the processor architecture can improve the energy efficiency by an order of magnitude. The Green Flash project has taken steps down this path with the use of customized embedded processors for large-scale climate simulation [3]. One approach that has

the potential to simplify the programming of HPC systems and make possible more efficient customization is super-optimization using domain-specific knowledge captured by a domain specific language (DSL) [2].

### 3. LOOKING FORWARD

The battle that is being fought by the HPC community bears resemblance to what the EC community has gone through two decades ago. In fact, the HPC and the EC communities face many similar obstacles in both software, hardware and application implementation in the resulting systems. Both communities are dealing with ever-increasing hardware capabilities with significant architectural changes, have many overlapping figures of merits, and not to mention the growing design productivity gap. Realizing these factors has helped generate greater interest in hardware/software co-design in the HPC community [4, 5].

There is much to be gained by mining co-design research that has been done by the EC community. However, the methodology will need to be developed further for HPC because of several complicating factors: (1) architectures must be optimized to support many applications, (2) HPC applications can be very complex (many have millions of lines of code), (3) supercomputers are very complex, and (4) applications will change significantly during the next decade (a new programming model will be adopted and applications will need to manage on-node locality and deal with an explosion in parallelism). These challenges will bring exciting opportunities to both EC and HPC design communities and propel the hardware/software co-design research area to a new height.

### 4. REFERENCES

- [1] J. Ang, D. Doerfler, S. Dosanjh, K. Koch, J. Morrison and M. Vigil, "The alliance for computing at the extreme scale," *Proceedings of the Cray Users Group Meeting*, 2010.
- [2] H. Chafi, Z. DeVito, A. Moors, T. Rompf, A. Sujeeth, P. Hanrahan, M. Odersky and K. Olukotun, "Language virtualization for heterogeneous parallel computing," *Onward!*, October 2010.
- [3] D. Donofrio, L. Oliker, J. Shalf, M. F. Wehner, C. Rowen, J. Krueger, S. Kamil and M. Mohiyuddin, "Energy-efficient computing for extreme-scale science," *IEEE Computer*, Vol. 42, No. 11, pp. 62-71, November 2009.
- [4] S. Dosanjh, "Exascale computing and the role of codesign," *International Advanced Workshop on High Performance Computing, Grids and Clouds*, Cetraro, Italy, 2010. <http://www.hpcc.unical.it/hpc2010/ctrbs/dosanjh.pdf>
- [5] A. Geist and S. Dosanjh, "IESP exascale challenge: co-design of architectures and algorithms," *International Journal of High Performance Computing*, Vol. 23, No. 4, 2009, pp. 401-402.
- [6] J. Tomkins, R. Brightwell, W. Camp, S. Dosanjh, et. al., "The Red Storm architecture and early experiences with multi-core processors," *International Journal of Distributed Systems and Technologies*, Vol. 1, Issue 2, 2010, pp. 74-93.