

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/220475922>

Codesign for Systems and Applications: Charting the Path to Exascale Computing

Article in *Computer* · November 2011

DOI: 10.1109/MC.2011.334 · Source: DBLP

CITATIONS

5

READS

46

3 authors, including:



Vladimir Getov

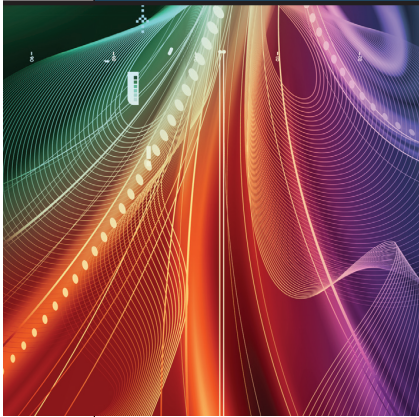
University of Westminster

164 PUBLICATIONS 1,294 CITATIONS

[SEE PROFILE](#)

All content following this page was uploaded by Vladimir Getov on 07 May 2021.

The user has requested enhancement of the downloaded file.



Codesign for Systems and Applications: Charting the Path to Exascale Computing

Vladimir Getov, *University of Westminster*

Adolfy Hoisie, *Pacific Northwest National Laboratory*

Harvey J. Wasserman, *Lawrence Berkeley National Laboratory*

The clock speed benefits of Moore's law have ended, and researchers must codesign future exascale HPC systems and applications concurrently in an integrated manner to achieve higher performance under stringent power and reliability constraints.

Computational science has become a vital tool in the 21st century, central to progress at the frontiers of nearly every scientific and engineering discipline, including many areas with significant societal impact. A persistent need for more computing power has provided an impetus for the high-performance computing (HPC) community to embark upon the path to exascale computing.

The challenges associated with achieving efficient, highly effective exascale computing are extraordinary. Past growth in HPC has been driven by performance and has relied on a combination of faster clock speeds and increasingly larger systems. Achieving exascale performance under reliability and power constraints and in the presence of levels of parallelism increased by orders of magnitude will change the path of system and application development.

A recent DARPA study showed that even if it were technically feasible, exascale systems built following the current trajectory would require an energy budget in the hundreds-of-megawatts-per-hour range and reliability estimates that would render them impractical.¹ Thus, the clock speed benefits of Moore's law have ended, and the emphasis must now unavoidably yield to the goal of achieving performance under stringent power and reliability constraints.

EXASCALE COMPUTING CHALLENGES

The issues researchers will encounter on the path to exascale HPC are equally critical for all large-scale computing architectures and facilities, not just the largest ones or only those related to scientific computing. Workloads may differ, but energy challenges are common. Because power is the overriding hardware concern, energy efficiency will be essential across all computing scales. Furthermore, energy issues will affect all levels of the computing system, including processors, interconnects, algorithms, software, and programming models.

Given the complexity of the increasingly daunting constraint space under consideration, successful optimization requires a new tack, a new approach, and a new set of design methodologies. For example, given the overwhelming performance and energy cost of data movement,² efficiency requires minimization of data movement—a task for all layers of the stack, from the hardware to the application software.

Similarly, optimization of the performance/power/reliability triad mandates rethinking of algorithms, programming models, and hardware in concert and requires an unprecedented level of collaboration and cooperation in hardware, system architecture, system software, and application codesign. This requires a completely new approach based on concurrent development and engineering in an integrated manner to a set of consistent overall design metrics, employing accurate, quantitative design methodologies.

THE CODESIGN APPROACH—BACKGROUND

For embedded systems,³ codesign traditionally has meant partitioning concepts in the design process to produce systems meeting stringent performance, verifi-

cation, and other specifications within a shorter design cycle. The goal and methodology for doing this, as well as the benefits of this approach, have been well-established for many years. The key concept is meeting system-level objectives by exploiting tradeoffs between hardware and software through an integrated concurrent design process. An additional benefit accrues from automation or semiautomation of this concurrent design process, but the crucial part of the definition is concurrency: developing hardware and software at the same time on parallel paths.

What was perhaps left undefined was the precise nature of the interaction between hardware and software. This interaction evolved over the years with increasing use of improved design automation tools, faster application-specific integrated-circuit development tools that allow quick and inexpensive implementation of complex algorithms in silicon, and the use of reduced-instruction-set computing technology that allows the implementation of traditional hardware functionality in software.



The key concept is meeting system-level objectives by exploiting tradeoffs between hardware and software through an integrated concurrent design process.

Codesign in embedded systems came about in large part because a variety of factors led to the use of software in systems that had previously been entirely hardware-based. This increased the complexity of that software in microcontrollers, digital signal processors, and even general-purpose processors. Other factors included the decreasing cost of microcontrollers, rapidly increasing numbers of available transistors, the availability of advanced emulation technology, and the improved efficiency of higher-level language compilers for use in embedded systems. A key motivation was the need to support the growing complexity of embedded systems, which has an obvious parallel in exascale computing.

Embedded systems are characterized by running only a few applications that are completely known at design time, not being programmable by end users, and having fixed runtime requirements—meaning that additional dynamic computing power is not useful.⁴ Codesign considerations for such systems include cost, power consumption, predictability, and meeting time bounds.

In contrast, general-purpose computing systems are characterized by running a broad class of applications, being programmable by end users, and having the characteristic that faster is always better, which requires including cost and peak speed in their design criteria.⁴

The essence of the codesign challenge for HPC and exascale systems is to use the key design criteria of embedded systems—cost and power consumption—while creating systems that are useful and effective over the broad range of applications needed to advance science. “One-off” exascale systems will suggest failure.

CODESIGN FOR HPC SYSTEMS

In the HPC arena, codesign has also been used recently,⁵ and therefore it is not entirely new to exascale computing. Both the IBM BlueGene/L supercomputer (*IBM J. Research and Development*, vol. 49, no. 2/3, 2005) and IBM’s PERCS project for DARPA’s High-Productivity Computing Systems (HPCS) program⁶ have adopted the codesign approach. Two additional excellent examples of codesigned special-purpose supercomputers for molecular dynamics applications are the MDGrape system⁷ and the Anton supercomputer⁸ built by D.E. Shaw Research.

The IBM RoadRunner⁹ pointed the way toward the use of hybrid architectures by its inclusion of coprocessing elements along with general-purpose processors to accelerate a specific workload. The heterogeneity of the resultant architecture, which required a mixture of several programming models, posed significant challenges to ensure the utility of the coprocessor approach for designing HPC systems that are to be truly effective for a wide range of scientific applications. Still, the metric for success in these codesign examples was performance, without regard to power and reliability.

The HPC community currently finds itself needing to apply codesign methods on the path to exascale systems and applications.¹⁰ Therefore, key concepts that apply include

- employing a high level of abstraction to describe the system;
- using models to allow analysis and exploration of the system architecture, validate assumptions regarding the architecture, explore the design implementation performance parameters, and verify that tradeoffs made using high-level system models were worthwhile; and
- creating codesign methodologies and tools that designers can use to “tinker” with the platform, adding, subtracting, or changing parameters to determine the effect on the architecture and system performance.

A novel exascale concept is related to the necessity of rethinking the application software itself, including optimizing the algorithms and the codes for minimizing data movement for energy efficiency or for implementing resilience mechanisms. Hence, these virtual testbeds need to support initial optimization by both system and appli-

cation designers, before expensive and time-consuming actual implementations are necessary.

IN THIS ISSUE

The articles in this special issue have been selected to cover a cross-section of the codesign space and of the relevant concerns and challenges.

Architectural simulation

"Rethinking Hardware-Software Codesign for Exascale Systems" by John Shalf, Dan Quinlan, and Curtis Jansen describes a set of high-accuracy simulation tools that researchers can employ for low-level hardware and architecture codesign for a simplified application workload.

High-speed interconnects

In "Codesign for InfiniBand Clusters," Sreeram Potluri and coauthors discuss a codesign approach that takes advanced features from the commodity InfiniBand network, incorporates the design into a state-of-the-art message-passing interface communication library, and then modifies applications to leverage these new features.

High-end systems

"Codesign Challenges for Exascale Systems: Performance, Power, and Reliability" by Darren J. Kerbyson and colleagues describes a comprehensive codesign methodology that uses analytical modeling to achieve maximum performance, power, and reliability for full systems and applications.

In its simplest definition, codesign is about anticipating and changing the future.¹¹ Early intervention in hardware designs, optimizing what is important, influencing the design, redesigning algorithms and system software, devising languages and programming models that reflect abstract machine models, writing code generators and autotuners, and modeling all of the above are the essence of the craft. Successfully meeting these challenges is essential for continued progress in computing performance. 

References

1. P.M. Kogge, ed., "ExaScale Computing Study: Technology Challenges in Achieving Exascale Systems," tech. report TR-2008-13, CSE Dept., Univ. of Notre Dame, 2008; www.cse.nd.edu/Reports/2008/TR-2008-13.pdf.
2. W. Dally, "Power, Programmability, and Granularity: The Challenges of ExaScale Computing," *Proc. 25th IEEE Int'l Parallel & Distributed Processing Symp. (IPDPS 11)*, IEEE, 2011, p. 878.
3. T.A. Henzinger and J. Sifakis, "The Discipline of Embedded Systems Design," *Computer*, Oct. 2007, pp. 32-40.
4. L. Thiele, "Hardware/Software Codesign," AS 2011 course materials; www.tik.ee.ethz.ch/tik/education/lectures/hswcd.

5. M. Wolfe, "Compilers and More: Hardware/Software Codesign," *HPCWire*, 2 Nov. 2010; www.hpcwire.com/hpcwire/2010-11-02/compilers_and_more_hardware_software_codesign.html?layout=print.
6. R. Rajamony, L.B. Arimilli, and K. Gildea, "PERCS: The IBM POWER7-IH High-Performance Computing System," *IBM J. Research and Development*, vol. 55, no. 3, 2011, pp. 3:1-3:12.
7. T. Narumi et al., "A 55 TFlops Simulation of Amyloid-Forming Peptides from Yeast Prion Sup35 with the Special-Purpose Computer System MDGrape-3," *Proc. Conf. High-Performance Computing, Networking, Storage and Analysis (SC 06)*, IEEE, 2006; doi:10.1145/1188455.1188506.
8. D.E. Shaw et al., "Millisecond-Scale Molecular Dynamics Simulations on Anton," *Proc. Conf. High-Performance Computing, Networking, Storage and Analysis (SC 09)*, ACM, 2009; pp. 39:1-39:11.
9. K.J. Barker et al., "Entering the Petaflop Era: The Architecture and Performance of Roadrunner," *Proc. Conf. High-Performance Computing, Networking, Storage and Analysis (SC 08)*, IEEE, 2008, pp. 1:1-1:11.
10. D. Maliniak, "Hardware/Software Co-Design Comes of Age," *Electronic Design*, 10 July 2008; <http://electronicdesign.com/print/embedded/hardware-software-codesign-comes-of-age19301.aspx>.
11. K. Yelick, "Software and Algorithms for Exascale: Ten Ways to Waste an Exascale Computer," invited talk, Oil and Gas High Performance Computing Workshop, Rice University, Houston, Texas, Mar. 2011; www.og-hpc.org/Rice2011/Workshop-Presentations/OG-HPC%20PDF-4-WEB/Yelick%20Exascale-SW-10ways-OilGas.pdf.

Vladimir Getov is a professor of distributed and high-performance computing at the University of Westminster, London. His research interests include parallel architectures and performance, autonomous distributed computing, and high-performance programming environments. He is a member of IEEE and ACM, a Fellow of the BCS, and Computer's area editor for high-performance computing. Contact him at v.s.getov@westminster.ac.uk.

Adolfy Hoisie is a Laboratory Fellow and director of the Center for Advanced Architectures at the Pacific Northwest National Laboratory. His research focuses on performance analysis and modeling of systems and applications, areas in which he has published extensively. Hoisie is a member of IEEE. Contact him at adolffy.hoisie@pnnl.gov.

Harvey J. Wasserman is a member of the User Services Group at the National Energy Research Scientific Computing Center, the primary computing center for the US Department of Energy's Office of Science, located at Lawrence Berkeley National Laboratory. Wasserman's research focuses on workload characterization, benchmarking, and system evaluation. Contact him at hjwasserman@lbl.gov.

For videos related to this topic, see the following:

- "ASCR Discovery—Codesigning for Exascale," www.youtube.com/watch?v=gXvh7WezxEG
- "Reconfigurable Exascale Computing," www.youtube.com/watch?v=zTr4bepr5Xc