Manifesto from Dagstuhl Perspectives Workshop 12212

Co-Design of Systems and Applications for Exascale

Edited by

Arndt Bode[1], Adolfy Hoisie[2], Dieter Kranzlmüller[3], Wolfgang E. Nagel[4], and Christian Straube[5]

1 **TU München and LRZ Garching, DE,** `bode@in.tum.de`
2 **Pacific Northwest National Laboratory, US,** `Adolfy.Hoisie@pnnl.gov`
3 **LMU München and LRZ Garching, DE,** `kranzlmueller@ifi.lmu.de`
4 **TU Dresden, DE,** `wolfgang.nagel@tu-dresden.de`
5 **LMU München, DE,** `straube@nm.ifi.lmu.de`

Abstract

The Dagstuhl Perspectives Workshop 12212 on "Co-Design of Systems and Applications for Exascale" is reaching into the future, where exascale systems with their capabilities provide new possibilities and challenges. The goal of the workshop has been to identify concrete barriers and obstacles, and to discuss ideas on how to overcome them. It is a common agreement that co-design across all layers, algorithms, applications, programming models, run-time systems, architectures, and infrastructures, will be required. The discussion between the experts identified a series of requirements on exascale co-design efforts, as well as concrete recommendations and open questions for future research.

Perspectives Workshop May 20–25, 2012 – http://www.dagstuhl.de/12212
1998 ACM Subject Classification B. Hardware, D.2 Software Engineering
Keywords and phrases Exascale, Co-Deisgn, Scalability, Power Efficiency, Reliability
Digital Object Identifier 10.4230/DagMan.4.1.28

Executive Summary

The term "exascale" itself describes not only the performance of future systems in terms of "ExaFLOPS" (10^{18} Floating Point Operations Per Second), but instead covers all areas of High Performance Computing (HPC), starting from the fact that future systems will continue to scale up on all characteristics from todays leading supercomputers. Consequently, the endeveaour towards exascale computing faces challenges not only in fundamental, methodical aspects, but also practical aspects when operating these machines.

Correspondingly, the development and operation of exascale systems faces a series of different problems, starting from the concurrency and complexity of future systems, through reliability and corresponding resilience, towards power consumption and total cost of ownership. It is expected that future exascale systems will consist of hundreds of millions of processor cores and billions of parallel executable threads, which should work together as efficiently as possible for a wide range of scientific applications.

The joint vision of the experts participating in the workshop has therefore been described as follows:

> "To provide exascale capabilities to scientific and engineering applications."

The role of the experts and thus their mission is to "co-design systems, such that they reach exascale capabilities within the given technological and non-technical boundaries". The

Co-Design of Systems and Applications for Exascale, *Dagstuhl Manifestos*, Vol. 4, Issue 1, pp. 28–39
Editors: Arndt Bode, Adolfy Hoisie, Dieter Kranzlmüller, Wolfgang E. Nagel, and Christian Straube
Dagstuhl Manifestos
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

activity "co-design" itself can be defined as „two or more distinct activities collaborating on and across different layers to design a system architecture for a specific goal".

On the way to exascale systems, co-design across all layers has been idenfitied as the most important approach. It requires collaboration of experts on exascale systems design and operation, and across all layers, from algorithms, applications, programming models, run-time systems, architectures and infrastructures. More specifically, the most important requirements for exascale co-design have been identified as follows:

Requirement 1: Co-design requires collaboration between tool developers, software developers and users to allow post-mortem analysis for tuning and online introspection.

Requirement 2: Co-design requires collaboration between the HPC computer architecture community and the realiability/resilience community to coordinate all levels of resiliency throughout the entire exascale stack.

Requirement 3: Co-design requires the joint development of computer center infrastructures, computer systems architecture, software, and application development.

Today, there are already a number of initiatives targeting each of these requirements, mostly in isolation. However, it might be necessary to address all or a large subset of the requirements together to enable exascale computing.

Table of Contents

1 Introduction

Ever since the beginning of computing, performance has been an important issue. Faster and faster computing has been the goal. Today, petascale computing is a reality, and the focus has shifted to the next barrier: exascale computing. With future systems even more powerful by orders of magnitude, new possibilities and challenges are approaching. This Dagstuhl workshop has addressed the many scientific, technological, and financial challenges of exascale-level computing with the hypothesis that exascale computing is only possible by co-designing across different levels of software, hardware, and the surrounding infrastructure.

The vision as derived during the workshop and based on the requirements of the scientific community is thus "to provide exascale capabilities to scientific and engineering applications", where it is important to notice that exascale means extreme scale or large scale, not the particular barrier of exaFLOPS performance looming ahead.

With this vision at hand, the participating experts identified their particular role and mission as follows: "to co-design systems such that they reach exascale capabilities within the given technological and non-technical (social,. . .) boundaries". While each expert has been knowledgeable on a distinct layer of the exascale architecture, the mission requires expertise combined across all layers (algorithms, applications, programming models, run-time systems, architectures and infrastructures). Exascale computing requires involvement from a number of different and even distinct areas of computer science in order to perform exascale co-design of hard- and software, including also different levels of software working closely together with hardware and the interfacing to the underlying infrastructure. This has lead to the definition of co-design, where two or more distinct activities collaborate on and across different layers of an exascale architecture to design a system for a specific goal.

In summary, the workshop has reflected on the current state of petascale machines providing multiple examples from world-leading machines and using them to derive the barriers on the road towards exascale computing. Looking beyond the current research into the future, where exascale computing will become feasible, the roadmap to exascale needs to be identified with intermediate goals and pitfalls, and leveraging the combined forces of computer science to overcome them.

2 Challenges and Requirements

The main challenges of exascale computing can be grouped into development, operation, application, and usage. An additional important challenge is the organization and training of user skills, which is of course included as a result of the above mentioned tasks. The requirements to effectively and efficiently use exascale systems in the area of computer architecture, systems software and application and algorithm design call for solutions potentially well beyond the observable scientific evolution.

At first glance, the lion's share of the discussed challenges seems to be not new, but instead well known for several years. In fact, some of these challenges are already solved for petascale systems. Nevertheless, there is a very special and new aspect of exascale systems that demands new solutions: all areas pass a certain limit of scale, whereby a new dimension of existing problems or entirely new problems are introduced.

Remark: It is important to note that "exascale" must not be confused with "exaFLOPS", because the latter focuses only on a very narrow aspect of HPC, i. e. number of operations per second, whereby "exascale" covers all areas of HPC. Hence, not only challenges while increasing floating point operations per second (FLOPS) should be analysed, but the challenges of all HPC areas, including for instance storage or interconnect. As a consequence, the challenges to consider range from fundamental, methodical aspects (like programming models) to very practical problems (like the number and size of log-files being produced on such machines).

2.1 Critical System Level Parameters

As mentioned in the International Exascale Software Project roadmap (IESP) [1] regarding the technology trends, the following three critical system level parameters have been defined, that relate directly to the development of exascale system architectures and their operation:

- Concurrency and complexity of the systems
- Reliability / resilience of the systems
- Power consumption and costs

Each of these aspects is described in more detail below.

2.1.1 Concurrency and complexity of the systems

Analyzing the development of the number of processors and functional units can be done with the systems listed on the TOP500 [2] supercomputers list. Already in 2012/2013, first systems with more than a million processor cores and in the order 10 millions of functional units have been brought into existence. By the time of the first implementation of an exascale computer architecture, systems with hundreds of millions of processor cores and billions of parallel executable threads must be expected.

These numbers indicate that solutions for system parameters such as reduced interconnectivity between processor cores and the development of new highly scalable algorithms and programming technologies are needed. Each of these parameters must not only be seen in isolation, but affects other parameters as well. For example, reduced interconnectivity of the architecture will require applications that are latency tolerant for communication and synchronization.

The expected complexity, heterogeneity, and failure behavior of exascale computer systems will introduce working conditions for the entire software stack, which will have to react appropriately to the behavior of the system's hardware. For each individual element of the exascale software stack as well as for the entire cooperation model of the stack, it will be necessary to develop new powerful tools to help software developers and the application users in mastering these complex systems.

Requirement 1. Co-design is required between tool developers, software developers, and users to allow post mortem analysis for tuning (required to obtain a maximum of performance) as well as online tools for introspection and direct links to adaptive runtime systems to support the decision making processes.

2.1.2 Reliability / Resilience of the systems

System dependability comprises several aspects, but within the workshop, we focused mainly on reliability, resilience, and logging. These three aspects face new challenges on exascale systems as described below.

Simplified, a component's reliability can be quantified by the "Mean Time Between Failures" (MTBF) and the total reliability of a system is evaluated as a combination of individual component values. Consequently, the aforementioned significant growth in component count results in correspondingly higher error rates. Due to the adding of system components (such as more cores, more memory, and more disks), the probability of failure increases proportionally to the number of components. This aspect is critical for exascale systems, since they comprise significant more components than todays petascale systems and hence, probability of failure is much higher. With current technologies, MTBF would be decreased down from hours to minutes or even less on exascale systems. Extrapolating system complexity from petascale systems predicts that the probability of failures will grow with at least three orders of magnitude.

Besides the new challenge of extreme scale in exascale systems, system's reliability is additionally faced with problems induced by power constraints. Smaller circuit sizes and lower voltages increase soft error vulnerability, for instance bit flips caused by thermal and voltage variations as well as radiation. Additionally, power management cycling as offered by a number of microprocessors already today decreases component lifetimes due to increased thermal and mechanical stresses. Summarizing, problems appear more often in terms of magnitude due to scale and power constraints. Therefore, reliability and resilience of exascale systems will not be realizable by common approaches. In fact, a number of additional problems is emerging.

Redundancy or checkpointing as applied on petascale systems is also not feasible due to scale, complexity and power constraints: redundancy means more running components (increases power consumption) and overhead (increases runtime and thereby again power consumption). Conducting root cause analysis by checking log files is nearly infeasible due to file sizes. The absence of strategies for silent data and code corruption will cause applications to produce erroneous results, hangups, or crashes.

Exascale systems and software will be of a complexity in hard- and software that has never been realized and tested before. Consequently, applications must be inherent fault tolerant, and this requires not only solutions for applications and algorithms, but also for the basic system software.

Future research and development in exascale systems resiliency requires that the results obtained for general purpose computer architectures regarding reliability and resiliency must be transferred and extended.

Requirement 2. Co-design between the HPC computer architecture community and the reliability/resilience communities is required to analyse how to coordinate all levels of resiliency throughout the entire exascale hard- and software stack.

2.1.3 Power consumption and costs

Since electric power consumption became a hot-topic within the last years, several approaches and improvements in order to lower energy consumption were developed. Examples include adaptive clock speed, low voltage technologies [3], and new cooling concepts such as hot-water colling in SuperMUC [4].

Nevertheless, with the cost of power increasing (often dramatically) in most parts of the world, the power budget for any high performance computing system will be limited. For this reason, DARPA suggested a maximum value of 25 MWatt for top-level supercomputers. Based on the electricity cost in most of the western countries, this figure would already represent costs well over 100 million US$ to cover the running period of 4–5 years. Yet, as shown in several analyses, scaling today's architectures to exascale will lead to power consumptions in dimensions ranging from hundreds of megawatts to over a gigawatt [5, 6]. This includes the electricity required for cooling and operating infrastructure.

As a consequence, the limitation for power is one of the most important barriers for the development of future supercomputers. In fact, there is an urgent need to decrease power consumption of today's petascale systems in order to gain more expertise and knowledge for lower energy consumption. Advances in the following areas are expected to contribute to the reduction of power consumption:

- Energy efficient computer architecture
- Energy efficient data center implementation
- Systems software supporting application programs in minimizing energy to solution
- Energy aware efficient algorithms making best use of hardware resources and specific power features

Despite the fact that there are selective and effective optimizations in each of the aforementioned areas, overall energy savings are still far behind the required levels. A reason for this stems from the lack of synchronisation of improvement and optimization efforts. For instance, scheduling algorithms cannot be optimal if applications do not provide all necessary information to the scheduler.

An important example emphasizing the urgent need to combine the above mentioned areas of research and development is the design of the computer center infrastructure itself, which must be taken into account similar as the other hard- and software factors [4].

Requirement 3. Co-design must be extended to the joint development of computer center infrastructures, computer systems architecture, software, and application development.

2.2 Applications of Co-Design

During co-design, two or more factors are optimized in concert to achieve better solutions. For exascale systems and applications, the performance and the requirements above define a multidimensional space of optimization. A number of important factors in co-design contributing to the optimization of performance, power and reliability are:

Algorithms: Multiple algorithms or mathematical techniques can be used for a calculation, and may exhibit different computational characteristics. For instance, using a uniform resolution of a data grid may lead to an implementation with regular memory and corresponding communication characteristics but with computation exceeding that of a more complex adaptive refinement implementation.

Applications: The application represents the implementation of a particular method and comprises a component of the overall workload of interest. Multiple applications may be used in concert to explore multiple aspects of a physical system, such as climate simulations considering land, sea, and atmospheric components in conjunction.

Programming Models: The programming model underlies the application and defines the way in which computation is expressed. Two common approaches are used for expressing parallelism: process-centric such as the Message Passing Interface (MPI) in which the inter-process communication is explicitly expressed, and data-centric in which access to any data across the system may occur from any location e. g. Global Arrays, Unified Parallel C (UPC), and Co-Array Fortran (CAF).

Runtime system: The runtime is responsible for ensuring that application requirements are dynamically satisfied and mapped onto the system resources. This includes process and data management and migration.

Architecture: This includes the micro-architecture of a processor-core, arrangement of cores within a chip, memory hierarchy, system interconnect, and storage subsystem. Advances in technology are continually allowing innovations.

Until today, no co-design process has covered all factors in a comprehensive fashion. However some notable cases have addressed a subset of factors and the corresponding tradeoffs. Several existing co-design experiences have been presented at the workshop, which resulted in improved performance, power efficiency, and reliability. An overview of observations is included below:

Optimization of "application to architecture": For a system architecture already implemented, this process requires mapping of application workload onto architecture characteristics. This process is commonplace in application development and software engineering and is not considered co-design.

Optimization of "architecture to application": Given applications that have already been implemented, the process here is to steer the design of the architecture so as to achieve high performance. Given that only one factor is optimized this is also not considered co-design.

Co-design for performance: Enabling application and architecture to best match each other unlocks the potential to achieve the highest performance on a new system. (This process has been illustrated at the workshop in a presentation describing the design of an application and of the first peta-flop system – the IBM Roadrunner machine.)

Co-design for energy efficiency: The energy consumed by extreme-scale systems will increasingly become a design constraint and notable cost factor – see above. Co-design for energy means designing an application to provide information on expected periods of idleness, and defining the runtime to lower overall power consumption.

Co-design for fault-tolerance: A critical factor in the operation of extreme-scale systems is the detection and handling of faults. Traditional methods using checkpoint-restart mechanisms do not scale well with future system sizes. Selective checkpointing methods, such as replicating only critical data across system memory, can be used to reconstruct state from failed nodes and enable job execution to continue.

Co-design is driven by modeling (or modeling is the tool of co-design).

The complexity of the aspects described above leads to the necessity of optimization to multiple criteria. This requires sophisticated modeling and simulation (modsim) methodologies and tools.[1]

3 Recommendations and Open Questions

The above discussed challenges and roadblocks on the way to enabling exascale systems are addressed with three necessary activities as identified during the Dagstuhl workshop:

- Application of co-design across all layers
- Simplification of HPC system usage
- Education of HPC users

---

[1] A significant area of emphasis in co-design continues to be related to modsim activities. Some of the important future directions along these lines have been mapped by the modsim community and available in the Report on the ASCR Workshop on Modeling and Simulation of Exascale Systems and Applications.

All three activities focus on strategic aspects and synergies of different fields. The Dagstuhl workshop participants agree that the three activities should be employed conjointly to facilitate development, operations, and usage of exascale systems. Guided by a common pattern, the following subsections describe the three activities: For each of them, the general findings are discussed in terms of concepts, thoughts, and involved technologies. Afterwards, a list of the most interesting and promising research questions is provided, which cover not only technical, but also economical and organizational aspects.

3.1 Application of Co-Design

Applying the co-design methodology was considered as the most important approach and as a vital element on the way to exascale systems. Co-design vehicles are required because there are several possible paths to exascale system with many associated design choices along the way. The main objective is to bring different aspects together, and to develop a common solution, where aspects can (positively) influence each other. Because of its generality, the definition of co-design can be used within different topics as well as spanning several topics, e. g., co-design of software and tools, or co-design of hardware and software. Additionally, it is not restrained to realms, but it can cope with different timelines, teams, and organizational aspects.

In fact, the definition of co-design raises a double-edged sword: On the one hand, all aspects, topics, and realms can be covered. On the other hand, the severe question of trade-offs between generality, performance, and costs arises. Hence, the first open research question is:

> *On which aspects, topics, and realms should co-design be applied to optimize the trade-off between generality, performance, and costs?*

The following, non-exhaustive list outlines a few possible combinations for the application of the co-design methodology:

- Performance, power, reliability – co-design of capabilities
- Applications, hardware, software – co-design of disciplines
- Algorithms, software, hardware, applications – co-design of disciplines and domains
- Applications, software-stack/execution-stack – co-design of domains and paradigms
- Teams – co-design of people

After having selected the elements for the application of co-design, the next group of questions investigates how to bring the selected elements together, especially in terms of synchronisation and interfaces. An exemplary question in the realm of co-design of disciplines would be as follows:

> *How can the short life cycle of hardware be synchronized with the relatively long life cycle of software?*

Again, a lot of possible follow-up questions are generated on the realm of co-design of paradigms, in particular of the software stack and performance monitoring. A non-exhaustive list of possible questions is as follows:

- What is the right kind of interface for performance introspection?
- What kind of data flows between the layers?

- What kind of monitoring, first-person or third-person, is more suitable, and how can both approaches be integrated?
- How can system monitoring be included in an overall approach?
- What part of the software stack is missing for exascale?
- How can application developers be encouraged to adopt exascale and corresponding technologies?
- Which metrics other than performance are needed on exascale systems, e. g. energy consumption?
- Which abstractions and layered design approaches are convenient, keeping in mind their costs in performance?

Bringing all these questions into context leads to a group of research questions that can be summarized as follows:

> *How to co-design two different elements of an exascale system?*

The above described group of research questions introduces one very special situation, which is considered here separately because it heavily involves psychology, social, and organizational science:

> *What to do with partners, who contribute fundamental elements to the co-design process, but are not willing to comply with the policies and restrictions of the co-design process?*

Clearly, this issue jeopardizes the overall co-design benefits, and it is crucial to investigate solutions. An example for this situation, which became prominent within the last years, is the involvement of HPC hardware vendors and the issue of not influencing the hardware part of HPC systems through science.

After dealing with strategic aspects of co-design, the last group of research questions focuses on the concrete (technical) operations of co-design methodology, e. g., "What kind of tools are missing and how to make them work?". This group of research questions can be summarized as follows:

> *How to support the application of the co-design methodology operationally?*

All these research questions will need to be addressed in order to enable exascale computing. The Dagstuhl workshop initiated many discussion and established a basis for further investigations on the way to exascale.

4 Participants

- Arndt Bode
TU München and
LRZ Garching, DE

- Hans-Joachim Bungartz
TU München, DE

- Kent Czechowski
Georgia Institute of Technology –
Atlanta, US

- Karl Fürlinger
LMU München, DE

- Hans Michael Gerndt
TU München, DE

- Vladimir S. Getov
University of Westminster –
London, GB

- Adolfy Hoisie
Pacific Northwest National
Laboratory – Richland, US

- Jeffrey K. Hollingsworth
University of Maryland – College
Park, US

- Darren Kerbyson
Pacific Northwest National
Laboratory – Richland, US

- Alice E. Koniges
Lawrence Berkeley National
Laboratory – Berkeley, US

- Bettina Krammer
Université de Versailles
St-Quentin-en-Yvelines, FR
- Dieter Kranzlmüller
LMU München and LRZ
Garching, DE
- Erwin Laure
KTH Royal Institute of
Technology – Stockholm, SE
- Allen Malony
Univ. of Oregon – Eugene, US
- John Mellor-Crummey
Rice University – Houston, US
- Barton P. Miller
University of Wisconsin –
Madison, US
- Bernd Mohr
Jülich Supercomputing
Centre, DE
- Daniel Molka
TU Dresden, DE
- Wolfgang E. Nagel
TU Dresden, DE
- Cherri M. Pancake
Oregon State University –
Corvallis, US
- Michael M. Resch
Universität Stuttgart, DE
- Sabine Roller
German Resarch School for
Simulation Sciences –
Aachen, DE
- Martin Schulz
Lawrence Livermore National
Laboratory – Livermore, US
- Stephen L. Scott
Oak Ridge National Laboratory –
Oak Ridge, US
- Christian Straube
LMU München, DE
- Achim Streit
Karlsruher Institut für
Technologie / Steinbuch Center
for Computing – Karlsruhe, DE
- Shinji Sumimoto
Fujitsu – Kawasaki, JP
- Jan Treibig
Univ. Erlangen-Nürnberg, DE
- Felix Wolf
German Resarch School for
Simulation Sciences –
Aachen, DE
- Zhiwei Xu
Institute of Computing
Technology of the Chinese
Academy of Sciences –
Beijing, CN

References

1 Jack Dongarra, Pete Beckman, Terry Moore, Patrick Aerts, Giovanni Aloisio, Jean-Claude Andre, David Barkai, Jean-Yves Berthou, Taisuke Boku, Bertrand Braunschweig, Franck Cappello, Barbara Chapman, Xuebin Chi, Alok Choudhary, Sudip Dosanjh, Thom Dunning, Sandro Fiore, Al Geist, Bill Gropp, Robert Harrison, Mark Hereld, Michael Heroux, Adolfy Hoisie, Koh Hotta, Zhong Jin, Yutaka Ishikawa, Fred Johnson, Sanjay Kale, Richard Kenway, David Keyes, Bill Kramer, Jesus Labarta, Alain Lichnewsky, Thomas Lippert, Bob Lucas, Barney Maccabe, Satoshi Matsuoka, Paul Messina, Peter Michielse, Bernd Mohr, Matthias S. Mueller, Wolfgang E. Nagel, Hiroshi Nakashima, Michael E Papka, Dan Reed, Mitsuhisa Sato, Ed Seidel, John Shalf, David Skinner, Marc Snir, Thomas Sterling, Rick Stevens, Fred Streitz, Bob Sugar, Shinji Sumimoto, William Tang, John Taylor, Rajeev Thakur, Anne Trefethen, Mateo Valero, Aad van der Steen, Jeffrey Vetter, Peg Williams, Robert Wisniewski, and Kathy Yelick. *The International Exascale Software Roadmap*. International Journal of High Performance Computer Applications, 25(1):3–60, 2011. DOI: 10.1177/1094342010391989

2 http://www.top500.org

3 Ram K. Krishnamurthy and Himanshu Kaul. *Ultra-Low Voltage Technologies for Energy-Efficient Special-Purpose Hardware Accelerators*. Intel Technology Journal, 13(4):100–117, 2009.

4 Ludger Palm. *LRZ awarded German Data Centre Prize*. Inside – Innovatives Supercomputing in Deutschland, 10(1), 2012. http://inside.hlrs.de/_old/htm/Edition_01_12/article_04.html

5 David Jensen and Arun Rodrigues. *Embedded Systems and Exascale Computing*. Computing in Science Engineering, 12(6):20–29, 2010. DOI: 10.1109/MCSE.2010.95

6 Daniel A. Hitchcock and Lucy Nowell. *Advanced Architectures and Critical Technologies for Exascale Computing*, U.S. Department of Energy – Office of Science – Office of Advanced Scientific Computing Research, Program Announcement to DOE National Laboratories LAB 10-255, 2010. http://science.energy.gov/~/media/grants/pdf/lab-announcements/2010/LAB_10-255.pdf

7 Darren J. Kerbyson, Abhinav Vishnu, Kevin J. Barker and Adolfy Hoisie. *Codesign Challenges for Exascale Systems: Performance, Power, and Reliability*. Computer, 44(11):37–43, 2011. DOI: 10.1109/MC.2011.298

8 Adolfy Hoisie, Darren Kerbyson, Robert Lucas, Arun Rodrigues, John Shalf, Jefffrey Vetter, William Harrod, and Sonia R. Sachs. *Report on the ASCR Workshop on Modeling and Simulation of Exascale Systems and Applications*. U.S. Department of Energy – Office of Science, 2012. http://science.energy.gov/~/media/ascr/pdf/program-documents/docs/ModSim_Report-2012_AH_5.pdf