Conclusion

Emmanuel Jeannot
Inria, LaBRI, Univ. Bordeaux

Jesus Carretero
UC3M - Universidad Carlos III de Madrid

December 2019

As we have seen in the former chapters, facilitating the adoption and usage of sustainable ultrascale computing systems, will need providing innovative solutions to advance the knowledge of designing sustainable ultrascale software and systems, which will be the basic facilities for new discoveries in science and technology and will have a direct impact on economic growth, society, and environmental aspects.

The main conclusion is that it is important to **enable Ultrascale computing** by supporting the evolution of Ultrascale systems towards on-demand computing across highly diverse environments by providing domain-specific, but interoperable tools to enable high productivity of human – computer interaction, leading towards robust solutions through multi-domain cooperative approaches using energy efficient hardware – software co-design principles. . However, there are still important points to be addressed, as pointed out in this book.

One of the key point will be the availability of programming abstractions for the different fields of Exascale such as data analysis, machine learning, scientific computing, Big Data management, smart cities, that will be based on asynchronous algorithms for overlapping communication and computation. To reach this overlap, parallel applications (such as the MPI-based one) will need to be optimized using platform topology and performance information. One crucial research topic will be programmability of UCS as applications will run millions of parallel execution flows. New workflow programming for very large plate forms will be needed. But interoperability and sustainability will only be reached when code will be prevented to be platform specific and still efficient on different platforms. From a broader point of view, the scale of UCS will lead to Supercomputing on demand leading to a better use of the vast amount of available resources. The efficiency will be linked to researches on performance evaluation, modeling and optimization of data parallel applications on heterogeneous HPC platforms. Management of such large distributed systems will be based on future researches on complex systems modeling, self-organizing systems and cellular automata.

In addition, the characterization of hardware and software faults is essential for making informed choice about research needs for the resilience of Ultrascale systems and for developing a standardized fault-handling model to provide guidance to application and system software developers about how they will be notified about a fault, what types of faults they may be notified about, and what mechanisms the system provides to assist recovery from the fault. This should pave the way towards improved fault prediction, containment, detection, notification, and recovery, which is needed to cope with scale.

This book has also studied the close relationship between HPC and data analysis in the scientific computing area and possible procedures to achieve the desired unification, including the appearance of new storage device technologies that carry a lot of potential for addressing issues in these areas, but also introduce numerous challenges and will imply changes on the way data is organized.

Models and simulation of energy consumption have been proposed for ultrascale systems, including metrics, models of energy consumption of heterogeneous hardware, and energy simulators. They have been complemented with the valuation of renewable energy usage for an Exascale computer, which can help to significantly reduce energy costs and carbon footprint of ultrascale systems, a review of cooling techniques in data centers, and full-cost model for estimating the energy consumption of computing infrastructures

Finally, the book has addressed the need of reformulation of algorithms and applications from different areas of research towards their usage for Ultrascale systems, creating scalable parallel algorithms for several application areas, such as numerical solution of problems and to cope with the computational complexity of super-diffusion problems and 3D relaxation gravity, or massive parallelization of the maximum clique problem. Moreover, how to provide application specific analytical energy models has been presented

As a result os the work made in the book chapters, and the research associated in the NESUS COST Action, it is clear that there are still challenges to be solved to arrive to Ultrascale Computing Systems. Achieving those challenges would require to:

Improve the programmability of complex systems. New programming paradigms are needed to help the programmer. These paradigms will solve the impossibility to have a global view of the whole system as the complexity of software workflow and hardware explodes and reach millions of heterogeneous entities.

Break the wall between runtime and programming frameworks. . There is a need to adapt generic high level code to the underlying infrastructure by giving feedback to the programmers during development. This feedback will help programmer to have insight on the performance and capabilities of the targeted platform and to make informed decisions.

Enabling behavioral sensitive runtime. The ability to provide behavioral information along-with applications will help runtime to take the most relevant decisions in function of its context such as other applications or characteristics of the execution platform. Runtime will be informed and will be able to allocate the right amount of resources at the right time but also will be able to reconfigure the application in the most relevant way.

Developing new programming abstractions for resilience and standardized evaluation of fault- Efficient tools and methods for characterization of both hardware and software faults are needed. Comprehensive and standardized fault-handling models for analysis of resilience of systems to improve fault prediction, containment, detection, notification, recovery mechanisms and strategy for ultrascale systems is crucial in their operation.

To enforce the convergence of HPC, Ultrascale and Big Data worlds. Storage, interconnection networks and data management in both HPC and Cloud needs to cope with technology trends and evolving application requirements while hiding the increasing complexity at the architectural, systems software, and application levels. Future work needs to examine these challenges under the prism of both HPC and Cloud approaches and to consider solutions that break away from current boundaries.

To design and develop intelligent data access mechanisms. Future applications will need more sophisticated interfaces for addressing the challenges of future Ultrascale computing systems. These novel interfaces should be able to abstract architectural and operational issues from requirements for both storage and data. This will allow applications and services to easier manipulate storage and data, while providing the system with flexibility to optimize operation over a complex set of architectural and technological constraints.

Adoption of intelligent methods for modeling and improving energy efficiency. We envision the wide use of machine learning techniques not only for understanding, but also for managing Ultrascale systems. A methodology for modeling the whole system based on its subset must be created to allow extrapolating the overall energy efficiency. A multi-layered approach allows feeding the models and management software with fine-grained measurements for selected part of the system when needed without deterioration of the whole system performance.

Increasing awareness and focus on energy efficiency. To achieve significant impact on the energy efficiency of large systems in real life, appropriate incentives must be provided for all stakeholders including users, developers, and providers. Relevant metrics, going beyond Flops/W, focusing on ultrascale systems energy must be proposed and widely adopted. We recommend to put efforts into innovative usage and business models to provide incentives for energy-efficient use of resources, e.g. by methods to increase awareness, appropriate metrics, pricing models, energy-related SLAs, etc. These efforts must also include means (e.g. interfaces, APIs) to allow effective exchange of energy-related data and incentives within large collections of heterogeneous services that will be common application of Ultrascale systems.

Designing software taking the advantage of heterogeneous hardware and infrastructure. Without careful integration of new hardware and infrastructure solutions, including optimization of software, significant reduction of energy consumption will not be possible. Therefore, energy-aware software development techniques must be developed (including autotuning, co-desing, etc.). New methods of resource management for heterogeneous systems are needed in order to find the best hardware configuration for specific applications. Finally, we propose to put more efforts into achieving energy savings from synergy of IT and infrastructure, including integration of IT management with cooling and heat re-use systems (and environmental data), the use of renewable energy sources, energy markets (e.g. applying demand response programs for IT) and other external systems.

Enabling complex Ultrascale computing applications. Develop complex applications based on complementary utilization of numerical and non-numerical, deterministic, stochastic and hybrid, multiscale and multi-physics, direct and iterative methods and algorithms. Support sustainable storage of Big Data and Big Data analytics including real-time multi-stream processing, processing of insecure, uncertain, incomplete and unreliable data. Integrate software tools providing fault-tolerance and resilience, self-correcting, automatic adaptation and generation of codes for heterogeneous architectures including accelerators.

Towards total efficiency of Ultrascale computing applications. Develop novel architecture-aware methods and algorithms that expose as much parallelism as possible, exploit heterogeneity, avoid communication bottlenecks, respond to escalating fault rates, and help meet emerging power constraints. Use domain-specific languages with specialized compilers to generate efficient codes for different Ultrascale computing architectures enabling self-adaptivity, deep machine learning, and complex socio-technical environments and systems. Integrate the complex chain of modeling, simulation, optimization, Big Data analytics, and decision making. Develop integral measures of global efficiency including the scalability issues related to total solution of the problems.