A Roadmap for Research in Sustainable Ultrascale Systems

NESUS Roadmap Version 1.0

Editors:
Leonel Sousa, Peter Kropf, Pierre Kuonen, Radu Prodan, Tuan Anh Trinh, Jesus Carretero

June 1, 2017

NESUS
Network for Sustainable Ultrascale Computing
cost IC1305

EXECUTIVE SUMMARY

Europe has made impressive progress in different High Performance Computing (HPC) research areas that are crucial for the goals of the H2020 framework and beyond. This includes also the organization building of the European HPC community in general and network of sustainable ultrascale computing systems in particular. The ultimate goal is to pursue European HPC leadership on a unified basis, expanding the scientific and industrial access to and use of supercomputers, and launching initiatives to strengthen the European HPC supply chain toward a sustainable ultrascale computing systems. As a result, to achieve the designated goals, there is an urgent need for the European HPC research community to have a clear roadmap for the next stage: short-term, mid-term, and long-term perspectives.

The objective of this research roadmap is to provide the readers with a set of research recommendations identified by COST Action IC1305, Network of Sustainable Ultrascale Computing Systems (NESUS), as key for achieving the ambitious goals set out in the Action.

As such, this research roadmap document aims at providing Sustainable Ultrascale Computing Systems input to the relevant stakeholders (H2020 Work Programme, ESF, COST programme leaders, industry, ...) for developing Sustainable Ultrascale Computing Systems. The content presented in this research roadmap is the result of wide discussions within NESUS. It aggregates the input provided by NESUS Members, obtained through working groups meetings, online consultation, and discussion by the Management Committee of the Action. Relevant challenges and opportunities for Sustainable Ultrascale System are identified in this document, as large-scale complex systems integrating heterogeneous parallel and distributed computing systems will be two to three orders of magnitude larger that today systems.

The COST Action IC1305 (NESUS) proposes in this research roadmap research objectives and twelve associated recommendations, which in combination, can help bring about the notable changes required to make true the existence of sustainable ultrascale computing systems. Moreover, they are useful for industry and stakeholders to define a path towards ultrascale systems.

A possible short and midterm research plan, that could be implemented in the EU to achieve those objectives and recommendations, is also proposed to coordinate European efforts for developing realistic solutions addressing major challenges of building sustainable ultrascale computing systems. This research plan should be centered on the development of software and applications for Ultrascale Computing Systems, which should be a priority given that Europe (industry, academia and other public entities) is a strong partner in software for HPC and distributed systems, while it is not a leader hardware for large-scale software systems.

RECOMMENDATIONS

1. Enabling Ultrascale computing.	Support the evolution of Ultrascale systems towards ondemand 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 codesign principles.
2. 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.
3. 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.
4. 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.

5. Developing new programming abstractions for resilience and standardized evaluation of fault-tolerant approaches.	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.
6. 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 in-creasing 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.
7. 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.
8. 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.

9. 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.

10. 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 programmes for IT) and other external systems.

11. Enabling complex Ultrascale computing applications.	Develop complex applications based on complementary utilization of numerical and non-numerical, deterministic, stochastic and hybrid, multiscale and multiphysics, 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.
12. 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.

Contents

Introduction

1.

1. Introduction

Ultrascale systems are envisioned as large-scale complex systems joining parallel and distributed computing systems, maybe located at multiple sites that cooperate to provide solutions to the users, that will be two to three orders of magnitude larger that today s systems.

The COST Action IC1305 (NESUS) aims at collaboratively rethinking the current basis of development of system software for scalable computing systems in order to pave the way towards a sustainable future scale growth by improving the coordination of efforts between complementary communities. Todays scientific community envisions and supports the emergence of Exascale systems within the next decade. In parallel, many companies and researchers are engaged in efforts of scaling data centers and system software to meet the requirements of diversifying on-line cloud applications and services. Both communities are already facing big data challenges and are developing their particular solutions to similar problems. While there is an emerging cross-domain interaction (for instance the need for high-performance in clouds or the adoption of distributed programming paradigms such as Map-Reduce in scientific applications), the cooperation between HPC and distributed system communities towards building the ultrascale systems of the future is still weak.

The main objective of the Action is to coordinate European efforts for proposing realistic solutions addressing major challenges of building sustainable ultrascale computing systems, as well as developing collaborative activities among the involved research groups to target cross-layer design issues to offer a unified view of ultrascale platforms.

This Research Roadmap is an important result of the NESUS action. Its main purpose is to identify the key research challenges facing sustainable Ultra-scale Computing Systems and a vision on how to approach and solve the problems as well as the time-frame involved. In preparation, the NESUS action has issued a report on the State-of-the-Art, Software

Techniques to Increase Sustainability in Ultrascale Systems, and Roadmap [7].

This research map includes contributions from many bibliographic sources. NESUS action has provided several published contributions. Some of them may be found in First Workshop on Techniques and Applications for Sustainable Ultrascale Computing Systems (TASUS 2014) [9], Proceedings of the Second International Workshop on Sustainable Ultrascale Computing Systems NESUS 2015 [2], and the Special issue on Sustainability in Ultrascale Computing Systems in the International Journal of Supercomputing Frontiers and Innovations [4].

COST Action IC1305 - Network for sustainable ultrascale systems

2.

2. COST Action IC1305 - Network for sustainable ultrascale systems

2.1 COST programme

COST European Cooperation in Science and Technology-is an intergovernmental framework aimed at facilitating the collaboration and networking of scientists and researchers at European level. It was established in 1971 by 19 member countries and currently includes 35 member countries across Europe, and Israel as a cooperating state.

COST funds pan-European, bottom-up networks of scientists and researchers across all science and technology fields. These networks, called "COST Actions", promote international coordination of nationally-funded research.

By fostering the networking of researchers at an international level, COST enables break-through scientific developments leading to new concepts and products, thereby contributing to strengthening Europes research and innovation capacities.

COSTs mission focuses in particular on:

» Building capacity by connecting high quality scientific communities throughout Europe and worldwide;

» Providing networking opportunities for early career investigators;

» Increasing the impact of research on policy makers, regulatory bodies and national decision makers as well as the private sector.

Through its inclusiveness, COST supports the integration of research communities, leverages national research investments and addresses issues of global relevance. Every year thousands of European scientists benefit from being involved in COST Actions, allowing the pooling of national research funding to achieve common goals.

As a precursor of advanced multidisciplinary research, COST anticipates and complements the activities of EU Framework Programmes, constituting a bridge towards the scientific communities of emerging countries. In particular, COST Actions are also open to participation by non-European scientists coming from neighbour countries (for example Albania, Algeria, Armenia, Azerbaijan, Belarus, Egypt, Georgia, Jordan, Lebanon, Libya, Moldova, Montenegro, Morocco, the Palestinian Authority, Russia, Syria, Tunisia and Ukraine) and from a number of international partner countries.

COSTs budget for networking activities has traditionally been provided by successive EU RTD Framework Programmes. COST is currently executed by the European Science Foundation (ESF) through the COST Office on a mandate by the European Commission, and the framework is governed by a Committee of Senior Officials (CSO) representing all its 35 member countries.

More information about COST is available at www.cost.eu. In particular, the COST Vademecumprovides all the administrative and financial rules with respect to the man-agement and implementation of COST Actions and associated activities. The COST Vademecum is a legally binding document approved by the ESF and follows the rules established by the CSO.

2.2 Network for sustainable ultrascale systems (NESUS)

This research agenda is result of the COST Action 1305 -Network for sustainable ultrascale systems (NESUS).

The goal of the NESUS Action is to establish an open European research network targeting sustainable solutions for ultrascale computing. Thus, NESUS Action focus on a cross-community approach of exploring system software and applications for enabling a sustainable development of future high-scale computing platforms. In details, the Action work is focused in the following scientific tasks:

» First, the current state-of-the-art on sustainability in large-scale systems has been studied. The Action strives for continuous learning by looking for synergies among HPC, distributed systems, and big data communities in cross cutting aspects like programmability, scalability, resilience, energy efficiency, and data management.

» Second, the Action explores new programming paradigms, runtimes, and middlewares to increase the productivity, scalability, and reliability of parallel and distributed programming.

» Third, as failures will be more frequent in ultrascale systems, the Action looks for new approaches of continuous running in the presence of failures, trying to find synergies between resilient schedulers that handle errors reactively or proactively, monitoring and assessment of failures, and malleable applications that can adapt their resource usage at runtime.

- Fourth, future scalable systems will require sustainable data management for addressing the predicted exponential growth of digital information. The Action explores synergistic approaches from traditionally separated communities to reform the handling of the whole data life cycle, in particular: restructure the Input/Output (I/O) stack, advance predictive and adaptive data management, and improve data locality.
- Fifth, as energy is a major limitation for the design of ultrascale infrastructures, the Action has addressed energy efficiency of ultrascale systems by investigating and promoting novel metrics for energy monitoring, profiling, and modeling in ultrascale components and applications, energy-aware resource management, and hardware/software codesign.
- Finally, the Action has identified applications, high-level algorithms, and services amenable to ultrascale systems and investigated the redesign and reprogramming efforts needed for applications to efficiently exploit ultrascale platforms, while providing sustainability.

NESUS COST action is composed by more than 75 research institutions from 35 EU countries, and 10 non-EU countries.

How are ultrascale computing systems different?

3.

3. How are ultrascale computing systems different?

With the spread of the Internet, applications and web-based services, distributed computing infrastructures, local parallel systems and the availability of huge amounts of dispersed data, software dependent systems will be more and more connected, more and more networked, leading to the creation of "supersystems". The phrase "Ultrascale Computing Systems (UCS)" refers to this type of IT "supersystems". However, to really speak of UCS we must consider several orders of magnitude increase in the size of data, in the computing power and in the network complexity relative to what is existing now. By making explicit the characteristics of UCS, we better understand at which extent these systems are different from what exists today. Some of the main characteristics of UCS are the following [16]:

- decentralization: Because of their size UCS are inherently decentralized in many aspects: development, computing power, data storage, maintenance and operations, to mention few.
- continuous evolution: New features and capabilities will be deployed, and unused capabilities will be dropped while the system will continue to be operational.
- heterogeneous and incompatible: Usually an UCS system will not be built from uniform parts and, if yes, it will evolve towards partly unsuitable and incompatible parts because of its expansion and repairs.
- normal failures: Software and hardware failures will be the norm rather than the exception.
- blurring of system boundary: parts and users of the systems can affect the system itself leading to overall emergent behavior.

Last but not least, many, if not all, of the envisioned UCS will integrate HPC infrastructures up to the the level of exascale machines. Indeed, the complexity of the addressed problems and the constant increase of data size needed to solve these complex problems, will lead to strong needs of HPC capabilities. A typical example of such UCS are Cyber Physical Systems (see COST Action IC1404 Multi-Paradigm Modeling for Cyber-Physical Systems). Because of their size and their complexity, the traditional centralized engineering approach will be no longer adequate nor can it be the primary way to develop, operate and to maintain UCS in a sustainable way. The challenges that sustainable UCS posed require a change in perspective. We need to replace the satisfaction of requirements using traditional approaches based on rational top down engineering by the orchestration of complex, decentralized systems.

Addressing this challenge as a whole is most probably an un-realistic approach and does not correspond to the characteristics of UCS. Indeed, by definition, UCS cannot be addressed as a whole. These type of systems are too large and complex to be encompassed by a single human, a single team and even a single organization. A more treatable approach is to identify the main domains in which we need to improve our knowledge and technologies in order to provide the necessary means that will make sustainable UCS being a reality. This is why the NESUS COST action has defined 6 working groups, each working on a specific domain with the objective of proposing, in their specific domain, a roadmap towards sustainable UCS.

The six working groups defined are the following :

- WG1: State of the art and continuous learning in Ultra Scale Computing Systems
- WG2: Programming models and runtimes
- WG3: Resilience of applications and runtime environments
- WG4: Sustainable data management
- WG5: Energy efficiency
- WG6: Applications

The structure and relation of the NESUS action working groups is shown in Figure 3.1. What is needed is to define a layer of Ultrascale Computing Services than can allow to create application for UCS using the underlying facilities. All the working groups have contributed to this roadmap document. Their specific contributions are found in the Sections 3 to 8 in the ordering of the workgroup numbering.

Section 3 summarizes the state of the art and initiatives related to UCS. The subsequent Sections 4 to 8 discuss important research topics in UCS from three perspectives :

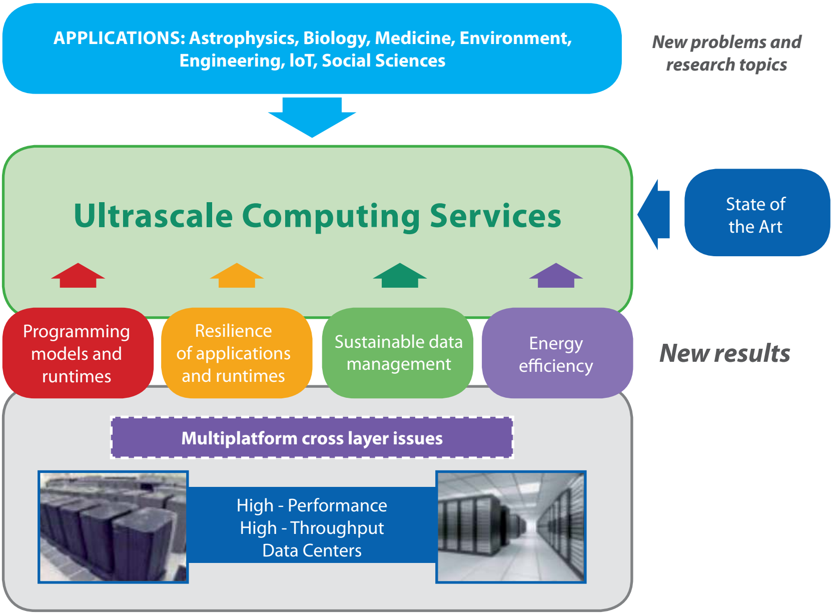
APPLICATIONS: Astrophysics, Biology, Medicine, Environment, Engineering, IoT, Social Sciences
New problems and research topics
Ultrascale Computing Services
State of the Art
Programming models and runtimes
Resilience of applications and runtimes
Sustainable data management
Energy efficiency
New results
Multiplatform cross layer issues
High - Performance
High - Throughput
Data Centers

Figure 3.1: Structure of the NESUS working groups

1. Present research topics and challenges;
2. Future research topics and challenges anticipated for the next 2 to 5 years;
3. Possible approaches to address the future research topics and challenges at longer term.

Systems, Technologies, and Use-cases: State of the Art and challenges

4.

4. Systems, Technologies, and Use-cases: State of the Art and challenges

A large number of worldwide research teams and projects investigate system software solutions for supporting scalability in Cloud and Exascale computing. Research in Cloud computing is currently addressing a large variety of challenges related to sustainable scalability such as economy-of-scale, agile elastic scalability, energy-efficiency, scalable storage, automatic management of resources, resilience, hybrid Clouds, etc. At the same time, several Exascale research projects explore novel approaches for advancing the system software towards new scalability levels, while addressing the main challenges of energy efficiency, high-performance, programmability, and resilience. International initiatives, such as the International Exascale Software Project (IESP) or the collaboration between European PRACE and US XSEDE, are mostly devoted to large scale parallel systems and scientific data systems. The advent of the Big Data challenges has generated new initiatives closely related to Ultrascale computing systems in large scale distributed systems. There are some initiatives running on Ultrascale computing systems, such as CUCIS and ULS in USA and LSCITS in UK, and some research activities on extreme-scale systems funded at international level, but they are fragmented in research communities (Exascale, Cloud computing, Big Data, energy efficient data centers) with complementary strengths.

It is worth to mention here that big countries like USA, China and Japan have significant research programs in the area of Ultrascale computing systems, addressing software challenges and providing scalable solutions. In particular, in 2015 the US President established the National Strategic Computing Initiative (NSCI) to maximize the benefits of HPC for US economic competitiveness and scientific discovery [15]. The Department of Energy (DOE) is a lead agency within NSCI to execute a joint program focused on an Exascale computing program emphasizing high performance on relevant applications. DOE has been funding researches related to Exascale challenges for more than 5 years and the Exascale Computing Project (ECP) has being recently launched. EPCs goal is not to procure Exascale hardware, but to develop software technologies and applications for Exascale systems. Also Japan has active research programs for developing Exascale computing systems. Currently Japan has HPCI, a nation-wide HPC infrastructure into which the supercomputers and large-scale storages are connected. The

Japanese Cabinet set up a comprehensive strategy on science, technology and innovation that includes Exascale systems. The Exascale strategy includes the development of hardware, software and applications. Finally, also China has significant activities in the area of HPC and Exascale systems. As well known, currently the two fastest supercomputers in the world are located in China. Chinas programs planned that, according to the national plan for the next generation of high performance computers, their first Exascale supercomputer will enter service by 2020. Also in this program, research activities in the area of Ultrascale software and applications have a primary role. Recently, Prof. Depei Qian, disclosing new information regarding HPC development and Exascale plans of China, declared that ”WG2 are in urgent need for the system software, for the domestic processor, for the tool software and also the application software. Without an ecosystem around the domestic processors, we will not succeed in this respect”.

The following subsections summarize current challenges, systems, technologies, and use-cases of research in Ultrascale Computing Systems.

4.1 Challenges

In this section we describe some of the challenges that we face on the state-of-the-art roadmap to sustainable Ultrascale systems.

Data distribution and data location. Distribution and locality is an increasingly critical problem, since the energy and performance of data movement is not expected to improve at the same pace as the floating point operations. Moreover, the new trend of Big Data represents a real challenge, since moving terabytes to petabytes of data over wide area networks is no longer a feasible solution. New solutions are required to minimize the data movement, for example by offloading code to the data that brings additional optimization and security issues. Other solutions require to dynamically optimize the data placement to handle memory limits or to reduce or filtering using real-time stream processing techniques.

Scalability, heterogeneity, resource discovery, selection and integration. Achieving high scalability in dynamic heterogeneous environments reaching Exascale performance is a real challenge requiring new optimization techniques at the runtime and resource management levels. Ultrascale systems are expected to exhibit billions of threads, as a result of decreasing clock frequencies to preserve power consumption, a scalability never encountered and understood in the past. Moreover, automatic porting of parallel codes across heterogeneous platforms (multicore CPUS, manycore accelerators) while preserving performance and computational efficiency is a challenge that remains to be solved. Moreover, complementing and merging Exascale HPC infrastructures with elastic virtualized Clouds for improving resource utilization and energy efficiency is a technology awaiting market penetration and worldwide deployment. Other new hosting paradigms like dynamic on-the-fly-created micro data centers or embedded HPC systems should be researched too.

Codesign-aware systems. Since systems are getting increasingly large due to the high degree of parallelism, it is important to preserve the same levels of parallelism within algorithmic approaches, parallel programming technologies, system software, and hardware.

Model-based systems. New modeling techniques are need to be researched allowing to specify application needs, network and computer configuration, and system software through higher-level abstractions by providing system-independent specification that would be later mapped to a specific system.

Green and energy-efficient architectures. The major Cloud infrastructures are deployed in huge data centers that consume an enormous amount of energy, as large as 3% of the global energy supply and responsible for the 2% of the global greenhouse emissions, according to recent studies. In order for the Cloud innovation to achieve a breakthrough at a worldwide scale, it must be sustained by green computing technologies combining renewable and clean energy sources with energy efficient hardware manufacturing design and energy efficient data center operation.

4.2 Systems

In this section we describe some of the research challenges when designing next generation sustainable Ultrascale systems.

Rethinking the memory stack. The memory stack, especially with respect to caches, needs to be rethought and redesigned to deal with the remote heterogeneous Big Data processing tasks. As memory bandwidth and size are not able to cope with the increase in scalability and performance, new technologies that can preserve the amount of memory per core need to be researched. Similarly, the I/O bandwidth will not keep pace with the machine speed.

Revisiting the data-flow approach for edge computing and streaming. The challenge of dealing with Big Data applications from an increasing number of sources requires novel decentralized solutions for content gathering and processing in networked Clouds, data transformations across a multitude of formats and encodings, and real-time stream processing to extract compact meaningful information or reduce its dimensionality. The extreme heterogeneity and scalability dimensions of the Ultrascale systems require moving the computation towards the far peers of the network, often referred nowadays as next generation edge and fog computing paradigms. Revisiting data-flow based approach is required.

Facilitate programming using DSL on parallel and distributed systems. We anticipate that domain-specific languages (DSL) specialized to a particular application domain will present great potential to facilitate programming of Exascale architecture, as well as scalability and performance of applications

Robust and transparent resilience methods. The de-facto MPI standard for programming parallel architectures lacks fault tolerance support. Sustainable Ultrascale systems require new robust and transparent resilience methods to support the runtime environments of their programming languages, whether general-purpose MPI or DSL. Autonomic properties such as self-awareness, self-adjusting, self-anticipating, self-organizing, self-recovery, as well as reliability, security and interoperability are required.

Time and QoS-sensitive execution. Time-constrained HPC within Quality of Service (QoS) parameters (colloquially known as “real-time HPC”) becomes ever more important open issue to be addressed and rediscovered for with every new application and computer architecture. More fundamental approaches that move from besteffort execution to enforced QoS-enabled are required.

4.3 Technologies

In this section we describe some of the main technologies that need to be researched towards sustainable Ultrascale systems.

New memory technologies To facilitate fast processing of increasingly amounts of Big Data, new persistent memory technologies like non-volatile RAM (NVRAM) and 3D technologies are required to be researched.

Heterogeneous devices and systems Designing heterogeneous computing systems com-bining some of the flexibility of software with the high performance of heterogeneous hardware, for example through very flexible high speed computing fabrics like FP-GAs.

Software stack and accelerator kernels Accelerators are currently going strong and will further evolve in the next generation supercomputers. Extending the software stacks to efficiently support them and enable automatic and performance-portable translation across devices is required. Programming languages today are still decoupled from each other and miss a uniform abstraction.

Lighter and optimized virtualization technologies To improve utilization in data centers, there is an increasing trend to move towards lighter technologies such as containers and unikernels to replace virtual machines that consume not only a full copy of an operating system, but also a virtual copy of all the hardware that the operating system needs to run.

Quantum and optical technologies Transistors are now approaching sizes as small as 10nm or below that they will not follow the normal laws of physics such as gravity. They will soon be impacted by "quantum effects" which means that their behavior becomes unpredictable. Quantum and optical technologies are disruptive innovations and technological changes that still require an enormous amount of research to achieve maturity, general-purpose acceptance and use, and ultimately market penetration and deployment.

Interconnection networks In order to support ultrascale computing, decentralization is an important aspect, thus interconnection networks play an essential role in the architecture of HPC systems, distributed datacenters and clouds, as the large number of processing and storage interconnected nodes enable to meet higher computing and storage demands. We need interconnection networks and container technology which allows to not only control latency, but provide ultra-low latency networking and high communication bandwidth, to avoid data bottlenecks and to guarantee a balanced computation-communication system that can results in the adequate application scalability. Innovative interconnection technologies are needed to improve high speed communication performances, reliability and fault tolerance by retaining the power consumption on the current level.

4.4 Use-Cases

In this section we describe some important use cases use for validating the sustainable Ultrascale systems.

Multi-physics simulations Using Ultrascale computing architectures for scientific discoveries like multi-physics simulations remains an interesting research use case.

Big Data and machine learning Discovering patterns in Big Data requires advanced machine learning-based techniques. In this context, deep learning is currently a strong trend that requires Ultrascale computing capabilities for fast, efficient and accurate training. Also, other machine learning techniques need scalable computing systems to analyze very large data sets.

Modeling biological systems Human brain modeling, gene sequencing or personalized medicine are some examples of areas that could be used to test Ultrascale systems and to integrate several of the paradigms devised in this area.

New specialized end-user devices The eventual end of the Moore's law requires rethinking the way end-user devices are designed, especially the mobile technologies that do not necessarily need raw computing power, but can offload most of the processing and data storage in the Cloud. When the chips just cannot get any smaller, there will be a need to design devices that look and behave differently, for example optimized for battery power, energy efficiency, better connectivity and design, instead of performance.

Energy-efficient data center operation It is predicted that the greenhouse gas emissions will increase by 16% by 2040, with potential harmful effects on ecosystems, biodiversity, and livelihood of people by 2047 according to a recent article in Nature. In order for HPC technologies to be sustainable at an Ultrascale level, it must be supported by energy-efficient computer manufacturing design and data center operation, following Cloud computing principles.

4.5 Recommendations

Recommendation 1

Enabling Ultrascale computing. Support 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.

Programming model and environments to express massively parallelism, large scale distribution, heterogeneity, data locality

5.

5. Programming model and environments to express massively parallelism, large scale distribution, heterogeneity, data locality

Challenges in the domain of programming models and tools faced in the context of UCS are extremely difficult[3]. Indeed, todays programming languages focus largely on Von Neumann execution models and software design methods are largely using the model of composition of black-box abstractions based on minimal external interfaces and implemented by hidden inner mechanisms. In addition, most existing programming languages or tools treat software as an isolated, closed-world formal system and are targeted to the programming of computers. Unfortunately, UCS systems will be deeply embedded in the real world. These systems will comprise not only information technology (IT) components (computers), but also machines of many kinds, single and networked sensors, information streams and mass storage of huge amount of data, high level manmachine communication, and so forth. Last but not least, as already mentioned in the section 2 of this document, UCS will also integrate HPC infrastructures for which programming models and tools must allow developers to efficiently used the underlying very complex and parallelized and distributed computing hardware. As a consequence, it is most likely that UCS will be defined and implemented in many languages, each with its own abstractions and semantic structures as well as its own programming paradigm and tools. We can summarize the above using the following sentence:

> UCS will be large scale, parallel, distributed, heterogeneous, decentralized, high performance, robust and evolving systems.

5.1 Present research topics and challenges

In order to be able, tomorrow, to program such complex systems, the community of researchers in the field of programming model and tools currently address the following specific topics:

Cloud/ Fog/ Dew Big Data Computing topics: Compared to past systems, UCS are not only larger but of greater diversity and heterogeneity. UCS will comprise heterogeneous systems using hybrid hardware, but will also use more distributed systems such as Cloud, Fog and Dew one. The Dew Computing is positioned as the ground level for the Cloud and Fog computing paradigms. Vertical, complementary, hierarchical division from Cloud to Dew Computing satisfies the needs of high-and low-end Big Data computing demands in everyday life and work. Researchers are addressing tools and runtime to abstract these complex and overwhelming heterogeneous infrastructures, but also to extract a maximum performance out of them. One step ahead, some researches evaluate the links between other platforms such as IoT and UCS. Computer science community addresses the scalability framework models and tools and performance models to support this effort. Researches evaluate how cloud can provide services to UCS by integrating workflow and map-reduce paradigms and other programming tools for scalable data analysis. The distributed nature of large scale distributed UCS are addressed by researches on security and privacy but also by the management infrastructure such as the monitoring using for example wireless sensor data acquisition and processing (possibly local) and storage, as exemplified in the Dew Computing paradigm.

HPC topics: Concerning dedicated HPC hardware infrastructure, the main research is combining big data and HPC to process large volumes of data on a cluster of modern hybrid compute nodes. Other research concerns the adaptation of classical HPC library to UCS. These libraries are necessary for using UCS from the low level communication one such as MPI to higher level one such as BLAS.

Application-driven topics: From services such as Map-Reduce to Application such as meteorological simulations, researches adapt software to UCS. Most scientific communities are trying to deal with UCS: DNA sequencing, geophysical inversion, meteorological simulations, satellite imagery, but also the one dealing with more abstract subjects such as cellular automata or percolation theory. Researchers also propose services in order to simplify the usage of UCS for particular usage like data analysis.

Tool-driven topics: Researchers are using higher level languages such as DSL, OpenMP or Chapel to address the heterogeneity, the complexity and performance needs of UCS. Other initiatives such as open specification like SYCL or SPIR provide unified way to address this complexity. Indeed, as UCS will be constituted by numerous different complex parts, each addressing specific problems, an approach only based on general purpose programming languages is not anymore suitable. Due to the scale of UCS, programmers need such languages to abstract from the complexity coming from faults and heterogeneity. Interoperability is required because of the distributed and the decentralized nature of UCS. Indeed UCS will not be developed and operate as a whole but, on contrary, will grow organically through continuous and independent developments and updates. The distributed nature of large scale distributed UCS are addressed by research on the management infrastructure such as the monitoring, deployment automation, migrations and the large scale of threads and transient micro-services.

5.2 Future (anticipated) research topics and challenges

Cloud/Fog/Dew Big Data Computing: In the future the highest opportunities lie in the availability of massive scale cloud infrastructure which will be omnipresent. To effectively use these available resources, massively federated and scalable software with orchestration through network awareness will be necessary. As an extension of links between UCS and Clouds, data access models for data mining in Exascale systems will be a key research topic. The integration will be between Cloud systems but also Fog and future type of infrastructure, leading to need on machine-to-machine computing and Cloud computing integration. Heterogeneity of such system will continue to increase, leading to the need to be able to integrate warehouse-scale computing using purpose-designed chips. Integrating the lowest, Dew-level devices will present additional challanges due to the extreme quantities of Physical Edge Devices, their severely low processing power and communication means, and the huge amounts of data generated.

HPC: 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.

Application-driven topics: With the aim of harnessing the power of UCS, scientific community will be able to improve dramatically the quality of models. One key example will be the research focus on meteorology beyond wind simulation (Interfacing between different software packages and data formats, necessary for integration of simulations for complex tasks). New tools will be needed to use UCS for scientists from diverse fields, but tools only available to computer scientists will be needed such as the Hardware/Software Co-design models to guide together the development of hardware and software infrastructure.

Tool-driven: Several tools will be needed to use efficiently UCS. Some tools can be provided by software, but also abstract models and new programming paradigms helping programmers to better use the available resources are helpful. Due to the scale of the systems, one key element will be resource-efficient models for automatic recovery from minute-to-minute failures. As security is often forgotten by programmers, software-defined security models will be needed on large scale distributed infrastructure to simplify its usage. One way to increase security and privacy will be to create new secure Privacy-Preserving data management algorithms such as machine learning. To address code sustainability and adaptation evolution on code production is needed such as source-to-source translators and MDE (Model Driven Engineering) in order to adapt to the underlying hardware.

5.3 Possible approaches to address the future research topics and challenges

Investigations in the following technologies would help to find solution to above listed topics.

Cloud/Fog/Dew Big Data Computing: New abstractions will help to simplify their usage such as software-defined data centers, services abstracted from infrastructure and agnosticism from the infrastructure service API. Data management will be a key of this usage, using group-level data aggregation, locality-based data selection and analysis, data-driven local communications and data processing on limited groups of cores could be approaches to follow.

HPC: The main approach will be to use higher level and hardware independent programming models, but also using adaptive and reactive runtimes. This approach can be based on innovative libraries such as GASPI for asynchronous distributed computing. More generally, improving communication library models will be a key challenges. They can be optimized with model-based innovative network-aware communication models or by using new implementations of the MPI standard focused on performance and collective operations. HPC will be implemented by integrating heterogeneous systems and one way to generalize this fact will be to incorporate reconfigurable accelerators such as FPGA.

Application-driven: The key approach will be to make already available more interconnected using open and widely accepted framework for data formats to support easy format conversion. New algorithmic paradigms will also be necessary such as using cellular automata in engineering and bioinformatics to increase the parallelism of other research fields.

Tool-driven: One way to preserve privacy will be to use homomorphic computing and to provide integrate such computing transformation in the different layers, from low level computing libraries such as linear algebra to higher level one such as Big Data.

5.4 Proposed recommendations

Recommendation 2

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.

Recommendation 3

Break the wall between runtime and programming frameworks. It will help 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.

Recommendation 4

Enabling behavioral sensitive runtime. The ability to provide behavioral information along-with applications will help runtime to take the most relevant decisions based on 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.

Secure operation and resilience of ultrascale systems

6.

6. Secure operation and resilience of ultrascale systems

As discussed in the introduction, Ultrascale computing is a new computing paradigm that comes naturally from the necessity of computing systems that should be able to handle massive data in possibly very large scale distributed systems, enabling new forms of applications that can serve a very large amount of users and in a timely manner that we have never experienced before. It is very challenging to find sustainable solutions for UCS due to their scale and a wide range of possible applications and involved technologies. For example, we need to deal with cross fertilization among HPC, largescale distributed systems, and big data management. One of the challenges regarding sustainable UCS is resilience. Traditionally, it has been an important aspect in the area of critical infrastructure protection (e.g. the traditional electrical grid and the smart grids). Furthermore, it has also become popular in the area of information and communication technology (ICT), ICT systems, computing and large-scale distributed systems. The existing practices of dependable design deal reasonably well with achieving and predicting dependability in systems that are relatively closed and unchanging. Yet, the tendency to make all kinds of large-scale systems more interconnected, open, and able to change without new intervention by designers, makes existing techniques inadequate to deliver the same levels of dependability. For instance, evolution of the system itself and its uses impairs dependability: new components ″create″ system design faults or vulnerabilities by feature interaction or by triggering pre-existing bugs in existing components; likewise, new patterns of use arise, new interconnections open the system to attack by new potential adversaries, and so on.

6.1 Present Research Topics and Challenges

Based on the above mentioned background, we identify the following research challenges for resilience in Ultrascale systems:

Characterization of hardware and software faults in Ultrascale systems Characterization of hardware and software faults is essential for making informed choice about research needs for the resilience of Ultrascale systems. From the hardware perspective if silent hardware faults are exceedingly rare, then the hard problem of detecting such errors in software or tolerating their impact can be ignored. If errors in storage are exceedingly rare, while errors in compute logic are frequent, then research on mechanisms for hardening data structures and detecting memory corruptions in software is superfluous.

Development of a standardized fault-handling model Development of a standardized fault-handling model is key to providing 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. Applications running on todays high performance computing systems are not even notified of faults or given options as to how to handle faults. If the application happens to detect an error, the computer may also eventually detect the error and kill the application automatically, making application recovery problematic.

Improved fault prediction, containment, detection, notification, and recovery Scale is a major opportunity for applications Ultrascale computing Systems. However, the larger the scale, the higher the probability of a failure in some part of the system. To build such a systems, resilience is a must, and that means the we need better fault prediction mechanisms, containment measures and recovery from failures to allow the applications keep-on working even if a specific component fails. [6]

Programming abstractions for resilience in Ultrascale systems Programming abstractions for resilience will be able to grow out of a standardized fault handling model. Several programming abstractions will need to be developed and supported in order to develop resilient Ultrascale applications. The development of fault-tolerant algorithms requires various resilience services.

Standardized evaluation of fault-tolerance approaches Standardized evaluation of fault tolerance approaches will provide a way to measure the efficiency of a new approach compared with other known approaches. It will also provide a way to measure the effectiveness of an approach on different architectures and at different scales. The latter will be important to determine whether the approach can scale to serve the needs of Ultrascale systems...

6.2 Future (anticipated) research topics and challenges

Based on the above mentioned background, technology and service trends, we identify the following research challenges for resilience in Ultrascale systems:

- Convergence of HPC, Cloud, and Big Data brings complex system resilience challenges rather than just adding the challenges of HPC, Cloud and Big Data
- Unprecedented in scale of IoT and mobility, large scale sensor networks in extreme networks such as cyber-physical systems and smart grids.
- Reorientation of resilience challenges brought by implementation of Software-Defined Networking
- Security, trust, and privacy issues with end-users heavily involved in system construction, management and maintenance.

There has been a convergence of the different technologies in terms of HPC, Cloud, Big Data in order to face new challenges in terms of large scale computing, simulation and data analytics.

This convergence is foreseen to be extended in order to take over the opportunities offered by the Internet of Things (IoT) including mobile networks and large scale sensor networks such as those systems deployed in smartgrids.

Many of the security and resilience challenges opened in IoT area are shared by the Ultrascale domain including secure data management, trust and privacy.

6.3 Possible approaches to address the future research topics and challenges

It is requested to anticipate the upcoming applications enabled by this new global system. The communications versus computation challenges and the related network complexity raised by such new platforms request more distributed solutions such as Fog computing, Software-Defined-Networks (SDNs), etc.

In terms of security, we will need more distributed solutions for trust management and more efficient and extended homomorphic encrytion primitives.

The new systems should be able to handle partial inconsistencies by using meta-heuristic/probabilistic approaches. In particular, to sustain the expected scalability and dynamics of these systems, the foreseen solutions would probably have to rely on unstructured Peer-to-Peer (P2P) approaches, where the supported overlay network relies on random and self-organized topologies to handle group membership and communications in an automated and distributed way.

The self-healing properties and proven resilience against the most important types of faults met in such large-scale systems (namely crash faults and cheating fault) render these approaches ideal candidates to sustain the development of robust systems able to handle Secure operation and resilience of Ultrascale systems.

6.4 Proposed recommendations

Recommendation 5

Developing new programming abstractions for resilience and standardized evaluation of fault-tolerant approaches. Efficient tools and methods for characterization of both hardware and software faults must be studied and designed. 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.

Data management for Big Data in the Exascale Era

7.

7. Data management for Big Data in the Exascale Era

Ultrascale systems will require sustainable data storage and data for addressing the predicted exponential growth of digital information. Current Big Data frameworks, such as Apache Hadoop and Spark, are used widely to solve certain types of data analytic problems, aiming mainly at high productivity. On the other hand, HPC data storage and access approaches for data-intensive application are built with different requirements in mind and achieve higher efficiency for data access and processing. Going forward, it is necessary to explore synergistic approaches from between HPC and Data Analytics approaches, in an effort to reform the handling of the whole data life cycle and achieve the best of both worlds.

7.1 Present research topics and challenges

The main challenges today in data and storage management evolve around designing infrastructures that combine features from both datacenter and scientific worlds to keep up with application requirements while taking advantage of new technologies. Generally, there is increasing agreement that the ecosystem created around Apache software stack can significantly boost productivity of some classes of applications [1]. However, even though some progress has been recently made, there is still significant need to improve the performance of many components of this stack. Therefore, there is an open discussion about the suitability of integrating Big Data Analytics and HPC platforms [17]. Some tools, such as Apache Hadoop, have already found a large popularity in commercial data centers and in the academic domains. However, despite evidence that they are fit for application domains such as astronomical image analysis, medical image analysis, and genome analysis, there are several obstacles for their adoption in the HPC data centers, including the poor integration with the batch schedulers or the specific storage requirements [12].

The main aspects of storage and data management systems that need to be reconsidered are the following:

Scale-out: The ability to scale in multiple dimensions (diverse applications, access patterns, infrastructure size) remains an important problem. Datacenter frameworks achieve scale-out with some form of partitioning, while the scientific world has limited scalability in an effort to maintain a single, consistent view of all the data and metadata.

Consistency: Typically, this refers to having a consistent view while there are concurrent read and write operations to the data. Combining consistency with scaling is an ongoing challenge. Datacenter frameworks give up on consistency by partitioning data, while scientific libraries rely on strong filesystem semantics for updates and synchronization at the expense of scaling.

Heterogeneous data integration: The infrastructure should be able to cope with very large and high-dimensional data, stored in different formats at a single data center, or geographically distributed across multiple infrastructures. The problem is especially prevalent in large enterprises, which have many systems along with an abundance of unstructured data under management. Current Big Data and HPC I/O solutions rely on a large variety of I/O interfaces, making it difficult to integrate multiple data sources. Future frameworks should expose their mechanisms, features, and services over diverse data using standard technologies, to make them usable as building blocks for high-level APIs, software components, and applications.

Elasticity: The infrastructure must be able to handle a growing workload in an agile manner, by dynamically allocating the required physical resources (processors, storage, network). Conversely, as soon as the workload shrinks, the infrastructure should release unneeded resources. A characteristic of the data-analytics frameworks is the difficulty to predict the needs of storage resources. Therefore, although there is still significant work to be done, datacenter solutions have striven to achieve elasticity under the pressure of load variability. On the other hand, scientific frameworks have mostly evolved from traditional scalable processing solutions that use fixed and static amounts of resources.

Data access and analysis as a service: Data analytics tend to involve complex data accesses and processing approaches that cannot be handled directly by users of data. Therefore, there is a need to provide end users with ready-to-use services of data storage, access, analysis algorithms, or ready-to-use knowledge discovery tools/apps, accessible through the network and through different interfaces.

Interoperability of data management and processing frameworks: Today, the popular service-oriented paradigm allows running large-scale distributed applications on heterogeneous platforms along with software components developed using different programming languages or tools. However, till now applications, algorithms, and services developed in different frameworks do not interoperate. Programming frameworks, languages and tools must be designed/ adapted/extended to allow a wider integration of multiple data analytics frameworks.

Data locality: In very large-scale systems, the cost of accessing and moving data could be very high, limiting scalability of applications. Data locality techniques are investigated to provide control over where data values are stored and where tasks are executed. Thus, designers and developers can ensure parallel computations execute near the variables they access, or vice-versa.

Security: Data protection, identity management, and privacy are becoming particularly important as infrastructures are being consolidated for efficiency purposes and as they scale to large sizes to cope with exploding application requirements.

7.2 Future (anticipated) research topics and challenges

Data storage and management are today coupled with the programming model and the runtime system. Given the current trends in data storage and access technologies, there is a number of issues that need to be addressed in the longer term at the boundary of data access and data processing:

Large scale data access and processing: Data processing (analytics) can be categorized broadly with respect to two dimensions: (a) The relative size of the dataset with respect to main memory (DRAM) size, leading to two broad categories of in-memory vs. out-of-memory (or out-of-core) analytics. (b) The complexity of the required processing over the data, again broadly broken down to linear vs. higher-order computation. We believe that analytics on data sets that do not fit entirely in memory, and in particular analyzed that require higher-order processing, constitute one of our biggest challenges for modern infrastructures.

Efficiency: The infrastructure should minimize the resource consumption for a given task and an amount of data to be accessed and processed. Additionally, the infrastructure should achieve high utilization for all available components. Efficient data processing is a fundamental requirement in the future data-analytics frameworks. Existing datacenter solutions have evolved as modular frameworks with multiple layers and interfaces, resulting in high overheads per unit of work. Scientific libraries and software stacks on the other hand, tend to be more lean at the expense of also being more monolithic.

Quality of Service: has so far focused mostly on high-throughput without consideration of response times and other QoS aspects. Typically, both data analysis and data processing have been carried out in a batch fashion. Thus, the different computing iterations were applied over a set of stored data, but without having concurrent and independent workloads running on the same infrastructure. In recent years, domains that need a set of constantly refreshed information have gained greater importance: scientific computing research [19], environmental research by means of sensor networks [10], social network analytics [8], and many others. Scaling data access and processing over shared resources makes it necessary to minimize the interference across system layers and tasks, which have a detrimental effect on user-perceived QoS [18].

Storage device technology: Future infrastructures should encompass different generations of technologies, such as different types of processors, storage devices, and network components, because the required scale cannot be achieved and maintained using a single generation of all system components. Nowadays there is a lack of solutions that adequately support heterogeneous infrastructures for data storage in the datacenter or scientific worlds.

Programming abstractions: Existing approaches to data management, processing, and analytics, such as MapReduce and workflow models often used on HPC and clouds still lack in expressiveness, flexibility, and analysis tools. As a result, performing efficient data analysis today remains a challenge and an arduous task. Research activities in this area can also borrow for the HPC world, especially as the number of processing elements increases in parallel systems and datacenter alike. Models, such as PGAS and its variant need to be investigated and adapted for supporting efficient Big Data analytics developments.

7.3 Possible approaches to address the future research topics and challenges

We believe, that, at a high level, approaches to address these problems should:

HPC and data analysis Understand and realize the close relationship between HPC and data analysis in the scientific computing area and advances in both are necessary for next-generation scientific breakthroughs. To achieve the desired unification, the solutions adopted should also be portable and extensible to future Ultrascale systems. These systems are envisioned as parallel and distributed computing systems, reaching two to three orders of magnitude larger than today's systems [5].

Embrace and cope with new storage device technologies The appearance of new storage device technologies 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, handled, and processed, throughout the storage and data management stack.

Shift from performance to efficiency Instead of only or mostly considering absolute performance as the driving force for new solutions, we should shift our interest to considering the efficiency at which infrastructures are operating. This is becoming more important as the size at which future infrastructures are required to operate continues to scale with application requirements.

7.4 Proposed recommendations

Recommendation 6

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.

Recommendation 7

To design and develop efficient 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.

Energy awareness and efficiency for sustainable UCS

8.

8. Energy awareness and efficiency for sustainable UCS

Significant reduction of energy consumption in complex large scale systems requires addressing a number of challenges [13].

First of all, energy-related data must be effectively measured, collected, integrated and analyzed. To this end, appropriate system architecture is needed to allow the monitoring system to provide sufficiently detailed information avoiding communication bottlenecks. In UCS energy monitoring will have to be accompanied by accurate models and estimation methods that are required to reduce number of sensors and meters, and consequently cost and complexity of such systems.

Fast progress in terms of energy efficiency has been done within hardware itself, for instance in the adoption of low power CPUs or accelerators such as GPUs and FPGAs. However, to make the most out of such hardware, optimization that is focused on energy efficiency is needed on the software side. Therefore, software development tools along with patterns that will help to achieve high energy efficiency are needed especially for heterogeneous hardware and accelerators that will be present in USS. Advances in resource management techniques will be needed to cope with this heterogeneity at large scale.

We also anticipate that the role of virtualization technologies in large scale systems will be increasing. Therefore, further research on lightweight virtualization, co-location, and dynamic consolidation and rescheduling is needed.

It is worth noting that the substantial part of the overall energy consumption of large computing systems comes from the infrastructure. Therefore, to improve efficiency often optimizing integration of IT and infrastructure is essential. For example, energy consumed also depends on infrastructure and environmental parameters such as temperature, cooling air/liquid flow, humidity, cooling system power usage, etc.

Finally, as we envision future USS not only as centralised isolated supercomputers but also federations of data centers, edge computing systems, and surrounding infrastructure, the research on energy efficiency of highly distributed systems, federated data centers or integration with renewable energy sources and smart grids can be interesting direction.

8.1 Present research topics and challenges

Ultra large scale profiling and monitoring for reporting energy usage: There is ongoing work on methods and tools for fine-grained monitoring of energy consumption by various components, e.g. using Intel RAPL, PAPI, EML, etc. This work also includes code instrumentation, application profiling, methods to define performance using performance counters as well as energy monitoring and profiling of physical and virtualized infrastructures (virtual machines). Important research challenge is to provide monitoring for multiple layers of software stack and techniques for propagation of metrics between the layers [11].

Multi-objective energy metrics: Current research on metrics concentrates on energy efficiency metrics for evaluation of whole and specific parts of computing systems, adaptivity of data centers to changing load and energy availability, and cooling performance

Models and simulation of energy consumption of UCS: There is ongoing work on simulators such as GroudSim, DCworms, GreenCloud, Philharmonic, SimGrid and/or trace-based analysis of applications. One of the aspects included in current research is simulation of clouds. Due to the difficulty for researchers to manage complex data center infrastructures, we need Cloud simulators. Such simulators should take into account all actors (e.g., VMs, physical hosts, network hardware) and activities (e.g., VM migration, powering off physical hosts) of workload consolidation. Among all activities, VM migration is one of the most widely used, because it provides the capability of moving the state of running VMs between physical machines, thus it allows to dynamically adjust the workload. Therefore, current research challenge is to design an accurate model for VM migration. First of all, since VM migration is a very network-intensive process, we need models for energy consumption of network transfers. On this model, energy consumption models for VM migration can be built and used in Cloud data centers simulators, allowing us to provide more accurate prediction of energy consumption in data centers.

Energy efficient resource management and scheduling: Current research focuses on energy-aware resource management and scheduling techniques including techniques such as on/off, DFVS, power capping, allocation to suitable hardware, etc. There are also efforts to finding trade-offs between performance and energy consumption, estimating costs. Researchers approaches the whole data center optimization taking into account thermal issues, cooling, electricity conversion. For virtualised infrastructures workload dynamic consolidation is seen as a solution that can allocate a run-time the physical resources such as to reduce the number of servers required to execute applications. This is enabled by run-time virtualization mechanisms which allow fine and coarse grained resource allocation. The workload consolidation have to deal with two important objectives: (1) minimization of servers required for running applications; (2) minimization of applications performance degradation.

Energy efficient algorithms and eco-design of applications: There is much effort on manual porting applications on heterogeneous platforms with multicore processors and accelerators. It includes optimizing applications by improving parallelization, use of cache and memory, etc. So far the primary objective was performance.

8.2 Future (anticipated) research topics and challenges

Ultra large scale profiling and monitoring for reporting energy usage: Overhead of the measurement and monitoring must be minimized to the limits as otherwise is im-practical in ultrascale systems. We need to provide application and hardware energy profiles that are scalable to millions of cores and be able to identify sources of excess energy consumption to facilitate its improvements. This should also apply to large virtualized infrastructures including techniques for lightweight virtualization (e.g. containers).

Multi-objective energy metrics: It is important to reflect energy efficiency of ultrascale systems but with respect to other aspects such as fault tolerance, reliability/resilience, security, code sustainability (issues hard to guarantee in large systems and affected by some energy efficiency techniques such as switching on/off, low voltage processing, use of heterogeneous systems and complex accelerators, high temperature processing). As data movement will be becoming more and more important compared to computing the common metrics such as Flops/Watt will be not sufficient anymore. Hence, there will be need for metrics assessing efficiency of data movement in the system.

Models and simulation of energy consumption of UCS: We need simulators for ultrascale both large scale supercomputers and distributed clouds (including the whole infrastructure: cooling, electrical equipment). As future ultrascale systems may be very complex and heterogeneous, high frequency monitoring and profiling will be very difficult. Therefore, accurate and effective simulation models are needed. Energy simulator should also cover virtualised infrastructures (clouds). A cloud simulator should provide simulations at three level: the physical level, concerning the physical machines and the network infrastructure. Most of the existing models for physical machines are based on the assumption that CPU is the only parameters that matters, regarding energy consumption of PMs. This assumption may lead to inaccurate results when talking about communication intensive workloads, therefore we need to investigate more detailed models of energy consumption of physical machines. Another interesting research topics may be the investigation of how containers work. Containers are a technology that provides a level of virtualization that is closer to the PM hardware, therefore giving the possibility to offer higher energy efficiency compared to VM. However, at this time, no simulator tried to model the way container works. Providing models for containers may provide an interesting advancement to the current state-of-art.

Energy efficient resource management and scheduling: One of the main future challenges is to cope with high number of failures and assuming inaccuracy of information. Sharing resources on fine-grained level to achieve energy efficient datacenters and increase the performance per watt ratio to the ultrascale level. Another aspect is ultrascale cloud management and large virtualised systems. This should include efficient multi-cloud management techniques, which require fully distributed resource management and scheduling, as well as approaches managing a large number of lightweight virtual machines. An important challenge is to study how energy efficiency influences pricing models, which may lead to new pricing models and methods to engage users in energy savings. Even in the case of the use of energy efficient computing components the energy consumption of the whole ultrascale system can be large. Therefore, important challenge is to solve problems including the whole energy flow including aspects such as variability, price and carbon footprint of energy (e.g. use of renewables), heat re-use, new cooling methods -various types of liquid cooling.

Energy efficient algorithms and eco-design of applications: The important challenge is the autotuning of applications on heterogeneous systems. The computing systems of today provide a set of heterogeneous resources per computing node, such as, multicore processors and accelerators. An automatic and energy efficient execution of an application requires that the runtime environment is able to partition the application in order to optimize the execution time and cost.

8.3 Possible approaches to address the future research topics and challenges

Ultra large scale profiling and monitoring for reporting energy usage: Possible approaches include lightweight monitoring approaches using hardware support, accurate models allowing energy estimation (as in the case of RAPL), big data / machine learning techniques to cope with data sizes and complexity.

Multi-objective energy metrics: Metrics relevant for ultrascale, for example power pro-portionality, scalability per J, trade-off with resilience, efficiency scalability, data movement efficiency on all levels.

Models and simulation of energy consumption of UCS: In general applying simulations to ultrascale systems will require simplified but accurate empirical modeling, stochastic approaches, and machine learning. Research may include machine learn-ing approaches, such as decision trees, linear regression, regression trees and artificial neural network. Energy simulators should include models of energy consumption of heterogeneous hardware to simulate it in ultrascale. Such approaches will work with data collected in heterogeneous cloud infrastructure, running different types of hardware configurations (meaning different CPUs and different types of network hardware and topologies) and different types of virtualization technologies. Energy simulators should also model lightweight virtualisation techniques.

Energy efficient resource management and scheduling: Possible approaches may include approximate algorithms taking into account risk of failures and assuming inaccuracy of information. They could use stochastic and machine learning methods for management. On single resources level today technology allows the logic isolation of jobs at a still significant performance costs. Therefore, low cost context switch and low cost process reallocation are features that need to be improved in order to achieve effective resource sharing policies. For virtualised infrastructures the workload dynamic consolidation techniques should take advantage of very lightweight virtualisation techniques and attempt to allocate large numbers of virtualised tasks in such a way to efficiently use the shared (possibly heterogeneous) resources. Additionally, adjusting execution to energy demand response and local renewables characteristics can help to significantly reduce energy costs and carbon footprint of ultrascale systems.

Energy efficient algorithms and eco-design of applications: Applications should be automatically analyzed and represented as a graph where nodes represent tasks that can be compiled and run in any of the computing elements of the system. Many bibliography addresses the scheduling of such kind of graphs but it is still a challenge to automatically generate quality graphs from applications code, especially with a focus on maximising energy efficiency.

8.4 Proposed recommendations

Recommendation 8

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 parts of the system when needed without deterioration of the whole system performance.

Recommendation 9

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.

Recommendation 10

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 programmes for IT) and other external systems.

Reformulating science problems and refactoring solution algorithms for ultrascaling computing

9.

9. Reformulating science problems and refactoring solution algorithms for ultrascaling computing

The needed reformulation of algorithms and applications from different areas of research towards their usage for Ultrascale systems and platforms has to address different challenges that arise from the different application areas, algorithms and programs [14]. Challenges include the scalability of the applications programs to use a large number of system resources efficiently, the usage of resilience methods to include mechanisms to enable application programs to react to system failures, and the inclusion of energy-awareness features into the application programs to be able to obtain an energy-efficient execution. As a large number of heterogeneous resources has to be controlled when using Ultrascale systems, the availability of suitable programming models and environments also plays an important role for Ultrascale applications. Programming models for Ultrascale computing should provide enough abstractions such that the application programmer does not need to deal with all low-level details of an efficient execution of the (parallel) programs and the control of the execution resources of the platform. On the other hand, the programming models should enable the application programmer to concentrate on the algorithmic aspects and problem-specific issues of the specific application area such that program development is supported as far as possible.

These topics are addressed in the previous sections, while this section is concerned with the usage of the corresponding research results in the context of large application programs from different areas, including coupled multiphysics problems, multilevel/multigrid methods for discrete multiscale problems as well as data science and simulation methods. An important issue is the integration of the techniques developed to address the different aspects into the application programs. For existing application programs it would be beneficial if such an integration could be performed without a significant change in the program code. The provision of corresponding libraries could be a useful step towards such a seamless integration.

The usage of Ultrascale systems and platforms also allows for extension of existing application models and programs such that larger problems can be considered and more accurate solutions can be computed with Ultrascale systems, giving the users from the different application areas a potentially significant benefit from Ultrascale computing. However, the underlying algorithms, methods and techniques have to be suitable for an execution on Ultrascale systems and platforms, i.e., the required scalability, resilience, and energy-efficiency requirements have to be fulfilled at this level. Only if these requirements are fulfilled, it can be expected that the resulting application program is suitable for Ultrascale computing. A successful implementation of an Ultrascale application program may also require a switch to new algorithms and simulation techniques that are more suitable for Ultrascale computing than existing approaches. Reasons may be a better inherent scalability of the new algorithms, a better locality of reference behavior or less dependencies between the computations of the algorithm, thus enabling a more efficient parallelization also for heterogeneous Ultrascale architectures. However, such a switch may involve a significant re-formulation of the application program, since the algorithmic structure is usually deeply intertwined with program control.

The rest of this section contains a list of application areas and approaches that are likely amenable for Ultrascale computing as summarized in Figure 9.1.

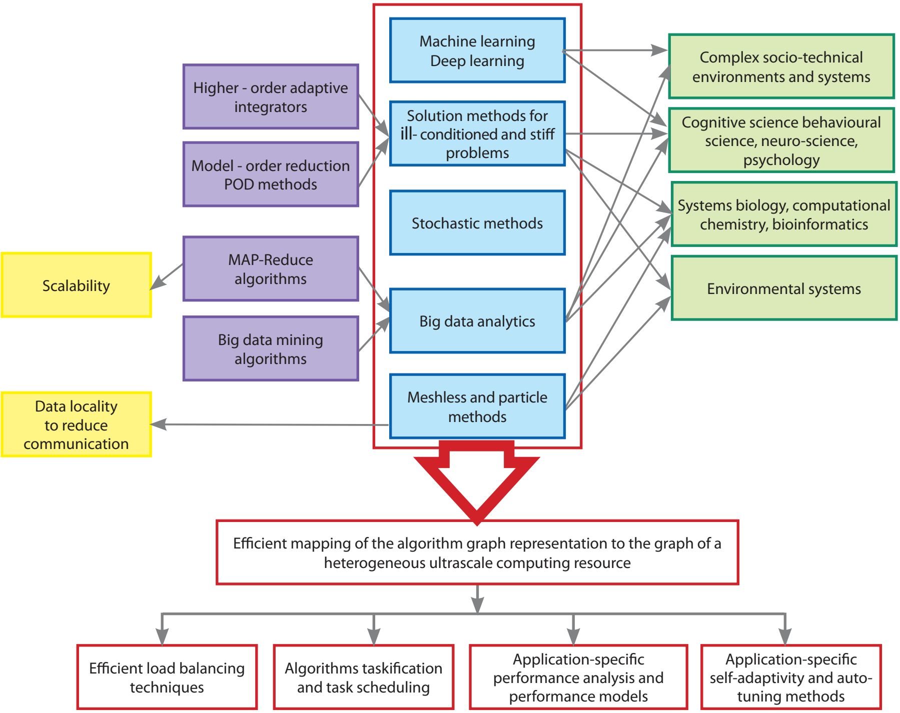
Machine learning
Deep learning
Solution methods for ill- conditioned and stiff problems
Stochastic methods
Big data analytics
Meshless and particle methods
Higher - order adaptive integrators
Model - order reduction POD methods
MAP-Reduce algorithms
Big data mining algorithms
Scalability
Data locality to reduce communication
Complex socio-technical environments and systems
Cognitive science behavioural science, neuro-science, psychology
Systems biology, computational chemistry, bioinformatics
Environmental systems
Efficient mapping of the algorithm graph representation to the graph of a heterogeneous ultrascale computing resource
Efficient load balancing techniques
Algorithms taskification and task scheduling
Application-specific performance analysis and performance models
Application-specific self-adaptivity and auto-tuning methods

Figure 9.1: Application areas amenable for UCS

9.1 Present research topics and challenges

Methods and algorithms for coupled multiphysics problems

» Operator splitting methods and algorithms; stable semi implicit time stepping techniques; splittings for multidimensional problems in space; splittings/ decompositions with respect to physical processes; challenge: find fully parallelizable stable splitting methods

» Monolithic methods; fully adaptive implicit time integration and asynchronous time integration; composite block preconditioned iterative methods for strongly coupled problems;

- Optimal control methods and multi-objective optimization; challenge: find approximations and heuristics to solve the problem in polynomial time with sufficient accuracy;

Multilevel/multigrid methods and algorithms for discrete multiscale problems

- Robust multilevel methods for ill conditioned problems and/or problems with strongly heterogeneous coefficients; hierarchical basis methods; local/additive Schur complement approximations;
- Multilevel methods for structured and unstructured grids; additive and multiplicative methods and algorithms; geometric and algebraic methods and algorithms;
- Meshless and particle-based methods, including intrinsic and extrinsic enrichment, and point collocation methods; scalability and communication reduction by data locality;
- Model reduction methods; singular value decomposition and moment matching based methods; proper orthogonal decomposition; state-space truncation methods for parallel model reduction; parallel model reduction of large dynamical systems;

Parallelism

- Balancing communications and computations for basic classes of numerical algorithms; reducing global communications; avoiding transposition of data, that is avoiding large-scale FFT routines;
- Challenges include: the lack of efficient new algorithms for new heterogeneous architectures with accelerators, difficulties with parallelisation and performance engineering of existing codes;

Data science methods

- From big data mining to big data analytics; compressive sampling, matrix completion, low-rank models, and dimensionality reduction; efficient learning and clustering; robustness to outliers; convergence and complexity issues; performance analysis; scalable, online, active, decentralized, deep learning and optimization;
- Big data analytics relying on MapReduce type methods: scalable, distribute computing, MapReduce on Multi-Core, GPU, hybrid distributed environments; opportunistic/heterogeneous computing; programming model;

- From text ontologies to video and multimedia analytics: video semantic content analysis framework based on ontology combined multimedia standards providing functionalities to enable generation of audiovisual descriptions;

Simulation

- Conservative methods for coupling quantum/particle and continuum methods; examples include extension of the chip analysis to the gate level, coupling with photon-based thermal models;
- Discrete event simulations of large circuits with conservative and optimistic methods; synchronization schemes for parallel simulation; adaptive protocols for parallel discrete event simulation;
- Scalable methods and algorithms for simulations for the internet of things problems, including hardware adaptation and parallel processing in ultra-low power multi-processor systems on chip; example: processing of large-scale time series;
- Break-through in simulation in molecular dynamics is among the current challenges; example: avoiding global communication caused by FFT in Poisson-Boltzmann solvers;
- Stochastic simulations in systems biology; methods in computational chemistry and bioinformatics; drug discovery and in-silico drug design; simulations for environmental systems; simulations related to global climate changes;

9.2 Future (anticipated) research topics and challenges

Methods and algorithms for coupled multiphysics problems

- Efficient solution of ill conditioned and stiff problems; towards extremely scalable methods and algorithms for strongly coupled nonlinear problems; robust parallel methods in time and in space;
- Higher order time integration schemes using implicit methods; higher order implicit-explicit Runge-Kutta schemes; super linearly convergent parareal time-parallel time-integration schemes;
- Covering of heterogeneous architectures and efficient mapping and load balancing techniques for these architectures;

- Taskification of algorithms and task scheduling methods for heterogeneous architectures; dynamic task scheduling; utilization of directed acyclic weighted representing the dependency among tasks, based on their execution time and communication time;
- Application-specific performance analysis and performance models as basis for application-specific scheduling and mapping techniques; application-specific self-adaptivity and autotuning methods;

Parallelism

- New fault-tolerance and resilience algorithms for ultrascale computing with robust self-correcting mechanisms providing guaranteed stability and global accuracy/convergence.
- Automatic adaptation of execution of algorithms to the specifics of different ultrascale architectures, decoupling of specification of algorithms from their execution on an hardware platform
- Virtualization and cloudification: moving existing codes to cloud systems: performance preservation, data privacy, resource management enabling unlimited scalability
- Automatic generation of code for accelerator components providing portability and efficiency.

Data science methods

- New approaches for data analytics, inclusion of real-time stream and multistream processing, consideration of insecure, uncertain, incomplete and unreliable data
- 3Vs requirements (Volume, Velocity, Variety) call for increased scalability; 3Vs describe a set of data and a set of analysis conditions that define a concept of Big Data; example: 1-divide and conquer using Hadoop; 2-brute force using an appliance such as the HANA (High-Performance Analytic Appliance);
- Sustainable storage of very large data and algorithms for their collection, generation, analysis and visualization

Simulation

- As an example, molecular dynamics requires a complex redesign of methods, algorithms and huge software implementations to get significantly better stability to increase significantly the currently used time step of 2-5 fs. The related implicit parallel solvers should totally avoid the global communications.

9.3 Possible approaches to address the future research topics and challenges

Monolithic methods

- New algorithms, possibly implemented on novel architectures (such as dataflow)

Multilevel methods

- Performance engineering of current codes aimed at multicore architectures; Multilevel heterogeneous methods and algorithms and their efficient mapping on the graph representation of heterogeneous ultrascale computing systems; Multilevel multiscale methods and algorithms for multiphysics applications in strongly heterogeneous media and uncertain data;

Parallelism

- Use of domain-specific languages (DSL) for specific application areas with specialized compilers to generate efficient code for different HPC architectures (clusters, GPU, accelerators, FPGA, MIC and others); challenge: how can the target architecture be captured such that efficient code can be generated?
- Automatic inclusion of self-adaptivity features in existing algorithms, methods and implementations; self-healing and self-repairing methods and algorithms;

Data science methods

- Machine learning and deep learning; advanced techniques and algorithms to parameterize deep neural network structures; example: artificial neural networks with many hidden layers and parameters; Machine learning.

Simulation

- Complex socio-technical environments and systems; functional resonance analysis in hazard identification; simulations in cognitive science, behavioral science, psychology, neuro-science, simulation of social behavior;
- Simulations in cognitive science, behavioral science, psychology, neuro-science, simulation of social behavior

9.4 Proposed recommendations

Recommendation 11

Enabling complex Ultrascale computing applications. Develop complex applications based on complementary utilization of numerical and non-numerical, deterministic, stochastic and hybrid, multiscale and multiphysics, 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.

Recommendation 12

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.

Approaches to address these challenges

10.

10. Approaches to address these challenges

To address the former challenges and to provide solutions for the recommendations suggested in this Research Roadmap, there is a need of a coordinated research plan at the European scale. This plan, that should be coordinated with other efforts in Europe, will allow to continue the efforts of the NESUS Action in a near med term, while opening the landscape to all the research community through open calls that could be part of the H2020 program in a near future.

- Specific Research and Development projects. Proposal: 10 years program.
- Coordination and support action. Proposal: 6 years program.
- Center of Excellence program. Proposal: 6 years for establishment.
- Technological transfer through a Sustainable Ultrascale Software Center. Proposal: 6 years.
- Program for ultrascale computing education. Proposal: 5 years. Major research activities and funding to accomplish this plan should be centered on software. The main reason is that the HPC Systems sales covering the top 500 supercomputers (www.top500.org) indicates a contribution of European companies in the order of low single digit percentage value over a time period of the last 10 years. In contrast to that, 83 % of HPC application SW is made in Europe, which clearly shows the strong role of HPC SW in Europe. The real strength in HPC in the last several decades in Europe is clearly centered in software development and research, which is also confirmed by a much larger number of companies and academic groups that are active in SW development and research for large scale systems compared to those active in HW development or research for that kind of systems.

Conclusions

11.

11. Conclusions

NESUS action is fostering collaboration of very active European research groups in the field of sustainable ultrascale computing, increasing the cross-fertilization of scientists coming from different communities to structure and federate a disseminated community in Europe and to bridge the gap between theoretic research and real world applications by including stakeholders and industry players. This Research Roadmap is a result of the contribution of all those communities to set up the basis that will provide significant advances for the current challenges of ultrascale computing.

The major purpose of this Research Roadmap is to facilitate the adoption and usage of sustainable ultrascale computing systems, by 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 environment at European level.

The main conclusions of this work have the form of a set of recommendations that are shown at the beginning of the document. In our opinion, not reaching solutions that are able to overcome the issues posed in the recommendations in a short-mid term will create potential risks in the path towards ultrascale computing systems. Thus, we recommend to set up a research program as an approach to solve the problems and reach the recommendations issued here.

References

[1] Angelos Bilas, Toni Cortes, Domenico Talia, María S. Perez, Javier Garcia-Blas, Pilar González-Férez, Andr Brinkmann, Stergios Anastasiadis, Malcolm Muggeridge, Carmela Comito, Sai Narasimhamurthy, Anna Queralt, Florin Isaila. Data storage for big data in the exascale era: Challenges and prospects. Technical report, COST Action IC1305, EU, December 2015.

[2] Carretero, Jesús and García-Blas, Javier and Wyrzykowski, Roman and Jeannot, Emmanuel. *Proc. Second International Workshop on Sustainable Ultrascale Computing Systems* NESUS 2015. University Carlos III of Madrid, 2015.

[3] G. D. Costa, T. Fahringer, J. A. R. Gallego, I. Grasso, A. Hristov, H. Karatza, A. Lastovetsky, F. Marozzo, D. Petcu, G. Stavrinides, D. Talia, P. Trunfio, and H. Astsatryan. Exascale machines require new programming paradigms and runtimes. *Supercomputing frontiers and innovations*, 2(2), 2015.

[4] Editor: Jesus Carretero. Special issue on Sustainability in Ultrascale Computing Systems. *SUPERCOMPUTING FRONTIERS AND INNOVATIONS International Journal*, 2(2):131, 2015.

[5] Jesús Carretero et al. Memorandum of understanding. In *Network for Sustainable Ultrascale Computing (NESUS)*, page 30, 2014.

[6] A. Gainaru, F. Cappello, M. Snir, and W. Kramer. Fault prediction under the microscope: a closer look into hpc systems. In SC, 2012.

[7] Jesus Carretero, Radu Prodan, Leonel Sousa. State-of-the-Art, Software Techniques to Increase Sustainability in Ultrascale Systems: a Roadmap. Technical report, September 2015.

[8] S. Khopkar, R. Nagi, and A. Nikolaev. An Efficient Map-Reduce Algorithm for the Incremental Computation of All-Pairs Shortest Paths in Social Networks. In *2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 1144–1148.

[9] Lopes, L., ilinskas, J., Costan, A., Cascella, R.G., Kecskemeti, G., Jeannot, E., Cannataro, M., Ricci, L., Benkner, S., Petit, S., Scarano, V., Gracia, J., Hunold, S., Scott, S.L., Lankes, S., Lengauer, C., Carretero, J., Breitbart, J., Alexander, M. (Eds.). *First Workshop on Techniques and Applications for Sustainable Ultrascale Computing Systems (TASUS 2014)*. LNCS 8805 and 8806, Springer, 2014.

[10] K. G. S. Madsen, L. Su, and Y. Zhou. Grand Challenge: MapReduce-style Processing of Fast Sensor Data. In *Proceedings of the 7th ACM International Conference on Distributed Event-based Systems*, DEBS '13, page 313318. ACM.

[11] Marcos D. Assuncao , Jorge Barbosa , Vicente Blanco , Ivona Brandic , Georges Da Costa , Mateusz Jarus , Helen D. Karatza , Laurent Lefevre , Ilias Mavridis , Ariel Oleksiak , Anne-Cecile Orgerie . On energy monitoring and analysis of ultra scale systems: State of the art and future directions. Discussion paper, COST Action IC1305, EU, June 2016.

[12] S. Michael, A. Thota, and R. Henschel. HPCHadoop: A framework to run Hadoop on Cray X-series supercomputers. In *Proceedings of Cray user group*, 2014.

[13] Michel Bagein, Jorge Barbosa, Vicente Blanco, Ivona Brandic, Samuel Cremer, Sbastien Frmal, Helen D. Karatza, Laurent Lefevre, Toni Mastelic, Ariel Oleksiak, Anne-Ccile Orgerie, Georgios L. Stavrinides, Sbastien Varrette. Energy efficiency for ultrascale systems: Challenges and trends from nesus project. Technical report, COST Action IC1305, EU, September 2015.

[14] NESUS Working Group 6. Report on the requirements for ultrascale systems from the applications perspective. Technical report, COST Action IC1305, EU, December 2014.

[15] B. Obama. Creating a National Strategic Computing Initiative. Executive Order, July 2015.

[16] Peter H. Feiler, Kevin Sullivan, Kurt C. Wallnau, Richard P. Gabriel, John B. Goodenough, Richard C. Linger, Thomas A. Longstaff, Rick Kazman, Mark H. Klein, Linda M. Northrop, Douglas Schmidt. *Ultra-Large-Scale Systems: The Software Challenge of the Future*. Software Engineering Institute, USA, June 2006.

[17] D. A. Reed and J. Dongarra. Exascale computing and Big Data. *Communications of the ACM*, 58(7):56–68, 2015.

[18] Y. Sfakianakis, S. Mavridis, A. Papagiannis, S. Papageorgiou, M. Fountoulakis, M. Marazakis, and A. Bilas. Vanguard: Increasing server efficiency via workload isolation in the storage i/o path. In *Proceedings of the ACM Symposium on Cloud Computing*, SOCC '14, pages 19:1–19:13, New York, NY, USA, 2014. ACM.

[19] R. Stoica, M. Frank, N. Neufeld, and A. C. Smith. Data handling and transfer in the LHCb experiment. 55(1):272277.

List of contributors

Working Group Leaders:

WG1:

Leonel Sousa, Instituto Superior Tcnico (IST), Universidade de Lisboa, Portugal. Radu Prodan, University of Innsbruck, Austria.

WG2:

Alexey Lastovetsky, University College Dublin, Ireland. Georges DaCosta, University Paul Sabatier III -IRIT, France.

WG3:

Pascal Bouvry, University of Luxembourg, Luxembourg. Tuan Anh Trinh, Corvinus University, Hungary.

WG4:

Angelos Bilas, FORTH and University of Crete, Greece. Toni Cortes, Barcelona Supercomputing Center (BSC), Spain.

WG5:

Ariel Oleksiak, Poznan Supercomputing and Networking Center, Poland Laurent Lefevre, Inria, University of Lyon, France

WG6:

Gudula R¨unger, Chemnitz University of Technology, Germany. Svetozar Margenov, Bulgarian Academy os Sciences, Bulgaria.

Other Contributors:

Andr Brinkmann, Johannes Gutenberg University Mainz, Germany.

Anna Queralt, Barcelona Supercomputing Center (BSC), Spain.

Anne-C´ecile Orgerie, CNRS, IRISA, France.

Bilijana Stamatovic, University of Donja Gorica, Montenegro.

Carmela Comito, ICAR-CNR, Italy.

Dana Petcu, University of Timisoara, Romania.

Domenico Talia, University of Calabria.

Florin Isaila, University Carlos III Madrid, Spain.

Francisco Almeida, La Laguna University, Spain.

Helen Karatza, Aristotle University of Thessaloniki, Greece.

Ilias Mavridis, Aristotle University of Thessaloniki, Greece.

Javier Garcia-Blas, University Carlos III of Madrid, Spain.

Jean-Marc Pierson, IRIT, University Paul Sabatier of Toulouse, France.

Jorge Barbosa, University of Porto, Portugal.

Juan Antonio Rico Gallego, University of Extremadura, Spain.

Malcolm Muggeridge, Seagate, United Kingdom.

Manuel F. Dolz, University Carlos III of Madrid, Spain.

María S. Perez, Universidad Politecnica de Madrid, Spain.

Mario Kova, Faculty of Electrical Engineering and Computing, University of Zagreb, Croatia.

Maya Neytcheva, Uppsala University, Sweden.

Milan Mihajlovic, University of Manchester, United Kingdom.

Neki Frasheri, Polytechnic University of Tirana.

Pilar González-Férez, University of Murcia, Spain.

Radim Blaheta, Institute of Geonics, CAS, Ostrava, Czech Republic.

Raimondas Ciegis, Vilnius Gediminas Technical University, Lithuania.

Roel Wuyts, IMEC, Belgium.

Roman Trobec, Jozef Stefan Institute, Slovenia.

Roman Wyrzykowski, Czestochowa University of Technology, Poland.

Sai Narasimhamurthy, Seagate, United Kingdom.

Sebastien Varrette, University of Luxembourg, Luxembourg.

Stergios Anastasiadis, University of Ioannina, Greece.

Thomas Rauber, University of Bayreuth, Germany.

Vicente Blanco, La Laguna University, Spain.

Vladimir Voevodin, Moscow State University.

The roadmap document has been discussed during plenary sessions of various NESUS meetings. We thank all the participating members for their constructive input to the document.

2017

NESUS

Network for Sustainable Ultrascale Computing

cost IC1305