ETP4HPC

European Technology Platform for High-Performance Computing

Strategic Research Agenda 2015 Update

European Technology Multi-annual Roadmap Towards Exascale
Update to 2013 Roadmap

ETP 4 HPC

SUPPORTED BY THE **EXDCI** PROJECT WITH FUNDING FROM THE EUROPEAN UNION'S HORIZON 2020 RESEARCH AND INNOVATION PROGRAMME UNDER THE GRANT AGREEMENT **NO 671558**

ETP4HPC is an industry-led think tank and advisory group of companies and research centres involved in High-Performance Computing (i.e. Supercomputing) technology research in Europe. It was formed in 2011 with **the aim to build a world-class HPC Technology Supply Chain in Europe**, increase the global share of European HPC and HPC technology vendors as well as maximise the benefit of HPC technology for the European HPC user community. **ETP4HPC is also the EC's partner in the HPC contractual Public-Private Partnership** (cPPP[1]) which monitors and manages the European HPC research investment programme supported by a €700M investment by the EC within the Horizon 2020 programme.

ETP4HPC believes that a competitive European HPC technology value chain will help Europe achieve economic leadership through the use of HPC tools for product creation and service delivery, for the development and supply of world-class technologies and sub-systems, and for the supply, and administration of HPC systems themselves.

According to multiple reports, investment in HPC can take a variety of forms: a) Buying HPC platforms and services, b) Developing technologies to feed into HPC systems, software and services, c) Using HPC to improve business processes, d) Using HPC to improve products and e) Supply of HPC support services, including compute cycles. **HPC is able to generate huge rates of return** (e.g. an IDC report[2], which has a particular European context, argues that 'each euro invested in HPC on average returned €867 in increased revenue/income and €69 in profits'). The strategy adopted by the European Commission in 2012[3] highlights the need for intensified HPC technology provision in Europe and calls for an integrated and concerted effort by the entire European HPC industry. HPC is also one of the pillars of the Digital Single Market (DSM)[4] strategy adopted by the EC in 2015, which includes Big Data and Cloud and aims to build a basis for the long-term growth of the European 'digital' economy. This SRA serves as a tool to achieve the HPC-related objectives set in these strategies.

[1] www.ec.europa.eu/digital-agenda/en/high-performance-computing-contractual-public-private-partnership-hpc-cppp

[2] IDC, 'High Performance Computing in the EU: Progress on the Implementation of the European HPC Strategy', 2015 (Commissioned by EC)

[3] European Commission, 'High-Performance Computing: Europe's place in a Global Race', 15 Feb 2012

[4] European Commission, 'A Digital Single Market Strategy for Europe', 6 May 2015

CONTENTS

1 **Foreword** 6
2 **Executive summary** 7
3 **The European HPC Ecosystem** 8
3.1 SRA Update 2015 10
3.2 The Process 11
3.3 The elements of the Horizon 2020 HPC Roadmap 12
4 **New trends in HPC challenges, use and technology** 14
5 **Technical Research Priorities** 24
5.1 HPC System Architecture and Components 26
5.2 System Software and Management 29
5.3 Programming Environment 34
5.4 Energy and Resiliency 37
5.5 Balance Compute, I/O and Storage Performance 41
5.6 Big Data and HPC usage Models 45
5.7 Mathematics and algorithms for extreme scale HPC systems 47
6 **Research milestones** 52
6.1 HPC System Architecture and Components 54
6.2 System Software and Management 55
6.3 Programming Environment 56
6.4 Energy and Resiliency 58
6.5 Balance Compute, I/O and Storage Performance 59
6.6 Big Data and HPC usage Models 60
6.7 Mathematics and algorithms for extreme scale HPC systems 61
7 **End-user and ISV requirements** 62
8 **Extreme-Scale Demonstrators** 66
8.1 Approach 68
8.2 Proposal of ETP4HPC for the EsD calls 71
9 **Ecosystem at large - stakeholders and initiatives** 72
9.1 European Extreme Data and Computing Initiative 74
9.2 Eurolab4HPC 74
9.3 Centres of Excellence in Computing Applications 75
9.4 Big Data Value Association 76
10 **Conclusions and outlook** 78
11 **Glossary** 80
12 **Contributions and Acknowledgements** 82

1. FOREWORD

Since the first ETP4HPC Strategic Research Agenda (SRA) issued in 2013, High Performance Computing has gained even more recognition as a key technology for the future of science, industry and society. In addition to its role to provide an insight through large simulations in industrial and scientific fields, it is widely accepted that the digital economy will heavily benefit from HPC. The data economy, drawing on data provided by the Internet of Things and other sources, may improve the way we use resources and deliver services if HPC is used to analyse the data and make valuable judgement from it.

Developing this technology is a strategic move to accelerate the innovation and the deployment of the digitalised industry and data economy. Having a strong HPC ecosystem will enable the European players to benefit from the added value of the HPC value chain and will help to position HPC users at the forefront of their domains. Today, the HPC market is a fast growing one and represents a significant fraction of the IT infrastructures. HPC technology also has the potential to supply solutions to other IT fields, reinforcing its strategic position in a world that increasingly relies on IT. As one of the main trends of this domain, the interaction between users and technology is becoming crucial in order to take advantage of all the benefits of the advances made in HPC. Having strong expertise and skills in Europe is the key for maintaining a leading position in the use of HPC. We are confident that the contractual Public Private Partnership established between the European Commission and ETP4HPC will succeed in developing a world-class European HPC ecosystem.

This document is the first update of our Strategic Research Agenda. It aims to identify the research priorities for the European HPC technology community and reflects the point of view of the experts associated with ETP4HPC. Compared to the previous full issue, this SRA maintains the same overall direction but due largely to technology evolution, some of the milestones have changed and others have been revised.

To achieve this work, we have consulted with over 170 European HPC experts and I would like to thank them all for sharing their vision with us. We hope that this SRA Update will receive the same warm welcome as the first one and that it will help to shape leading edge research projects in the scope of the Horizon 2020 programme.

We (all) believe that this research program will be a key element in advancing Europe's global position in HPC and central to maximising the economic benefits from its exploitation.

ETP4HPC Chairman
Jean-François Lavignon

2. EXECUTIVE SUMMARY

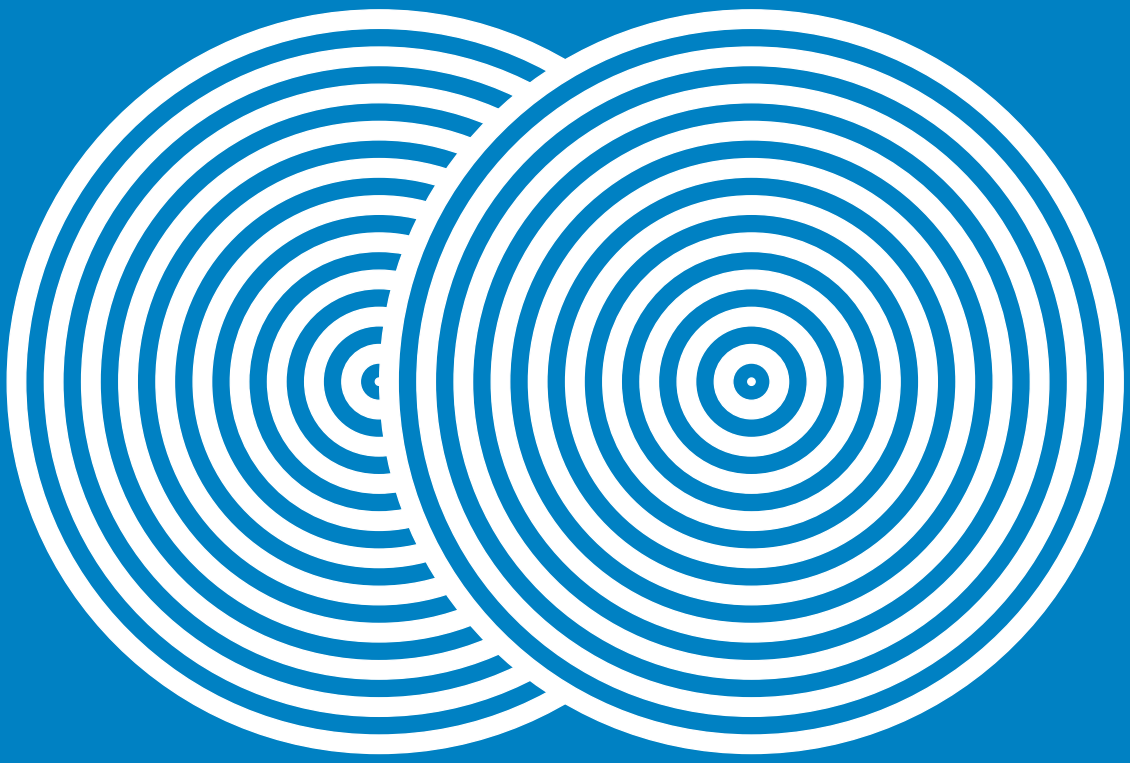
This document is an update to the European High-Performance Computing (HPC) Technology Strategic Research Agenda (SRA) issued in 2013 by ETP4HPC, the European HPC Technology Platform. ETP4HPC issues and maintains this SRA as a mechanism to provide contextual guidance to European Researchers and Businesses but also to guide EU priorities for research in the Horizon 2020 HPC programme. This SRA delineates a roadmap for the achievement of European exascale capabilities focusing on the following areas: HPC System Architecture and Components, Energy and Resiliency, Programming Environment, System Software and Management, Big Data and HPC Usage Models, Balance Compute, I/O and Storage Performance, Mathematics and algorithms for extreme scale HPC systems and Extreme-Scale Demonstrators. It is the work of experts associated with ETP4HPC and it also includes the input of industrial end-users, independent software vendors and other IT communities (e.g. Hipeac and BDVA). This SRA update will be followed by a full issue in 2017.

First, the strategic context of this roadmap is defined. These initial chapters help understand the elements of the European HPC ecosystem, the process of creating this roadmap and its role in the Horizon 2020 programme. Then, an update on the trends and challenges of HPC technology is presented. The details of the areas of HPC technology targeted are in Chapter 5, which also includes references to the individual milestones listed in Chapter 6. Finally, Chapter 8 introduces the topic of Extreme-Scale Demonstrators which will serve to prove the context and relevance for the various technologies emerging; and build a European community in their supply-chain.

Whilst the research priorities presented in Chapter 5 and 6 apply to the FETHPC 2016-17 calls, the implementation of Extreme-Scale Demonstrators should take place in 2018-2019. ETP4HPC considers this concept as an important mid-term validation point of the research related to HPC technology within H2020, even though its implementation may require different funding mechanisms, or a new combination of the existing ones.

Whilst this report points heavily towards the H2020 HPC program for its delivery, the primary objective is to develop Europe's HPC ecosystem and maximise its exploitation. Accordingly, businesses and researchers who independently undertake these objectives to the benefit of Europe are to be applauded.

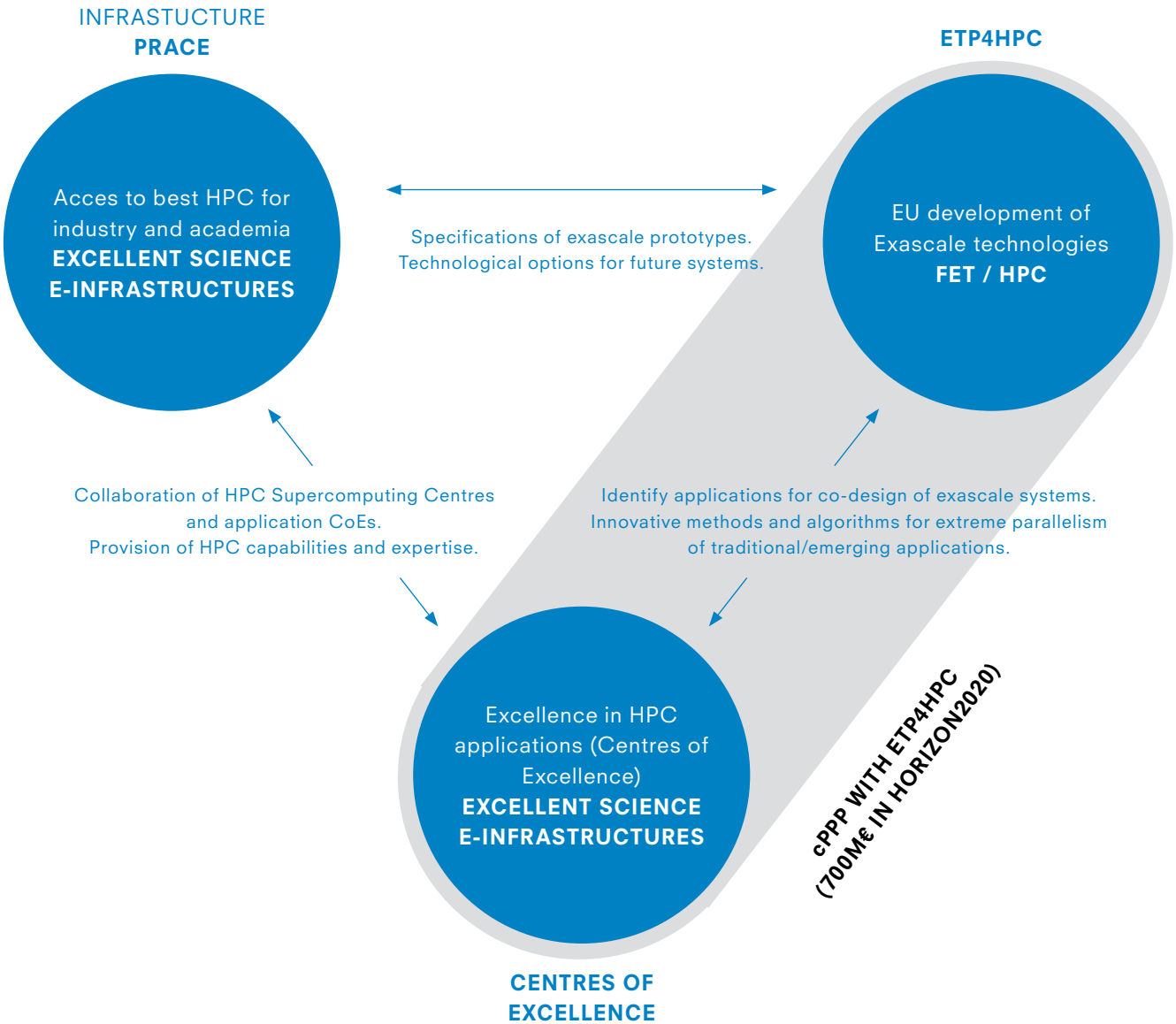
3.



THE EUROPEAN HPC ECOSYSTEM

The European HPC Ecosystem aims to develop world-class HPC technologies, platforms and applications, leading to exascale systems and their advanced use, thus creating jobs, new products and more efficient companies as well as enabling scientific discoveries. This will contribute to the economic competitiveness of the European economy as a whole and also to the well-being of the European citizen by equipping our scientists, economists, sociologists, agriculturalists, politicians and engineers to address the Grand Societal Challenges that the continent faces.

Figure 1
The three pillars of the European HPC Eco-system and the interactions between them. The HPC cPPP covers the areas of technology provision and application excellence. The FETHPC5 programme of the EC supports the development of European HPC technology while the EINFRA6 calls include the operation of the Centres of Excellence in Computing Applications. The separately funded EXDCI project provides mechanisms for the coordination of the entire strategy.



[5] The H2020-FETHPC-2016-2017 call text is available at:
www.ec.europa.eu/research/participants/portal/desktop/en/opportunities/h2020/calls/h2020-fethpc-2016-2017.html#c,topics=callIdentifier/t/H2020-FETHPC-2016-2017/1/1/1&callStatus/t/Forthcoming/1/1/0&callStatus/t/Open/1/1/0&callStatus/t/Closed/1/1/0&+identifier/desc

[6] The EINFRA-21-2017 call text is available at:
www.ec.europa.eu/research/participants/portal/desktop/en/opportunities/h2020/topics/2122-einfra-21-2017.html

This European HPC Ecosystem (i.e. Technology Provision represented by ETP4HPC, Research Infrastructure represented by PRACE[7] and Application Expertise in the form of Centres of Excellence for Computing Applications) has entered a stage of rapid progress. The entire process is monitored by the contractual Public-Private Partnership (cPPP) for HPC, one of the eight European structures of this type.

The first FETHPC (Future and Emerging Technologies - HPC) technology research projects with a total combined value of almost €100M are due to be rolled out within the second half of 2015. These projects are based on the guidelines defined in the first ETP4HPC Strategic Research Agenda (SRA 1, issued in 2013) of ETP4HPC. The first Centres of Excellence for Computing Applications[8][9] (CoEs) with a total investment value of €40M have the start of their operation scheduled for the third quarter of 2015. These efforts form part of a €700M investment package from the European Commission in this technology within the Horizon 2020 Research and Development Programme.

The EXDCI (European Extreme Data and Computing Initiative) coordination and support action (CSA) project is led by PRACE, in close partnership with ETP4PC. It embodies the expertise and experience of EESI[10] and a portfolio of independent experts aims to specifically stimulate and coordinate the European HPC strategy. It is scheduled to operate from September 2015 for two and a half years. Its mission is to monitor, coordinate and support this strategy of the European HPC Ecosystem, providing tools for: technology 'roadmapping', analysing academic needs, international collaboration and education and training across the three pillars of European HPC (Infrastructure, Technology and Applications).

A separate coordination project (Eurolab-4-HPC[11], see also Chapter 9) will operate in parallel with EXDCI and will focus on Excellence in High-Performance Computing Systems and longer-term research in computing architectures and HPC beyond 2020.

3.1 SRA UPDATE 2015

This Strategic Research Agenda (SRA) Update is a part of the continuous process of maintaining the European HPC Technology Roadmap by ETP4HPC. The SRA document is an important tool for establishing the European HPC Technology Roadmap and guiding the subsequent H2020 funding programmes. For this reason, it needs to be continually revised. ETP4HPC plans to carry out a full SRA revision every four years and interspace them with mid-term updates.

The first issue of the SRA was written in 2013 (SRA 1) and its recommendations form the basis of the first FETHPC (Future and Emerging Technologies HPC) research call within the Horizon 2020 programme. This is the first mid-term update, and should be read in conjunction with the 2013 release. The SRA is a living document and will be used to define the subsequent research calls. ETP4HPC has embarked on the process of issuing this document (i.e. SRA 2) as an update to SRA 1, reflecting the changes that we know have occurred, i.e.

· The first round of FETHPC calls (FETHPC-1-2014) addressed a number of milestones identified in SRA 1. The projects resulting from that call are now in progress and they are expected to produce solutions that future projects might choose to build on.

· The HPC Ecosystem has witnessed technology developments within and beyond Europe that need to be mirrored in the roadmap. A separate chapter is dedicated to those changes, which have an effect on the original milestones defined in SRA 1.

· SRA 1 was designed in the early stages of the operation of ETP4HPC. The membership base of the organisation has multiplied since and this update allows new members to include their point of view in the roadmap.

[7] www.prace-ri.eu

[8] A summary of the Centres of Excellence in Computing Applications is available at: www.ec.europa.eu/programmes/horizon2020/en/news/eight-new-centres-excellence-computing-applications

[9] The EINFRA-5-2015 call text including the Centres of Excellence in Computing Applications is available at: www.ec.europa.eu/research/participants/portal/desktop/en/opportunities/h2020/topics/329-einfra-5-2015.html

[10] www.eesi-project.eu

[11] www.cordis.europa.eu/project/rcn/197540_en.html

· In a similar vein, the European HPC Ecosystem has evolved significantly since 2013. Measures have been put in place to monitor and coordinate its strategy. The contractual Public-Private Partnership between the EC and ETP4HPC aims to ensure the commitment of all stakeholders in the development of the Ecosystem. The EXDCI project (see Chapter 9.1) synchronises the efforts of the Ecosystem order to achieve the agreed objectives. This SRA update reflects this maturity through the increased involvement of all stakeholders (with an emphasis on industrial HPC end-users).

Whilst constituting a ~90 page document, this SRA 2 is still an update of the first 2013 SRA and mainly provides refreshed technical content in Chapter 5. As scheduled, this update will be followed by a full issue of the SRA in 2017 (SRA 3) and a further update in 2019. We look forward to constructive contributions to those from a wider community in due course.

3.2 THE PROCESS

The process of preparing this update began in January 2015. First, a global analysis of the trends in HPC challenges, use and technology was performed. This analysis is presented in Chapter 4. From this work emerged eight areas of the European HPC Technology Roadmap:

· HPC System Architecture and Components

· System Software and Management

· Programming Environment

· Energy and Resiliency

· Balance Compute, I/O and Storage Performance

· Big Data and HPC Usage Models

· Mathematics and algorithms for extreme scale HPC systems

· Extreme-Scale Demonstrators

All ETP4HPC members were invited to participate in technical working groups, mirroring the categories above. The working groups were led by selected members of ETP4HPC, who produced the content of Chapters 4 to 6. The topics of "Mathematics and Algorithms for extreme scale HPC systems" and "Extreme-Scale Demonstrators" are introduced by this update as they did not appear in SRA 1.

In June 2015, two workshops involving external experts were organised to verify the validity of the document's assumptions taking into account the requirements of the market. First, a workshop involving industrial HPC end-users was held to obtain their feedback in relation to their system requirements. Then, a similar event with the involvement of Independent Software Vendors (ISVs) took place (see Chapter 7).

After two successful project phases and a final conference in May 2015 in Dublin, EESI/EESI2 coordination and support actions published final recommendations on strategic European actions with a particular focus on software key issues improvement[12]. EESI2 made a selection of key recommendations in the areas of Tools & Programming Models, Ultra-Scalable Algorithms and Data Centric Approaches. These recommendations are complementary to and consistent with the ETP4HPC ones. A workshop between EESI2 and ETP4HPC was held in Dublin and different EESI experts participated in the SRA workgroups.

The Big Data community has been consulted during the preparation of this SRA. A workshop was held in July 2015 to discuss the converging areas of technology addressing the needs of both communities. This input has been used to propose some of the milestones of the R&D roadmap.

[12] www.eesi-project.eu
an interim deliverable is at: www.www.eesi-project.eu/wp-content/uploads/2015/05/EESI2-ALL-RECOM-July2014.pdf

3.3 THE ELEMENTS OF THE HORIZON 2020 HPC ROADMAP

Over the last two years the Horizon 2020 roadmap for HPC related work programmes and other actions have evolved significantly. ETP4HPC's current view of the timing of this Programme's various components is outlined in Figure 2.

· The Work Programme 2014/2015 (WP14/15) has resulted in nineteen accepted projects being launched at the time of writing this document and beginning their operation in Q4 2015.

· The call for WP2016 will open in the Q4 2015 while the call for 2017 will open a year later.

· The two following Work Programmes are shown tentatively and at this moment they should be considered as 'under evaluation'.

· The Coordination & Support Action (CSA) called "EXDCI" (see Chapter 9.1) started in September 2015. ETP4HPC has proposed to continue this type of support throughout the entire Horizon 2020 (CSA 16/17).

· As mentioned in the SRA 1, ETP4HPC strongly recommends building HPC System prototypes, in the pre-exascale timeframe (2018/19), as proof-points for the effectiveness of the Work Programmes 2014/15 and 2016/2017. This concept called "Extreme-Scale Demonstrators" is outlined in Chapter 8.

· ETP4HPC is committed to renewing the SRA document for each new Work Programme call to ensure that the parties interested in submitting a proposal have access to an up-to-date roadmap including the relevant research topics and the supporting actions. This SRA 2 (an update of SRA 1) is therefore planned to be released by 20th October 2015.

HPC — HORIZON 2020 ROADMAP

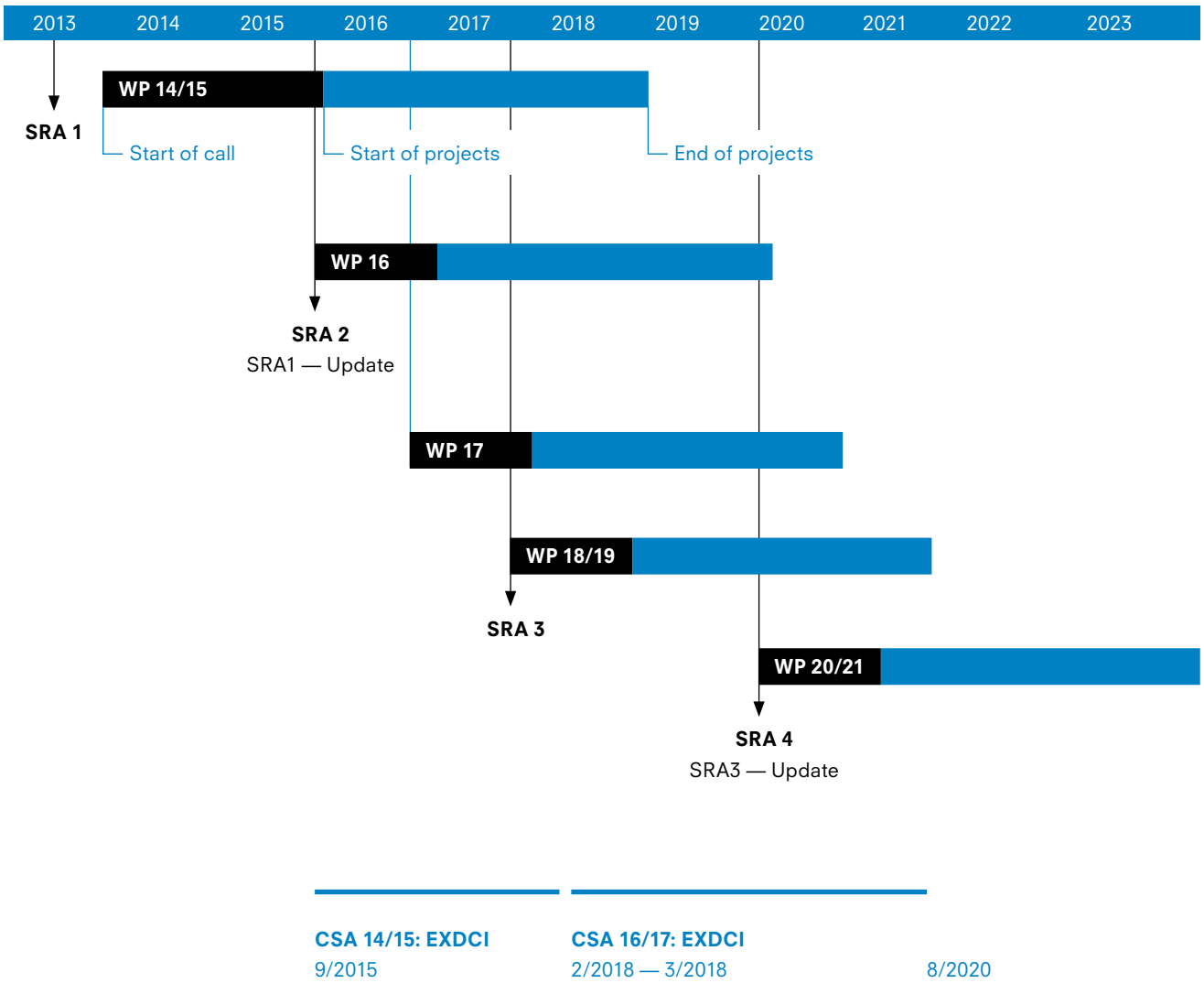


Figure 2
ETP4HPC's view of the Horizon 2020 HPC roadmap: the timelines of the relevant Work Programmes (WPs).

4.

NEW TRENDS IN HPC CHALLENGES, USE AND TECHNOLOGY

This chapter provides a high-level view on the latest trends identified in the various fields of HPC technology — from chip to system — and new ways of using HPC infrastructure.

The "Multi-dimensional HPC Vision" introduced in Chapter 4 of the SRA 1 is still valid and has proven to be useful in dealing with the multiple facets of HPC technology.

There is a demand for R&D and innovation in both extreme performance systems and mid-range HPC systems. Almost all scientific domains and some industrial users want to achieve extreme-scale performance systems as soon as possible. At the same time, there is a need, particularly expressed by industrial users and ISVs, for more flexible, easier-to-use, more productive and cost-effective HPC systems delivering mid-range performance.

The ETP4HPC HPC technology providers are also convinced that to build a sustainable ecosystem, their R&D investments should target not only the exascale objective. This market will be too narrow to yield a sufficient return on investment and support sustainable technology development, and that such a strategy would weaken the European players. On the contrary, an approach that aims at developing technologies capable of serving both the extreme-scale requirements and mid-market needs can be successful in strengthening Europe's position.

As a consequence, the SRA has two dimensions: one targeting the R&D aiming at developing the new technologies able to offer more competitive and innovative HPC systems for a broad HPC market, and another to enhance these technologies with the right characteristics to address the extreme-scale requirements.

The third element is the trend of developing new HPC applications. Besides traditional HPC workloads, more and more Big Data applications will need to be addressed with HPC solutions. There is also a request from some domains to use HPC systems for the control of complex systems, such as smart grids. The Cloud delivery model is yet another trend that will impact the features of future HPC solutions. Accordingly, the SRA has a dimension to address all these new usages.

There is also a major concern that some of the stakeholders of HPC development could be limited by some barriers, namely shortage of skills, insufficient availability of services to fill the gap between potential user demands and HPC solution offerings, usability of solutions or flexibility and vitality of the ecosystem. The fourth dimension identified in this SRA reflects these concerns.

Some changes need to be introduced to these four dimensions to take into account some of the evolutions of the last two years. Figure 3 provides an updated version of our four-dimensional view of HPC technology development:

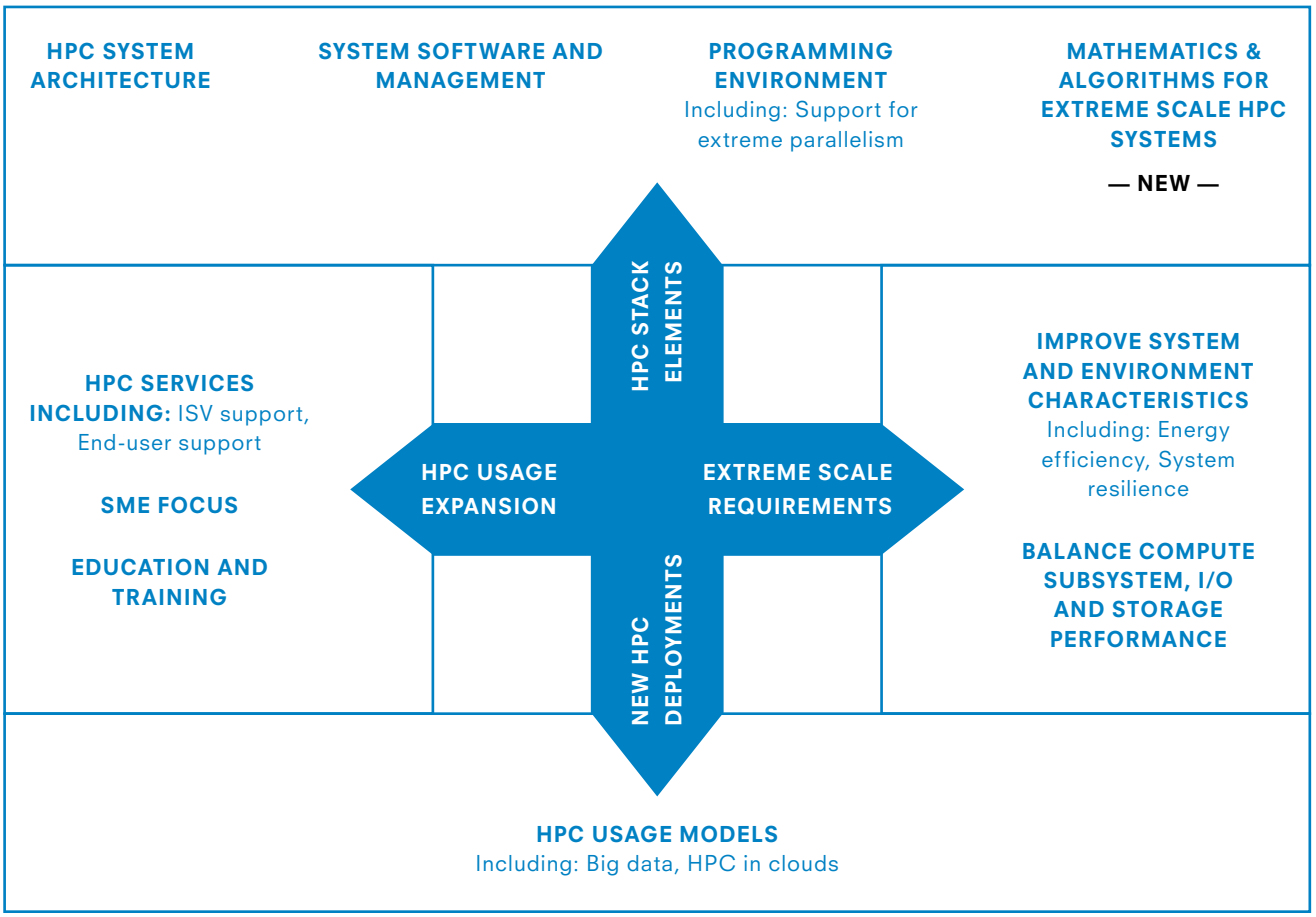


Figure 3
The new four-dimensional model of European HPC technology development based on its first version as defined in SRA 1 (2013)

The following additions and amendments seemed necessary:

· A new research field was added to the dimension "HPC Stack Elements" called "Mathematics and algorithms for extreme scale HPC systems". Mathematical methods and algorithms play an increasingly important role for both, providing HPC technologies as well as using HPC technologies efficiently. By adding this new dimension, ETP4HPC acknowledges the key role of mathematical methods and algorithms for future development of HPC architectures and technologies. It will engage in this broad and active area of research, taking into particular account the impact on the commercial stakeholders. One example is the methods for scalable data analytics, an area which is predominantly driven by commercial operators and where huge progress in algorithms is required to make them scalable.

· The two topics "usability" and "affordability", although still essential and among the top priorities of academic and industrial end-users need to be dealt with in a very integrative way on the HPC architecture level. They both

are basic drivers for new HPC stack technology updates, especially in the areas of system software, programming environments as well as power efficiency. The seven technology domains in Chapter 5 spell out new research priorities taking demands for usability and affordability into account. Thus, these two aspects will not be handled separately any more.

There are three topics, not or only marginally addressed in the SRA 1, receiving increasing attention: "Security in HPC infrastructures", "resource virtualisation" and "HPC in clouds":

· **Keeping** computer systems, and the information stored/processed on them secure by preventing unauthorised access, use or modification is clearly a very important task across all fields of computing. Hence, a great variety of R&D projects on architectures, systems and measures that enhance system and data security, are being carried out by many academic and commercial players, and results are increasingly put into operational use. The prevalent HPC usage Model has profited from advances in authentication and authorisation of users, in securing high-speed data communication, and in access control for datasets. Since all three topics are shared with most other fields of computing, it does seem reasonable to continue to leverage advances in these fields. No specific and unique requirements from HPC were identified here; and, of course, it will be important for HPC R&D projects to be aware of said security provisions and accommodate them across the full stack (from hardware to services). Since this is a horizontal and mainly implementation concern, we do not propose specific security priorities, objectives or milestones in this document. Emerging HPC use cases (Cloud, Big Data) do add security concerns not present in the classical HPC environment. The attack surface grows through the use of services, increased commercial value at stake and the handling of valuable and personal or commercially relevant data. Therefore, R&D targeting these use cases needs to accommodate a high-level of data protection against outside as well as inside attacks; existing and continually improved methods for end-to-end encryption, dynamic enforcement of fine-grained access privileges, and detailed data access logs can be leveraged here. Again, we do not propose specific security priorities, objectives or milestones here; requirements to provide support for security layers is mentioned where appropriate.

· **CPU, device and network virtualisation** has enabled significant efficiency and reliability in Enterprise computing. Its uptake in HPC has been extremely limited, mainly due to concerns about performance and vendor lock-in. With advances in network virtualisation, an example of a compelling use case is emerging: being able to move all or parts of a "virtual HPC Cluster" to different HW transparently to the workloads in case of impending failures, or to accommodate partial system maintenance which would substantially increase availability of a system, and thus, address resiliency concerns. This will require a careful combination of CPU virtualisation (or containers) and network virtualisation, and is discussed in Chapters 5.2 and 5.5.

· **Leveraging Cloud mechanisms** to provide universal and simple access to HPC systems and services is by no means a new idea. Commercial players (e.g. Amazon) have had good initial success in offering Web-Service based access to HPC-class resources in certain niches, e.g. genome analysis workflows. For most other usage areas of HPC, the uptake of Cloud is still in a very early phase. The "Fortissimo" FP7 project, for instance, has created an HPC marketplace, which makes HW and SW resources at several HPC centres available for small and medium-sized enterprises in the manufacturing area. About 30 targeted experiments have shown the value of such a model to the end users, and have resulted in prototypes for HPC services.

The following paragraphs provide an overview of the current status and the latest trends perceived in the seven SRA technical domains:

· HPC System Architecture and Components

· System Software and Management

· Programming Environment

· Energy and Resiliency

· Balance Compute, I/O and Storage Performance

· Big Data and HPC usage Models

· Mathematics and algorithms for extreme scale HPC systems

This overview allows the Reader to put the research priorities listed and explained in detail in Chapter 5, in the perspective of the overall picture presented here.

To keep up with the ever demanding HPC user needs, the performance of HPC systems at all scales must grow faster than the regular progress of computing technology. A disruptive approach is needed and it requires changing **the architecture of HPC systems,** from chip level to complete systems by integrating new technologies for processing units, memory, storage, networking, etc.

Most significantly, the recent use of HPC accelerators (e.g. GPUs, Many-core CPUs) has resulted in a significant performance boost for some applications. To enlarge the application scope of these HPC accelerators they must be better integrated in the architecture of the HPC nodes or at the system level. To deliver performance, these HPC processing units must access data with a much higher bandwidth than is available with today's DRAM memory technology. This is possible with the introduction of high-bandwidth memory in addition to or in lieu of DRAM. To overcome the limited capacity of these fast memories and to fully leverage their potential, a complete re-engineering and re-architecting of HPC applications might be required.

New non-volatile memory technologies (NVRAM, also referred to as "SCM-storage class memory") appear to be on the near horizon. It is expected that they will offer a much larger capacity than current DRAM, a good fraction of DRAM bandwidth and similar endurance. This opens interesting opportunities for the design of HPC systems. Long-term, the new NVRAM could replace DRAM altogether in compute nodes and become the base for ultrafast storage at the same time. These are technologies which would greatly improve the ability to check-point/restart runs which have failed.

HPC systems are highly parallel. The many thousands of nodes available in an HPC system must be tightly coupled by an efficient network which also integrates storage. The HPC system network must scale with the number and performance of compute nodes and storage devices, requiring more bandwidth but also cutting latencies through better integration. The HPC network should also simplify the application development providing a simple access to the whole system resources.

As mentioned above, virtualisation is making its way into the HCP system design and is essential for a more flexible usage of HPC systems; it will also increase system resiliency through Check-Point/Restart and help to deliver improved fault tolerance and robustness, e.g. by allowing defective nodes to be excluded. Network Virtualisation will be an important part of this. The improved flexibility will also facilitate access to HPC as a cloud resource, enabling new business and usage models through agile, on-demand infrastructures.

Finally, new High Performance Data Analysis (HPDA) applications are emerging alongside with HPC. There seems to be quite of lot of common requirements between these two domains. The new HPDA field could profit significantly from using HPC technology. Conversely, HPC architecture should take into account HPDA specifics and requirements as they emerge.

During the creation of SRA 1 in 2013, important stages of the evolution of **system software and management** building blocks were clearly identified. Unfortunately, it seems that some expected research goals are delayed.

The operating system currently has to take into account new intra-node architectures with many cores, mesh system bus and a heterogeneous memory model. New NUMA-type topologies for memory, I/O, and sockets are proposed. The impact of this node architecture and the intensive use of Multi-threaded programming models increase the need for tools, instrumentation and actuators to solve the scalability and concurrency issues introduced by more complex node architectures. This is seen as a critical research topic.

Sharing of computing resources between some HPC and Big Data domains is increasingly considered. The Big Data "revolution" has put more emphasis on new system software requirements, such as virtualisation support, data centric management, and close-to real-time capabilities. Of course, supercomputer-class storage and I/O subsystems, interconnects and processing capabilities can solve many issues of Big Data, yet progress in the system software field is required here.

Virtualisation, data security at hardware and system level becomes a critical challenge for exascale infrastructure which could address new Big Data use cases. High Performance Data Analytics (HPDA) will explore the

convergence of those two domains. System software will need enhancements to support it.

In HPC, Interconnect research development has been mainly driven by performance and scalability. Network Virtualisation and QOS are still open issues for HPC interconnect fabrics. It could be an interesting challenge to find the balance between performance and virtualisation capabilities.

New interconnect hardware based operation and new memory (NVRAM) capabilities combined with large scale simulation have a strong impact on "in-situ" processing requirements. In-situ processing use cases are mainly concentrated in visualisation, snapshotting or real-time domains.

High bandwidth links offer new capabilities for interactions in heterogeneous nodes enabling new data transfer capabilities and shared memory access. At the system and network level, it will require specific investments in tools and low level APIs to offer access to counters and registers in order to understand global data move issues (congestion, locking, routing trouble shooting, etc.).

The evolution of Cluster Management tools to introduce on-the-fly data analysis and post-mortem data mining started in 2015 but is still incomplete. Event-driven health-checking and introspection is vital for stable operation. Modularity and heterogeneity of the system configuration require the development of an advanced integration model.

Resource management and job scheduling are a critical piece of software for a Big Data framework as well as for HPC supercomputers. Several improvements will be necessary to reach - with a good level of scalability - the multi-criteria objective such as highest allocation flexibility, tightly coupled with application- and software environment deployment. Alternatively, new allocation criteria could be taken into account (e.g. network topologies, interconnect bandwidth, I/O associated workflow, CPU architectures and new memory hierarchy architectures). Significant software evolution will be required in resource and task scheduling to support the run-time environment of new systems.

In the domain of **Programming Environment**, to allow portable performance of applications at extreme scales, we require the development of more productive programming models and environments (including support for domain-specific languages), the easier combination of different programming models, programming models supported throughout the developer tool chain, and the coherent adaptation of applications in co-design with the lower-level environment.

A most promising development is the inclusion of increased intelligence throughout the programming workflow, from the top-level application down to low-level system software. Important examples of how intelligence in the programming environment can facilitate enhanced performance and enhanced productivity in application software development are:

· Application-level abstractions can be used to allow the run-time system (where run-time system is understood here and in the corresponding section of Chapter 5 to encompass programming model support, communication middleware, realisations of APIs and application support libraries) to optimise data layout and data movement, improving performance and energy use and avoiding complex, platform-dependent application software adaptation.

· Hint frameworks within the programming environment enable the application user to make assertions about potential concurrency or locality.

· Application-level abstractions and hint frameworks may be used by the run-time system to enable dynamic resource use (malleability and load balancing, discussed in detail in Chapter 5.3).

· Using techniques from data analytics, performance tools can integrate increased intelligence to provide insight into the behaviour of full production software at scale in order to improve scalability, performance or energy efficiency.

Enhancing the productivity of application software developers for large-and extreme-scale systems is central to a number of trends: domain specific languages (DSLs) should allow the application developer to concentrate on domain-specific problem-solving, whilst allowing lower-level software components to generate optimised code across different architectures; generalisation and extension of past progress in auto- and self-tuning parallel libraries can lead to meta-programming environments and corresponding meta-scheduling support (run-time) systems; interoperability and composability of programming models improves comprehension and provides flexibility for the software developer and should ensure, at the same time, that the system software efficiently exploits the physical resources.

There is a common growth in the deployment of heterogeneous, parallel computing architectures (including, but not limited to, FPGAs, customised processors system-on-chips (SoCs) in embedded and HPC systems, which can benefit the developments towards exascale. There is a significant potential for harnessing developments in the programming environment to facilitate software portability and to benefit from the enlarged developer and user communities.

Finally, while resilience is a topic addressed as a separate topic within the SRA, it is ensuring the resilience of applications executing at scale impacts directly on all sub-themes within the Programming Environment area. Indeed, it could be considered as a key-topic orthogonal to those described above.

In a computer, **electrical energy** is converted to heat by the function of the switching gates and leakage current, producing human interpreted computational results as a side effect. "Energy efficiency" of this process has two meanings: efficient extraction of the emerging heat and reduction of the electrical energy spent while obtaining the computational results by increasing the efficiency of data manipulation and computation itself (Flops/watt).

The focus in the reduction of the spent electrical energy is twofold. Firstly, reduction of losses. Being addressed today by techniques such as the Point-of-Load (PoL) voltage conversion utilising solid-state switched voltage conversion devices. Losses can be reduced by tenfold when combined with adequate cooling for heat extraction. Secondly, reduction of data movement results in huge energy savings, and this is achieved by architectural changes in computers and better organisation of the computational process.

For the computational process – it is a challenge and necessary research topic to render energy spent in computer proportional to the perceived computational advance. The notion of computational advance as a function of energy spent needs better quantification before methods are implemented to make the two proportional.

Another possible energy reduction measure is the acceptance of lower precision result in HPC calculations, trading accuracy for energy usage. Whether this approach can be used will depend on the specific application and algorithm, making this a challenge for energy efficient scientific libraries.

Modern HPC system designs come with increased levels of memory hierarchy and a drastically increased number of hardware processing units running in parallel with the goal to reach exascale performance. With a larger number of processing units, the hardware failure rates increase and **resiliency research** explores ways to minimise impact of failing components on computation. The organisation of a checkpoint is one way to restart the computation after a failure, as well as a way to expand or contract computation on a varying number of processing elements and manage it based on the partial outcome. The challenge is to find a balance between hardware support and software transparency to adopt this technology in a broader manner.

Resiliency is gained if failures can be predicted; the challenge being to determine which data must be collected to make such predictions. Specific failure statistics must be gathered for this research to advance.

The increased efficiency in heat extraction is addressed by replacing air through liquid with higher heat capacity (e.g. water). This presents a tremendous opportunity for computer cooling infrastructures. There are many ways to eliminate air from its current heat-carrying role with industry moving towards the best price/performance technology. The industrial challenge here is the completeness of heat capture into water; a mixture of air-cooled and water-cooled technology diminishes advantages of both. Data centre efficiency is gained when water can be cooled by natural conduction in atmosphere (free hot water cooling) instead of phase-change cooling, requiring extra energy. Water handling infrastructure for "free hot water cooling" is well established, but its usage in computer centres is a novelty that needs a larger number of use cases to support wider adoption. A special research topic is the re-using of the heat instead of dissipating it into the atmosphere. This may be addressed by co-location computer centres with consumers of such heat sources, but more use cases are needed to quantify the perceived advantages.

Storage and I/O performance improvements typically lag behind compute performance especially with the arrival of increasing heterogeneity in the compute subsystem (much greater now than in 2013 SRA 1), with the proliferation of multi-core and many core architectures and the arrival of new technologies in the HPC ecosystem such as GPGPUs. Complexity and parallelism in the compute subsystem has increased by about an order of magnitude in the last few years. However, storage subsystem has still not (in the time

leading to this SRA update) kept up with the needs of applications that have started to exploit the benefits of such heterogeneous and highly parallel compute architectures. There have been research initiatives, but industry roadmaps for storage and I/O are still largely making incremental improvements to incumbent solutions – these will soon start to reach fundamental limitations as they were architected for a different era when compute subsystems and applications were very different. We are starting to see many activities in object storage technologies that offer a flat namespace[13], and aim to address some of the limitations of parallel file systems. Some of the open source objects storage technologies are now starting to gain adoption in HPC. However, even object storage technologies today are far too simplistic in their approach, and are mainly architected for cloud archival workloads, and not for highly performance critical and highly parallel HPC workloads. We anticipate a major architectural revamp of object storage technologies in response to increasing heterogeneity and parallelism.

Storage sub-systems (perhaps driven by advanced object store software) will need to continue to show balanced I/O performance with respect to compute capability and the increasing lag between compute and I/O has to be immediately addressed. Otherwise, we will risk having very powerful compute subsystems, which have to heavily compromise on I/O, workload that is going to be a big detriment for big science and innovation, especially, with Big Data entering the HPC realm.

The use of new solid-state non-volatile devices, starting today with the early use of FLASH technology in HPC deployments, is expected to include other technologies with performance closer to RAM and with longevity suitable for these environments. Exactly when such technologies will reach a maturity, scale and cost suitable for deployment in the massive scale of HPC is still unclear but it is believed to be a necessary component of any exascale solution. Smaller capacity, higher cost but highly performing devices must co-exist in a system with a range of other storage devices (solid state, disk or tape based) and a significant challenge will be created in the management of data across such systems.

There is a need for the storage subsystem to become more "intelligent" in their ability to do computations, and hence, achieve better task sharing between compute and I/O. The portion of energy budget for storage and I/O, which are continuing to creep up because of Big Data I/O, has to be kept under check. There is also a need to maintain or improve systems resiliency with continuously failing I/O and storage components. Some of these storage components may be an entirely new class of NVRAM based storage technologies, residing in multi-layered I/O structures along with existing storage device technologies. The reliability behaviour of these new devices is not very well understood, especially at scale.

Dealing with extreme data is a situation that manifests itself with fast increasing frequency of HPC systems. The root causes stem from two main categories:

Traditional HPC simulations (e.g. fluid dynamics, computational chemistry and physics, cosmology among others) have been benefitting from the tremendous increases in computational capacity of HPC systems and are now using models of unprecedented realism and accuracy. These models represent the real world using trillions of discrete degrees of freedom, which require huge RAM and scale out systems. On the other hand, researchers need to apply advanced and highly complex analytics and processing (including visualisation) on this data, which simply means that off-loading this data to remote platforms is simply not an option. Thus, data analytics needs to take place in-situ, and perhaps in synchrony with tightly coupled synergistic computing platforms (e.g. visualisation engines).

On the other hand, new applications arise as potential HPC clients. Big Data applications, in which data is not generated from some sort of model but rather is collected, accumulated or even streamed, and comes with computational complexity that already sets the computational needs to the petascale or even to the exascale region, even after local pre-processing. Big Data systems, that have been primarily developed for scale out to distributed, non-reliable, resources, are simply too coarse in efficiency and cost effectively to cope with such computational complexity. Thus, HPC solutions, with lots of memory and very fast networks start to appear very appealing and are the next natural step for Big Data users. Indeed, Big Data systems have already started to be influenced from HPC architectures and practices (e.g. multi-threading, parallelism, memory modelling etc.). On the other hand, it is also clear that data will be

[13] For example: Ceph Website www.ceph.com, Accessed November 2014

highly distributed as it originates from distributed sources. Thus, synergy of Big Data systems and tightly coupled HPC systems is foreseen in forming a larger, hybrid, computing resource that combines local and global processing to serve the needs of the new kind of Big Data applications.

Advances in **mathematical methods and algorithms** will be essential in order to produce robust applications that can leverage future high-performance exascale architectures and to reach the goal of improving energy efficiency by two orders of magnitude. Significant efforts in this area are required to allow applications to become even more parallel, scalable and robust, and to optimise for data locality on architectures with deepening and heterogeneous memory hierarchies. New challenges arising from emerging application areas that require both, high-performance computing resources, as well as the ability to manage and process extreme amounts of data should be addressed. The impacts of research on mathematical methods will not be confined to applications, but will equally influence the design of future exascale system software like compilers, communication libraries and programming environments. Notably, the area of optimisation and scheduling will benefit from new mathematically motivated approaches. The interaction with exascale programming environments will be important, as those environments will provide the platform through which new mathematical methods and algorithms will be realised in software applications.

Mathematical methods and algorithms is a new area addressed within the ETP4HPC's SRA. The agenda will focus on research required to achieve extreme scale performance, and in particular, take into account the impact on the commercial stakeholders that include:

· Users of HPC resources that benefit from the development of new methods for solving numerical and extreme data challenges, in particular, in the area of industrial and engineering applications as well as emerging Big Data applications;

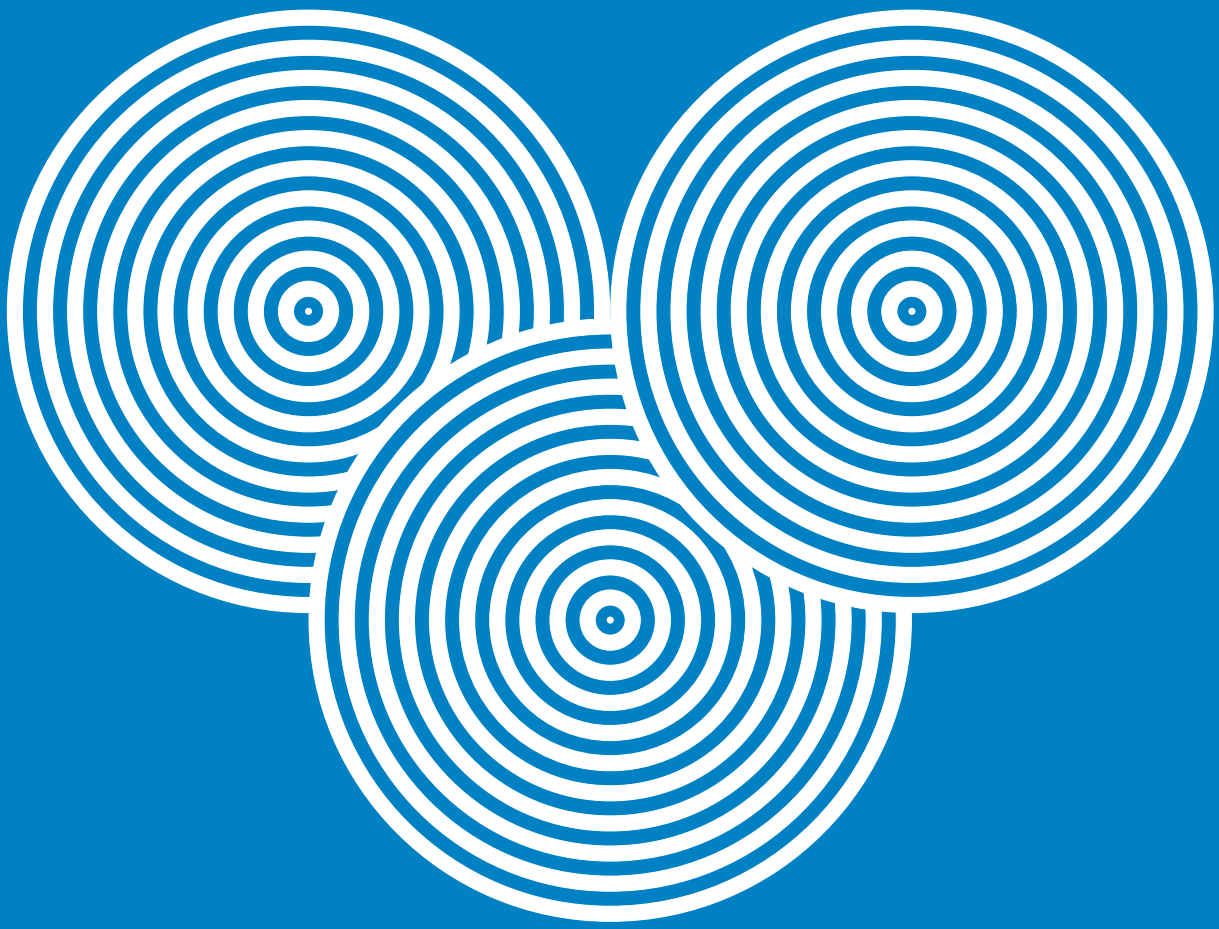
· HPC hardware system vendors who look for algorithms that enable most efficient exploitation of their solutions;

· Software solution and service providers (e.g. ISVs), for which new and robust mathematical methods and algorithms are crucial in providing more competitive software solutions and to enable SaaS services, e.g. based on HPC in Cloud concepts.

This new area of the ETP4HPC SRA and other efforts within the European HPC ecosystem, in particular, the work of the "European eXtreme Data and Computing Initiative" (EXDCI), are interdependent. This makes the cross-coordination of activities crucial. Furthermore, the emerging Centres of Excellence will aggregate significant expertise in mathematical methods and algorithms and will be a key player in driving further research in this field.


Exascale
Productivity

5.



TECHNICAL RESEARCH PRIORITIES

This chapter contains the essence of the updated research roadmap. Compared with the first SRA, it provides substantially more topics spanning a broader area of technical challenges to be mastered. It also provides more insight into the specific research requirements. The research topics are linked to "Research Milestones" listed in Chapter 6. Please also refer to Chapter 4 for a general introduction into and an overview of the required work outlined in this chapter.

The overall goal of the research proposed is defined by the following five objectives:

- **Exascale and extreme scale:** to substantially improve the ability for technology providers in Europe to deliver exascale-capable HPC technologies across the entire HPC stack by 2023. As already mentioned in Chapter 4, besides being in a position to provide exascale compute infrastructure to top-end HPC users in Europe, the HPC industry should be in a position to master the entire spectrum from low to high-end scale sizes, adopting and re-using the technology advancements from the extreme (exa)-scale at the high-end.

- **Ease-of-use:** there are two dimensions of ease-of-use becoming most relevant: being able to "handle", manage and efficiently use extreme scale HPC infrastructures on one side and facilitate the generation and porting of applications on much higher scale compute structures, heterogeneous compute nodes and multi-layer memory subsystems on the other.

- **Efficiency:** Energy efficiency is a key element here, reducing the Total Cost of Ownership (TCO) and is certainly another important goal, especially in the light of extreme scale HPC infrastructures as well as enabling a higher utilisation of all resources.

- **Enabling extreme data & extreme computing:** contribute to the development of a "data-centric" focus across the entire stack and facilitate HPC technology, becoming the underpinning compute infrastructure of choice of high demand Extreme data applications.

- **Broadening of HPC use:** A new term, "democratisation of HPC", has been coined, referring to HPC becoming available by a much broader set of industrial users, in particular SMEs and research organisations. Affordability, down-ward scalability, ease-of-access, and effective programming environments play a key role here.

The following Sections (5.1 – 5.7) describe the areas of European HPC technology in detail. Chapter 6 lists the milestones to be accomplished. These milestones are referred to in the text of Chapter 5 using the following convention, e.g.: [M - AREA CODE – NUMBER].

5.1 HPC SYSTEM ARCHITECTURE AND COMPONENTS

To meet the ever demanding requirements for performance, HPC systems must keep evolving. Simply following technology evolution is not enough as improvements are too slow in meeting targets. Disruptive approaches are needed and the use of new technologies is required. In particular, HPC systems will feature much faster nodes using HPC processing units, faster/larger memory and storage devices, and also, better interconnect. Power efficiency is the main roadblock towards exascale, so all such performance gains should not increase power needs.

5.1.1 Compute nodes — HPC processing units

Recently HPC systems have been deployed using HPC accelerators such as GPUs or Many-cores accelerators. These units are highly parallel and use hundreds/thousands of threads to deliver the performance. These cores run slower than those in generic CPUs, and thus, will not be usable for single thread performance sensitive applications. To improve their efficiency, these HPC processing units must be better integrated into the system architecture than they are as accelerators. The current versions use a generic PCIe interface; it will be improved at the hardware level with coherent interface or as a standalone processor. Even with these improvements, the architecture might be too complex for large HPC applications. Programming Environment should provide a standard interface to hide this complexity; at the hardware level the appropriate features should be available. The simple flat parallelism model with a compute thread corresponding to an MPI task will not scale. A hybrid approach MPI+X (where X could be OpenMP, OpenAcc) or PGAS +X should be enabled. [M-ARCH-1]

5.1.2 Data Access — HW components

Although CPUs and HPC accelerators have improved drastically, the peak performance of the compute nodes and the delivered performance at the applications level have been lagging. The growing gap between theoretical and delivered performance is directly connected to slow improvement in memory speeds; this is often referred to as the "memory wall". To overcome this limitation, fast memory (GDDR, HMC, HBM…) is been embedded with HPC accelerators. These fast memories have a smaller capacity than DRAM memory, and thus, are adding an extra level in the memory hierarchy. To add complexity, these fast memories must be addressed explicitly by the programmer and/or run-time systems, if not, just used as a functionally transparent caching layer that may be of little use. In general, delivering improvements in bandwidth and latency can have major impacts on code efficiency[14]. Thus, the challenge is to find the right balance between future memory bandwidth, latency, size, power consumption and cost. [M-ARCH-2]

Upcoming Non-Volatile Memory (NVRAM) technologies are opening new opportunities for HPC systems. New NVRAM will feature much larger byte-addressable capacities as DRAM: hundreds of GBs vs tens of GBs. Their performance (read or write BW) will be much better than current FLASH based NVRAM, approaching DRAM levels. Furthermore, their endurance should be comparable with DRAM at least in combination with some hidden wear-levelling technology. They could be used in HPC systems, both as main memory and ultra-fast IO, and would completely change the system programming model which distinguishes memory and storage (files), see also Multi-tier storage in Balance Compute, I/O and Storage. [M-ARCH-3]

[14] www.memsys.io/wp-content/uploads/2015/09/p31-radulovic.pdf

5.1.3 HPC Systems Interconnect

HPC systems are composed of a large number of nodes, from 100 for a departmental system to 10,000s for the Top10 systems. The application performance relies on parallelism and depends directly on the efficiency of the interconnect unifying the compute nodes into a single system. The HPC system interconnection network must scale together with the compute nodes and the storage performance. The HPC networks bandwidth is planned to grow from a standard 100 Gb/s in 2016, to 200, then 400 Gb/s in the following years.

With such transmission rates, the possible range for electrical connections will shorten and optical links will become prominent. Underlying network technology improvements are expected in areas such as silicon photonics and photonic switching, which should enable scale performance whilst keeping consumption under control. [M-ARCH-4]

Independent of the BW increase just mentioned, Network efficiency is another topic of interest. Reducing communication latencies will require integrating Network components with compute and storage resources. Other improvements will come from system fabric optimisation, with new topologies, dynamic routing and resource scheduling, in order to avoid network congestion. [M-ARCH-5]

Direct access to the whole system memory should enable new ways to program parallel applications. In particular, the evolution of today's PGAS (Partitioned Global Address Space) programming languages will simplify the implementations of HPC applications. [M-ARCH-6]

5.1.4 Global Energy efficiency

Energy efficiency is the major issue for the design of exascale HPC systems design. Although, this theme is an essential motivation in all system components development (Compute processors, memory, storage, interconnect…), the global system budget must be checked and balanced to provide good system performance for all HPC applications. The successive milestones, as specified in the first version of the SRA (100, 45 and 20kW/Pflops), should be reassessed. As technological difficulties arise, they are being delayed. The first 100kW/Pflops milestone at the end of 2015 is now expected to be met in mid-2016. The intermediate 45kW/PFlops in 2018 is now seen as being very challenging. There could be two intermediate steps before reaching 20KW/Pflops in 2022/2023, with possibly 60kW/Pflops in 2018 and 35kW/Pflops in 2020. [M-ARCH-7]

5.1.5 Virtualisation

Virtualisation is an important tool for improving HPC systems ease-of-use, reliability and security. At the node level, containers can be set-up to facilitate system administration. Containers will provide a flexible way to tailor the run-time environment for each user and application. They will also enforce better security as applications will be insulated from system software and other applications running on the system.

In the network, virtualisation will allow for a good Quality of Service (QoS). It will arbitrate between concurrent users, applications and data flows and their respective priorities. Another important aspect is that it could help improve system resiliency with an easier implementation of Check-Point/Restart at the system level.

It is necessary to develop virtualisation at all levels of the HPC systems and in a coherent way. [M-ARCH-8]

5.1.6 New application domains

High Performance Data Analysis (HPDA) is a good example of new application domains which could benefit from the HPC experience. HPDA applications have emerged in recent years and have been, so far, mostly based on the Map-Reduce distributed algorithm. New classes of problems are now requiring more sophisticated approaches (e.g. graph analysis and real-time analysis), and the necessary tools, HW, SW and development environment, are being investigated. There seems to be a lot in common with HPC as these new algorithms require a much tighter programming environment. HPDA could make use of HPC technology. And HPC architecture could take into account HPDA specificities. [M-BDUM-DIFFUSIVE-1]

5.1.7
New disruptive HPC architectures

Besides HPC accelerators (GPUs, Many-cores…), other types of processing elements such as FPGAs, DSPs… have also been proposed for various dedicated applications. They seem well suited for in-flight data processing. However, the specificities of their programming model have so far limited their adoption. The situation must be reassessed in light of the new developments underway, which integrate these devices more tightly with the rest of the system's resources. Ultimately, completely new architectures could be proposed for HPC systems in general or for important subsets of HPC applications. For example, Processor in Memory (PiM) has been suggested several times as a way to break the memory wall. A super-efficient interconnection network using photonics, for example, could completely change the way the system's resources (processors/memory/storage) are organised. [M-ARCH-10]

5.2 SYSTEM SOFTWARE AND MANAGEMENT

5.2.1 Operating system

The purpose of an operating system is to manage efficiently the resources of a computer system and to bridge the gap between the physical resources that make up such a system and the abstract view of a system, which run-time systems and programming models require. For HPC systems, operating systems typically operate on each node independently, with co-ordination executed through higher layers of Cluster operating or management systems. Given the rapid change in resources and programming models, a basic operating system must be defined for the exascale community. A suitable abstraction and a common set of APIs implementing it will be defined and can be used by a run-time system to support the management of resources, including adaptive and dynamic management policies that identify and react to load imbalances and the intermittent loss of resources.

The node architectures in supercomputers are becoming more and more complex. Increasing levels of parallelism in multi- and many-core chips, complex memory hierarchies and emerging heterogeneity of computational resources, coupled with energy and memory constraints, force a re-evaluation of our approaches towards operating systems and run-time environments.

To enable optimal use of emerging HPC architectures, evolutionary improvement of existing operating systems should be explored.

· Kernel scheduling policies must be designed and implemented which solve multithread contention [M-SYS-OS-1].

· New memory management policies and libraries for complex memory hierarchies need to be developed which enable applications to use these [M-SYS-OS-3].

· Offload programming model support is needed for optimal use of heterogeneous architectures [M-SYS-OS-5].

· To avoid limitations in scalability, reliability issues and kernel overhead OS decomposition [M-SYS-OS-6], container or virtualisation usage [M-SYS-OS-4] should be explored.

The great challenge for future operating systems is mainly on interoperability with future run-time systems, middleware, and tools. The role of OS and run-time system interaction is critical to minimise power usage, improve fault tolerance and increase performance. The interfaces between the layers will have to be studied and clearly specified. Today, experimentation is spread in many small and incomplete research projects.

The low-level APIs exposed by the OS and the various services implemented by the run-time system must be seen in the context of their use by programming models and applications. The complex interactions between components do require a holistic view in order to achieve best scalability and performance – it is not possible to attain this by defining APIs and optimising components in isolation.

System security support is a horizontal critical topic which needs to be accommodated at each significant hardware or software evolution step. Today, most OS levels are mainly based on SE Linux, and it will be mandatory to investigate how to integrate existing and emerging security mechanisms with OSs suitable for HPC. This should take into account the HPC-specific security requirements [M-SYS-OS-7].

Research topics:

· Offer a simple but efficient abstraction of the hardware to enhance the programming models,

· Define mechanisms to accommodate extended APIs and allow passing of hints from applications to the OS/RTS, e.g. for I/O and management of data and threads,

· Improve hardware abstraction for compilers and support new compiler features, e.g. optimisation and parallelism extraction,

· Extend run-time systems by developing mechanisms for passing hints from applications and new algorithms for better managing data locality, scheduling, I/O, coherency, etc.,

· Provide programming model developers with an efficient API for run-time services,

· Revisit APIs to support inter- and intra-node communication, thread management, and explicit management of the memory hierarchy provided by the entire system,

· APIs to support energy monitoring, management and resiliency will be essential (see also Chapter 5.4 and milestones M-ENR-MS-1 and M-ENR-MS-2 in Chapter 6.4),

· Fault tolerant/masking strategies for collective OS services are indispensable,

· Remove scalability limitations (such as OS Jitter) inherent in current operating systems by de-constructing their implementations, and leveraging concepts such as dedicating system resources to application and various OS functions,

· Hypervisor, Virtualisation and container support at HW and system level;

· Offload programming support at OS level.

Expected results:

Capabilities to build different operating system instances and run-time systems, adapted to the needs of applications and programming models with respect to emerging exascale hardware, e.g. heterogeneous compute elements, complex memory hierarchies, use of accelerator embedded cores, etc.

5.2.2 Interconnect management

Interconnect topology management and routing is expected to react on message requests from applications or I/O in real-time and create optimal traffic patterns. First, this challenge has an impact on routing algorithms which will need to become dynamic. Secondly, optimising and scaling of application requests will require leveraging low-level communication acceleration based on hardware [M-SYS-IC-1]. A new borderline between low-level system control and application execution in user space on top of interconnect interfaces needs to be specified.

Identifying and fixing congestion situations and trouble-shooting errors [M-SYS-IC-2] of these networks is a real challenge that will require new management solutions to be developed.

At the interconnect adapter-level, driver and low-level interfaces will evolve depending on hardware technology (in particular closer integration of adapters and nodes) and on new programming model needs. Here, a tight cooperation between system software developers, system architects, and programming model designers will be necessary to integrate new low-level protocol capabilities and define a well-adapted API for higher levels of programming software.

The support of a virtualisation environment over the entire HPC infrastructure is a new issue. Low level support of virtualisation, such as the definition of virtualised networks is currently not fully supported.

Exascale interconnects are mainly performance driven, but it could be interesting to explore how to keep performance capabilities and develop compliance with virtualisation requirements e.g. for QOS routing or security purposes [M-SYS-IC-3].

· Adaptive and dynamic routing algorithms,

· OS-bypass and new low-level hardware interface API [M-SYS-IC-1],

· Congestion control and concurrent access regulation [M-SYS-IC-2],

· Scalable interconnect management tools,

· Intra-node and inter-node network low-level operation and addressing model;

· Network virtualisation support or compliance [M-SYS-IC-3].

5.2.3 *Cluster management software*

Maintaining a clear view of the configuration and status of HPC systems is indispensable as they are exceedingly complex and susceptible to small perturbations that can have an extraordinary impact on performance, consistency and usability.

Today's ability to manage and scale to tens of thousands of nodes is solved by innovative cluster management frameworks. The new challenge of cluster management is driven much more by the multiplicity of applications execution contexts and by the flexibility required for I/O and data space management. Virtualisation and containerisations become a way to define specific and secure execution contexts. Cluster management, whatever the size of supercomputer, needs to integrate the management of this new environment.

Research topics:

· Flexible execution context configuration and management: The variety of hardware, programming environment, and applications' requirements will demand new tools to compose, integrate and store system images, containers definitions or virtual machines [M-SYS-CL-1],

· "Real-time" failure diagnostics and prescriptive maintenance based on Big Data analytics technics [M-SYS-CL-2],

· System security: system management also has to ensure system integrity, a major factor of which is system security [M-SYS-CL-3],

· Performance counters: design and implement performance counter APIs that provide scalable access to sufficiently detailed and up-to-date information for exascale performance analysis tools (see also M-ENR-MS-1 in Chapter 6.4)

Expected results:

A complete and consistent framework, which offers strong integration and configuration capabilities and powerful supervision tools.

5.2.4 *Resource management and job scheduling*

Resource management and job scheduling are a critical function for efficient use of Big Data infrastructures and HPC systems. New mechanisms or improvements of existing ones will be necessary to reach: a sufficient level of scalability, the objectives of efficient utilisation of system resources and energy, reliable fulfilment of application SLAs; and, highest allocation flexibility. These have to be tightly coupled with applications and software environment deployment requirements (such as co-scheduling of coupled applications) and should take new allocation criteria into account, whilst collecting fine-grained information on job execution context.

Research topics:

Centralised control will not be able, from now on, to scale up and manage the complexity of scheduling and orchestrating modern large-scale application workflows.

Resources heterogeneity and the multiplicity of resource-selection criteria such as power consumption, or data locality are real challenges for an exascale resources manager.

With the extreme data focus, the role of data handling and heterogeneity is more and more critical on supercomputers (see M-SYS-RM-2). Data locality becomes a critical parameter in allocation criteria so that data-aware scheduling is necessary (see also Chapter 5.3). Scheduling needs to be data-aware but also tightly coupled with data movement from storage to memory.

During execution, the interaction between the job resource allocator and the run-time will be a key factor for performance and reliability of application execution. These dynamic capabilities require evolutionary improvements of job scheduling and resource manager.

With uptake of virtualisation and containerisation, the application's execution context becomes specific to each task. Next generation resource schedulers will need to support efficient deployment of this specific software environment prior to the execution of applications. There is also the requirement for elasticity (adding/removing resources to/from running jobs), migration and offload to a cloud.

So, new workflow scheduling models and resource allocation policies need to be architected and developed.

Multi-objective adaptive scheduling in a dynamic environment [M-SYS-RM2] will probably be a way to deal with most of the above issues. However, the complexity of the scheduling will be enormous and its scalability will be a tough challenge.

Scheduling is also impacted by cross-layer issues such as energy efficiency, resiliency and security. In particular, since energy usage and power control mechanisms are themselves highly dynamic and application-dependent, adaptive scheduling has a high potential to help with the energy challenge (see also Chapter 5.4 and M-ENR-MS-2 in Chapter 6.4).

For a highly interactive use of HPC systems, job pre-emption is desirable. In this way users could get access to the resources when they need them, rather than when the scheduler provides them. To be of value, this feature will require efficiency in suspend-and-resume job support.

Exascale systems require a failure-tolerant application environment, implementing at least effective checkpoint/restart features [M-SYS-RM-3 and M-ENR-FT-6 and M-ENR-AR-4]. The system software layers will need to provide the required basic mechanisms for storing, managing and retrieving state, in close collaboration with the I/O subsystem. Extensions of such mechanisms could support the emerging use case of highly interactive access to HPC resources.

Expected results:

Dynamic resource allocation capabilities and powerful multi-criteria scheduling algorithms.

5.2.5 Visualisation software

Current and future supercomputers enable numerical simulations of unprecedented fidelity and scale. The petabytes and soon exabytes of data produced by these simulations are increasingly difficult to analyse. Powerful tools are, therefore, required to analyse and interactively visualise data without - mainly for energy efficiency reasons - moving it over the HPC interconnect fabric or copying it in memory.

Specific requirements are:

· Visualisation performance must scale directly with the simulation performance, allowing for simulations at unprecedented scale and with previously unattainable visualisation accuracy,

· Location of simulation and visualisation minimises data movement and enables efficient workflow execution, reduced time to solution and overall system power efficiency,

· Flexibility of use of compute and visualisation resources supports dynamic scheduling,

· Global shared access infrastructure (i.e. network, storage, resource scheduling) is provided;

· On-the-fly analysis, often referred to as “in-situ visualisation” [M-SYS-Vis-1] or “in-situ analysis” is supported in an efficient way, enabling the emerging interactive supercomputing use case.

Today, visualisation and analytics in HPC is dominated by the open source tools Paraview and VisIt, both funded by the US DOE. Both were started more than a decade ago. Whether the requirements for visualisation can be better achieved by evolving these or by creating entirely new frameworks will remain to be seen.

Research topics:

The exascale time frame and the confluence of HPC with Big Data needs, provide a good opportunity to design new visualisation tools exploiting the node architecture and memory hierarchy of present and future supercomputer architectures, in combination with the use patterns of the

different application domains. These tools would then need to support emerging usage scenarios, such as "interactive supercomputing" and co-visualisation.

One of the challenges when rendering visualisations in parallel is the compositing phase, which merges the independently rendered partial frames into a final image, especially when running at high concurrency and rendering large frames compositing is one of the limiting factors in achieving interactive frame-rates. Research in compositing algorithms and efficient pipelining is, therefore, needed to fully exploit the rendering performance of future supercomputing systems for interactive visualisations [M-SYS-VIS-2].

With increasing amounts of data, advanced volume visualisation techniques become critical. Rather than treating volume visualisation as an add-on, a modern tool should be designed, from the ground up, with volume visualisation techniques in mind. In addition, modern hardware enables high quality visuals via real-time ray-tracing, providing improved user perception of the visualised data. Ray-tracing capabilities should, therefore, be at the core of a new tool as well [M-SYS-VIS-3].

In order to solve the I/O bottleneck of large scale simulations with in-situ techniques, the visualisation application needs to be tightly coupled with the simulation code, often requiring instrumentation of the simulation code with function calls to the visualisation application. While the widely used visualisation tools provide such functionality, the application interfaces are cumbersome to use and lack a common standard. Simplifying the task of instrumenting massively parallel applications for in-situ visualisation, and is therefore, an area in extreme demand for active research.

Modern tools have also to embed the support for features like:

- remote visualisation,
- hardware encoding/decoding,
- encryption/decryption capabilities,
- real-time analysis/visualisation,
- infrastructure components (e.g. firewalls and resource management systems);
- workflow optimisations.

With the convergence of architectures for HPC and Big Data, data analysis and visualisation tools need to be able to cope with very high dimensional data, graphs and other complex data topologies [M-SYS-VIS-4]. It will also be necessary to analyse and visualise complex data/graph structures originating from unstructured environments. Rather than trying to retrofit existing HPC visualisation software for the new challenges, a unique opportunity emerges to develop new tools designed, from the ground up, for modern, heterogeneous architectures and covering a broad range of data models as encountered in HPC and Big Data.

Such a tool can also be designed from the outset with an in-situ workflow in mind, simplifying the interaction between simulation and visualisation application. This can help to gain more traction for the in-situ workflow with a broader user base, helping to reduce the pressure on the file system, and ultimately, leading to an accelerated scientific discovery process.

5.3 PROGRAMMING ENVIRONMENT

The development of efficient and productive programming models and environments, including interoperability, remains a priority. Increased intelligence throughout the programming environment, including run-time management of data layout and movement, is considered to be most important. In particular, this would be necessary to effectively support increased system architecture complexity, heterogeneous systems, and increasing architectural variability between systems. The development of programming environments must be performed coherently and in unison with the adaptation of applications; importance being placed on ensuring that a broad enough range of applications is addressed. The support for legacy applications is included within that scope, as is the need to meet the requirements of application workflows and coupled applications (e.g. for multi-scale or multi-physics co-simulation).

An important aspect linked to the run-time systems is the need for separation of core algorithmic issues from run-time concerns (again an issue for co-design between the software developers at multiple levels in the application to system software hierarchy).

While major progress has already been made in dealing with resiliency for hardware components, significant advances are needed to provide a real fault-tolerant programming environment to application developers and a fault tolerance must be handled at all levels: a DSL should include constructs (e.g. hints) to help a run-time take decisions in case of a fault; programming languages and libraries should embed fault-tolerance mechanisms; and, tools are required to facilitate the validation of application behaviour in the case of failure.

5.3.1 Intelligent Data Placement

Computing is cheap, while data movement dominates performance and energy. Thus, data layout and data movement should be optimised. If suitable abstractions are deployed at the application level, then these complex and platform-dependent optimisations can be facilitated by the run-time system. Programming tool intelligence should be based on cost models that propagate (throughout the software stacks) information about energy, load-balancing and communication requirements.

The priority research topics for this are as follows.

- **Data transfers:** handling of implementation details and low-level APIs, automatic caching, and prefetching.
- **Locality:** locality-aware scheduling and data reuse.
- **Sub-setting:** accurately computing dependencies when later work depends on a subset of the data structure (sub-array, subtree, etc.) and minimising data transfers.
- **Optimising in-memory layout** (array-of-structures vs structure-of-arrays, row/column major, blocking, packing, other cache-aware transformations).

The application should specify local needs rather than implementation, and performance should be dominated more by bandwidth than latency.

Related milestones are: M-PROG-RT-3 and also M-PROG-API-2, -API-4, -LIB-1, -RT-1, -RT-2, -RT-4, -RT-5

5.3.2 Malleability and dynamic load balancing

Malleability is the ability of an application to adapt to changing resource availability while it is running; e.g. as nodes become available/unavailable. This allows the system software to release nodes to an urgent job or make use of temporarily idle nodes. Dynamic load balancing is the use of malleability to reassign resources between applications, controlled by a run-time system. Research should target programming models that enable malleability in a natural way. This can lead to dynamic load balancing between

applications, controlled by the run-time system and enabled by the job scheduler.

This topic has a strong relationship with the system software area of the SRA, requiring support in the job scheduler and benefitting from OS support (hand-off scheduling and fast context switching).

Related milestones are: M-PROG-RT-2 and also M-PROG-RT-1, -RT-4, -RT-5

5.3.3 Higher-level Programming environments

Progress already made in auto- and self-tuning parallel libraries needs to be extended and generalised (from specific application fields) to meta-programming environments and corresponding meta-scheduling support (run-time) systems. As the new algorithms theme addressed in Section 7.7 develops, there will be a need for supporting programming environments – here co-design between applications-oriented and programming environment/paradigm, and also, DSL developers will be essential – and the development of mini-applications. For the former, there is significant potential in (re-)investigation of functional/high-level programming. For the latter, there is a need to go beyond the abstraction of current algorithms to the compact specification of the newly developed computational methodologies.

Related milestones are: M-PROG-API-7 and also M-PROG-API-1, -LIB-3

5.3.4 Programming model interoperability and composability

Interoperability refers to performance tools, debuggers, verification tools, etc. that understand the programming model abstractions. It also refers to the interface(s) with scripting languages and workflow tools, couplers, and the use of persistent objects. Composability is the ability to build new programming models out of existing programming model elements, leading to hybrid programming models (MPI+X, PGAS+X, MPI+X+Y). The various "levels" should cooperate among themselves and with the system software to efficiently exploit the shared physical resources. Linked to the interoperability aspect, performance tools, debuggers, verification tools should present information in terms of the programming model used by the programmer. Resolution of interoperability semantics will become a key aspect (as heterogeneity of system – and system use – grows). Both interoperability and composability aspects are particularly important for PGAS and task-level developments, wherein Europe demonstrates strength.

Related milestones are: M-PROG-API-3, -DC-4, -PT-2 and also -API-6, -LIB-2

5.3.5 Domain-specific language frameworks

A domain-specific language (DSL) is tailored for problems in one domain; e.g. SQL, Matlab and R, providing rapid prototyping and a separation of concerns between domain knowledge on the one hand, and numerical analysis and parallelisation/optimisation on the other. Similarly, DSLs (or domain-specific extensions to existing languages), tailored to a specific scientific domain or class of application, potentially provide reusable syntax and components to enhance programmer productivity, whilst allowing lower-level software components to generate optimised code across different architectures. A DSL allows compiler and run-time system transformations for performance (e.g. parallelisation) and correctness (e.g. numerical stability and preconditioning). DSL frameworks reduce the cost of producing an optimising compiler, run-time system, performance tools and debugger for a specific DSL, by building on a general-purpose programming model and run-time supporting data management, scheduling, etc.

DSL developments should try to identify common core architectural components and common, re-usable features. Current research should explore a variety of approaches, but in the long term it is hoped that a proliferation of DSLs can be avoided through the development of a small number of generic DSLs (some examples: the use of stencil-based approaches in many applications, algorithmic skeletons and high-level paradigms like BSP) and integration into general purpose programming languages and environments.

The interaction with run-time system developments is crucial as it is needed to support both the application and the run-time system in enabling resiliency.

Related milestone: M-PROG-API-5

5.3.6
Information transfer programming model/API to run-time

The information transfer (from Application Programming Interface through to the run-time system) needs to be multi-layered and support flexible run-time hierarchies occurring in dynamic and heterogeneous systems. A specific example would be hint frameworks, in order to allow the programmer to annotate potential concurrency, locality, significance of the computation, sensitivity to device reliability, etc., including in legacy codes, where research priorities range from the definition of semantics to techniques to exploit the information.

Related milestones are: M-PROG-API-2, -API-4, -LIB-1, -LIB-2, -RT-1, -RT3

5.3.7
Performance analytics

There is an increasing need for intelligent performance/system energy tools using techniques from data mining, clustering and structure detection. The goal is to obtain real insights on production codes, where the number of threads and quantity of data makes it impossible to understand a traditional trace. The gap between the tool's output and the necessary source code changes should be reduced, by mapping to the source code structure. Scalability should be improved, including through data reduction: summarised data, reduced precision, spectral analysis, sampling, profiling and filtering. Application scalability prediction is important, including mathematical models and trace-based tools.

Related milestones are: M-PROG-PT-4 and also M-PROG-API-2, -API-4, -RT-2, -PT-1, -PT-2, -PT-5

5.3.8
Debugging and program correctness

Debugger technology is needed which can support applications that have been developed on and for heterogeneous computing systems, using both current and non-conventional programming models, languages and APIs, and deployed on the full range of target systems towards exascale. A key topic with increasing importance (as application complexity grows) is program/software verification. Advances have been made in parallel program verification/verified programming, including models that account for cost measures. There should now be concerted efforts to apply to specific and innovative European know-how.

Related milestones are: M-PROG-DC-1, -DC-2, -DC-3, -DC-4

5.4 ENERGY AND RESILIENCY

As pointed out already in Chapter 4, improving energy efficiency by several orders of magnitude is a prerequisite on the path to exascale. The research which is necessary in the context of 'energy' needs to target four different dimensions: (1) reducing the "innate" electrical power consumption of the next generation HPC system's hardware infrastructure, (2) more efficiently extracting the heat generated, (3) reducing the data center overall power consumption and (4) possibly re-using the heat extracted. To improve the innate energy efficiency by orders of magnitude, significant advances by the compute chip industry in the underlying CMOS processes, or even an adoption of an altogether different implementation of "compute devices" are required. This chapter focuses on domains (2) – (4). Based on the HPC Challenges we have devised the following research areas:

5.4.1 Quantification of computational advance and energy

Most common computational metric in HPC is Flop/s. This may not reflect all computations; some application dependent measure is needed to quantify advances per job, subroutine or block of code. A way must be found to measure energy spent for the same computational metric. Some techniques exist already, such as the data reported by the power supply. The challenge here is the delayed response from the power line to an increased load on the processor side. Reliable statistics for application dependent measurement must be presented for processor and accelerated computation. This may require development of additional sensors and measurement methods.

The goal of this research is to deliver application specific measure of computation done per unit time and energy spent. Requirements for specific sensors may emerge as a demand for the IT industry. Note that this consideration also includes data movement and storage activities, as proposed in all BDUM-METRICS-(1-3) milestones.

The expected result is a library or reporting system that provides the combined metric to system/user.

Related milestone: M-ENR-MS-1

5.4.2 Methods to manage energy used for computational advance

This research area covers methods to manage the energy spent in the computer for the most efficient computational advance. Based on the results gained by the previous research area (see M-ENT-MS-1), propose methods to set energy consumption to be proportional to the computational advance. This may be an extension to the first research topic, however, taken separately, because methods to deal with energy management may require a different approach. Measurement is for past events; here we require a way to predict the optimised energy strategy for a most efficient computation with respect to user policy.

The goal of this research is to obtain a prediction strategy and energy consumption management methods based on the application specific computational advance. Note this consideration may have specific bearing on the algorithmic research, such as M-ALG-6.

The expected result is a library and/or a system to set the energy needed for calculation of a job, subroutine or block of code in the most efficient way with respect to user policy.

Related milestone: M-ENR-MS-2

5.4.3 Throughput-efficiency increase

Use of idle time to increase efficiency of computer throughput: In scheduling instructions in processor, idle time is an inevitable consequence of resource constraints. In parallel execution, synchronisation points amass idle time. Minimisation of idle time is known as the "Slack Allocation Problem" and is widely researched. The research area M-ENR-MS-2 asks for methods to lower energy spent on programs containing slack. This research topic asks for methods to use slack to schedule additional work. This has been researched in a different context for virus and parasitic computing, but we ask for a controlled environment. The overall computer throughput will increase.

The goal of this research topic is to demonstrate increased throughput of computer by execution of additional code using idle time of programs. Implementation of this paradigm will increase the overall efficiency of the computer system, as more jobs will be able to pass per unit of time. Research in this area may be helped by development of work scheduling methods, see M-SYS-IC-1.

The expected result is a description of methods and practical demonstration of a system to run additional code without slowing down execution of other programs.

Related milestone: M-ENR-MS-3

5.4.4 Use the new levels of memory hierarchy to increase resiliency

Discover methods to use the new levels of memory hierarchy to increase resiliency of computation: New memory hierarchy may be composed of a layer with non-volatile characteristic (Flash memory). This memory may be used to store the checkpoint for the computation, but likely, there are even more powerful ways to use it to increase the resiliency.

The goal of this research is to advise methods to use the non-volatile memory to store state that may be used to recover from a failure. Note that this research increases the corresponding architectural development, such as M-ARCH-3.

The expected result is a description of methods and/or demonstration implementation that can survive single component failures of various kinds and is equal to, or more efficient than simple checkpoint.

Related milestone: M-ENR-AR-4

5.4.5 Collection and Analysis of statistics related to failures

Collection and Analysis of statistics related to failures in computers: A fast database of log statistics exists, but these are not immediately related to failures. Classification of failures is needed and events prior to these failures need to be explored for patterns that may correlate to failures. Obvious data to look for are the temperatures and voltages on the hardware boards, but other measurements may be needed for better prediction strategies.

Such work has been done specifically for memories and disks, where failures can be easily detected. A better and more complete classification is needed to account for all failures, including software errors that may be correlated to hardware events in the system. Due to unpredictable and rare occasions of some failure patterns, this may be a multi-year effort.

The goal of this research is discovery of hardware events and patterns of events correlated to failures in the system.

The expected result is lists of patterns that probably lead to failures and methods to collect the related measurement and statistics proving the case.

Related milestone: M-ENR-FT-5

5.4.6 Prediction of failures and fault prediction algorithms

Prediction of failures in computers and fault prediction algorithms: This is an extension of the research topic M-ENR-FT-5 but is treated separately, as it requires a different methodology. This is mostly a computational/mathematical work needed to develop prediction algorithms based on statistically significant patterns for failures. Such work has been done before, but in absence of statistically significant patterns, the impact is low.

The goal of this research is development of prediction algorithms. Note that deduction of hazardous pattern may require methods of High-Performance Data Analytics. These methods will be called for in different parts of the SRAs and the machine must be optimised for this activity.

The expected result is a system that predicts probability of a specific failure occurrence within a given time frame,, if certain events or patterns of events are measured in the system. This may be implemented as system daemon with notification capability.

Related milestone: M-ENR-FT-6

5.4.7 Prevent Application crash on unreliable system

Based on the availability of Failure Prediction (as explored in milestone M-ENR-FT-6) and possible tools for managing application, in a dynamic environment [M-PROG-RT-2], this milestone will establish robustness of applications. Other means of establishing application survival probability even if M-ENR-FT-6 target is not reachable may be established by this research milestone. In addition, this milestone may be researched in a different context of virtualisation and architectural support for the virtualisation (cf. M-ARCH-8).

The expected result is a system that demonstrates survival of applications as resources that the application uses once they become unavailable.

Related milestone: M-ENR-FT-10

5.4.8 Quantification of savings from trade between energy and accuracy

Quantification of savings from trade between energy and accuracy: reduction of accuracy of the computation, e.g. usage of single precision computation instead of the double precision leads to energy savings as less data needs to be transferred. This may be explored in algorithms that increase the accuracy iteratively, or in other ways, tread accuracy against energy usage. The quantification of the energy saving may come with methods developed by the research topic M-ENR-MS-1.

The goal of this research is demonstration of algorithms' energy saving ability when less accuracy is required for the result of the computation. This may be highly applicable and/or algorithm dependent and part of the results in this area may come from the research towards milestone M-ALG-8.

Expected results are measurement results for several significant applications and/or computational kernels. Implementation of this strategy specifically for scientific libraries is a proof case.

Related milestone: M-ENR-AR-7

5.4.9 Power efficient numerical libraries

Power efficient numerical libraries: An obvious extension of the research area M-ENR-MS-1 and M-ENR-MS-2 is the use of these techniques for energy efficient scientific libraries. One of the requirements of this research is bit-reproducibility of the results under various run conditions. It is expected that either energy or accuracy can be specified as termination criterion for the algorithm.

The goal of this research is that it advances towards highly scalable scientific libraries that can deliver results of specified accuracy with quantifiable and efficient energy usage. Much of this research may rely on the results obtained from algorithms development, such as M-ALG-8 and programming support libraries M-PROG-LIB-3.

Expected results are libraries for linear algebra, FFT and others, demonstrating the desired characteristic.

Related milestone: M-ENR-AR-8

5.4.10 Highly efficient HPC installation

Demonstration of a sizable HPC installation with efficiently cooled components, while energy losses are small and under control. This is a wish list for work carried out in this research for at least a one-year period of running such machinery:

· Demonstration of the "free-cooling" all year round and energy reuse.

· Measurement and demonstration of the common metric like Power Usage Effectiveness (PUE), Effectiveness of computational energy (TUE[15]).

· Collection of energy usage data for job statistics and energy saving measures for specific jobs

· Methods to reuse heat produced in such installation.

· Study this usage-model for Total Cost of Ownership (TCO) and exploitation characteristics of such installation, taking infrastructure into account.

[15] www.eetd.lbl.gov/sites/all/files/isc13_tuepaper.pdf

· Work related to other Chapters demonstrating progress towards exascale systems.

· Work related to Research Areas that have not been resolved at the time of the installation.

This computer installation should demonstrate PUE~1.05 (i.e. 5% infrastructure load) and TUE~1.1 (i.e. <5% power conversion losses) which would mark a significant improvement in Point of Load (PoL) efficiency.

There are successful installations that have significant water cooling component and energy reuse, most notably LRZ in Germany. There are no installations that are entirely water cooled, i.e. excluding any air movement. If air cooling is excluded, it will provide an opportunity to measure a different Total Cost of Ownership model, needed for the comparison and study of the usage characteristics towards exascale systems.

The goal of this research area is to measure usage characteristics of computer installation without any air movement requirement, with free cooling and energy reuse.

Expected results are measurements and reports related to energy efficiency and TCO.

Related milestone: M-ENR-MS-9

5.5 BALANCE COMPUTE, I/O AND STORAGE PERFORMANCE

Balanced I/O at extreme scale: The focus for balanced I/O should be for the storage and I/O subsystem to keep up with the increasing parallelism and heterogeneity of the compute subsystem – in terms of dealing with both, checkpoint I/O and application I/O. It should be noted that the applications will start to generate a lot more data, and will need to deal with increasing 'data analysis' (heavy reading of often randomly accessed data) workloads – this differs from traditional HPC applications that have existed in the past. The I/O performance requirement needs will continue to dramatically increase scaling to handle billions of parallel processes doing simultaneous I/O, and I/O architectures need to be designed to handle very graceful scaling with core counts. Stop gap solutions are not going to work here, if we are to continue to maintain balanced I/O with compute over longer time scales.

Storage architectures incorporating new device hierarchies and exploiting storage abstractions such as advanced object stores will be important in the continued need to Balance I/O performance with computational performance and scale.

5.5.1 Extreme data processing

Traditional high end HPC storage with performance specified to handle the needs for checkpoint data for systems reliability. There are now new requirements in evidence stemming from the need to generate science out of Big Data. Within HPC, Big Data requirements are driven by the needs of vast "instruments" such as the Square Kilometre Array[16] which will become active in the next few years. Big Data analytics and HPC driven simulations were treated as completely separate infrastructures. However, this is changing as Big Data analysis decisions will now need to be fed back into running simulations, sometimes even in 'real-time' control loops. This imposes new needs on I/O system architectures. A "single" system has to do both simulations and extreme data processing, changing the workload profiles and the storage and I/O subsystems, which can potentially take-up some of the load of doing the data processing since data resides locally – avoiding the need to move data back and forth between the compute and storage subsystems. [M-BIO-5, M-BIO-9]

5.5.2 Active Storage / On-the-fly / in-transit data manipulation / In-Situ processing

As an extension to the above argument, storage systems need to develop new capabilities in being able to process data in-situ with on-going computations in the compute sub-system. Such "active storage" capabilities can be introduced anywhere in the I/O stack (network, storage device, NVM storage layer such as "burst buffers", etc.) or 'on-the-fly' between storage tiers. Active storage at the device level existed as research in the early 2000s[17], but did not establish itself through a lack of extreme data processing use cases and inflexible software architecture. This is expected to change through research in this area and storage should incorporate the ability to perform specific user defined tasks and/or full-fledged applications.

Deep I/O hierarchies, combined with advances in networking infrastructures will offer improved opportunities for in-transit data manipulations and transformations. For example, some memory and limited processing capability are expected to be available in next generation interconnection networks, which can be exploited for on-the-fly data manipulation, in parallel with in-core computations. Some interesting possibilities are data selection, data compression or de-duplication and applying simple transformations on data.

The ability for elements of the data manipulation, especially for the 'Big Data' applications to be performed in storage elements of a widely distributed system or close to the data sources potentially remote from HPC centres, should be considered to effectively aggregate their processing power. Related milestone: M-BIO-6

[16] C. Broekema et al., 'Exascale High performance Computing in the Square Kilometer Array', AstroHPC'12, June 2012

[17] Erik Riedel, 'Active Disks - Remote Execution for Network-Attached Storage', Technical Report CMU-CS-99-177, Doctoral Dissertation, Pittsburgh, PA, November 1999

5.5.3 Energy reduction

Energy is a major issue in computer architecture and technology, and equally if data storage is not to become a much larger portion of the energy needs of systems as we process and manipulate extremely large data sets. Thus, this must be addressed in both the data storage and the I/O subsystem. The energy required to move data may be orders of magnitude higher than doing computation[18]. This has not been such a significant problem to date, as traditional HPC is typically checkpoint I/O based (with only occasional data transfers due to checkpoints). This scenario is going to significantly change with the needs of Big Data processing within HPC. Avoiding the movement of data in the compute + storage subsystem is extremely valuable if we are to achieve any realistic energy reduction. This can be addressed through in-situ processing of data sets in the storage subsystem, as well as in-situ processing in memory as addressed in the Chapters 5.1, 5.2 and 5.4.

5.5.4 Resiliency and Reliability

Infrastructure faults and failure of storage and I/O components (not just in the compute) will be a norm as we approach exascale simply as a consequence of the number of components involved. This means that infrastructures need to be dynamically adapted to continuously occurring failures, to keep application downtime to the very minimum. There is, hence, research needed in advanced high availability mechanisms within the infrastructure, minimising overheads of detection, communication and recovery of faults.

For dealing with application reliability, we need to address adaptive check-pointing that provides system resiliency based on the reliability "metrics" of the underlying storage infrastructure. This should, however, be counterbalanced by the need to reduce checkpoints as much as possible, reducing energy footprint and application downtime. Hierarchical checkpoint schemes, as an advancement of burst buffer approaches[19], need to be investigated.

See also Energy and Resiliency (Chapter 5.4) discussions on this subject area.

5.5.5 Multi-tier storage

Storage device technology landscape is showing the signs of continuing diversification. Flash based technology is now a mainstay in the enterprise and has early deployments in HPC initiatives in "burst buffer" type tiers [BurstBuffer]. Apart from Flash, we are seeing continuing signs of a new generation of device technologies driven by Non Volatile RAM (NVRAM) or Storage Class Memory (SCM) type devices such as MRAM, STTRAM, RRAM, etc. Each of these devices has different operating profiles, and offer different combinations of performance/cost/capacity/resiliency characteristics, as they are at varying levels of commercialisation or scale. They are generally characterised to sit between main memory and disk based storage. Very little is still known about how these devices can be reasonably employed in HPC. There can be a large number of trade-offs that can be studied when different combinations of these devices are introduced at various points in the I/O stack. Energy and cost have always to be kept under check when introducing these devices. With a multiplicity of storage mediums, each with its own characteristic combinations of performance, reliability, endurance and, of course, cost, it is important that data management frameworks (for example: Information Lifecycle management) and applications themselves (including by Big Data tools), can efficiently and effectively ensure the overall system is operating at its best. There are interesting possibilities of optimising the system for various parameters such as cost, performance and power when different combinations of these devices are used in the deep I/O hierarchy, giving way to a much more complex optimisation problem than exists in today's systems architectures.

Related milestones are: M-BIO-1 & ARCH-4, M-BIO-3 and M-BIO-4

[18] Steve Conway and Chirag DeKate (IDC), High Performance Data Analysis HPC meets Big Data, March 2013

[19] DoE Fast Forward Program Documents, www.wiki.hpdd.intel.com/display/PUB/Fast+Forward+Storage+and+IO+Program+Documents, Accessed May 2015

5.5.6 Fine-grained shared Quality of Service

There has been some research in addressing I/O quality of Service for HPC parallel file system solutions such as Lustre. However, to date, industry has been very slow to catch up in exploiting this feature for improved time to solution. Prioritising workloads from different applications/tasks/processes: continues to be extremely important in the light of increasing parallelism. Millions of processes having the same priority for I/O access (which will be a very precious resource) or uncontrolled processes can severely impact overall system performance. Multiple classes of priorities need to be assigned for applications/tasks/processes. Furthermore, quality of service needs to be finer grained (with pre-defined defined latency, bandwidth, etc. for I/O accesses) if we are to expect graceful scaling with compute parallelism and multi-tenancy applications.

Related milestone: M-BIO-7

5.5.7 Layouts/Views/Transformations of data

Storage and I/O capacity requirements in a single managed supercomputing storage system (whether distributed or centralised) is bound to cross the thresholds of an Exabyte with the proliferation of instrument and sensor data that needs to be stored for further processing, and inputs of this processing created by simulations or fed back into running simulations. Today, there are various middleware libraries and data formats, each with its own view of data. Multi-tenant applications needing access to the same data sets from the system, but with each of their own middleware components, will lead to unnecessary data replication and sprawl that will complicate the existing data volume problem. This also has very negative implications in terms of cost, I/O bandwidth and energy. Hence, it should be possible for data to interoperate between various middleware libraries, presenting different "windows" or "views" to essentially the same data, without any copying. This requires research in holistic underlying software I/O middleware infrastructures.

Furthermore, the data access and placement should be optimised for various parameters (access throughput, access latency, cost, etc.) based on the needs of the applications. For example, to achieve very low latency, most of the data needs to be kept in NVRAM and main memory or conversely a long term 'cold' dataset may be on tape, disk or distributed remotely. Hence, different data sets have the needs of different "layouts" within the storage infrastructure that can be dynamically changed, based on application needs.

5.5.8 Adaptive/autonomic storage

Storage is a "fixed" resource today with limited "smarts" for dynamically adapting itself to different performance and reliability criteria (for example: dynamically carving a very high performance virtualised "container" to be readily available once an application is scheduled). Adaptive/autonomic storage though used as a term in the enterprise for some time now, has not been achieved in HPC. The use of Hints (from applications or other sources) or self-management could provide significant system improvements if correctly applied.

Related milestone: M-BIO-10

5.5.9 Standardisation of APIs

There are still non holistic I/O access APIs that cater to a very wide range of applications and a very wide class of data storage infrastructures underneath (which includes features such as application guided I/O). The different I/O middleware technologies and libraries, such as HDF5, NetCDF, ADIOS[20], etc. are very fragmented and there has been no effort to unify them under a common I/O access API, which is suitable for scaling applications to exascale. There is a need for research in this area as standardisation has a very long cycle time, and although included in the 2014/15 calls, no proposals were selected. This does not, however, remove the necessity for it.

Related milestone: M-BIO-4

[20] ADIOS ORNL website, www.olcf.ornl.gov/center-projects/adios, Accessed May 2015

5.5.10 Data Management

With the proliferation of data in HPC, there is a wide class of tools, methods and techniques needed for data management functions such as data integrity checking, information life cycle management, better data sharing functions, tracking data provenance, metadata organisation and management, providing better audit trails etc. Tools, hitherto, in the enterprise, are completely unsuitable for dealing with the levels of volume scale outs as will be seen in data intensive HPC. There is a rethink needed not just for these individual data management functions, but also for techniques to deploy these functions and easily add data management functions as part of a common data management framework that scales to exascale (see also Chapter 5.6.2).

Related milestone: M-BIO-8

5.5.11 Role of the network

Interconnects between compute and storage are going to play a key role in achieving the targets of performance, energy and cost in future exascale subsystems. Very low latency and high throughput networking will need continued R&D. Scaling existing solutions, for example, Ethernet beyond 400Gb/s with very low latency, should also be considered, aligning with industry roadmaps[21]. Possibility of doing on-the-fly processing within the network should be studied. New network topologies may be required to deal with massive scale outs of storage and computation. Effective aggregation of and coordination of extremely large but lower performance networks may also be necessary.

The networking technologies are further addressed in the Chapter 4, 5.1 and 5.2 of the document.

5.5.12 Manageability

All aspects of storage systems management are complicated by the use of multiple tiers of storage, the vast volume of data, and the size and complexity of the systems, making the need for improved tools and techniques very important. Topics could include; Scheduling with storage system and data content awareness; and, contention of resources; Telemetry, Analysis and simulation of systems; and, workload, prediction and synthesis at exascale.

Related milestone: M-BIO-10

5.5.13 Understandability

There is a pressing need to understand the various possible architectures at scale out, considering the incorporation of new device tiers, new software abstractions such as advanced object stores, etc. There is also a need to understand the behavioural response of the storage system to various classes of application workloads. It is not reasonable to wait for full deployment of these systems to gain more insights into their behaviour. There is, hence, an ever pressing need for storage and I/O system simulation, both fine grain (at device and component level) and coarse grain (at larger topology abstractions). Storage systems can also be understood by very deep I/O subsystem infrastructure telemetry analysis. This area of research should also be pursued.

Related milestone: M-BIO-2

5.5.14 Security

This area is always very important from a data and storage point of view, with the increasing use of potentially sensitive consolidated 'Big Data' with issues of authenticity, ownership and rights to use, combined with multi-tenancy of super-computing facilities with access portals through cCoud services, amplify the potential risks and complexity. This is discussed in Chapter 4.

[21] Ethernet Alliance Roadmap, www.ethernetalliance.org/roadmap, Accessed May 2015

5.6 BIG DATA AND HPC USAGE MODELS

The convergence of HPC and Big Data is a trend that is happening fast and is bound to influence the HPC world. Data is becoming central even for traditional HPC domains in one way or another, whilst new HPC clients are by default data centric. These trends can be seen as a challenge; however, it is much more useful to be considered as opportunities. The HPC community needs to formulate a plan as to how to address them and benefit from them the most.

5.6.1 Performance Metrics

It is clear that the huge data volumes are here to stay and it is not a transient trend, either because the increased fidelity of simulations is creating more data or because Big Data systems will keep "pouring" data into HPC systems. Thus, we need to rethink the optimality criteria which we have been using to design HPC systems. Firstly, data centric applications are typically less computationally intense. Thus, high flop/s or flops/watt metrics need to be completely redesigned. The HPC community has reacted already by introducing the HPCC and the Graph500 benchmark. It is clear though that these need to be enhanced and augmented for the data centric applications, adopting, for instance, benchmarks from Machine learning such as deep learning, support vector machines or similar devices. Secondly, the overall computation can be split in phases that alternate between Big Data and HPC systems, or follow a certain workflow between them. Thus, the overall distributed nature of the computations and the data handling needs to be accounted for in the new metrics.

Related milestones are: M-BDUM-METRICS-(1-3) and M-ENR-M-1

5.6.2 Data Centric Memory Hierarchies/Architectures

Moving data will be the primary problem. This includes introducing data in the system, but also moving data across the memory hierarchies. It is clear that disk needs to be used only as a last resort and thus non-volatile memory will become of central importance. In addition, the data structures used in the data analytics algorithms will often be completely different from those used in the compute bound part. Therefore, efficient data structure transformation will be a key factor. In particular, when considering heterogeneous systems, we will need to adopt and further develop solutions that allow data coherency between the various compute engines. The HPC community needs to offer solutions that are easy to be used by the millions of Big Data users and developers for high-performance data analysis (HPDA). Tools for HPDA need to extent common Big Data tools in levering the efficiency and performance of the underlying HPC infrastructure by maintaining generality for complex workflow patterns. Furthermore, the memory hierarchies on the HPC system need to be efficiently linked with those of distributed Big Data systems. Determining efficient data flows that keep as much of the computation local will be a key factor.

Related milestones are: M-BDUM-MEM-(1-2)

5.6.3 Research in Algorithms

Research in algorithms that trade computation with data accesses will be of key importance. This means that we need to develop algorithms that minimise data access by relying on more computation that remains in-situ. To this end, a systematic analysis of key Big Data applications is required and a breakdown of the data flows to reusable pieces that form a pipeline. Similar activities have led to great advances in traditional HPC applications (e.g. Berkeley Dwarfs). In addition, we need to rethink distributed computation in Big Data applications in view of the availability of HPC systems that are coupled to these. Even theoretical advances pioneered years ago need to be revisited in view of extreme parallelisation potential available today.

Related milestones are: M-BDUM-ALGS-(1-2)

5.6.4 Programming Models

Programming models and languages for data centric computing will also need to become central in HPC. That is to say, programming models and languages that are heavily used in main stream Big Data research and practice (i.e. hadoop, scala, Java etc.) can serve as an inspiration for the

needed adaptations of HPC practices in view of Big Data. The HPC community needs to offer solutions that are easy-to-use by the millions of Big Data users and developers. Furthermore, mixed programming models will be crucial in bridging Big Data and HPC environments.

Related milestones are: M-BDUM-PROG-(1-3)

5.6.5 Virtualisation of HPC

Convergence of HPC and Cloud computing is a crucial step. We will start to increasingly see HPC users wanting to use an elastic way of storing data and using resources, since this is far more economic and a reality in mainstream data analytics. The question is now how to bind and merge HPC resources with Cloud architectures.

Related milestones are:
M-BDUM-VIRT-(1-2), M-BDUM-DIFFUSIVE-1

5.6.6 Diffusive Supercomputing: Bridging computation and data acquisition

HPC needs to come closer to data, both in its generation and its consumption. Thus, we need to research as to how to develop heterogeneous HPC data processing systems (HPC and Big Data hybrids), which are flexible in allowing breakthrough simulations and data analytics at the same time. This means that HPC design needs to become much more holistic, considering simultaneously computation, data storage and computation on storage, data acquisition and data serving. We can envisage HPC systems to be used closer to where data is generated and use these 'Big Data' systems as a crucial pre-processing stage of the data analysis pipeline.

Related milestones are: M-BDUM-DIFFUSIVE-(1-2)

The aforementioned trends and target areas, which clearly show that the new HPC uses which are data centric, will demand research on the full hierarchy: algorithms, system S/W and H/W, languages and programming models and storage. Thus, we see this subject as a horizontal one, running across almost all other areas of research focus rather than an isolated vertical theme.

5.7 MATHEMATICS AND ALGORITHMS FOR EXTREME SCALE HPC SYSTEMS

The development of future HPC architectures is strongly driven by several technological trends which in particular adhere to power constraints. New mathematical methods and algorithms are important ingredients in ensuring efficient usability of these future architectures and technologies, as well as scalability from the mathematical methods level, through algorithms down to system levels. They have to cope with increasing parallelism at the growing number of levels, as well as with a deepening of memory hierarchies that can also be heterogeneous. Comparing the top system on the Top500 lists, shows a trend which is expected to continue: As of November 2004, a Blue Gene/L node performed up to 8 Flop per clock cycle, whilst ten years later each node of the Tianhe-2 system is capable of executing about 3,100 Flop per clock cycle. As clock frequency will not increase, the level of concurrency at different levels of any future HPC architecture will increase further, with most added parallelism expected at a node level. With power consumption becoming a major factor in total cost of ownership at almost any location, also unconventional hardware architectures, based, e.g., on FPGA designs that allow for extremely energy efficient, and application specific acceleration, are becoming a competitive option in important application areas. Once again these solutions rely on concurrency becoming more abundant, and thus, there is a need for pushing scalability as well as efficiency limits of applications, with an emphasis on industrial and commercial organisations in Europe, to improve their products or their usage. This is particularly challenging for those cases which are not embarrassingly parallel and feature a high level of irregularity. To keep the system architecture balanced in terms of compute versus, both, memory performance and capacity, a deepening of memory hierarchies is to be expected. Algorithms and mathematical methods will have to play a critical role to enable efficient exploitation of these new architectures.

A vital area for research and development is where the need for scalable computing resources is linked to huge amounts of data, which are generated through simulations or originate from external sources, and also, where the need for new mathematical methods and algorithms for extreme data challenges is critical.

Robustness and ease-of-use are further aspects which need to be addressed. The growing importance of computational methods in industrial processes leads to a growing importance of robustness and reliability. Robustness has multiple facets including numerical robustness, propagation of uncertainties in complex work-flows, or robustness with respect to undetected hardware errors. For broader uptake of HPC solutions, ease-of-use is a critical aspect and should facilitate uptake of new algorithms and mathematical methods by Software-as-a-Service (SaaS) providers.

5.7.1 Robust methods and algorithms enabling extreme scalability

To exploit the performance of future massively-parallel architectures, for many problems new algorithms are required which allow for: a fixed problem size, the increase in the level of concurrency and facilitate the usage of different levels of parallelism or specific hardware accelerators. New mathematical methods may lead to innovative approaches in the use of computation in problem solving which generate new levels of concurrency. Algorithms may have to be hierarchical to reduce communication as well as synchronisation and to simplify dynamic task scheduling. On the other hand, performance variability of the system components is expected on future exascale systems. This makes global communication hiding algorithms such as pipelined Krylov solvers, even more important. At some point a bulk synchronous approach may not be a feasible option anymore. Run-time instantaneous response to the system variability will be required, thus, new dynamic parallelisation and load balancing capabilities should be included on the algorithms design. Mathematical methods such as stochastic and hybrid ones (stochastic/deterministic and deterministic/deterministic ones), including hybrid Monte Carlo and quasi-Monte Carlo, and, asynchronous methods and algorithms, as well as multilevel multiscale hybrid Monte Carlo, domain decomposition methods and algebraic multi-level/multi-grid methods, are of particular interest. They may lead to innovative approaches in terms of increased scalability of the methods and algorithms

enabling reduced communications. They also may result in efficient algorithmic resilience and fault-tolerance. The latter renders them very suitable in problem solving for large class of problems that generate new levels of concurrency. The challenges of emerging exascale architecture can be met by enabling runs with mixed and variable precision, maintaining a high level of parallelism and through multi-level and multi-scale approaches, thereby, matching the hierarchical topology of new computer architectures, which may include compute nodes comprising accelerators that are both, highly parallel and highly hierarchical[22].

Finally, an important aspect to be addressed is robustness. Robustness has several aspects including: fault resilient algorithms which enable the algorithms to run on large and error-prone systems, reproducibility of numerical results and numerical stability of algorithms. Additionally, robustness enables many opportunities to trade performance for accuracy, which makes it possible to consider approximate computing scenarios where the exactitude of some computations can be tuned to maximise compute or memory access performance, whilst keeping the results' quality within an acceptable margin.

Here, also data-related uncertainties need to be considered and addressed, e.g. within a Verification Validation and Uncertainty Quantification (VVUQ) framework.

The following specific research topics are proposed:

1. Extreme-scale algorithms for forward in time computing: Implicit and semi-implicit algorithms for the numerical integration of non-linear dynamical systems have several decisive conceptual advantages over explicit schemes. These advantages include superior conservation properties, robustness for arbitrary time step sizes, and improved adherence to dominant physical balances. Implicit algorithms generally require more communication in parallel implementations, so that their advantages may be offset by diminishing efficiency on massively parallel future systems. Important open questions in this context which need to be addressed, concern the existence of algorithmic rearrangements that systematically improve the parallelism of implicit schemes, or alternatively the development of innovative approaches that would render explicit time discretisation competitive with implicit ones, both in terms of energy-to-solution and time-to-solution, as well as preserving desirable conservation and mimetic properties, robustness, and preserving fundamental balances, underlying the simulated dynamical systems.

2. Trading performance for accuracy through means of improved robustness: Robustness of algorithms becomes more important since it brings opportunities to trade performance for accuracy. Possible options are to use inexact, stochastic and hybrid, however, fast, communication-avoiding methods for computing approximate solutions at intermediate steps or to reduce the impact of some global operations, such as reductions and/or dot-products, by decreasing their arithmetic accuracy or by skipping some irrelevant computations (see also M-ENR-AR-7).

Related milestones are M-ALG-1 and M-ALG-2.

5.7.2 Methods for scalable data analytics

Significant challenges arise from extreme data challenges, which require a paradigm shift from a compute centric view to a more data centric view. Algorithms for data discovery, such as graph analytics, require highly scalable compute resources and are not embarrassingly parallel. Accessing graph nodes across extremely large clusters may result in highly irregular memory and network access patterns and imposes a huge scaling challenge. Graph analytics are in particular necessary for discovering hidden connections and patterns within data and for identifying non-conforming objects in massive data sets (the needle-in-the-haystack problem). Other methods require analysis and visualisation to be embedded in highly-parallel simulations or real-time processing of massive data streams for event detection and support of ad-hoc decision making. Such operational and mission-critical applications in data analytics require specially balanced computer architectures which have particular bottlenecks removed and are not necessarily available on the standard IT market, also including new server design, I/O subsystems, and storage technologies. Some industry consortia have started discussing blueprints of special system architectures which will match their needs in data analytics. Analytic approaches are, therefore, heavily constrained by technological limitations. Progress, thus, has to be achieved through co-design activities involving, both designers of architectures and technologies, as well as developers of new mathematical methods and algorithms.

[22] Dongarra et al, Applied Mathematics research for Exascale Computing, March 2014, Report, US Department of Energy, 2014

The following specific research and development topics are proposed:

· Scalable graph-based analytics: Graph-based analytics features irregular memory access and communication patterns whilst a growing number of use cases start to mandate supercomputing resources. Further research is required to develop and establish methods which will allow to scale out problems on massively-parallel HPC architectures.

· Enabling co-design of mathematical methods for data analytics and HPC technologies and architectures: In order to mitigate the technical limitations of today's analytic approaches on today's architectures, more efforts on co-designing mathematical methods for data analytics and application optimised HPC technologies and architectures are required. These efforts should result in a parametrisation of relevant data analytics use cases towards exascale performance levels.

Related milestones are M-ALG-3 and M-ALG-4.

5.7.3 Mathematical support for data placement and data movement minimisation

As memory hierarchies of computers become increasingly complex and heterogeneous, the problem of how to decompose user data onto the various levels and how/when to move data between levels becomes an exponentially difficult problem. This problem is exacerbated by new memory technologies that support multiple usage models, and the introduction of data "persistence" (in non-volatile memories) as a new modelling requirement. The true complexity of the data problems being solved is not yet known, but seemingly simple data layout problems such as the tile-size selection problem, bear striking similarities to heterogeneous partition problems such as the MaxGrid problem, which can be shown to be NP-complete[23], as can many other static scheduling and partitioning problems. When inter-process communications and the effects of multiple data structures are combined, the complexity of data partition problems is likely to be NP-complete. For this reason, some advance work on the complexity analysis of data-driven algorithms must first be performed. When/if the general problems are shown to be of high complexity, then sensible heuristic, approximation or special-case problems must be defined instead.

The programming environment will seek to solve these data-related issues via additional language constructs, user controls and automation, e.g., of caching and pre-fetching (see "5.3.1 Intelligent Data Placement"). However, without advancements at the mathematical level, such technical features will (at best) add burden to the application development life cycle. It is expected that a combination of new technologies will be required to support the data optimisation needs of the various classes of algorithms. However, currently, such algorithmic classes cannot be ascertained easily. Advances are required in such algorithmic categorisation, as well as in specific fields of I/O Lower bounds, Static (affine) scheduling algorithms[24] and dynamic (task-level) scheduling.

The following specific research and development topics are to be addressed:

· The various classes of data, partitioning and scheduling problems and the classes of data movement problem that stem from key applications should be studied and categorised.

· The complexity of the data problems should be ascertained, and specifically, the likely availability of a polynomial time solution to each category of problem, or alternatively, a list of suggested heuristic, approximation or special-case problems to solve.

· Existing work on the defining of I/O lower bounds25, which is the most appropriate measure of complexity for data problems, should be extended and deepened. The end goal of such work is to define a framework that can be adopted by programming environments so that data optimisations can be placed in context to the I/O lower bound. Ultimately, I/O lower bounds for arbitrary loop nests should be derived.

· Mathematical support for static scheduling of codes should be developed in the areas of Lattice Optimisation, integer linear programming and other mathematical solvers, for solving data-driven scheduling problems at run-time.

· Mathematical and algorithmic approaches for the scheduling of tasks on abstract resources should be developed and deepened. The applicability of the described approaches should be clear and the conditions under which the approaches are recommended should be equally clear. The approaches should extend to scheduling of problems over deep and heterogeneous memory hierarchies.

[23] Casanova, Henri, Arnaud Legrand, and Yves Robert, 'Parallel algorithms', CRC Press, 2011

[24] Bastoul, Cédric, et al. 'Putting polyhedral loop transformations to work' Languages and Compilers for Parallel Computing. Springer Berlin Heidelberg, 2004. 209-225.

[25] Irony, Dror, Sivan Toledo, and Alexander Tiskin. 'Communication lower bounds for distributed-memory matrix multiplication', Journal of Parallel and Distributed Computing 64.9 (2004): 1017-1026.

· The mathematical support for the optimisation of multiple memory levels must be simultaneously developed.

Related milestones are M-ALG-5 and M-ALG-6. For reaching these milestones the architecture roadmaps for the next couple of years need to be assessed in respect to data storage and movement capabilities, including e.g. node-level memory hierarchy organisation. The outcome of this research is expected to impact work in programming environments.

5.7.4 How can the algorithmic and mathematical advances be leveraged in programming tools

The assumption in this section is that new mathematical and algorithmic work is required before tools are developed, and that without such theoretical advances an incorrect or incomplete toolset could be developed. As such, it is imperative that the work described herein, takes a technology-neutral stance and can be leveraged by multiple toolsets.

With that qualification in mind, there are several areas of the programming environment that would be boosted by the mathematical and algorithmic advances stated above. New programming languages and their compilers could use the data movement and minimisation algorithms directly. This would involve incorporating the proposed models into a form that the compiler's internal representation could understand, and that the compiler can execute the set of code transformations that would be recommended by the solutions to the data problem. These same changes can be leveraged in run-times, but only in combination with some higher-level abstractions since the problem requires context which is not available in typical run-time libraries. The proposed changes also potentially support the development of tools that would not fit into the current standard environment. In particular, domain-specific languages could be designed specifically to support this data-centric analysis and code transformation. In all three of the mentioned cases, additional infrastructure work would be required that constructs the models, configures optimisation problems, and interfaces to the programming environment software.

The following specific research and development topics are proposed:

Mathematical methods for compiler technologies, run-time environments and related tools: Leverage results from research and developments related to data movement minimisation, to enable new compiler technologies that support optimisation of data placement and minimisation of data movement. This will in particular requires a formulation of the proposed mathematical methods in terms of compiler's internal representations. These mathematical methods will also impact run-time environments and tools to enable optimal data placement and data movement minimisation at run-time. This work should result in specific models and formulation of optimisation problems which can be used in compilers, run-time environments and related tools.

Related milestone: M-ALG-7.

5.7.5 Algorithms reducing energy-to-solution

The potential of reducing energy-to-solution has been demonstrated for a number of cases where different methods for solving the same problem, revealed significant differences in terms of an energy-to-solution metric[26]. Minimisation of data movement is an important aspect in this context as (off-chip) data transport is the most expensive in terms of energy consumption. But, also other aspects need to be addressed, including effective exploitation of a given hardware architectures and the design of algorithms suitable for architectures, based on particular power-efficient technologies. Architectures are likely to become more heterogeneous to improve power efficiency at hardware level. A further aspect concerns improving our understanding of the interplay between algorithmic choices and the energy required to solve a particular numerical problem. HPC solutions at various levels are becoming capable of providing fine-grained information on power consumption, and thus, facilitate energy-to-solution measurements. Generalised, designing algorithms that minimise energy-to-solution and changing computing paradigms towards a still to be established energy efficiency paradigm is the next step. This is further addressed in Chapter 5.4 [M-ENR-MS-1]. Furthermore, these energy measurement capabilities open opportunities for reducing energy-to-solution through auto-tuning frameworks. Research is required to better understand how different energy-efficient algorithms work together, in particular, in those cases where different data layouts are required.

[26] Pavel Klavík, A. Cristiano I. Malossi, Costas Bekas, Alessandro Curioni, "Changing computing paradigms towards power efficiency", Philosophical Transactions A, 2014.

The following specific research and development topics are proposed:

Development of algorithms optimised for energy-to-solution: New algorithms should be developed which allow for a reduction of energy-to-solution on existing, emerging and future architectures. This research should include a characterisation and comparison of algorithms with respect to energy consumption (see also Chapter 5.4, M-ENR-AR-8).

Related milestone: M-ALG-8.

5.7.6 Vertical integration and validation of mathematical methods and algorithms

Efforts are required to ensure for any of the developed mathematical methods and algorithms that these can be efficiently implemented on different types of HPC architectures and can easily be used by a broad user community. This requires a cross-cutting approach together with other areas of the SRA to:

- Investigate the scalability of particular mathematical methods and algorithms at all relevant levels for relevant use cases, and work-loads and extrapolate scalability for upcoming exascale architectures;

- Explore and tune algorithmic parameters to maximise scalability and performance of the algorithms;

- Improve agility in the design, implementation or porting, tuning and optimisation process for different algorithms on different architectures.

Vertical integrations should ensure scalability at all levels, i.e. at mathematical models/methods level, through algorithmic level, down to systems and architecture levels. The interest is in an integrated approach in developing scalable mathematical methods and algorithms that lead to scalable programming models and tools and scalable libraries, all these ensuring scalability at all levels and opening a path to exascale scalability[27].

A suitable balance between architectural and algorithmic performance needs to be identified as only the combination of both determines the overall performance in terms of minimal time-to-solution. This will also require a suitable tuning of algorithmic parameters support by auto-tuning techniques.

The results of such efforts can only be exploited if a careful design process for next generation mathematical algorithms is in place. This becomes critical, in particular, for leveraging the potential of extreme scale algorithms when ported on the emerging heterogeneous computing architectures. Some of the limiting factors to the wide spread use of heterogeneous computing include, the relatively expensive and architecture-specific porting process, the lack of separation-of-concern in user applications; and, developers productivity and ease of usability.

The following specific research and development topics are proposed:

- **Vertical integration and validation**: Algorithms and mathematical methods are to be tested and validated with respect to scalability at all levels of the architecture as well as ease of implementation, tuning and optimisation on different architectures. This must involve exploration of full vertical integration into lower levels of system hardware to upper levels of system software architectures.

- **Tuning of algorithmic parameters for exascale**: The parameter space for tuning algorithms to maximise their scalability and performance on current, emerging, as well as future exascale architectures should be investigated.

Related milestones are: M-ALG-9 and M-ALG-10.

[27] Alexandrov V. 'Scalable Stochastic and Hybrid Methods and Algorithms for Extreme Scale Computing', Procedia Computer Science, V. 29, Volume 29, pp. 1888—1892, Elsevier 2014.

6.

RESEARCH MILESTONES

A milestone in the tables below is defined as a tangible **research subject (content)** achieved by a **certain point in time** (**'availability date'**). The results of the research topic referenced by a milestone are available at the availability date and might be needed to start the work of some other milestone(s) in the same or another of the seven domains.

A milestone in a particular domain can have two relationships with one or more milestones in different domains:

· A milestone can be a **co-requisite** with one or several milestones of other domains (i.e. they all have the same or a "close-by" date). These milestones should be worked on in unison.

· A milestone can have one or more milestone from other domains as **pre-requisites**. This means that the results of these other milestones are needed in order to work on this specific milestone (the "other milestones", thus, have an earlier date!).

6.1
HPC SYSTEM ARCHITECTURE AND COMPONENTS

SRA-2 Milestones	Availability date	Co-requisites	Pre-requisites
M-ARCH-1: New HPC processing units enable wide-range of HPC applications.	2018	M-PROG-API-1, M-PROG-API-3	
M-ARCH-2: Faster memory integrated with HPC processors.	2018	M-PROG-RT-3	
M-ARCH-3: New compute nodes and storage architecture use NVRAM.	2017	M-BIO-1, M-BDUM-MEM-2, M-ENR-AR-4	
M-ARCH-4: Faster network components with 2x signalling rate (rel. to 2015) and lower latency available.	2018		
M-ARCH-5: HPC networks efficiency improved.	2018	M-SYS-IC-1, M-SYS-IC-2	
M-ARCH-6: New programming languages support in place.	2018	M-PROG-API-6, M-BDUM-PROG-2	
M-ARCH-7: Exascale system energy efficiency goals (35kW/PFlops in 2020 or 20 kW/Pflops in 2023) reached.	2020-2023	M-ENR-MS-9	M-ALG-8, M-ENR-MS-2, M-ENR-AR-8
M-ARCH-8: Virtualisation at all levels of HPC systems.	2018	M-SYS-IC-3	
M-ARCH-10: New components / disruptive architectures for HPC available.	2019	M-PROG-API-7	

6.2 SYSTEM SOFTWARE AND MANAGEMENT

SRA-2 Milestones	Availability date	Co-requisites	Pre-requisites
M-SYS-OS-1: Kernel scheduling policy.	2016		
M-SYS-OS-2: OS Low level standard API with run-time.	2017	M-PROG-RT-4	
M-SYS-OS-3: New memory management policy and libraries.	2017	M-PROG-RT-3	
M-SYS-OS-4: Container and virtualisation support; Hypervisor for HPC.	2016		
M-SYS-OS-5: Offload programming model support.	2017-2019	M-PROG-API-2, M-PROG-API-3, M-PROG-API-7, M-PROG-RT-3, M-BDUM-PROG-2	
M-SYS-OS-6: OS decomposition to add application performance and flexibility.	2019		
M-SYS-OS-7: Investigate HPC specific security requirements on OS level.	2017-2019		
M-SYS-IC-1: OS-bypass and hardware interface integrity protection.	2016	M-ARCH-5	
M-SYS-IC-2: Interconnect adaptive and dynamic routing algorithm and congestion control, power management.	2017	M-ARCH-5, M-ENR-MS-2	
M-SYS-IC-3: Network virtualisation compliancy.	2017	M-ARCH-8	
M-SYS-CL-1: Flexible execution context configuration and management (from image to containers).	2018	M-PROG-RT-2, M-BDUM-VIRT-2	
M-SYS-CL-2: Prescriptive maintenance based on Big Data analytics technics.	2016		
M-SYS-CL-3: Infrastructure security.	2017-2020		
M-SYS-RM-1: New Scalable scheduling enhancement, with execution environment and data provisioning integration.	2017		
M-SYS-RM-2: New multi-criteria adaptive algorithms: Heterogeneity-/memory- and locality-aware.	2017	M-PROG-RT-4, M-PROG-LIB-1	
M-SYS-RM-3: Resilient framework.	2020		
M-SYS-Vis-1: Scalable “in situ” visualisation.	2016		
M-SYS-Vis-2: Scaling for the compositing phase.	2017		
M-SYS-Vis-3: Ray-tracing capabilities.	2018		
M-SYS-Vis-4: High dimensional data, graphs and other complex data topologies.	2018		

6.3 PROGRAMMING ENVIRONMENT

SRA-2 Milestones	Availability date	Co-requisites	Pre-requisites
M-PROG-API-1: Develop benchmarks and mini-apps for new programming models/languages.	2016	M-ARCH-1	
M-PROG-API-2: APIs and annotations for legacy codes.	2017	M-SYS-OS-5	
M-PROG-API-3: Advancement of MPI+X approaches (beyond current realisations).	2017	M-SYS-OS-5, M-ARCH-1	
M-PROG-API-4: APIs for auto-tuning performance or energy.	2017	M-ENR-MS-2	
M-PROG-API-5: Domain-specific languages (specific languages and development frameworks).	2018		
M-PROG-API-6: Efficient and standard implementation of PGAS.	2018	M-ARCH-6	M-ARCH-5
M-PROG-API-7: Non-conventional parallel programming approaches (i.e. not MPI, not OpenMP / pthread / PGAS - but targeting asynchronous models, data flow, functional programming, model based).	2019	M-ARCH-10, M-SYS-OS-5, M-BDUM-PROG-3	
M-PROG-LIB-1: Self- / auto-tuning libraries and components.	2018	M-SYS-RM-2	
M-PROG-LIB-2: Components / library interoperability APIs.	2017		
M-PROG-LIB-3: Templates / skeleton / component based approaches and languages.	2019		
M-PROG-RT-1: Run-time and compiler support for auto-tuning and self-adapting systems.	2018		
M-PROG-RT-2: Management and monitoring of run-time systems in dynamic environments.	2018	M-SYS-CL-1,M-BDUM-VIRT-1	
M-PROG-RT-3: Run-time support for communication optimisation and data placement: data locality management, caching, and prefetching.	2019	M-SYS-OS-5, M-SYS-OS-3, M-ARCH-2, M-BDUM-METRICS-1, M-ALG-6	
M-PROG-RT-4: Enhanced interaction between run-time and OS or VM monitor (w.r.t. current practice).	2018	M-SYS-RM-2, M-SYS-OS-2	M-SYS-OS-4
M-PROG-RT-5: Scalable scheduling of million-way multi-threading.	2020	M-ALG-5	
M-PROG-DC-1: Data race condition detection tools with user-support for problem resolution.	2017		

SRA-2 Milestones	Availability date	Co-requisites	Pre-requisites
M-PROG-DC-2: Debugger tool performance and overheads (in CPU and memory) optimised to allow scaling of code debugging at peta- and exascale	2018		
M-PROG-DC-3: Techniques for automated support for debugging (static, dynamic, hybrid) and anomaly detection, and also, for the checking of programming model assumptions.	2018		
M-PROG-DC-4: Co-design of debugging and programming APIs to allow debugging to be presented in the application developers original code, and also, to support applications developed through high-level model descriptions.	2018		
M-PROG-PT-1: Scalable trace collection and storage: sampling and folding.	2018		
M-PROG-PT-2: Performance tools using programming model abstractions.	2018		
M-PROG-PT-4: Performance analytics tools.	2018		
M-PROG-PT-5: Performance analytics at extreme scale.	2019		M-BDUM-METRICS-1

6.4 ENERGY AND RESILIENCY

SRA-2 Milestones	Availability date	Co-requisites	Pre-requisites
M-ENR-MS-1: Quantification of computational advance and energy spent on it.	2017	M-BDUM-METRICS-1, M-BDUM-METRICS-2, M-BDUM-METRICS-3	
M-ENR-MS-2: Methods to steer the energy spent.	2017	M-ALG-6, M-PROG-API-4, M-SYS-IC-2	
M-ENR-MS-3: Use of idle time to increase efficiency.	2018		M-SYS-IC-1
M-ENR-AR-4: New levels of memory hierarchy to increase resiliency of computation.	2017	M-ARCH-3	
M-ENR-FT-5: Collection and Analysis of statistics related to failures.	2018		
M-ENR-FT-6: Prediction of failures and fault prediction algorithms.	2019		
M-ENR-FT-10: Application survival on unreliable hardware.	2019		M-PROG-RT-2
M-ENR-AR-7: Quantification of savings from trade between energy and accuracy.	2018	M-ALG-8	
M-ENR-AR-8: Power efficient numerical libraries.	2019		M-ALG-8 M-PROG-LIB-3
M-ENR-MS-9: Demonstration of a sizable HPC installation with explicit efficiency targets.	2019	M-ARCH-7	

6.5
BALANCE COMPUTE, I/O AND STORAGE PERFORMANCE

SRA-2 Milestones	Availability date	Co-requisites	Pre-requisites
M-BIO-1: Tightly coupled Storage Class Memory IO systems demo.	2017	M-ARCH-3	
M-BIO-2: Common I/O system simulation framework established.	2017		
M-BIO-3: Multi-tiered heterogeneous storage system demo.	2018		
M-BIO-4: Advanced IO API released: optimised for multi-tier IO and object storage.	2018		
M-BIO-5: Big Data analytics tools developed for HPC use.	2018		M-BDUM-DIFFUSIVE-1
M-BIO-6: 'Active Storage' capability demonstrated.	2018		
M-BIO-7: I/O quality-of-Service capability.	2019		
M-BIO-8: Extreme scale multi-tier data management tools available.	2019		
M-BIO-9:Meta-Data + Quality of Service exascale file i/o demo.	2020		
M-BIO-10: IO system resiliency proven for exascale capable systems.	2021		

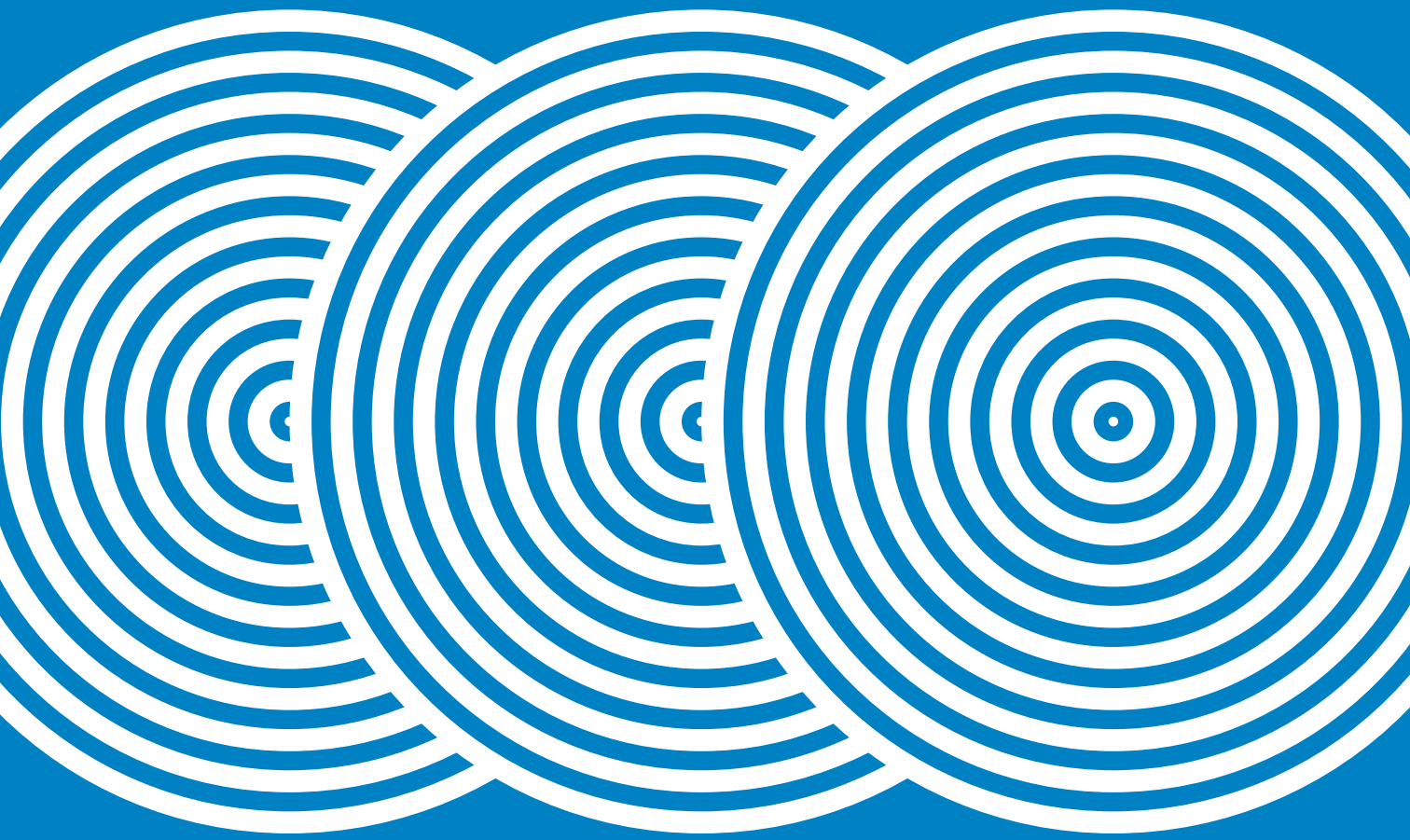
6.6 BIG DATA AND HPC USAGE MODELS

SRA-2 Milestones	Availability date	Co-requisites	Pre-requisites
M-BDUM-METRICS-1: Data movement aware performance metrics.	2017	M-PROG-RT3, M-ENR-MS-1	
M-BDUM-METRICS-2: HPC like performance metrics for Big Data systems.	2017	M-ENR-MS-1	
M-BDUM-METRICS-3: HPC-Big Data combined performance metrics.	2018	M-ALG-4, M-ENR-MS-1	
M-BDUM-MEM-1: Holistic HPC-Big Data memory models.	2017		
M-BDUM-MEM-2: NVM-HPC memory and Big Data coherence protocols and APIs.	2017	M-ARCH-3	M-ARCH-2
M-BDUM-ALGS-1: Berkeley Dwarfs determination for Big Data applications.	2017		
M-BDUM-ALGS-2: Implementations of Dwarfs in Big Data platforms.	2019	M-ALG-6	
M-BDUM-PROG-1: Hybrid programming paradigms HPC-Big Data.	2017		
M-BDUM-PROG-2: Hybrid programming paradigm with coherent memory and compute unified with Big Data programming environments.	2018	M-ARCH-6, M-SYS-OS-5	
M-BDUM-PROG-3: Single programming paradigm across a hybrid HPC-Big Data system.	2021	M-PROG-API-7	
M-BDUM-VIRT-1: Elastic HPC deployment.	2018	M-PROG-RT-2	
M-BDUM-VIRT-2: Full virtualisation of HPC usage.	2021	M-SYS-CL-1	M-ARCH-8, M-SYS-OS-4
M-BDUM-DIFFUSIVE-1: Big Data - HPC hybrid prototype.	2017		
M-BDUM-DIFFUSIVE-2: Big Data - HPC large-scale demonstrator.	2020		M-ARCH-2

6.7
MATHEMATICS AND ALGORITHMS FOR EXTREME SCALE HPC SYSTEMS

SRA-2 Milestones	Availability date	Co-requisites	Pre-requisites
M-ALG-1: Scalability of algorithms demonstrated for forward in time computing for current architectures.	2017		
M-ALG-2: Multiple relevant use cases demonstrated for improving performance by means of robust, inexact algorithms.	2018		
M-ALG-3: Scalable algorithms demonstrated for graph-based analytics.	2019		
M-ALG-4: Processes established for co-design of mathematical methods for data analytics and of HPC technologies/architectures.	2019	M-BDUM-METRICS-3	
M-ALG-5: Classes of data, partitioning and scheduling problems categorised and their complexity ascertained.	2019	M-PROG-RT-5	N-BDUM-ALGS-1
M-ALG-6: Mathematical and algorithmic approaches established for the scheduling of tasks on abstract resources and exploitation of multiple memory levels.	2020	M-BDUM-ALGS-2, M-PROG-RT-3, M-ENR-MS-2	
M-ALG-7: Research on mathematical methods and algorithms exploited for compiler technologies, run-time environments and related tools.	2018		M-ENRE-MS-1
M-ALG-8: Reduction of energy-to-solution demonstrated by means of appropriately optimized algorithms demonstrated for a set of relevant use cases.	2017	M-ENR-AR-7	
M-ALG-9: Process for vertical integration of algorithms established together with the validation of scalability, ease of implementation, tuning and optimisation.	2019		

7.



END-USER AND ISV REQUIREMENTS

ETP4HPC is committed to maintaining a continuous discussion with two key groups influencing the use of HPC technology: Industrial HPC end-users and Independent Software Vendors (ISVs). Two workshops have been facilitated in order to receive and discuss the point of view of these two communities respectively. Answers have been requested to the following questions:

· How do you support your business through the deployment of HPC?

· Are there any changes and trends in your company's use of HPC?

· What are your main three priorities in relation to the daily use of HPC infrastructure?

· How do you experience or see the link between "Extreme data" and "Extreme computing"?

· What role, if any, does "Data analytics" play in your use of HPC infrastructure?

· Do you see an increased need to dedicate "security" features in your HPC infrastructure?

· Do you see an increased advantage for "resource virtualisation" in HPC systems?

· How do you judge the increased necessity for 'real-time' HPC processing?

· For your business needs, what is your prognosis (+5 years) on the required improvement of:

— Scalability and total system performance
— Data volumes and data bandwidth
— Energy efficiency
— Resiliency of the HPC system stack

· How do you see the position and role of HPC technology providers (HW/SW), ISVs and HPC service providers in Europe?

· From your point of view, what could specific opportunities be for them for a w/w business?

The participants of the workshops represent a variety of sectors (i.e. Oil and Gas, Energy, Automotive, Aerospace, the Weather and Climate, Food-Industry, Consumer, Biomedical, Engineering services), various sizes and maturity stages (large organisations, SMEs and start-ups) and a wide range of HPC deployment modes (small and large HPC systems) and use cases.

The priorities identified reflect the diversity of the end-user and ISV community. It is not possible to identify one single system design point that the end-users or software makers are looking for. The following conclusions can be drawn:

· Scalability and total system performance is a priority but so is ease of use, TCO, flexibility of access to HPC resources.

· There is a consensus that energy efficiency, total power consumption, and system resiliency are extremely important. For some, resiliency is the first priority.

· The two communities are aware of the rising influence of 'Big Data' on the design of the upcoming system generations, and although some do not think HPDA would ever of importance to them, others are making their first tentative steps towards adopting their strategies.

In general, the more traditional technical computing community - represented by most of the participants - will continue advocating for an extremely broad set of priorities for future system characteristics, functions and features.

Several end-users and ISVs are interested in the following add-on features:

· In the case of mid-range and smaller users, flexible access to HPC resources through Cloud services becomes more relevant than at the time of the SRA 1 (e.g. October 2012, when a similar workshop was held).

· "Security", a topic typically remaining outside the core HPC technology domain in the past, becomes relevant considering potential attacks at HPC centres.

· Being able to change the workflow dynamically and also "in situ visualisation" serving to change the workflow in real time (e.g. in the area of catastrophe management in the case of natural or industrial hazards) rank higher than previously.

· The requirement for predictive performance (foreseeable time to solution) is considered important as well.

· The need for making the use of HPC easier, more accessible (the term "democratisation" of HPC is used in this context).

· Some users and ISVs identify the lack of skilled and trained personnel as an issue (problem the EXDCI project will focus on)

The consequence of this is that the ETP4HPC SRA needs to support this wide span of technology improvements. In order to be able to support the requirements of end-users and ISVs, every element of the HPC "value chain" needs to be addressed, ensuring a balanced evolution of the entire technological area. In line with this approach, adding the "Mathematics and Algorithms" domain to the SRA helps achieve a complete coverage of the technical areas required.

8.

EXTREME-SCALE DEMONSTRATORS

High-Performance Computing (HPC) is a crucial asset for driving Europe's product innovations and stretching its technology providers. The "Extreme-Scale Demonstrators" (EsDs) we propose, are vehicles to optimise and synergise the effectiveness of the entire HPC H2020 Programme through the integration of isolated R&D outcomes into fully integrated HPC system prototypes; a key step towards establishing European exascale capabilities and solutions. The primary focus of the EsD projects will be establishing proof-points for the readiness, usability and scalability potential of the successful technologies developed in WP2014/15 and WP2016/17, when deployed in conjunction with open market technologies at that time. Accordingly, this presents ETP4HPC's current view of the way these EsDs should progress.

There is an existing consensus between HPC centres and industrial members of the ETP4HPC that such projects should create "ready to use" systems commensurate with exascale commercial objectives. They should encourage a strong co-design approach between technology and applications providers. They would produce tangible results which validate the capabilities produced in the preceding H2020 work programmes. These EsDs should provide platforms deployed by HPC centres and used by CoEs for their production of new and relevant applications. The fully integrated EsD systems should not be confused with systems/subsystems prototyped as part of individual research projects but as synergetic and integrated platforms. At project end, the EsDs will have a higher TRL[28] (Technical Readiness Level) of 7-8 (compared to 6-7 of prototypes as part of the projects), thus, their stability and usability will enable stable application production at reasonable scale[29]. Therefore, the EsDs will be 'stepping stones' towards a more expedited and solid commercial exploitation of the underlying system design and technology. The subsequent commercial exploitation of EsD output in preparing exascale level products and/or component technologies is left to the participating industrial partners. Whilst for the integration of EsDs the target is to deploy technology developed in the FETHPC programmes, the EsD projects need to be open to also include relevant technology developed outside of this programme, i.e. in the other parts of the H2020 programmes or in the global market.

To be clear, the purpose of creating and using the EsDs is fundamentally different from procuring 'big commercially available production systems' e.g. by Tier 0 centres within PRACE. The EsDs are meant to validate and prove the advancements in R&D performed within the H2020 HPC work programmes and gather valuable feedback for future projects, whilst the periodically procured commercial HPC systems are geared towards providing a robust compute infrastructure available to large user communities. However, a fraction of cycles from the EsDs, once deployed and stable, should be made available to members of larger user groups, with well-established allocation mechanisms to also expose the technology to the wider community. The EsD projects should, therefore, help industrial (and also SME and Mid-Cap) users to prepare for the next step in their HPC usage.

[28] Metrics used according to
www.acq.osd.mil/chieftechnologist/publications/docs/TRA2011.pdf

[29] A performance target and size around 5% of the peak systems at that time is recommended.

In summary, the EsD projects will fill the following important gaps in the current HPC H2020 programme:

· Bringing the technologies developed in the FET-HPC programme and related H2020 R&I closer to commercialisation, thus, fostering exploitation and take-up of these technologies;

· Benefiting from targeted R&D efforts across many projects and combine components into an integrated system;

· Providing the missing link between the three pillars of the HPC strategy: technology providers, infrastructure providers, and user communities through projects that leverage their respective expertise to develop new high-end compute platforms.

8.1 APPROACH

ETP4HPC proposes that EsD projects will be set-up as 'dedicated R&I projects' within WP2018/19, maximising the level of co-design evolving since WP14/15. The current view of ETP4HPC is to have two sets of EsD calls, each one leading to one or two projects. Each of the EsD projects will be structured in two phases:

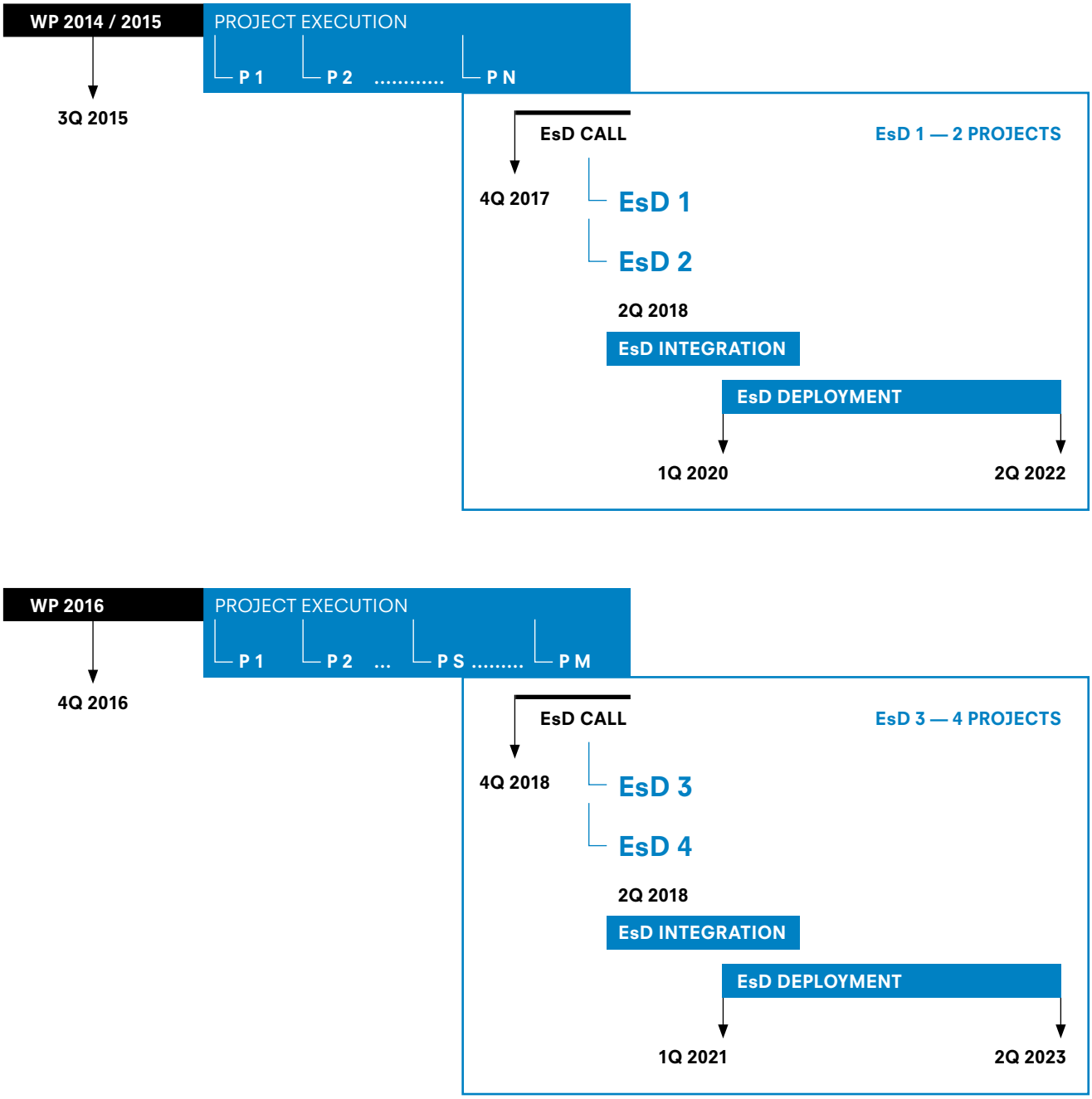
· **Phase (A)**: Development, Integration and Testing, involving little or no basic technology research projects, will have a substantial R&D focus mostly geared towards integrating and customising hardware and software components and sub-systems developed in the preceding R&D projects.

· **Phase (B)**: Deployment and Use, where the EsD is validated and operated by a hosting centre and made available to application owners for code porting and development to address numerical/extreme data challenges as well as characterisation and platform validation based on real use cases.

The following figure 4 shows the relationship between the EsD projects and the work programmes WP2014/15 and WP2016/17:

Figure 4
The relationship between the EsD projects and the work programmes WP2014/15 and WP2016/17

EXTREME SCALE DEMONSTRATORS CALL-INTEGRATION-DEPLOYMENT SCHEDULE



ETP4HPC suggests scheduling a block of EsD projects immediately after each of the first two work programmes. Each block should fund between one and two EsD projects, according to the available budget and the co-funding structure.


PE1

8.2 PROPOSAL OF ETP4HPC FOR THE ESD CALLS

The EsD calls will have a high dependency on the outcome of WP2014/15 and WP2016/17 projects regarding their timing, but mostly regarding contents. The portfolio of accepted projects in the work programmes must provide a sound technology basis for building EsDs, and the accepted projects should be actively encouraged to foster cross-project interlock. The structure of the WP16/17 should support cross-project integration, with particular regard to IP visibility and licensing clarity. Little coherence between accepted projects, too many disjoint focus areas, and insufficient technology options and readiness might otherwise jeopardise the success of the EsD calls.

It is proposed that calls should be announced within the WP 2018-2019. It is proposed that the EsD project calls will have a funding envelope compatible with a spending of €20-40M (30-50% R&D and 50-70% parts costs) per EsD project for phase A and €3-6M for phase B to cover utilities, operation-manpower and maintenance. Phase A should have a duration of 18-24 months and phase B of 24 months with a validation feedback checkpoint after 9 months. Therefore, total project duration of 32-48 months is envisaged.

The EsD characteristics will need to be further refined, however, they should deliver a high enough TRL to support a stable and effective production environment in their respective Phase B. Their impact on commercial product lines is not expected before 2020. Looking at the hardware characteristics, it is expected that the EsD architectures target scalability of applications up to 200 Pflops. This and other hardware characteristics (energy efficiency, I/O bandwidth, resiliency, etc.) will be detailed in the 2017 release of the SRA, also taking into account results from the FETHPC projects and requirements from the CoEs.

ETP4HPC recommends suitable projects to involve three types of partners for EsD projects: technology providers, application owners and HPC centres.

The role of **technology providers** (with a key role for system integration) will be to ensure the integration of technology, the project management, the testing and quality/performance assurance during phase A. The system integration will be the focal point for maintenance and service during phase B.

The role of the **application owner** will be to define application requirements and key challenges, which requires Tier-0 type resources, and that can be addressed by EsD during phase A. During phase B, they will port and optimise application(s) to EsD and use EsD productively.

The role of participating **HPC centres** will be to participate in the co-design process and to manage system deployment during phase A. Furthermore, they will operate the EsD, validate and characterise the system prototypes (in terms of performances, robustness, efficiency, etc.) during phase B.

9.

ECOSYSTEM AT LARGE — STAKEHOLDERS & INITIATIVES

Complementary to the R&D effort, the development and the coordination of the European HPC ecosystem is a key element for the success of European HPC policy. This section presents some of the initiatives currently in place and also the way ETP4HPC plans to interact with them. The vision of HiPEAC[30] will also remain an important source of reference in the area of general IT and computing technologies.

[30] European Network of Excellence on High Performance and Embedded Architecture and Compilation, an EU Framework Programme 7 project, www.hipeac.net, 'HiPEAC Vision 2015' is available at www.hipeac.net/assets/public/publications/vision/hipeac-vision-2015_Dq0boL8.pdf

9.1 EUROPEAN EXTREME DATA AND COMPUTING INITIATIVE

EXDCI (European eXtreme Data and Computing Initiative) is a project that supports the coordination of the Strategy of the entire European HPC Ecosystem (i.e. Technology Provision, Research Infrastructure and Application Expertise) by delivering the tools, measurements and other mechanisms needed to deliver that strategy between 2015 and 2017. The project is coordinated by PRACE and it also involves ETP4HPC. Many members of either organisation are third parties in this project. Also, members of the EESI and EESI2 (European Exascale Software Initiative), and previous projects, have joined EXDCI.

EXDCI will support the preparation of the next ETP4HPC Strategic Research Agenda (HPC Technology Roadmap), and the next issue of PRACE Scientific Case of 2012. Amongst its other activities are: the production of Key Performance Indicators monitoring the progress of the ecosystem, a training and education initiative aimed at attracting young talent and job creation as well as mechanisms for international collaboration and SME development. EXDCI also contains a task dedicated to cross-cutting issues to ensure that all 'roadmapping' activities across the entire ecosystem are synchronised. EXDCI will collaborate with Eurolab-4-HPC, which is presented in the next section.

EXDCI will also work with BDEC[31] (Big Data and Extreme-Scale Computing) including organising a European BDEC workshop and attending similar events in the US and Japan. This international initiative, (being a continuation of IESP - International Exascale Software Project) has gathered experts from the United States, the European Union, and Japan over the last three years. BDEC has embraced the *"idea that we must begin to systematically map out and account for the ways in which the major issues associated with Big Data intersect with, impinge upon, and potentially change the national (and international) plans that are now being laid for achieving exascale computing"*. This resonates with ETP4HPC's approach, e.g. as expressed in SRA 1 in 2013.

[31] www.exascale.org/bdec

9.2 EUROLAB4HPC

EXDCI runs in parallel with the Eurolab4HPC project in the area of Excellence in High-Performance Computing Systems. To compete internationally, Europe must bring together the best research groups to tackle the long-term challenges for HPC. These typically cut across layers, e.g., performance, energy efficiency and dependability, so excellence in research must target all the layers in the system stack. The EuroLab-4-HPC project's overall goal is to build connected and sustainable leadership in high-performance computing systems by bringing together the different and leading performance orientated communities in Europe, working across all layers of the system stack, and, at the same time, fuelling new industries in HPC. The aim is to boost European research excellence on the key challenges towards the next generations of high-performance computing systems (such as energy efficiency, complexity, dependability and cutting across all levels – hardware, architectures, programming, applications).

ETP4HPC and Eurolab4HPC will work together to issue consistent roadmaps and to maximise the impact of the research project especially by the creation of start-ups and the development of SMEs.

9.3 CENTRES OF EXCELLENCE IN COMPUTING APPLICATIONS

The 9.3 Centres of Excellence in Computing Applications (CoEs) form one of the three pillars of the European HPC Ecosystem and represent the European Application expertise[32]. The current CoEs are a result of a €40M EC H2020-EINFRA-2015-1[33] Call, which specifies the establishment of 'a limited number of Centres of Excellence (CoEs) necessary to ensure EU competitiveness in the application of HPC for addressing scientific, industrial or societal challenges. CoEs will be user-focused, developing a culture of excellence, both scientific and industrial, placing computational science and the harnessing of 'Big Data' at the centre of scientific discovery and industrial competitiveness.'

· User-driven, with the application users and owners playing a decisive role in governance;

· Integrated: encompassing not only HPC software but also relevant aspects of hardware, data management/storage, connectivity, security, etc.;

· Multi-disciplinary: with domain expertise co-located alongside HPC system, software and algorithm expertise;

· Distributed with a possible central hub, federating capabilities around Europe, exploiting available competences, and ensuring synergies with national/local programmes;

Each CoE is expected to deliver a tangible return on investment to its customers, with a view to develop a semi-sustainable operational model in the following call.

Following the closure of the call, at the time of writing this SRA Update, CoEs are being established in the following areas:

· **Energy**

· **Molecular biology**

· **Weather/Climate**

· **Tools/Performance**

· **Global systems**

· **Materials and Atomic/Molecular simulation**

ETP4HPC will be working to include the CoEs in the processes of the contractual Public-Private Partnership for HPC and synchronise their efforts with those of the other two pillars of the European HPC Ecosystem (i.e. ETP4HPC and the FETHPC projects, PRACE).

[32] The current CoEs are listed under: www.ec.europa.eu/programmes/horizon2020/en/news/eight-new-centres-excellence-computing-applications

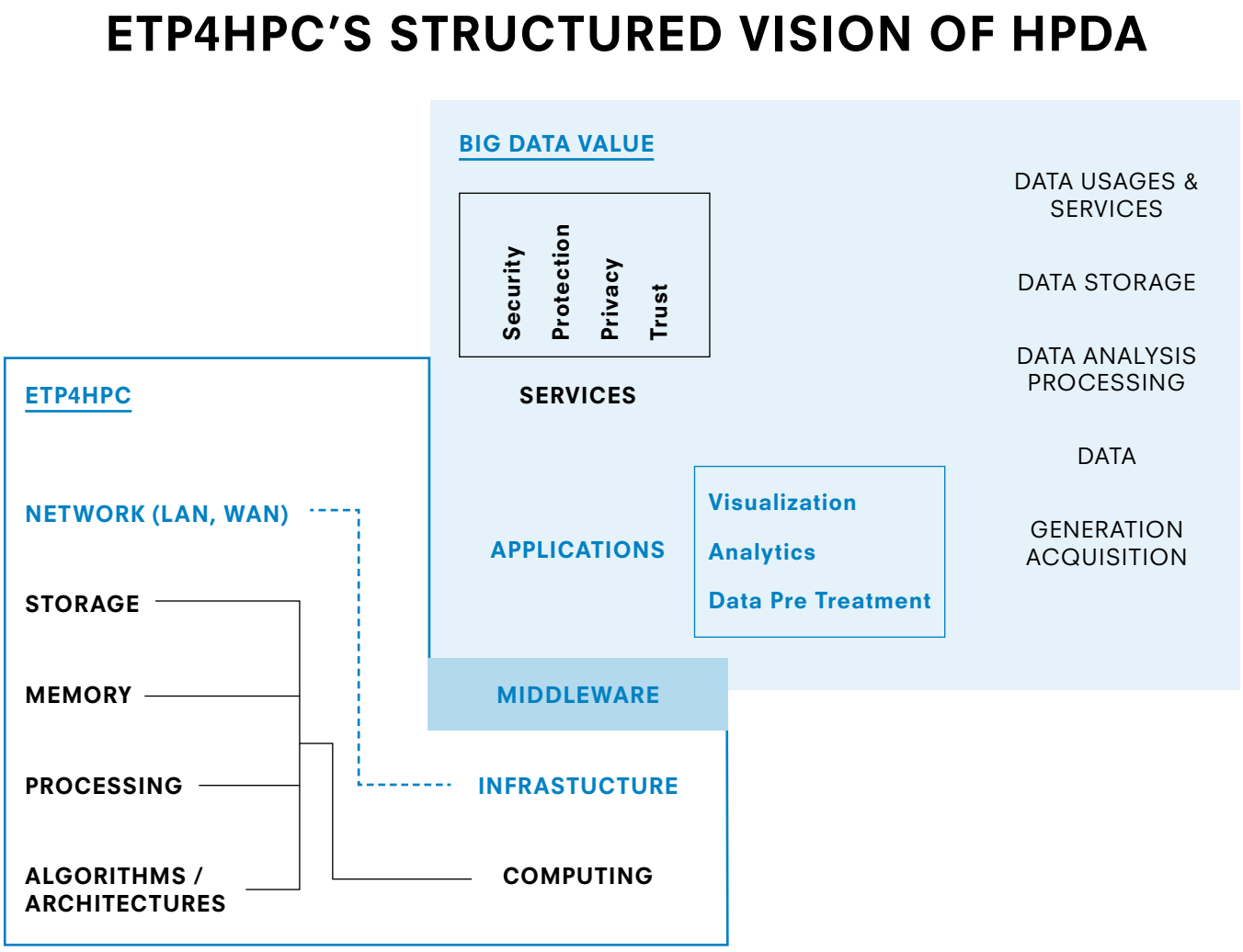
[33] www.ec.europa.eu/research/participants/portal/desktop/en/opportunities/h2020/topics/329-einfra-5-2015.html

9.4 BIG DATA VALUE ASSOCIATION

The Big Data Value Association[34] (BDVA) is an industry-led contractual organisation that is the European Commission's partner in the implementation of the Big Data Value cPPP.

BDVA and ETP4HPC have already held two working sessions with technical representatives of both parties, focusing on defining common ground for their mutual technical roadmap. The base understanding is that HPC technology is the underpinning compute infrastructure for High-Performance Data Analytics as shown in Figure 5:

Figure 5
ETP4HPC's structured vision of HPDA



[31] www.bdva.eu

BDVA has a multitude of non-technical challenges (e.g. addressing legal and social issues, building a business model). The advancements of the compute infrastructure are a long-term element. HPDA is using existing infrastructure and the process of optimisation for high demand use cases is only starting now.

The increasing use of HPDA will pose additional, in some areas much harder, requirements into the design of future system generations. Although there are no immediate bottlenecks with current HPC design points identified, today, the Big Data community is using whatever IT infrastructure is available to them. The HPC technology providers, ISVs, software vendors need to prepare for the handling of massive amounts of extremely diverse types of data coming from a multitude of sources in the next 5-10 years. The HPC ecosystem needs to be ready for this change driven by Big Data.

BDVA and ETP4HPC have agreed to synchronise BDVA's Strategic Research and Innovation Agenda (SRIA) and ETP4HPC's SRA in a top-down approach by analysing high demanding Extreme Data use cases and extracting HPC research priorities.

10.

CONCLUSIONS AND OUTLOOK

This SRA update has seen a significant increase in the participation of experts – this document is the collective work of 170 experts from over 45 member organisations, facilitated in 8 working groups. It is a sign of the raising awareness of HPC, HPC technology provision and the importance of the European HPC technology 'roadmapping' exercise. ETP4HPC is committed to maintain this process throughout the Horizon 2020 programme.

SRA 2 is an update to SRA 1, with a strong focus on renewing the technical content presented in Chapters 4, 5 and 6. The non-technical chapters of SRA 1 are still valid as the global strategic approach recommended by ETP4HPC.

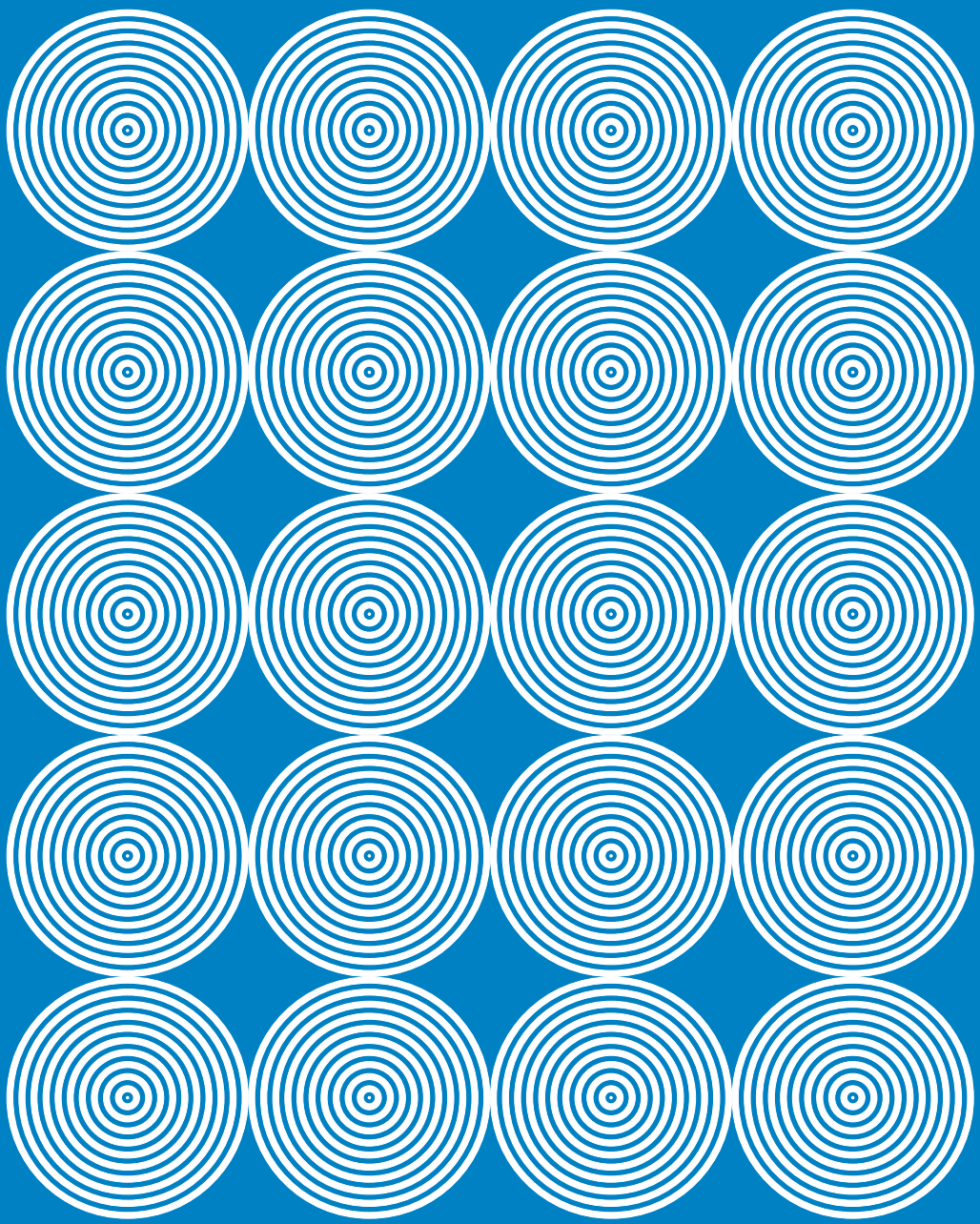
Besides the updated priorities in the six domains introduced in SRA 1, this SRA includes a new technical domain (Mathematics and algorithms for extreme scale HPC systems, Chapter 5.7) which is essential to align the HPC technology with the expanding HPDA use mode. In this document, ETP4HPC recommends the new concept of ''Extreme-Scale Demonstrators'', a critical proof-point of the effectiveness of the HPC research executed within the work programmes 2014/15 and 2016/17 (Chapter 8).

SRA 2 will also be the basis for further in-depth interactions with other HPC and HPDA, e.g. BDVA, the newly established Centres of Excellence, the partners in the EXDCI project and Eurolab4HPC. The goal is to further synchronise research agendas and priorities and extend the network of stakeholders in order to increase the cohesiveness of the research domains suggested in future calls.

The implementation of this SRA will be monitored by the HPC contractual Private-Public Partnership (which was established following the issue of SRA 1). It serves as an effective framework to accelerate the development of HPC in Europe. The goal of ETP4HPC in this partnership should be to facilitate the implementation of R&D projects addressing the research priorities presented in this SRA 2 to ensure European leadership in HPC. The success of this effort will benefit our HPC ecosystem and also our science, industry and society.

ETP4HPC will publish the final master version on its webpage (www.etp4hpc.eu) and will open a public call for feedback to all HPC stakeholders.

11.



GLOSSARY

API	Application Programming Interface
BDVA	Big Data Value Association
BDEC	Big Data and Extreme Computing
CAGR	Compound Annual Growth Rate
CAPEX	Capital expenditure
CERN	European Organization for Nuclear Research
CFD	Computational fluid dynamics
CIO	Chief Information Officer
CMOS	Complementary Metal Oxide Semiconductor
CPU	Central Processing Unit
CSM	Computational Structural Mechanics
DOE	Department of Energy
DRAM	Dynamic Random Access Memory
DVFS	Dynamic frequency and Voltage Scaling
EC	European Commission
ECC	Error Correcting Code
EESI	European Exascale Software Initiative
EOFS	European Open File System
ESA	European Space Agency
EsD	Extreme-Scale Demonstrator
ETP	European Technology Platform
FDSOI	Fully-Depleted Silicon on Insulator
FLOPS	Floating Point Operations per Second
GPU	Graphical Processing Unit
HLRS	Höchstleistungsrechenzentrum Stuttgart
HPC	High-Performance Computing
HW	Hardware
I/O	Input/Output
ICT	Information and Communication Technology
IDC	International Data Corporation
ILM	Information Lifecycle Management
ISV	Independent Software Vendor
ITIL	IT Infrastructure Library according to ISO 20001
KPI	Key Performance Indicator
MPI	Message Passing Interface
MTBF	Mean Time between Failure, a reliability indicator
NSF	National Science Foundation
NVRAM	Non-volatile random-access memory
OEM	Original Equipment Manufacturer
OPEX	Operational expenditure
PCB	Printed Circuit Board
PCM	Phase Change Memory, a new memory technology
PDE	Partial Differential Equation
PGAS	Partitioned Global Address Space, a programming model
PPP	Public Private Partnership
PRACE	Partnership for Advanced Computing in Europe
PUE	Power Usage Effectiveness
QoS	Qualities of Service
R&D	Research and Development
RAS	Reliability, Availability and Serviceability
RDF	Resource Description Framework
RT	Run-Time
SaaS	Software as a Service
SCM	Storage Class Memory
SDC	Silent Data Corruption
SIC	Application-Specific Integrated Circuit
SKA	“Square Kilometer Array” program (www.astron.nl/r-d-laboratory/ska/ska)
SME	Small and Medium-Sized Enterprise
SoC	System on a Chip
SOI	Silicon on Insulator, a CMOS technology term
SRA	Strategic Research Agenda
ST-MRAM	Spin-Torque Magnetic Random Access Memory
SW	Software
SWOT	Strengths, Weaknesses, Opportunities and Threats – a recognised strategic analysis tool
TCO	Total Cost of Ownership
UI	User Interface
WAN	Wide Area Network

12.
CONTRIBUTIONS & ACKNOWLEDGEMENTS

SRA Editorial Board
Michael Malms
Jean-Philippe Nominé
Marcin Ostasz

Steering Board Members
Bernadette Andrietti
Carlo Cavazzoni
Hugo Falter
Fabio Gallo
Jean Gonnord
Joerg Heydemueller
Jean-François Lavignon (ETP4HPC Chair)
David Lecomber
Thomas Lippert
Malcolm Muggeridge
Oliver Oberst
Franz-Josef Pfreundt
Ian Phillips
Francesc Subirada
Frank van der Hout

Workgroup coordinators
Costas Bekas
Paul Carpenter
Marc Duranton
Thomas Eickermann
Hans-Christian Hoppe
Guy Lonsdale
Malcolm Muggeridge
Jean-Pierre Panziera
Dirk Pleiter
Pascale Rosse-Laurent
Igor Zacharov

Document Editing
Catherine Gleeson

ISV AND END-USERS

Independent Software Vendor Representatives
Benedetto Risio
Gérard Lecina
Mark Loriot
Pierre-Aimé Agnel
Thomas Schumacher

End-User Representatives
Alain Beuraud
Serge Bogaerts
Ricard Borrell
Norbert Bourneix
Carlos Falconi
Alfred Geiger
Mauricio Hanzich
François Legrand
Gaël Mathis
Yves Mayadoux
Hugues Prisker
Bernard Querleux
Christian Saguez
Paul Selwood
Denis Wouters
Francesco Zen

Other Technical Contributors
Jean-Thomas Acquaviva
Chris Adeniyi-Jones
Pierre-Aimé Agnel
Vassil Alexandrov
Ahmed Al-Jarro
Piero Altoè
Manuel Arenaz
Mike Ashworth
Edouard Audit
Rosa Badia
Santiago Badia
Frank Baetke
Andrea Bartolini
Javier Bartolome
Valeria Bartsch
Peter Bauer
Rob Baxter
Toine Beckers
Francesco Benincasa
Martin Bernreuther
Jean-Yves Berthou
Balakrishnan Bhaskaran
Torsten Bloth
Thomas Blum
Francois Bodin
Thomas Boenisch
Francesco Bongiovanni
Ricard Borell
Daniele Bortolotti
Sven Breuner
André Brinkmann
Luigi Brochard
Jochen Buchholz
Mark Bull
Sébastien Cabaniols
Eddy Caron
Marc Casas
Gastone Castellani
Alexey Cheptsov
Nikolaos Chrysos
Steffen Claus

Guillaume Colin de Verdière
James Coomer
Toni Cortes
Stefano Cozzini
Pooyan Dadvand
Gabriele Dangelo
Patrick Demichel
Martijn DeVries
Björn Dick
Marc Dollfus
Iain Duff
Benoit Dupont de Dinechin
Joe Duran
Hartmut Fischer
Jose Flich
Bruno Franzini
Valentin Fuetterling
Rafael Gadea
Michael Gienger
Charles Gillan
John Goodacre
Klaus Gottschalk
Jose Gracia
Alan Gray
Daniel Gruenewald
Jordi Guitart
Gaetan Hains
Michael Hennecke
Andreas Hildebrandt
Bernhard Homoelle
Guillaume Houzeaux
Ally Hume
Adrian Jackson
François Jeanmougin
Nick Johnson
Yann Kalemkarian
Georgios Karakonstantis
Manolis Katevenis
Frank Kautz
Dmitry Khabi
Peter Kilpatrick
Axel Koehler
Samuel Kokh
Michael Krajecki
Jens Krueger
Martin Kuehn
Uwe Kuester
Jesus Labarta
Jacques-Charles Lafoucrière
Pierre Lagier
Timothy Lanfear
Stephane Lanteri
Kirill Larin
Erwin Laure
Laurent Lefevre
Kåre Løchsen
Mark Loriot
Herve Lozach
Michael Lysaght
Rui Machado
Paolo Maggi
Pekka Manninen
Filippo Mantovani
Manolis Marazakis
Xavier Martorell
Moreno Marzolla
Matteo Masetti
Giovanbattista Mattiussi
Iakovos Mavroidis
Eric Michel
Miquel Moreto
Thomas Moschny
George Mozdzynski
Sai Narasimhamurthy
Krzysztof Nawrocki
Christoph Niethammer
Dimitrios Nikolopoulos
Geraint North
Peter Oliver
Michael Ott
Adam Padee
Mark Parsons
Ursula Paul
Christian Perez
Tiago Quintino
Mirko Rahn
Holm Rauchfuss
Arnaud Renard
Alejandro Rico
Peter Robinson
Noam Rosen
Riccardo Rossi
Francesco Ruffino
Einar Rustad
Nico Sanna
Vijay Saravane
Marie-Christine Sawley
Heiko Schick
Bertil Schmidt
Bernhard Schraeder
Marc Alexander Schweitzer
Horst Schwichtenberg
Andrey Semin
Lorna Smith
Thomas Soddemann
Ivor Spence
Ingolf Staerk
Dimitar Stoyanov
Adrian Tate
Tolga Tekin
Ilian Todorov
Amitabh Trehan
Beppe Ugolotti
Osman Unsal
Hans Vandierendonck
David Vicente
Xuan Wang
Nils Wedi
Christian Weihrauch
Michèle Weiland
Gilles Wiber
Andreas Wierse
Niall Wilson
Bernd Winkelstraeter


TOSHIBA

IMPRINT

© ETP4HPC

Text:
ETP4HPC

Graphic design and layout:
www.workship.es

Paper:
Bio top 250 g/m²
Bio top 120 g/m²

Printed and bound in Barcelona in February 2016:
www.grafiko.cat

Contact ETP4HPC:
office@etp4hpc.eu

www.etp4hpc.eu