Deliverable D5.3

WP5 Final Report on

Cross Cutting Issues Working Groups

CONTRACT NO	EESI2 312478
INSTRUMENT	CSA (Support and Collaborative Action)
THEMATIC	INFRASTRUCTURE

Due date of deliverable:	31/07/2015
Actual submission date:	31/07/2015
Publication date:	31/07/2015

Start date of project: 1 September 2012 Duration: 30 months

Name of lead contractor for this deliverable: PRACE-CINECA

Authors: Giovanni Erbacci (CINECA), Francois Bodin (CAPS), Anne-Laure Popelin (EDF), Simon McIntosh-Smith (Bristol Univerity), Thomas Ludwig (DKRZ), Franck Cappello (INRIA), Carlo Cavazzoni (CINECA), Marie-Christine Sawlay (Intel).

Name of reviewers for this deliverable:

Abstract: This is the final report of EESI2 WP5 Cross Cutting Issues Work Groups. The document reports the outcome and the results of the five working groups identified in EESI-2-WP5 to built a vision and a roadmap on cross cutting issues.

Revision 1.0

Project co-funded by the European Commission within the Seventh Framework Programme (FP7/2007-2013)		
Dissemination Level		
PU	Public	X
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Table of Contents

Glossary

Abbreviation / acronym	Description
ABFT	Algorithm-Based Fault Tolerance for Linear Algebra
ADS	Adaptive Directional Stratification
CDF	Cumulative Distribution Function
CPU	Central Processing Unit
DOE	Designs of Experiments
DAKOTA	Design Analysis Kit for Optimization and Terascale Applications
DARPA	Defense Advanced Research Projects Agency
DDDC	Double Device Data Correction
DDDC+1	memory enhanced Double Device Data Correction
DDR	Double Data Rate
DIMM	Dual In-line Memory Module
DRAM	Dynamic Random Access Memory
DVFS	Dynamic Voltage and Frequency Scaling
ECC	Error Correcting Code
EEHPC	Energy efficient HPC
EESI	European Exascale Software Initiative
EIOW	Exascale I/O Workgroup Middleware
ESREDA	European Safety, REliability and Data Association
FEM	Finite Element Methods
FeRAM	Ferroelectric RAM
FFT	Fast Fourier Transform
FORM	First Order Reliability Method
GPU	Graphics Processing Unit
HPC	High Performance Computing
IESP	International Exascale Software Project
IO	Input Output
I/O	Input Output
LARS	Least-Angle Regression
MPI	Message Passing Interface
NTV	Near-Threshold Voltage
NVRAM	Non-Volatile RAM
PCE	Parametric Constrain Evaluation
PCM	Phase-Change Material
PCRAM	Phase-Change RAM
PDE	Partial Differential Equation

RAID	Redundant Array of Inexpensive Disks
RAM	Random Access Memory
RMA	Remote Memory Access
ReRAM	Resistive RAM
RBD	Random Balance Designs
RBM	Reduced Basis Models
R&D	Research & Development
SDCs	silent data corruptions
SPMD	Single Program, Multiple Data
ST-MRAM	Spin-Torque Magnetoresistive RAM
STTRAM	Spin Torque-Transfer RAM
TBB	Task Building Blocks
VVUQ	Verification, Validation, and Uncertainty of Quantification

1. Executive Summary

This document represents the final report of the activity of EESI 2 WP5 “Cross cutting issues Work Groups”.
The Cross cutting issues address themes transversal to the different activities from applications to technologies making the activity in WP5 synergic to the activity in WP3 “ Applications” and WP4.”Enabling Technologies”. The five working groups (WG) on cross cutting issues have addressed the following actions:

WG 5.1 Data Management and Exploration: One of the major challenge of Exascale applications addressing scientific discovery nowadays. The issue is central for the organization of the scientific discovery workflow and aims to set up actions to address end-to-end techniques for efficient disruptive I/O and data analysis, involving the full life-cycle of data.

WG 5.2 Uncertainties (UQ/V&V): Science in the Exascale era involves computational models and applications which are multidisciplinary and complex, involving a huge amount of parameters and variables, so the verification, validation and uncertainty quantification of computer models' results becomes fundamental both for industry and academia. Identify methodologies and enhance tools for the analysis of these uncertainty sources, is fundamental for the exploitation of Exascale applications.

WG 5.3 Power & Performance: The power monitoring and power management at all levels of the system architecture, addressing energy efficient performance of applications, is a crucial issue to address in the Exascale era. Guidelines have been addressed in this area underlining the need of standards in parallel with the urgency of formation of professional HPC developers experts in green programming methodologies.

WG 5.4 Resilience: Robust fault tolerance protocols as well as performing checkpoint/restart methods, to increase the efficiency of Exascale systems, are becoming urgent to manage the fore coming systems with millions of cores. The activity in this WG continued the gap analysis between existing reports and projection about the resilience challenge for Exascale simulation, started in EESI 1. The set of recommendations based on this gap analysis have been better focalized and an holistic approach for resilience has been recommended.

WG 5.5 Disruptive Technologies: The roadmap toward Exascale and beyond will be guided and modeled by disruption in semiconductor technologies, I/O and memory technologies, cooling technologies and facility management, networking and data transfer technologies. The disruptive technologies analysed have been further investigated and the software implication have been considered to address specific recommendations for supporting programmability, efficiency and productivity of tools and applications, energy aware, at Exascale.

This deliverable represents the final report of WP 5, summarizing the findings of the different WGs along the project lifetime and documents the contributes to the final recommendations that have been produced by EESI 2, mainly the following ones:

- Holistic approach for extreme heterogeneity management of Exascale supercomputers
- High productivity programming models for Extreme Computing
- Holistic approach to resilience
- Software Engineering Methods for High-Performance Computing
- Verification, Validation and Uncertainty Quantification tools evolution for a better exploitation of Exascale capacities
- Software for Data Centric Approaches to Extreme Computing
- In Situ Extreme Data Processing and better science through I/O avoidance in High-Performance Computing systems
- Declarative processing frameworks for big data analytics, extreme data fusion

2. WG 5.1 Data Management and Exploration

2.1 Objectives and origins of expertise

Workgroup 5.1 has addressed "Data management and exploration” in Exascale applications viewed as the organization of the scientific discovery workflow.

Data management is at the core of the design of Exascale applications. Efficiency at Exascale level requires breaking with the traditional scientific workflow where simulation data are stored on disk for later analysis. This disruption comes in syncronization with new memory technologies, new photonic networks as well as the decreasing cost of transistors. For instance new non-volatile memories (e.g. Memristors) hold the promise of providing persistent memories close to the CPU that are fast, large, energy efficient and at a reasonable cost. On the software side, big data and other in memory computing technologies may be providing new solutions to help scientists facing the coming deluge of data. Holistic approaches considering all data cycles from sensors capture to visualization, encompassing simulation, code coupling, in-situ, pre and post analysis can guarantee that no bottlenecks are introduced in the scientific discovery process. In particular, it is strongly wished that new systems simplify human-in-the-loop workflows and a data centric view on Exascale applications is fundamental.

This is the philosophy that has guided the work of this WG during the life time of the project as documented in [1] and [2]. Our vision was also reinforced by the participation to the last BDEC meeting (Barcelona, January 2015: http://www.exascale.org/bdec/documents/barcelona) with was a further possibility to share the EU, USA and Japan visions on these topics.

The working group consisted of a chair (Francois Bodin) and vice-chair (Giovanni Erbacci) and more ten experts chosen to cover the domains of interest. Their names and area of expertise are listed below:

Name	**Organisation**	**Country**	**Area of Expertise**
Francois Bodin (chair)	University of Rennes	France	Data Management, GPU
Giovanni Erbacci (vice chair)	CINECA	Italy	HPC infrastructures
Jean-Michel Alimi,	Observatoire de Paris	France	Big Data, Numerical Cosmology
Gabriel Antoniu	INRIA	France	Cloud Service, Storage systems: BlobSeer
Giuseppe Fiameni	CINECA	Italy	Big Data, Storage Infrastructure
Georges Hebrail	EDF	France	Data Mining
Jacques-Charles Lafoucrière	CEA	France	File system: Lustre
Malcolm Muggeridge	Xyratex	UK	Cloud Storage
Kenji Ono	Riken	Japan	Data reduction for large scale datasets
Stéphane Requena	GENCI	France	Big Data
Alex Szalay	Johns Hopkins University	USA	Large scalable Databases, Numerical modeling of Galaxy.
Jean-Pierre Vilotte	IPG	France	Computational Physics, Seismology

2.1 Data Centric View of Exascale Applications

Designing an Exascale application that make rational and efficient uses of communication, compute, storage resources requires engineer skills that are currently in shortage or just not available to scientists. New best practices will have to be defined and implemented. They will very likely require setting up interdisciplinary support team capable of addressing extreme parallelism, fault tolerance and IO issues.

The deluge of data requires new data analysis techniques. Big data technology may provide new disruptive methods for such task. These techniques need to be extended to take advantage of highly scalable parallel infrastructure. This may be a return contribution of HPC to the big data field. Behind this topic lies many complex and holistic issues such as: serialization/deserialization of data, design of data structures able to cope with highly asynchronous execution as well as compute / IO activities interleaving. More generally, data mining techniques must be extended to fit the file formats used in HPC (e.g. HDF5, netCFD) and bridges must be established between HPC and big data usual formats.

Metadata management and specification is also a critical challenge. They are keys elements in the science discovery process. Their design is particularly important to obtain a consistent end-to-end use of the data. Furthermore, they impact on sharing policy management implementation (e.g. at the core of the decision process concerning data to be set public, what storage migration, etc.).

Analysis and visualization of data produced by large-scale simulations are often sidelined in favor of pure computation performance. As we foresee Exascale systems in the next decade, the offline analysis approach shows its limits: more and more scientists see the scalability of their simulations dropping because of unmatched computation and I/O performance as well as higher I/O variability. However, in-situ[1] approaches (potentially more efficient) have difficulties in getting accepted, as scientists fear to dive into fundamental code changes in a simulation they have used for years. Defining the right tradeoff here is a challenge. Also related to the same limitation in I/O performance, HPC scientists predict fundamental changes in the way I/O and data management will be handled in the near future. In particular, the heterogeneous processor environment and memory hierarchy of the new platforms, together with the increasing use of GPU and accelerators, open new alternatives for data analysis.

This topic cannot be viewed only under the technology angle. Indeed, designing the applications requires finding tradeoffs between in-situ vs. ex-situ processing, selecting data format, access policy, data relocation, format changes, etc. These tradeoffs are not only driven by technology and performance but also by the ecosystem exposed to the researchers. Furthermore, It is important to note that a global efficient use of the Exascale resources can be contradictory with the objectives of individual research teams. Understanding the full cycle of data is probably the most important question to drive Exascale technology development.

2.2 New Technologies, New Challenges

The Exascale goal stimulates the seeking out of new solutions and requires the exploiting of new technologies being brought to market such as photonics communications, energy optimized processors, chip stacking, non-volatile memories (NVM), etc. In the following we discuss the case of NVM as this technology likely will introduce disruption in most of the elements of the software stack of an Exascale system and the data streams. NVM hold the promise of providing persistent memories close to the CPU that are fast, large and at a reasonable cost.

NVM have been announced since a long time (e.g. 2008) but still now the freely available information is not clear about the characteristics of such technologies. For instance, if we consider Resistive RAM (RRAM), also denominated Memristor by HP, different sources of information are still contradictory on the capabilities of such memories in terms of writing latency.

Nevertheless, despite the late arrival of this technology on the market, it cannot be ignored in the race to Exascale computers. HP that hopes to deliver operational systems with Memristor technology by 2018 has made the most aggressive announcement.

From the application point of view, having a large memory, with close to DRAM performance and persistence, is likely to introduce a revolution in application design.

[1] See the EESI recommendation on this specific topic.

As current roadmaps foresee that large NVMs attached to compute nodes will reach the market by 2018-2020, it is urgent to handle the software issues involved with this technology that very likely will be a disruption in the race to Exascale systems with wide implications on the design of Exascale applications.

Application development for Exascale systems is of a rare complexity. Complexity in scalability and in roadmapping the software. On one hand it is best if legacy codes can be reused, on the other hand it is likely that many codes will have to be deeply re-designed / re-developed. Domain specific approaches may be able to hide complexity to users but as they are more specific they address a smaller community. In the end, a tradeoff must be made between development cost (including the tools, API, maintenance, etc.) and the potential users base. As application software moves much slower than hardware technology we believe that anticipation is extremely crucial.

Applications must optimise the use of IO bandwidth thanks to interleaving compute and data transfers in a manner known/understood by the system. This requires new programming methods and tools.

2.2.1 Software stack design

The data management throughout the entire software stack is likely to be modified extensively when new memory technologies will be introduced. This will have to be taken into account in the development of the applications as well as the supporting software stack. In the following we list non-exhaustively some crucial aspects that will affect the software stack as a consequence of the new NVM technologies.

Compute vs storage: It has been anticipated that compute power will grow faster than storage capacity for HPC machine (number of core vs node main memory size). The availability of NVM may change this trend (it is important to note that the low energy consumption of NVM allows to provide very large space). However, as more memory is available inside the node, getting the data out may stress out the IO system.

IO stack: Speed of expected NVM is so fast that it requires revisiting the landscape of storage software stack: current storage software's latency (and corresponding energy) will become the main cost. In the end the status of NVM may neither be the one devoted to RAM nor the one devoted to usual storage. New application development will define what is expected from this new hardware capability.

Resilience: Constraint on the implementation of resilience can be fully revisited. Memristor performance may be fast enough to allow saving the processor state frequently enough in such manner that transient errors are completely handled at system lowest level and transparent to users. This means redirecting the research effort at higher-level functionalities, closer to the application.

Programming API: If NVM are to be used for application development it must be exposed to programmer with a standard efficient API. The list of challenges in designing this API is quite long since it must address data persistence (e.g. dealing with issues such as pointer address), resource management (e.g. NVM allocation and IO organization), energy management (e.g. since the data is persistent some part of a system can be turned off while participating temporarily), data sharing between nodes (e.g. PGAS, peer-to-peer exchanges) and performance.

In-situ analysis, pre/post processing: NVM technology carries the potential to turn compute centric-machine into data-centric systems. As a consequence the data analysis and processing tasks that where envision for a different kind of system (e.g. big data machine) can be efficiently implemented close to the compute part. This opens many opportunities for developing a new kind of scientific applications more data oriented. It should be noted that thanks to photonic-based network, it is possible to aggregate node NVM to built a very large memory space dedicated to data mining and other analysis tasks. Visualizing the data in such context is still to be clarified.

Code coupling: Code coupling may be made more efficient as well as simpler to implement. Current libraries for code coupling will need to be revisited to take into account this new storage. Furthermore, the coupling "frequency" might also have to be reconsidered at numerical scheme level.

Compiler and runtime technology: Because of energy management, heterogeneous hardware and system configuration compiler research has been studying auto-tuning and runtime libraries to adapt to runtime changing context (e.g. because some core are out, data size is different, etc.). This technique usually relies on code versioning, specialization and run-time code generation. All these require run-time performance analysis and code tuning during a discovering phase. Keeping local data on node over application execution (thanks to persistence) would help to reduce the cost of this phase that in many cases impact significantly performance.

Debugging, performance tools: performance tools such as Tau, Vampire, Paraver, Scalasca are based on tracing events on each node. With the increase of parallel activity storing the huge amount of events is challenging at Exascale level. The ability to store and process locality as much as keep performance history on each node is likely to help redesigning this tool to handle the massive parallelism (e.g. post-mortem analysis). Similar studies are also needed for designing the next generation for debuggers.

2.3 Recommendations

The issues and recommendations presented by this WG are intended to address "**End-to-end techniques for efficient I/O and data analysis**" to describe the full life-cycle of data for a set of applications in order to produce designs/workflows that are consistent all the way from the production to the analysis of the data while considering locality, structures, metadata, right accesses, quality of service, sharing etc. This action can encompass the following items:

- Research to simplify human-in-the-loop workflows for data-intensive science with virtual data facility and for heterogeneous full-service data lifecycle encompassing the full data provenance chain and reproducibility.
- Research to develop libraries of scalable data analytics and data mining algorithms - and software components for use in workflows - encompassing data abstractions, in-situ and out-of-core data processing modes, approximate data analytics, indexing-topological-statistical methods, uncertainty quantification.
- Research to support full-service data lifecycle management systems – and their wide diversity – supporting large distributed teams and internally organized scientific communities, encompassing on-the fly analysis, private and public reuse, archival and curation, together with metadata and metadata interoperability, data annotation and permanent identifiers, data and secondary data provenance chain, security and privacy, etc.
- Research in advanced data analytics algorithms and techniques, adopting new disruptive methodologies, to face the analysis of the big data deluge advancing in different scientific disciplines.
- Specification scenarios for technology deployment and the available options for organizing the data storage and processing flow.
- It is urgent to encourage the community to form multi-disciplinary research groups capable of handling the complete set of concepts necessary to design data centric approaches to Exascale computing. It is particularly important to also integrate ecosystem and economical issues.

During the second year, these recommendations have been further developed and integrated (see [3]), encouraging to address research activity toward:

- **In situ extreme data processing for better science through I/O avoidance in high-performance computing systems**. The ultimate goal of the in situ extreme data processing is to promote new data transformations and compressions that reduce drastically extreme raw data, generated during HPC simulations, by preserving the information required for a particular analysis while sacrificing most everything else and store the only relevant data.
- **Declarative processing frameworks for big data analytics**. Exascale systems provide an incredible huge amount of *synthetic* data that need to be processed in order to get a full understanding of what they simulate, and compared and processd with the incredible amount of diverse *real* data produced by the data acquisition system. Computer scientists and specialists of statistics used to manage and treat these data. The current Variety, Volume and Velocity of data imply a synergy and collaboration between different fields of science in order to extract full intelligence and knowledge from these data in close to real time.

3. WG 5.2 Uncertainties (UQ / V&V)

3.1 Objectives and origins of expertise

The ability to simulate very complex and inter-disciplinarily phenomena, taking into account the effect of a wide number of input parameters it is a chance which will heavily grow in the Exascale era, but, at the same time, we must point out that the quantitative uncertainty assessment of the results becomes a fundamental issue for assuring the credibility of computer model based studies, and represents a challenge too.

The most challenging point is to bridge the cultural gap between a traditional scientific and engineering deterministic viewpoint and the probabilistic and statistical approach which considers the result of a model as an "uncertain" variable. The step forward is to develop and to spread in the scientific and engineering community an enhanced unified framework for model verification & validation and uncertainty propagation, what is commonly called *VVUQ*.

HPC and uncertainty quantification have a two-sided relationship. From one hand, the ever increasing size of the computational data leads to increasing sources of uncertainties, due to the accumulation of numerical errors. From the other hand, HPC gives access to computational power that can be used to tackle explicitly the evaluation of uncertainties, be it by embedded methods or by design of experiments. The activity in WG 5.2, has explored the different aspects involved in the relationship between uncertainties and HPC.

The working group consisted of a chair (Vincent Bergeaud, substituted by Anne-Laure Popelin in the second year), a vice-chair (Alberto Pasanisi) and more eleven experts chosen to cover the domains of interest, among them, four newly specialists met in the second year of activity (Laurence Viry, Bruno Sudret, Stefano Marelli and Eric Phipps). The experts names and area of expertise are listed below:

Name	Organisation	Country	Area of Expertise
Vincent Bergeaud (1st year chair)	CEA	France	Uncertainty quantification, Uranie
Anne-Laure Popelin (2nd year chair)	EDF	France	Uncertainty quantification, Industrial Risk Management
Alberto Pasanisi (vice chair)	EDF	Italy	Uncertain Analysis, Bayesian Decision Theory, Numerical Simulations
Stefano Tarantola	JRC-ISPRA	Italy	Statistic, EU policies.
Christophe Prud'homme	University of Strasbourg	France	Applied Mathematics, Computer sciences,
Olivier Le Maitre	LIMSI, Duke University	USA	Uncertain Propagation, Chaos Theory, CFD, stochastic nonlinear problems
Renaud Barate	EDF R&D	France	Automatic Design, Uncertainty analysis tools
Bertrand Looss	EDF R&D	France	Monte Carlo methods, Environmental modeling, Geostatistics
Fabrice Gaudier	CEA	France	Data Mining, Modeling of Uncertainty, Nuclear Energy, URANIE Framework
Laurence Viry	University of Grenoble	France	Sensitivity analysis and quantification of uncertainty (stochastic approach)
Bruno Sudret	ETH Zürich	Switzerland	Risk, Safety and Uncertainty Quantification, Probabilistic engineering mechanics

Stefano Marelli	ETH Zürich	Switzerland	Risk, Safety and Uncertainty Quantification
Eric Phipps	SANDIA National Lab	USA	Uncertainty Quantification, Optimization

3.2 UQ challenges in HPC

To approach Exascale, it becomes fundamental to develop and to spread in the scientific and engineering community an enhanced unified framework for model verification & validation and uncertainty propagation. This unified framework need at the same time:

- multidisciplinary skilled teams (statistics & probability, numerical analysis, PDE, physicians),
- high computational power, as the statistical methods for calibration and validation need to evaluate several times a (possibly) costly numerical code.

The uncertainties in the numerical simulation process can arise from different sources:

- lack of knowledge on a physical parameter (epistemic uncertainty)
- parameter with a random nature (aleatory uncertainty)
- uncertainty related to the model (model error)
- uncertainty related to the numerical errors (numerical errors).

Taking into account these uncertainties is essential for the acceptance of numerical simulation for decision making. These uncertainties must be integrated in the verification and validation process of the simulation codes. Verification consists in checking that the equations underlying the code is correctly solved. Validation is the stage during which the predictive capability of the numerical model is checked against experimental data or a reference model.

During the activity of WP 5.2 an identification of the main methodologies for the analysis of the uncertainty sources have been accomplished: both the two main categories of embedded methods for uncertainty analysis have been investigated (adjunct methods and spectral methods), as well as the methodologies and the tools for the uncertainty analysis methods based on Designs of Experiments (DOE).

The realization of the DOE-based uncertainty analysis methods follows a pattern that is largely independent form the numerical models which are analysed. Therefore, cross-cutting tools have emerged that help the end user to perform the tasks associated to DOE-based tools:

- Problem specification
- Input variables uncertainty quantification
- Definition and realization of the DOE
- Computation of meta-models
- Computation of output statistical indicators.

A number of tools have emerged, some specific and other more generic which include only some of the aspects required for the uncertainty analysis procedure. The main tools analyzed are:

- **DAKOTA**: Multilevel Parallel Object-Oriented Framework for Design Optimization, Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis. (http://dakota.sandia.gov/papers/DAKOTA_Overview_Jan2010.pdf)
- **URANIE**: the Open Source Platform developed at CEA/DEN dedicated to the study of propagation uncertainties, sensitivity analysis or model calibration in an integrated environment.
- **OpenTURNS**: a open source software under LGPL licenses, specifically designed for non-intrusive uncertainty quantification.
- **UQLab project**: a distributed HPC environment Matlab-based platform for different applications in UQ and risk analysis in the fields of civil and mechanical engineering. The foreseen modules include revisited algorithms for reliability analysis, meta-modelling techniques (polynomial chaos expansions, Kriging, support vector machines), sensitivity

analysis (Sobol' indexes), Markov Chain Monte Carlo methods, etc. (see http://www.ibk.ethz.ch/su/research/uqlab_EN)

When dealing with UQ in challenge applications, some aspects must be considered and analyzed, i.e. the dimension input space, the non smoothness of the model responses and the need to accurately estimate some rare events probabilities. HPC brings some solutions to these problems.

A critical component of predictive simulation is computing derivatives of simulation responses with respect to model states and parameters. Such information is needed for steady-state nonlinear solves, implicit time integration, stability analysis. Moreover, derivatives can be useful for sensitivity analysis, optimization, meta-model approximation, Bayesian inference sampling, error estimation, and uncertainty quantification. However hand-coding derivatives is time-consuming, error prone, and difficult to verify, particularly for parameter derivatives, adjoins, and higher derivatives.

Automatic differentiation (AD) is a technique for computing analytic derivatives in simulation codes without hand-coding the derivative computation itself, and is based on simple mathematical and computer science principles (see http://www.autodiff.org). The main AD software tools are TAPËNADE, ADOL-C, ADIFOR. These tools are in Fortran and C but also some applications in interpreted languages such as Matlab and Python are considered in AD.

AD for C++ code is still a serious scientific challenge: Sacado, a Trilinos package for automatic differentiation of C++ codes, being developed at Sandia. Sacado is designed for incorporation into large-scale C++ codes and leverages the C++ language itself to implement AD using operator overloading and templating.

Another important point to address for UQ is the meta-model solution in the context of HPC. It is important to have a clear definition of the main processes involved, The following processes should be clearly defined:

- P1: parameterization,
- P2: sampling,
- P3: distribution of model input data,
- P4: results recuperation,
- P5: estimation of quantity of interest,
- P6: metamodel building if necessary,
- P7: model evaluation,
- P8: metamodel evaluation.

Then, for each process, must be described the characteristics of the calculation:

- CPU time cost Ti of one evaluation of the numerical model,
- Storage volume Vi,
- Communication (data exchange) volume Ci,
- Environnement software, process complexity.

All these methods are still in the process of investigation at research level. And not yet represent a consolidated methodology A huge activity must be undertaken to consolidate, enable and disseminate these methods for a wider community addressing different scientific and engineering applications.

3.3 Recommendations

3.3.1 Diffusion of tools and practices

Uncertainty analysis is a field that has drawn considerable interest over the past years. Advances in statistical analysis, numerics and computer science provide methods that are readily available and that are largely independent from the application domain. Software tools are therefore available that deal with different aspects of uncertainty analysis (Optimization, Surrogate Model creation, Sensitivity Analysis, Numerical Roundoff Error Accumulation, etc.).

The surge in computational power calls for taking into account uncertainty analysis in academic and industrial studies. The use of the uncertainty analysis methodologies require competence that is somewhat different from the ones required to develop a simulation code, and a key issue is that of training. The software tools are obviously very important for facilitating the uncertainty analysis dissemination in the numerical simulation community.

Incitation should therefore be given to make sure the tools keep up with the best practices in numerical methods, and to help the training effort required to make uncertainty analysis a common practice.

On top of the software tools, diffusion of methodologies amongst engineers and scientists can be accelerated via books and tutorials that offer good overviews of the methodologies.

3.3.2 Progresses in numerical analysis

Numerical methods exist to handle many aspects of the uncertainty analysis:

- Identification of uncertainty sources
- Propagation of uncertainty sources
- Sensitivity analysis
- Reliability studies
- Robust optimization
- Validation.

Adaptative design

Methods based on DOEs offer a framework which is largely independent from the numerical model and therefore enjoy a large success in the scientific and engineering community. The aforementioned methods are efficiently used on a very large variety of problems. The limit of such methods is the necessity to use hundrerds or thousands of simulations for one study, and therefore, the emergence of exascale computers will broaden the range of usability of these methods. However, for the applications for which the CPU-time consumption is very important, it remains crucial to be as effective as possible, and therefore to have design of experiments that are as efficient as can be.

For very computationally intensive applications, adaptative design of experiments can be useful to make sure that every new point in the design brings as much information as possible. Works on this domain should be encouraged.

Surrogate models

Another way to deal with computationnally intensive applications is the use of surrogate models or reduced models instead of the full computational models.

A traditional way to work is the use a metamodel reprensenting the relationship between the input variables and a few global output variables (kriging, neural networks, polynomial, etc.). Reduced basis models offer also interesting solutions for more complex cases in which the output cannot easily be restricted to a small number of variables (notably in the case of multiphysics couplings) : the complete solution is reconstructed from a learning set and a set of input parameters. Progresses remain to be achieved to better take into account the objectives of uncertainty analysis at the learning stage of the reduced basis methodology. Also, achieving the of use reduced basis methods in a non intrusive manner would significantly enlarge their potential scope of application and their usage by the scientific community.

Model error

The current techniques mostly focus the error related to parametric uncertainty, be it of aleatory or of epistemic nature. Validation process should take into account the numerical model errors in order to achieve better predictability and to gain understanding on the level of confidence of the codes. A significant methodological effort should be dedicated to this issue.

3.3.3 Specifications for future software and architectures

Taking into account DOE-based methods in middleware

When using supercomputer power, tools dedicated to DOE-based methods are closely connected with the batch systems of the machines. Generally speaking, developing generic solutions for exploiting supercomputers is made difficult by the heterogeneity of the batch systems deployed and the limitations imposed on the number of jobs available per user.

Middlewares that would allow good flexibility in terms of switching easily from large number of small jobs to small number of large jobs would make the exploitation of the DOE-based tools easier for the user.

DOE Checkpoint/restart

Another progress that must be achieved lies in the DOE tools themselves. They poorly take into account the problem of resilience to failures. Two problems are intermingled here: the tools have little capacity for rerunning points in design of experiments that have not completed. Also, tools have no capacity to distinguish between cases that failed for numerical reasons and cases that failed for reasons related to the batch. Progresses on this topic must definitely be made.

Multiple levels of parallelism

Last, modern multiphysics computations involve multiple levels of parallelism (domain decomposition, code coupling, multiscale, etc.). The platforms have yet to make progress to ensure these different levels of parallelism are well combined with the one related to the DOE for efficient parallelisation of the ensemble.

In the end the activity in WG 5.2t and these hrecommendations contributed to formulate the recommendation *“Verification Validation and Uncertainty Quantification tools evolution for a better exploitation of Exascale capacities”* as part of the *Tools & Programming Models Pillar.* See [3]. The recommendation aims at preparing an unified European VVUQ package for Exascale computing by identifying and solving problems limiting usability of these tools on advanced HPC systems, and furthermore to facilitate the access to VVUQ techniques for the HPC community.

4. WG 5.3 Power & Performance

4.1 Objectives and origins of expertise

In the quest to achieve Exascale systems in the 2020 timeframe, energy efficiency has become one of the primary challenges. Exascale designs projected unacceptably high power requirements in excess of 100 MW for each system, leading to a surge of research and development searching for breakthroughs in energy efficient hardware and software. Today, significant advances have been made in many areas, but there are many challenges still remaining that need to be addressed if we are to meet our goal of Exascale systems within a 20 MW power envelope. The activity in WG5.3 has the objective to address these challenges.

The working group consisted of a chair (Simon McIntosh-Smith) and vice-chair (Thomas Ludwig) and further eight experts chosen to cover the domains of interest, among them, three new experts added in the second year (Paul Carpenter, Daniel Hackenberg and Manuel Dolz). The experts names and area of expertise are listed below:

Name	Organisation	Country	Area of Expertise
Simon McIntosh-Smith (chair)	Bristol University	UK	Microelettronics, HPC, MIC, Machine Learning
Thomas Ludwig (vice chair)	DKRZ	Germany	High volume data storage, Energy efficiency, Performance analysis, Parallel computing, Meteo clime
Alex Ramirez	nVIDIA; former BSC	Spain	Energy efficient hardware, Computer architectures, ARM based HPC
Matthias Müller	RWTH, Aachen University	Germany	Energy efficient HPC, Scientific Computing, Performance tools
Jean-Marc Pierson	Paul Sabatier University	France	Energy aware HPC
Laurent Lefevre	Lyon University, INRIA	France	Energy efficient Computing and Networking
James Perry	EPCC, University of Edinburg	UK	Accelerator technologies, Embedded systems, Code optimisation
Paul Carpenter	Barcelona Supercomputing Centre	Spain	Parallel programming models, Energy-efficient supercomputing
Daniel Hackenberg,	TU Dresden	Germany	Accelerator technologies, Numerical algorithms, Code optimisation
Manuel Dolz	University of Hamburg	Germany	Energy efficient programming, TCO in HPC

4.2 Key challenges to achieve Exascale systems

The critical challenges to Exascale in the area of energy efficiency and power management have been investigated by WG 5.3. The challenges, rated as critical, important or nice are reported below (and further documented in [1] and [2]).

Ability to profile applications for energy efficiency (critical). It is increasingly apparent that as we progress toward Exascale systems, HPC is becoming energy limited, and so increasing the energy efficiency of a code will ultimately lead to increasing that code’s performance. Yet today the number of tools and techniques available to software developers to profile, understand and optimise the energy efficiency of the code running at scale is very limited, and what little is possible is via vendor proprietary solutions. We cannot improve the energy efficiency of software without addressing this fundamental problem. To solve it, we need to be able to accurately measure the energy consumption of a system at all levels of detail, from individual components within a CPU up to a system-wide view, which includes networking, storage and cooling. Appropriate levels of resolution are required for this energy monitoring. An accurate method of correlating application execution to the observed energy consumption is also imperative to enable an analysis of causal relation, eventually leading to control decisions for manipulation mechanisms. Fundamentally it is the lack of hardware support, standard APIs, and tools to gather and access this energy information in a meaningful way that is a threat to achieving Exascale systems within the 20 MW target power envelope.

Fine resolution power mode manipulation mechanisms in all devices (critical). While automatic systems for optimising energy consumption will achieve some success, components in a system need to have software-controllable mechanisms to switch them into low power consumption modes when being underutilized. This works for processors already but still needs to be implemented for many other components, e.g. main memory. We must enable the user-space runtime system and the application itself, to manage the power states of the hardware to optimize energy usage and limit power consumption. Currently this is left entirely to the hardware, or to the operating system, which must perform these management tasks based on heuristics and speculation, since they do not have any actual knowledge of what the application is doing.

Improving scalability to improve energy efficiency (critical). It is likely that clock speeds will have to be decreased in order to meet the power budget specified for Exascale systems. This means that overall concurrency of compute will have to be significantly increased, not only to bridge the gap between Petascale and Exascale, but also to offset the slower clock speed. It is likely this trend to rapidly increase core counts in place of increasing clock speeds will be long term, and so an initiative to improve the scalability of our commonly used HPC codes will potentially have a big positive impact on their energy efficiency.

Model power consumption (critical). For a given application we should be able to model and determine its power consumption behaviour. Appropriate knowledge will help guide scheduling decisions. The set of running applications will determine the overall power consumption of the HPC system. In future we want to control this in order to stay in a defined power budget.

Dynamic, energy aware load balancing across heterogeneous resources (important). As nodes and systems become increasingly parallel (more cores, wider vectors) and potentially heterogeneous (GPUs, Xeon Phi), being able to exploit all of these resources to maximise performance and performance per unit energy are unsolved problems. Recent advances such as dynamically varying frequency and voltage (DVFS) for processors further complicate this issue: a more energy efficient application may result in a lower operating temperature, which could in turn enable a higher operating frequency and thus higher performance. Research into how applications can best exploit this phenomenon is needed, and techniques are required which will be easy for mainstream HPC developers to adopt without having to reinvent this wheel for each application.

Conduct overall benefit-cost-ratio analysis (important). We also have to conduct an overall analysis that leads to a measure of cost per scientific result. Energy consumption is one factor here. However, one might find that it is better to invest more in people instead of in ever more hardware components. Energy consumption is currently one of the biggest contributors to the overall operation cost of a system – and the one with the largest growth rate. However, it does not make sense to consider energy efficiency without integration into TCO. A more energy efficient system only makes sense if the additional costs have a return on investment that is shorter than the lifetime of the system. An effort to improve the energy efficiency of a large application only makes sense if the development costs are smaller than the saved energy costs.

Develop application benchmarks to measure energy efficiency (important). To measure the energy efficiency of different computer architectures and to drive the further development it is crucial to have energy efficiency metrics beyond simple Flops/Watt. Proper application benchmarks including run rules how to measure the power consumption are necessary.

4.3 Current state of the art and Gap analysis

The current state of the art in the area of energy efficiency and power management toward Exascale has been fully investigated by WG 5.3 during the life time of the project, as described in [1] and [2]. The main topics analysed are:

- Hardware energy monitoring
- Performance analysis tools
- Power and Energy system profiling
- Standard API for accessing energy information
- Performance and operating states in latter CPUs
- Modelling power and energy consumption
- INRIA-Illinois-ANL-BSC-JSC-RIKEN/AICS Joint Laboratory on extreme-scale computing
- Exascale Projects funded by EU: DEEP, CRESTA, Mont Blanc, DEEP-ER, Mont Blanc 2
- HPC accelerators
- Application specific systems
- Energy efficiency benchmarks

For each challenge previously identified, the group of experts has provided a short gap analysis,

Hardware energy monitoring. The goal: *to be able to monitor energy-related information from a system at appropriate levels of granularity and resolution, from the individual core up to the complete system*. Recent progress has been good, with hardware vendors at the component and system level adding many-more hardware counters to enable energy-related profiling of software applications: see the latest counters in Sandy Bridge CPUs from Intel, and in the XC30 nodes from Cray, as good examples. There is still a gap to close in terms of making sure all main components are measured in a consistent manner (memories, networking, power supplies etc.), and that all main vendors of components and systems present such information in a consistent way and with appropriate resolution. This is more of a standardisation challenge than a technical one.

Energy profiling of applications. The goal: *to enable software developers to optimise their applications for energy efficiency*. This requires that widely used software development tools are enhanced to report information about energy efficiency alongside their more traditional performance measurements. In the last year this has started to happen in HPC, with Intel's Vtune now reporting an energy consumption timeline for an application. But we need this capability to become both mainstream and ubiquitous, and for developers to become as skilled in optimising their codes for energy efficiency, or “performance per unit energy” as they are in optimising them for speed, or “performance per unit time”. So the remaining challenge is to add energy profiling capabilities to the widely used software tools used by HPC developers, and to ensure developers have the skills and motivation to use them. We also need to understand how the power and energy are used in an HPC systems across different architectures (low-power cores, accelerators, high-end cores, etc.). We need to understand how much energy is spent on computing, memory, interconnect, storage, power supply, cooling, and how these factors relate to each other. Once this is understood, then we need to know how these factors relate to the applications, and how changes in power states affect power consumption and application performance (and hence energy). Once we have bridged this gap, we can use this knowledge to guide optimizations in applications and hardware, introducing new power states or management techniques.

Standard API for accessing energy information. The goal: *to make it possible for all HPC software developers to have accurate, comprehensive information about the energy consumption characteristics of their codes, available at appropriate resolutions and for all levels of the system hierarchy, from the cores in a processor, to components on system boards and up to a complete parallel program, including its networking and storage energy information*. Over the last 12-24 months we see piecemeal examples of this being demonstrated, but there is not yet any more towards gathering and presenting this information via a standard API, such as the PAPI hardware counters standard. A standardised API on top of vendor proprietary interfaces will accelerate the rate at which this information can be gathered and disseminated via software development tools, such as profilers, debuggers, compilers and auto-tuners.

Performance and operating states in future processors and systems. The goal: *It is well known that depending of the architecture and the nature of application there exists different configurations to* *tune the architecture for improved energy efficient operation*. Recent research in new architectures has demonstrated that CPU-bound operations are suitable to run at higher frequencies, while memory bound operations can be executed at lower frequencies without increasing the total energy consumption. The selection of the best frequency is completely run-time dependent and might be determined by the values of appropriate counters. The goal in this sense is the development of automatic selection of the optimal execution frequencies and voltages for each component of the system. Today the support for managing power states in the CPU through DVFS is very limited, and this is often disabled in HPC systems. There is little or no support for these mechanisms in the rest of the system: memories, interconnect, storage, etc. To solve this, next to the power monitoring API, there should be a power management API. It is critical to evaluate first what the potential impact of this management could be, and then make it as fast and low-overhead as possible to enable lower granularity state changes.

Modelling power and energy consumption in future architectures. The goal: *a battery of experiments to determine and measure the power consumption will enable the construction of analytical models for specific architectures.* The idea to know in advance a estimation of the energy consumption of the applications before their execution would help developers and administrators reduce the energy consumption of their future Exascale systems and data centres. Current research has demonstrated the feasibility of building energy and power models for complex numerical applications.

Deploying and managing large scale numbers of energy sensors. The goal: *Profiling the right metrics for analyzing applications and services across large systems.* Still to be addressed: Using green levers/power saving modes appearing on hardware. “Going beyond DVFS” on systems that will potentially have millions of sensors providing real-time information on energy consumption, temperature etc.

Increased concurrency to offset decreased clock speeds. The Goal: *Billions of cores in a single machine will be necessary to achieve Exascale performance*. Recent progress**:** over the past few years there has been roughly a doubling of the number of cores in the machine at number 1 in the top 500 each year. The number currently stands at 3,120,000. However, the number of applications that can efficiently run at this scale is still small. Remaining gap: significant work and research is required to enable the several orders of magnitude improvement in scalability required to enable applications to run efficiently on Exascale machines with hundreds of millions of cores.

Addressing whole-system power consumption. Goal: *To analyse and optimise the power consumption of other components of the system in addition to compute nodes (e.g. interconnect, cooling).* Recent progress: there has been some research into this area. For example, a paper at the High Performance Power Aware Computing Workshop 2013 demonstrates that in some cases total system power consumption can be reduced by up to 16% by powering off unused links in the interconnect. Remaining gap: research in this area has lagged behind research into power efficient compute nodes and more will need to be done to ensure that entire system power consumption is addressed.

4.4 Recommendations

Energy efficiency is a crucial challenge that must be successfully addressed if the benefits of Exascale supercomputing are to be realised. Yet little research is being stimulated in this area, and the fundamental APIs required are not being developed in an open, standard format. These shortcomings must be addressed through a combination of targeted funding calls and industrial engagements. Without them, Exascale machines will remain an unreachable goal.

There is an urgent need for standard interfaces for power monitoring and power management at all levels of the system architecture. It is an urgent need as it is estimated that will take 2-3 years after the interfaces are defined until them actually becomes available in systems, and it will easily take 5 years until them are widely adopted in HPC sites.

In this activity is important to involve industry and academia. This joined effort will have several outcomes. The first outcome could be an extension to the Performance Application Programming Interface (PAPI) (http://icl.cs.utk.edu/papi/). A second outcome could be a best practice or buyers guide for what a system needs to provide in order to be operated in an energy efficient manner. This effort should also produce energy efficiency benchmarks to verify the claims of vendors and to guide and monitor the improvements in energy efficiency. This discussion should be lead by industry vendors, but should also involve HPC centers and academia as end users, and main developers of monitoring and analysis tools.

Create a task force to look at the relevant software development tools from the embedded computing space. This could produce a valuable report describing what we might be able to leverage in HPC.

We will need a major training and education initiative to prepare developers to face the power wall challenge. This initiative should equip developers with

1) the ability to understand the energy consumption of their applications, and
2) the use of good programming techniques in order to reduce power consumption.

A manual of tips and tricks for green programming would an extremely valuable resource for the HPC community as it copes with the power wall. However, developers are already faced with the enormous challenge of writing efficient parallel programs that will scale to Peta then Exascale systems. If these developers also have to care about energy efficiency, they will be lost. We need more experts and professional HPC developers to support the wider community. This investment would easily pay off with the more efficient use of the expensive Peta and Exascale systems.

Performance tools exist, but the learning curve to make productive use of them is very steep, more so once they also profile energy consumption. Centres of Excellence in performance analysis should be created to help users get acquainted with the available tools, with one-to-one hands-on tutorials provided by tools experts. Ideally these would be based on the users' own code.

Furthermore, some other focus areas recommended for funding to stimulate high quality research and collaboration with industry have been identified:

- Modelling and prediction
- Foster competition on hardware development and new hardware architectures
- Budgeting based on kWh and not only on core hours
- Energy aware software (libraries, runtimes support, energy aware applications))
- Integrate all levels of hardware and software
- Investigate relationships between power and resilience
- APIs that are truly open and work with all Vendors.

Many of these recommendations have been synthesized in the recommendation presented in Deliverable D 7.2 (see [3]), in particular

- *Holistic approach for extreme heterogeneity management of Exascale supercomputers*
- *Software Engineering Methods for High-Performance Computing*
- *Holistic approach to resilience.*

5. WG 5.4 Resilience

5.1 Objectives and origins of expertise

Resilience addresses the increase of system failure rate due to the explosive growth in component count in supercomputers as well as the use of advanced technologies such as NTV (near threshold voltage). The main objective of WG 5.4 is to address how Exascale computers must dynamically compensate for failures:

- Understand the need for resilience BEFORE the system is built (some of the largest HPC systems on earth have not considered this issue…)
- Understand trends of resilience approaches, compare them qualitatively, quantitatively
- Understand how resilience, performance and energy impact each others
- Understand the need to involve the application developers.

The resilience challenge cannot be addressed in isolation looking at a single software or hardware component. Resilience needs to be addressed considering the whole system: all layers of the software stack, all hardware components constituting the Exascale system and all usages of this system.

WG 5.4 consisted of a chair (Franck Cappello) and eight experts chosen to cover the domains of interest. The experts names and area of expertise are listed below:

Name	Organisation	Country	Area of Expertise
Franck Cappello (chair)	INRIA, Argonne Natl. Lab.	France,	Resilience, Application workloads, Extreme-scale computers
Luc Giraud	INRIA	France	Parallel Algorithms, HPC simulations, Petaflop scalbility
Torsten Hoefler	ETH Zurich	Switzerland	Optimization of Parallel algorithms, Performance Modelling and Tuning, Large Scale Parallel Architectures, Resilience
Simon Mcintosh Smith	Bristol University	UK	Microelettronics, HPC, MIC, Machine Learning, Parallel Architectures
Christine Morin	INRIA	France	Distributed operating systems, Fault tolerance, Autonomic Computing,
Bogdan Nicolae	IBM Research Dublin	Ireland	Scalable Storage Techniques, Exascale Architectures, Data Resilience
Pascale Rossé-Lauren	BULL	France	Applications, Compilers, Programming Environments
Osman Unsal	BSC	Spain	Runtime Environments, HPC, Transactional Memory, Fault Tolerance
George Bosilca	UTK	USA	HPC, Accelerators, Common Communication Infrastructure, HARNESS

5.2 Current state of the art and Gap analysis

This section resumes the activity done in WG 5.4, from the group of experts, on the topics of resilience, during the life time of the project and documented in [1] and [2]. The work, based on the results produced in EESI 1, aimed at providing: i) a gap analysis between existing reports and projection about the resilience challenge for exascale simulation; ii) a set of recommendations based on this gap analysis.

Resilience is becoming a very hot topic in HPC, as addressed in different conferences and meetings. Just to mention a few:

- 2014 SIAM conference on parallel processing featured 17 talks covering many aspects of resilient algorithms.
- Dagshtull seminar on Resilience on September 2014
- Many papers on resilience presented at ACM HPDC2014 (Checkpointing intel MIC, RDMA message logging, etc.)
- Two Tutorials on Resilience have been presented at SC2014

These main topics analyzed are presented in the following.

- **Reliability, Availability Serviceability (RAS) system**. The analysis on RAS system for Exascale, has been addressed at different levels:
  - **At Node HW level**: At hardware level RAS of the node component many low level mechanisms have been added to increase reliability of a platform. Most of node hardware have today embedded fault tolerance capabilities but it much more dedicated to datacenter HW server than HPC due to the cost of this enhanced capabilities. To complete intra node data resilience researches have done on none volatile memory integration at node level. For PCIe I/O interface cyclic redundancy check checksums are used for data transmission/retry and data storage, e.g. PCIe Advanced Error Reporting, redundant I/O paths.
  - **At Node system level**. System software is less frequently a root cause of failures but system software plays a critical role in fault detection, containment and recovery. The fault detection and containment is done at each software stack level from firmware, OS, and middleware. Great effort are done today to develop interfaces between the hardware and the firmware or techniques allowing early fault detection and recovery.The hardware offers mechanisms to recover most of node non-fatal error and recently developed HW to “FW or SW” interfaces allow interactions with low level software to design more complex recovery or isolation solutions. **For nodes based on those hardware technology the next step will be to integrate those capabilities with the other layers of the software stack. FW and OS must be enhanced to handle those new RAS features.** For new hardware platforms based on co-processor integration or embedded processors (ARM), RAS techniques are less developed. **These types of plateforms have a critical need of fault aware software stack.**
  - **At interconnect level.** Interconnect reliability is critical for applications execution: multi path link and adaptive routing have been integrated to interconnect to limit hardware failure impact on message passing libraries or applications. At link level fault protection capabilities are similar to the ones used for internal link: for example, Link Layer Retransmission (LLR from Mellanox IB solutions) allows packet retransmission by lower layers due to physical errors without any impact on the transport layers. The remaining risk on interconnect is much more on silent error and data corruption. **The applications and associated message passing libraries such as MPI used on top of interconnect need to be fault aware**.
  - **At File system and storage level.** Many resilient features have already been developed at hardware and software level for file system and storage. The major issue for exascale is on data integrity, data corruption detection and correction. **Again research is needed to detect data corruptions in file system and storage devices.**

  Exascale RAS systems must be investigated not only at each level of the stack (hw, os , middleware) but also globally to investigate new fault tolerance methodologies and to enable RAS systems to meet their own resilience needs. **The challenge is to provide the reliability of an N-modular redundancy scheme at only a fraction of the current energy and hardware costs.**

- **Run-time.** Although runtime has been identified as a critical issue for Exascale resilience by several recent reports, there is a lack of detailed discussion on how the runtime (and programming models) can enhance system resilience. In the following runtime means the node-level runtime which can optimize local, node-level error detection and recovery policies. In the era of Exascale, we expect hybrid programming models such as OpenMP on the node-level and MPI on the system level to be utilized extensively. Recently, OpenMP was extended with a task based execution model. We think that compared to thread based models, task based programming models offer a good substrate for reliability due to their superior isolation properties. Moreover, tasks are easier to migrate to a different processing unit in the event of a fault, as well as being easier to schedule. Work stealing runtimes such as Cilk or Task Building Blocks (TBB) from Intel make it easier to implement efficient localized failure checkpoint/restart mechanisms in runtime that is a function of the extent of the error propagation rather than system size. Likewise dataflow based runtimes also offer efficient localized checkpoint restart. We expect these runtimes to be effective for Exascale fault tolerance as well. One development that we would see in the next couple of years is exposing even more reliability related information from the hardware to the runtime. It will be up to the runtime to exploit and utilize this rich set of diagnostic and preventive notifications. In the exascale timeframe, we expect these hardware error recovery mechanisms to be exposed to the runtime so that the optimal reliability decision could take into account the available system information, including application state that is available to the runtime. **The research in this domain has just started and more efforts should be put on understanding how to leverage and control by the runtime hardware resilience features.**

- **High performance checkpointing.** The increasing rate of failures and I/O bandwidth limitations of exascale systems pose a serious problem for checkpoint-restart: several modeling studies show that traditional approaches (i.e. blocking coordinated checkpointing to a parallel file system) will become completely unfeasible at such large scale. On the other hand, checkpoint-restart naturally fits into the current programming models and practices as a key fault tolerance mechanism. Thus, an important research direction is how improve the scalability of checkpoint-restart. This direction needs to be attacked from multiple angles: 1) increase asynchrony to avoid blocking during checkpointing; 2) reduce the checkpoint sizes in order to save them faster; 3) reduce coordination overhead; and 4) leverage local storage resources. With respect to asynchrony, recent results show that specific memory access patterns for certain applications can be leveraged to optimize the order in which checkpointing data is flushed, thus minimizing the need to block or create extra copies. **Further research is needed to better understand memory access patterns for various application classes and derive interesting properties that can enhance checkpointing asynchrony.**

  More research is needed to better understand how redundancy across multiple processes relates to data structures at application level in order to identify applications classes that can benefit from specific optimizations like clustering similar processes together and let them share unique memory contents.

  Also, more research is needed to minimize the cost of identifying and leveraging redundancy in order to make such techniques feasible. With respect to coordination overhead, more research is needed to provide viable alternatives to global coordination, which is already becoming prohibitively expensive but still widely used in practice, despite promising advances in alternative directions. Finally, local storage resources will be a key element in combating the growing scalability limitation of I/O bandwidth. **Priorities here are the need to specialize for checkpoint-restart beyond the classic parallel file system model**.

  With respect to failure prediction, increasing accuracy has been shown by combining off-line and on-line analysis of events generated by the machine. With respect to migration, most techniques used so far are off-line and closely resemble checkpoint-restart. This creates a long downtime during which the application cannot progress. To address this issue, other communities (notably virtualization/cloud computing) have extensively developed and improved live migration techniques at virtual machine level in order to overlap the virtual machine execution with the migration itself and thus minimize migration overhead. **Under these circumstances, more research is needed to understand how live migration techniques can be adopted at application-level (i.e. What memory content needs to be moved? In what order? How to minimize amount of transferred data? etc.).** Furthermore, an important barrier in such adopting complementary techniques is the lack of flexibility in current message passing libraries (in particular MPI implementations) with respect to how processes are managed, e.g. lack of obvious features such as the ability to detach ranks from individual processes and make it easy to dynamically replace them or create groups of processes for the same rank. More research is urgently needed to address this issue.

- **Multilevel checkpointing.** Checkpointing on remote file system raises performance and reliability issues. It is expected that the bandwidth between the compute nodes and the remote file system will not scale as much as the size of the memory for Exascale systems. Even with application level checkpointing, at some point the amount of data to save at each checkpoint will requires 10s of minutes to be stored on remote file system. There is a high risk of limiting drastically the execution efficiency if failures are frequent.
  The two main environments for multilevel checkpoint restart offer in memory checkpointing, remote memory checkpointing, several encoding algorithms (XOR and Reed Solomon), local storage on SSD devices and remote storage on file system. **More research is needed to decouple checkpointing from the failures of storage levels. If in memory checkpoint cannot be performed then this should not block the execution.**
  More research is also needed to understand how to copy checkpoint image between the different level with a minimum overhead on the execution. There are different techniques: inlining, pipelining with local resource, pipelining with remote resources that need to be compared.
  The emergence of new non volatile memory technologies generates many opportunities for multi-level checkpointing. These memory chips will likely be available on every node of the system. High performance non volatile memory could even replace DRAM within the next 10 years if the price per byte reaches the one of DRAM. Another important consequence of the availability of affordable non-volatile memory is that disks become useless. Some researchers consider that spin disks may replace the tapes for massive storage.
  More research is needed to understand how to make the best usage of future non-volatile memory to for fault tolerance.

- **Advanced fault tolerant protocols**. Fault tolerant protocols play a critical role in capturing and restoring a consistent state of a parallel execution. Progresses have been made in three directions since the publication of the IESP and EESI reports. The three directions concern 1) distributed recovery, 2) hierarchical protocol performance modeling, 3) clustering procedure. Hierarchical fault tolerant protocols rely on forming clusters of processes. They use coordinated checkpointing inside cluster and message logging between clusters. Until recently avoiding event recording implied a centralized recovery procedure.
  However research is needed to understand the sensitivity of simulation codes to state inconsistency. For example, considering collective communications and the reductions in particular, it is not clear that reduction operations need to be replayed during the partial recovery of a cluster exactly the same way as they were played by the cluster before the failure. In other words inconsistency in value (floating point numbers) may not mean incorrect state from the simulation application point of view.
  New clustering algorithms need to be designed to target user defined recovery speed/message logging trade-offs to address both fast recovery and limited message logging.

- **MPI and other programming models.** MPI-3.0 adds certain new concepts to the MPI standard that are not necessarily addressed by current Fault-Tolerance strategies. The two main concepts that were added and may require additional fault-tolerance investigation are neighborhood collectives ("build your own collective") and the updated remote memory access (RMA) specification. RMA allows the optimized implementation of a class of graph computations and is thus relevant to Big Data graph problems.
  reused multiple times. Optimizations (e.g., tree reordering or graph coloring to avoid congestion) are often performed during the creation of the collective. In practice, neighborhood collectives are created through weighted MPI graph topologies on special communicators. **Fault-tolerance research would need to investigate if the sparsity of operations (involved in neighborhood collectives) can be used for advanced message logging or other fault-tolerance protocols. Also, the persistence and determinism of those operations (once created) is a rather interesting property.**

The new remote memory access interface in MPI-3.0 enables direct hardware support without going through the messaging layer. This requires new and efficient schemes for fault tolerance support since the remote process is not aware that its memory is updated (which prevents it from logging messages efficiently). **However, due to the nature of RMA, logging can be performed through the same interface which allows to adapt RMA-specific message logging and recovery schemes that enable transparent uncoordinated checkpointing and recovery schemes.**

Due to MPI's lack of fault tolerance support, other high-performance programming systems have been developed. The most prominent example is probably **MapReduce** which convinces by it's simple (conceptual) structure and aggressive fault tolerance. However, while MapReduce enables efficient implementation of most of the important machine learning algorithms, it is not as efficient for many graph problems such as graph searches. Some alternative schemes, such as Google's Pregel and related tools (Apache Giraph etc.) have been developed but those do not offer FT schemes that are comparable with MapReduce. For example, Pregel uses a simple coordinated checkpointing scheme. **So new research is needed on new programming models for graph algorithm providing efficient fault tolerance.**

- **Failure prediction.** Failure prediction is an important highly speculative approach. If successful it can change drastically the way failures are tolerated. Progresses have been made in the understanding of the impact of failure prediction on execution performance in presence of failures (predicted or not). Another important progress is the understanding that failure prediction cannot handle 100% of failure and this technique should be coupled with some preventive techniques like checkpointing or replication. Thanks to recent performance modeling, we know that failure prediction can be used to extend significantly the checkpoint interval. Researchers have explored failure prediction associated with partial replication. However failure prediction algorithms are still in their infancy. The best performances are still around 95% of precision (95% of what is predicted is correct) and 45% of recall (45% of all actual failures are predicted) for predictions predicting time and location. **So an important research effort should be made to increase the recall to value around 80%**. **The main objective of failure prediction now should be on performing actual prediction, online, on real production systems**. The first experiments in this context are disappointing, essentially because real logs on today largest systems are far larger than logs used for academic research.

- **Resilient numerical algorithms.** In numerical algorithms as in other software components one should distinguished between hardware crashes and data corruptions (soft, silent, transient errors). For hardware crashes, alternatives to global check point restart exists for some numerical kernels and have started to be investigated mainly in the context of linear algebra (primary dense linear algebra) based on ABFT approaches with some computational penalties (Memory and CPU). Still in the context of numerical linear algebra, a few fault-oblivious linear equation solvers have been designed that have no overhead in fault free calculation and increasing penalty cost when the fault rate increases. **The performance crosscutting between algorithm specific check pointing and their fault-oblivious counterpart needs to be investigated to possibly decide at runtime what alternative deserves to be selected** (so interactions with the runtime may be needed).

  On the soft error side, much less works exist, often based on a checksum mechanism that enables to possibly detect a (no longer) silent error but does not necessary permit to recover the corrupted data. **If hardware existed to detect memory corruption, some numerical algorithms might be revisited to re-compute or recover the lost piece of data.**

  One feature that is not much exploited is some data redundancy exhibited in many parallel numerical algorithms that could enable a straightforward recovery of those data (lost or corrupted) and a possible re-computation of a subset if not all of the lost/corrupted information.

  The current efforts only address a few numerical linear algebra techniques and **studies should be extended to cover all linear algebra kernels first as well as other widely used numerical kernels such as for instance FFT.**

  Composability of the above mentioned techniques with other fault recovery solution to best exploit the computing capabilities of future computers should surely be considered.

5.3 Recommendations

The recommendations presented by WG 5.4 aimed at finding relevant effective and efficient solutions for Exascale resilience, addressing both fail stop errors and silent data corruptions, taking into account the multifaceted aspect of this problem. The recommendation are organised in six different tasks:

- SP1: Extend the applicability of Checkpoint/restart and migration
- SP2: Improve system efficiency and execution recovery in presence of fail stop errors through better fault tolerant protocols
- SP3: Investigate alternatives to checkpoint/restart: tasks based checkpoint/restart, migration and redundancy
- SP4: Fault aware software stack
- SP5: Develop failure prediction
- SP6: Resilient algorithms

As a further step, a new recommendation can gather the different points in a single vision aiming at promoting an "*Holistic approach to resilience for simulations and data analytics*". This new recommendation fits in the Pillar *Tools and Programming Models* and proposes the development of resilience API that will provide the required integration of resilience techniques and coordination of software resilience mechanisms and by improving critical resilience mechanisms:

- Understanding and modeling of fault propagation
- Push Checkpoint restart as far as possible
- Error detection
- Failure prediction
- Roll back and roll forward recovery
- Resilient Runtime, Resilient OS and Resilient Algorithms.

It is recommended not an integrated project on Resilience covering all layers from hardware to the applications, but a project on Integrated Resilience, adopting an holistic approach, covering from numerical algorithms to resilient supporting software (libraries, runtime, OS, etc.).

6. WG 5.5 Disruptive technologies

6.1 Objectives and origins of expertise

The last decade has seen significant changes in processor architectures to improve computing performances and to overcome the physical limitations of increasing the clock frequency. This allowed processors to still scale according to Moore's law. However, a new challenge is becoming relevant today: the ever increasing power and energy requirements for operating the latest generation of HPC systems are key factors in limiting the peek performances of newer HPC systems and might make Exascale computing unsustainable for both technical and economic reasons.

A possible solution is to identify disruptive technologies in terms of new hardware architectures and energy aware system software which maximize the performance of HPC systems within a given power or energy budget.

WG 5.5 focus on the search of disruptive candidates technology/components that have good potential to create a discontinuity on the current architectural trends while reducing the demands on other components of the HPC environment, especially regarding system density and efficiency.

The working group consisted of a chair (Carlo Cavazzoni) and vice-chair (Marie-Christine Sawley) and nine experts chosen to cover the domains of interest, among them, three new experts joined in the second year (Andrea Bartolini, Luca Benini, Cristina Silvano). The experts names and area of expertise are listed below:

Name	**Organisation**	**Country**	**Area of Expertise**
Carlo Cavazzoni (chair)	CINECA	Italy	HPC, MIC, GPU, Numerical Simulations, Energy efficient computing
Marie-Christine Sawley (vice chair)	Intel	Switzerland	HPC, Massive parallelism, Energy efficient computing
Shekhar Borkar	Intel	USA	Semiconductor Technology: near threshold voltage
Bruno Michel	IBM	Switzerland	Packaging: microfluidics
Patrick Demichel	HP	France	Memory: memristor, 3D-stack memory, NVRAM. Data Transfer: Silicon Photonics
Piero Vicini	INFN	Italy	Network Technology: Adapter free device
Giampiero Tecchiolli	Eurotech	Italy	Cooling & Engineering: High density high efficiency solutions
Malcolm Muggeridge	Xyratech	UK	I/O sub system
Andrea Bartolini	University of Bologna and ETH Zurich	Italy	Energy aware systems, Power management for HPC, HPC resource management
Luca Benini	University of Bologna and ETH Zurich	Italy	Microelectronics, HPC Architectures, Embedded systems, Power aware design
Cristina Silvano	Polytechnic of Milano	Italy	Computer Architectures, Monitoring of applications for many-core architectures

6.2 Possible sources of Disruptive technologies

The analysis done by WG 5.5 to identify and investigate possible disruptive technologies for HPC at Exascale focused first to the potential disruption coming from the hardware component of HPC architectures. Then the focus was directed to the the software components that may disrupt the HPC software stack. The main sources of disruption have been identified in the following technology components, which have a good potential to create a discontinuity on the current architectural trends toward Exascale.

Semiconductor Technology:

All microprocessors used to perform computations, from handsets to supercomputers, are using silicon based semiconductors, and no alternative is foreseeable for the next ten years. Unless the manufacturing processes substitute silicon with something else (not foreseeable in near future), the size of the transistors cannot be reduced any longer while keeping the same power dissipation and the same voltage. As it is well known, size, voltage and power dissipation of a semiconductor are not independent.

There is still a lot of room for energy improvement in today semiconductor technology, especially if we allow a different way to design application and manage workloads. It is possible to design a new chip architecture that is able to work at different regime (frequency and voltage) in order to accommodate the needs of different workloads and meet the requirements in term of efficiency. This Near Threshold Voltage (NTV) chip could be organized in a hierarchical way. It may contain two kind of cores: control cores and execution cores making up a block, then different blocks can be connected together with a network to form a cluster, then clusters can be connected together into a single chip, with a global shared non-coherent address space.

NTV chips may trigger a revolution in the supercomputer architectures and applications as well. HPC will require combining NTV chip with a bus for short distance (up to 5 mm), a multi ported memory to share memory locally and switches to long distance connections. Applications with good data locality will express better performance from this architectural change.

NTV chips with respect to the exascale roadmap prediction will force an increase of parallelism by at least a factor four.

Impact on programming and execution models need to be considered as well.

System software, combined with the availability of sensors should become “introspective” in order to be able to schedule threads close to the data upon which they have to operate.

In conclusion, NTV chips can allow to meet the energy constraints of an exascale system, but there are a number of challenges that need to be addressed as well:

- a revolutionary architecture, with cheap computational costs and expensive data movement costs;
- a refactoring and a rethink of algorithms and applications;
- a programming model to harness extreme concurrency;
- an introspective, self-aware, execution model;
- and last, but not least, resiliency to provide system reliability.

Packaging

As the transistor size decreases, the power dissipated per unit volume increases accordingly, thus generating additional heat in hot spots. With such hot spots, the microprocessor could not work or at least could not work at its best; therefore it is very important to remove properly this heat. It is clear that the way the heat is removed from the chip becomes a critical factor in allowing further reduction in system size and efficiency. The evolution of HPC architectures are going through three main paradigm changes:

1) From Cold Air Cooling to Hot Water Energy Re-Use

2) From Performance to Efficiency

3) From Areal Device Size Scaling to Volumetric Density Scaling

Common to all these changes there is the possibility to design a new packaging concept around “pervasive” water cooling, with the liquid entering directly inside the chip.

From the point of view of the whole datacenter, hot water cooling could enable the design of a "zero emission" data center, where all energy used to power the supercomputer and to perform computation is converted into heat, and then the heat can be directly re-used for other purposes, like space heating.

To address efficiency and density, a disruptive approach regarding packaging has to be explored, and computer components have to go 3D, stacking them one on top to the other. There is no point in having all chips on a planar board, except the fact that they can be assembled and cooled more easily. Heat removal is clearly the main obstacle, and one has to design the packaging leaving room for water, or other fluid to go through the chips to remove heat. In this respect, micro-channels technology appears to be the first candidate to be used to start stacking chips.

The Hybrid Memory Cube, is the first attempt to put this concept into practice, is a project sponsored by a number of vendors. Technologies similar to memory cube are going to appear in real product by the end of 2015. The first computer stacking memory, interconnect and processors are going to appear around 2020.

The ultimate density and efficiency challenge is reached when power supply wires are eliminated to free wires and space for communication. This is done by combining power supply and cooling system using electrochemistry to supply power where it is needed, and the same liquid flowing can be used to cool the chip. Putting together all these aspects, a peta-scale computer can be built in a volume of only 10 liters. In summary:

Impact:

- Improve computing efficiency by a large factor (up to 5'000)
- 50'000'00 times reduced compute core volume

Barriers:

- Cost for 3D stacks and TSV (through-silicon via), saturates after 2 logic layers
- Cost for interlayer cooled chip stacks introduction
- Cost for electrochemical power supply development
- Power density of electrochemical power supply
- Cost of optical links has to reach 25$ per Tbit/s

Timeframe:

- 2-5 years TSV and hybrid memory cube (pre exascale)
- 5-7 years optical interconnect on chip stack level (at exascale)
- 7 years interlayer cooled chip stacks (at or post exascale)
- 10+ years for electrochemical power supply (post exascale)

Data transfer

moving data in current microprocessors produces heat and wasting energy. A critical factor then is to find new ways to move data around without perturbing electrons, this can be obtained if electrons are substituted by photons. Photons are electrically neutral and weightless and (if not absorbed) do not dissipate energy while travelling, so they are the perfect candidate to move information around. Unfortunately because they are electrically neutral and with 0 spin momentum, controlling their behavior is not an easy task. Photons are the natural candidates to substitute Electrons to do this job even for short distance (inside chip).

It has been estimate that by using photons, it is possible to have 30 times more bandwidth at one tenth of the energy, this should imply that in the end all data transmission will be optical, at all levels. But to manage photons is not an easy task, and there are a number of technological challenges that need to be solved (develop good laser source with low power consumption; integrate into a silicon chip the modulator and resonator to code and decode information into the light being transmitted, etc.). To have all in place to be implemented in an end-user Exascale, or post Exascale) system it will require other 7 years of research and development. Then probably the first Exascale system will use already some photonic technology to carry data around, but not yet at all levels. For the Exascale system, we can expect a fully optical switch that can be used to connect, on the same ground, CPU with memory, GPU, or other system nodes.

Many different optical switches (xbar) can be assembled together to build a system fabric interconnect switch with superior performance with respect to electronic fat tree. If this switch will be available, can be coupled with photonic circuit at node and chip level and will allow the setup of an exascale system with a switch-based interconnect.

In conclusion, fully optical switching and interconnect technology can be a disruptive technology for the system architecture. The system will be more integrated than today petascale systems since optics switch will allow a more “flat” design, without topology. This technology will also open more degree of flexibility, since the different system components (Memory, CPU, GPU, etc.) do not need to be all integrated in the same node. One can imagine a central memory complex to be shared between different nodes. This should also have a positive impact on system programmability, with less architectural constraints than today.

Memory

Main system memory today is implemented with DRAM, which is powered regardless of the fact that data are changed or not. Ideally one would like to spend energy only if a new state is induced. DRAM then dissipates a lot of energy proportionally to the quantity of memory available, and not proportionally to the number of changes.

One disruption in memory technology, regards the possibility to stack memory chips one above the other, with gain in capacity efficiency and speed. A condition to allow dense packaging and fast and cheep (in term of energy) interconnect is to combine, in the same stack, together with memory and cores an optical switch to allow the required sustained system bandwidth (1Tbps per core). This can be done through nano-photonic technology as discussed in the data transfer section.

Another disruption, bridging memory and I/O, is the possibility to build different memory technology as alternative to DRAM and Disk. One of the most promising and disruptive technology is based on *memristors*, Memristor based memory devices have very good characteristics (high density, stackable, low cost, low power, high speed, etc.) and they addresses positively issues related to power, performance, architecture flexibility, fault tolerance, programmability, capacity and cost. Memristor based memory chips seem essential for exascale systems, as they are the most promising component to support application’s check-point/restart functionalities in substitution to external disk based sub-systems. Other today non volatile ram devices (Flash based) do not seem to have a future for high-end HPC due to their characteristics.

Looking at what is happening today In the field of memory device, an interesting co-design project, co-sponsored by different institution is Blackcomb (https://ft.ornl.gov/trac/blackcomb).

Network

As the number of node increases, to allow communication between them in order to support massively parallel applications, network becomes a critical factor. Both from the point of the physical implementation of the network and from the point of view of the routing (the algorithms used to deliver the messages).

I/O subsystem

HPC I/O subsystem are still deployed using spinning disks, which have mechanical limitations and, like for the DRAM, they consume energy even if their state is not changed. SSD solid state technology appears to be a possible alternative, but actual costs do not allow implementing data storage systems of the same size. Probably some hierarchical solutions can exploit both technology, but this does not solve the problem of having spinning disks spinning for nothing.

To manage an Exascale system, it can be estimated a need in the order of: 100K drives with the current technology. This is not acceptable, most data can no longer make it to disks, and what data management can help? In this respect what is the role of SSD technologies? In what respect can they be disruptive?. The trend seems clear, disk do not speed-up and so the I/O subsystem need to be tiered with something faster, while disk will be moved to “archive-like” tiers of the I/O subsystem. High speed disk are no longer competitive with the new flash technology, whereas, mainly due to the very low cost, “slow” SATA drive will continue to play a role in future I/O subsystems. Flash, tiered with disk can offer the IOPS rate and the bandwidth require by HPC applications and first of all the above use case, without a disruptive change in the behavior of applications, however the middleware needs to be reviewed.

Disk subsystem capacity will continue to increase and the same is true for stream performances, but this is not true for random access. Regarding power consumption, the idea of slowing down disk when not in use is not supported by the usage model, where to meet performance requirements file system blocks are stripped across a wide array of disks, so all disks are always in use.

One area of energy efficiency may come from the use of sealed (helium filled) disks which have lower operating power and potentially allow new cooling technologies to be implemented.

A few concerns surround current SSD technology (Flash) there are such as: lifetime and performance degradation with aging. As we have already discussed speaking about memory components, on the horizon there are new promising technologies that could hit the market such as: PCM, ReRAM, FeRAM and ST-MRAM. But it is not yet clear if they will reach the capacity and cost suitable for HPC. If they do, then the opportunity to support a “byte” addressable model, can allow a dramatic change in the model the I/O is used by applications (more or less like magnetic core memory). In this sense they will be disruptive for software development, I/O infrastructure and Exascale middleware.

The middleware software is going to be the most impacted component by new disruptive I/O device and tier structure. First of all the filesystem: simply they will not scale, mostly because the interface they provide is too low level, preventing “smart” application driven I/O read and write strategies, and locks and synchronization dominates.

HPC Storage system deployed today are mostly based on InfiniBand and/or SAS devices, but the bandwidth roadmap of these standards do not seem to keep the pace with the need required for Exascale systems. Moreover the bandwidth requirements are so large that optical interconnect will become competitive at all scales. Limit of copper are: Crosstalk, Reflections, Electro-magnetic interference, Dielectric Loss / “Skin effect”, Signal skew. Then also for the I/O subsystem optical interconnections are going to be embedded into storage device: Copper layers for power distribution, Copper layers for low speed communication and Optical layers for high speed communication.

Cooling

It is clear that supercomputers are a big source of heat, not equally distributed. The heat sources are localized in few hot spots with a huge heat density. Removing heat away from the computers then requires a lot of energy on cooling capacity, this energy is not productive since does not go in useful work, so it is of fundamental importance for the efficiency of the machine to reduce this energy as much as possible. Direct liquid cooling (in different flavors and degrees) seems a good candidate allowing cutting costs of cooling.

Liquid cooling will no more be an option but a must for HPC infrastructures (the most promising technologies appear to be “immersive cooling” and “hot liquid cooling”), for two reasons: budget constraints and ability to remove heat from dense infrastructure. Analyzing the Top500 trends and technological trends, one can argue that an Exascale system is really possible in term of performance and integration by the 2020, a part two critical parameters: the number of cores and the energy dissipated per node. The number of cores is expected to be in the order of one billion raising a big wall in term of programmability and scalability of applications. The dissipated power per node, from the projections possible with today technology trends is expected to rise from 0.3 KWatt/node to 1.3 KWatt/node, which is really too high to meet the power constraints of an Exascale system.

To overcome the above limitations a shift toward silicon photonic optical technology become mandatory, and, from the point of view of the programmability, scheduling of macro applications tasks will complement MPI, OpenMP and other lower layer protocols, to exploit multi racks systems.

But this is not enough to close the power budget gap; we need an holistic approach at system efficiency and energy reuse aggressing all components that “heat” energy. Even so we will came close to 0.6 KWatt/node (by 2020), then we need some further improvements with respect today known roadmaps to reach 0.3 KWatt/node and realize and Exascale system within the power and cost constraints.

Optical computers

Optalysys is a UK company developing an optical supercomputer that can perform mathematical functions such as FFTs and matrix multiplications using light focused by liquid crystal patterns. This contrasts with the electronic approaches used by today’s mainstream computers. The optical computer approach has the promise of higher performance and much greater energy efficiency. Optalysys is a spin out from the University of Cambridge and is at the prototype stage. For more information see: http://optalysys.com.

Quantum computing

The nature of quantum computers is totally different from the classical digital computers based on transistors, as they make direct use of quantum-mechanical phenomena such as superposition and entanglement to perform operations on data. The computation is based on qubits, object obeying to the rules of quantum-mechanics .

A classical computer has a memory made up of bits, where each bit represents either a one or a zero. A quantum computer maintains a sequence of qubits. A single qubit can represent a one, a zero, or any quantum superposition of those two qubit states. In general, a quantum computer with n qubits can be in an arbitrary superposition of up to 2^n different states simultaneously (this compares to a normal computer that can only be in *one* of these 2^n states at any one time).

From the hardware point of view, all the costs relative to maintaining operative a quantum computer is due to the cooling of the machine, whose temperature should be close enough to 0 K degrees.

From the algorithmic point of view, the challenge for the use of quantum computers, is how to map classical problems of interest of the HPC world to problems solvable with a quantum algorithm. This issue is the one that still makes difficult to consider quantum computing an approachable solution and probably will be the subject of the work in the next future.

D-Wave Systems, Inc. is a quantum computing company, based in Canada. In May 2013 it was released the 512-qubit D-Wave Two system. The last product of D-Wave, the D-Wave Two has been recently adopted by NASA, Lockheed-Martin, Google and USC to tackle several problems from machine learning to minima-finding problems. In June 2015 D-Wave announced a new quantum processor with over 1000 qubits designed to deliver a 1,152 qubit region out of a complete 2,048 qubit fabric. The promise of quantum computing is very exciting for some fields of applications.Once mature algorithms will be ready for the quantum computers, this technology will certainly be a solution of absolute interest for the high-performance computing industry.

Other potential disruptions

It is worth noting that there are other research topics which may disrupt the energy efficiency of HPC in the future. Many of these are materials topics. For example, **graphene** is being explored as a potential alternative or complement for silicon as the basis for microchips as we come to the end of Moore’s Law scaling for traditional materials. Other research topics which may lead to other potential disruptions are represented by the **EC Flagship Human Brain Project** activity. Research on neuroscience at the human brain level can address new technologies in the field on brain inspired computing as well as neuromorphic computing.

Software components that may disrupt the HPC software stack.

Energy efficiency and awareness is the main driver in the evolution of the software components, at all levels: firmware, operating system, scheduler (low and high level), monitoring, and applications.

Heterogeneous systems currently dominate the top of the Green500 list and this dominance is expected to be a trend for the next coming years to reach the target of 20 MW Exascale supercomputers. However, to fulfill this target, energy-efficient heterogeneous supercomputers need to be coupled with a radically new software stack capable of exploiting the benefits offered by heterogeneity at all the different levels (supercomputer, job, node) to meet the scalability and energy efficiency required by the Exascale era.

In particular Energy-efficient heterogeneous supercomputing architectures need to be coupled with a radically new software stack capable of exploiting the benefits offered by the heterogeneity at all the different levels (supercomputer, job, node) to meet the scalability and energy efficiency required by Exascale supercomputers.

For the management of Exascale supercomputers, an holistic approach must be adopted involving the different layers of the HPC architecture and environment.

Apart quantum computing and brain inspired computing that require longer research roadmaps, as a general outcome of the above analysis, it appears that three main areas of innovation can determine a disruption with respect to the current approaches to realize an Exascale system: 1) hybrid systems and processors integration; 2) new "high bandwidth" optical interconnect; 3) memories and I/O subsystem and in general data movement, based on new NVRAM technology. (see Figure 1).

All these three main areas require a huge disruptive effort of integration at software level at different layers from OS to API to application software.

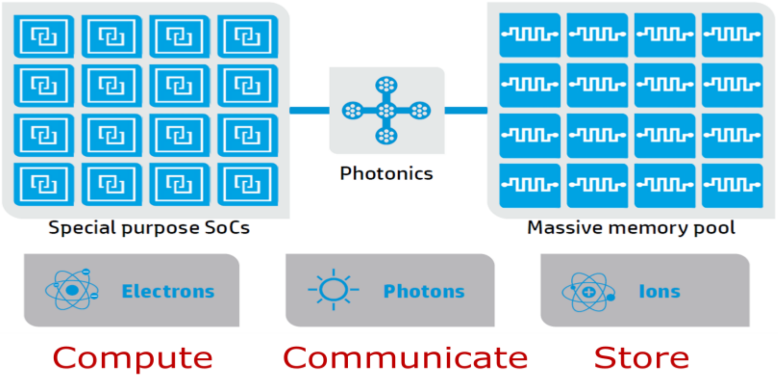

Figure 1: ***Disruptive technologies for Exascale systems***

6.3 Recommendations

Some of the analysed disruptive technologies (integration and packaging, efficiency of the architectures, new memory technologies optical interconnection, etc,) may allow dramatic redesign in system architecture and in new application paradigms, and could represent a great potential to enhance the Exascale process so it is recommended to push research and investment in these disruptive new technologies to short the path to Exascale.

These recommendations must be complemented by specific recommendations aimed to integrate the hardware architecture with the software layers at different levels. A disruptive holistic approach spanning all the decision layers composing the supercomputer software stack and exploiting effectively the full system capabilities (including heterogeneity and energy management) is recommended. A breakthrough approach is needed to express application self-adaptability at design-time and at runtime to manage and auto-tune applications for green and heterogeneous HPC systems up to the Exascale level.

Key elements of this approach are:

- scale out monitoring based on big data techniques and low level protocols

- control over the hardware

- domain specific languages

- separation of concern

The applications then need to be re-factored to exploit the separation of concerns with respect of the energy consuming part, and domain specific languages need to be developed to be able to cope with the lower layers allocating the workload to the available resources in order to optimize the efficiency.

All these considerations have led to the recommendation *“Holistic approach for extreme heterogeneity management of Exascale supercomputers”* as part of the *Tools & Programming Models Pillar;* See [3]. The recommendation aims at foster the research and development of:

- Hardware/Software APIs to manage the complexity and the programmability gap inherent of extreme heterogeneous Exascale level supercomputers;
- Design strategies for scalable and efficient heterogeneous-aware exascale applications;
- Scalable and efficient community scientific applications for exascale;
- System software to support efficient usage of exascale heterogeneous supercomputers in production.

This is furthermore of paramount importance nowadays after the recent action made by the US President Barack Obama which has signed an executive order setting up the National Strategic Computing Initiative (NSCI), to coordinate government agencies, academia and the private sector for the development of high-performance computing systems. One of the objectives of the NSCI will be to speed up the delivery of "a capable exascale computing system that integrates hardware and software capability to deliver approximately 100 times the performance of current 10 petaflop systems across a range of applications representing government needs.",(see [4]).

7. Conclusions

This document is the final report of the EESI2 WP5 *Cross Cutting Issues*. The WP is organized in five WGs: Data Management and exploration, Uncertainties (UQ/Verification & Validation), Power & Performance, Resilience, Disruptive Technologies.

The deliverable presents summary of the findings of the different WGs during the project lifetime. Between the topics that the experts have highlighted as challenges, we find:

- Data management represents one of the major challenges of Exascale applications addressing scientific discovery nowadays. Actions must be issued to address end-to-end techniques for efficient disruptive I/O and data analysis, involving the full life-cycle of data.
- For Exascale applications, verification, validation and uncertainty quantification of computer models' results becomes fundamental both for industry and academia. Identify methodologies and enhance tools for the analysis of these uncertainty sources, is fundamental for the exploitation of Exascale applications.
- Power monitoring and power management at all levels of the system architecture, addressing energy efficient performance of applications, is a crucial issue to address in the Exascale era. The definition of standards is urgent as well as the formation of professional HPC developers experts in green programming methodologies.
- Robust fault tolerance protocols as well as performing checkpoint/restart methods, to increase the efficiency of Exascale systems, are becoming urgent to manage the fore coming systems with millions of cores.
- The roadmap toward Exascale and beyond will be guided and modeled by disruption in semiconductor technologies, I/O and memory technologies, networking and data transfer technologies, energy aware and advanced cooling technologies and facility management. The entire Software stack (programming models, run-time, OS and system support software) needs to be fully innovated to support programmability and efficient performance/energy usage of the different resources and the computational models.

Many of these challenges have represented the basis to formulate the EESI 2 project recommendations issued in July 2013 and July 2014.

It is important to underline that the support for HPC actions at European level is of paramount importance also in light of the huge investments recently announced at US level aiming at . setting up the National Strategic Computing Initiative (NSCI), to coordinate government agencies, academia and the private sector for the development of a capable Exascale computing system."

8. Bibliography

[1] EESI2 Deliverable 5.1 “*First Intermediate report on cross cutting issues”*, September 2013.

[2] EESI2 Deliverable 5.2 “Second intermediate report on cross cutting issues”, April 2015.

[3] EESI2 Deliverable 7.2 “2014 Update Vision & Recommendations”, July 2014.

[4] THE WHITE HOUSE Office of the Press Secretary, Executive Order, *Creating a: National Strategic Computing Initiative*, July 29, 2015.