

Deliverable D5.1

WP5 First Intermediate Report

Cross Cutting Issues Working Groups

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

Due date of deliverable: <31 July 2013>

Actual submission date: <19 July 2013>

Publication date:

Start date of project: 1 September 2012

Duration: 30 months

Name of lead contractor for this deliverable: Giovanni Erbacci, CINECA

Authors: Giovanni Erbacci (CINECA), Francois Bodin (CAPS), Vincent Bergeaud (CEA), Alberto Pasanisi (EDF), Simon McIntosh-Smith (Bristol University), Thomas Ludwig (DKRZ), Franck Cappello (INRIA), Carlo Cavazzoni (CINECA), Marie-Christine Sawley (Intel).

Name of reviewers for this deliverable:

Abstract: First intermediate report on cross cutting issues, presenting the outcome of the first year of activity of the five working groups identified in EESI-2 WP5.

Revision 1.0

Project co-funded by the European Commission within the Seventh Framework Programme (FP7/2007-2013)		
Dissemination Level		
PU	Public	
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)	X

Table of Contents

1. EXECUTIVE SUMMARY	7
2. LIST OF EXPERTS AND WORK METHODOLOGY.....	10
2.1 WG 5.1 DATA MANAGEMENT AND EXPLORATION	10
2.2 WG 5.2 UNCERTAINTIES (UQ/V&V).....	11
2.3 WG 5.3 POWER & PERFORMANCE	12
2.4 WG 5.4 RESILIENCE.....	12
2.5 WG 5.5 DISRUPTIVE TECHNOLOGIES	13
3. WG 5.1 DATA MANAGEMENT AND EXPLORATION.....	14
3.1 INTRODUCTION.....	14
3.2 CHALLENGES	15
3.3 IMPACT ON APPLICATION DEVELOPMENT	15
3.4 FINDINGS.....	16
3.5 RECOMMENDATIONS.....	16
4. WG 5.2 UNCERTAINTIES (UQ / V&V).....	18
4.1 INTRODUCTION	18
4.2 CHARACTERIZATION OF UNCERTAINTY AND TERMINOLOGY	19
4.3 EMBEDDED UNCERTAINTY ANALYSIS METHODS.....	19
4.3.1 <i>Spectral methods</i>	19
4.4 DOE-BASED UNCERTAINTY ANALYSIS METHODS.....	20
4.4.1 <i>Methodologies</i>	20
4.4.2 <i>Tools</i>	22
4.5 FIRST RECOMMENDATIONS FOR EXASCALE.....	24
4.5.1 <i>Diffusion of tools and practices</i>	24
4.5.2 <i>Progresses in numerical analysis</i>	24
4.5.3 <i>Specifications for future software and architectures</i>	25
5. WG 5.3 POWER & PERFORMANCE.....	26
5.1 INTRODUCTION	26
5.2 THE REMAINING KEY ENERGY EFFICIENCY AND POWER MANAGEMENT CHALLENGES TO ACHIEVE EXASCALE SYSTEMS.....	26
5.3 CURRENT STATE OF THE ART	27
5.4 GAP ANALYSIS FOR EACH CHALLENGE.....	29
5.5 RECOMMENDED SPECIFIC ACTIONS FROM WG5.3	31
6. WG 5.4 RESILIENCE.....	32
6.1 INTRODUCTION	32
6.2 EESI1 RECOMMENDATIONS	32
6.3 RELIABILITY, AVAILABILITY SERVICEABILITY SYSTEM.....	33
6.3.1 <i>At Node HW level</i>	33
6.3.2 <i>At Node system level</i>	33
6.3.3 <i>At interconnect level</i>	33
6.3.4 <i>At File system and storage level</i>	34
6.4 RUNTIME.....	34
6.5 HIGH PERFORMANCE CHECKPOINTING	35
6.6 MULTILEVEL CHECKPOINTING	36
6.7 ADVANCED FAULT TOLERANT PROTOCOLS	37
6.8 MPI AND OTHER PROGRAMMING MODELS.....	38
6.9 FAILURE PREDICTION	38
6.10 RESILIENT NUMERICAL ALGORITHMS	39
7. WG 5.5 DISRUPTIVE TECHNOLOGIES.....	40
7.1 INTRODUCTION	40
7.2 SEMICONDUCTOR TECHNOLOGY	40

7.3	PACKAGING.....	43
7.4	DATA TRANSFER.....	46
7.5	MEMORY.....	49
7.6	NETWORK.....	50
7.7	I/O SUBSYSTEM.....	50
7.8	COOLING	55
7.9	NEXT STEPS	55
8.	BIBLIOGRAPHY	56
9.	ANNEXES	58
9.1	LIST OF EXPERTS IN WGS.....	58

List of Figures

Figure 1: WP5 Experts distributed per Country (43 Experts from 9 Countries)	10
Figure 2: Complex work flow of Exascale applications	14
Figure 3: Principle of spectral expansions.....	19
Figure 4: Uncertainty analysis methodology	20
Figure 5: Common methods for sensitivity analysis	21
Figure 6: New non volatile memory technologies	36
Figure 7: Potential architecture of an Exascale computer node.....	37
Figure 8: Cluster of core blocks of the NTV chip (courtesy of S. Burkar)	40
Figure 9: Hierarchical structure of NTV chip (courtesy of S. Burkar)	41
Figure 10: Typical power dissipated by moving data across different layer of the memory hierarchy...	42
Figure 11: Estimated power dissipated by data movement using "intelligent" tapering techniques	42
Figure 12: Microchannel based cooling system	44
Figure 13: Different physical layout of 3D stacking with microchannel cooling.....	44
Figure 14: Lab test sample of a microchannel cooled chip	44
Figure 15: Single die design with processor, optical interconnect and 3D memory.....	45
Figure 16: 3D packaging of optical interconnect, memory and processor	45
Figure 17: Draft schema of photonic data transport circuit	47
Figure 18: Estimated roadmap of photonic technology	47
Figure 19: Simulated performances of different numerical kernels, for different switch infrastructure	48
Figure 20: Optical interconnection internal to the node of an exascale system	48
Figure 21: Main characteristics of optical interconnect (projected for 2016 and 2020)	49
Figure 22: I/O "middleware" stack for a typical HPC application.....	52
Figure 23: System Diagnostics/Analytics/Simulation , big picture architecture.....	53
Figure 24: ELOW High Level Architecture Summary	54
Figure 25: Performance, distance and cost phase diagram for different physical transport media	54

List of Tables

Table 1: Agenda of the Workshop on HPC and Uncertainties,Paris, 22-23 April 2013	12
Table 2: Agenda of the Workshop on Disruptive Technologies, Milan, 15 April 2013	13
Table 3: Main characteristics of NTV chips	41

Glossary

Abbreviation / acronym	Description
ABFT	Algorithm-Based Fault Tolerance for Linear Algebra
ACPI	Advanced Configuration and Power Interface
ADS	Adaptive Directional Stratification
ANL	Argonne National Lab
API	Application Programming Interface
ARM	Advanced RISC Machine
BMBF	Federal Ministry of Education and Research (Germany)
CDF	Cumulative Distribution Function
CEA	Commissariat à l'énergie atomique et aux énergies alternatives
CINECA	Centro di Calcolo Interuniversitario. The Italian Supercomputing Centre
CMOS	Complementary Metal-Oxide Semiconductor
CMP	Chip Multiprocessor
CPU	Central Processing Unit
CRESTA	Collaborative Research into Exascale Systemware Tools&Applications
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
DEEP	Dynamical Exascale Entry Platform. EU funded Exascale FP7 Project
DIMM	Dual In-line Memory Module
DKRZ	Deutsches Klimarechenzentrum, Hamburg, Germany
DOE	Designs of Experiments
DoE	Department of Energy
DRAM	Dynamic Random Access Memory
DVFS	Dynamic Voltage and Frequency Scaling i
ECC	Error Correcting Code
EDF	Electricité de France
EDT	Event driven tasks
EESI	European Exascale Software Initiative
EIOW	Exascale I/O Workgroup Middleware
ESREDA	European Safety, REliability and Data Association
FDL	Free Documentation License
FEM	Finite Element Methods
FeRAM	Ferroelectric RAM

FFT	Fast Fourier Transform
FLOPS	Floating Point Operations per Second
FORM	First Order Reliability Method
FW	Firmware
GENCI	Grand Equipement National de Calcul Intensif
GPU	Graphics Processing Unit
HDF	Hierarchical Data Format
HOPSA	HOlistic Performance System Analysis
HPC	High Performance Computing
HW	Hardware
IB	Infiniband
IESP	International Exascale Software Project
INFN	National Institute of Nuclear Physics, Italy
INRIA	Institut National de Recherche en Informatique et en Automatique, France
I/O	Input Output
LANL	Los Alamos National lab
LARS	Least-Angle Regression
LGPL	Lesser General Public License
LHC	Large Hadron Collider
LLR	Link Layer Retransmission
MCA	Machine Check Architecture
MPI	Message Passing Interface
NetCDF	Network Common Data Form
NoE	Network of Excellence
NTV	Near-Threshold Voltage
NVDIMM	Non-Volatile Dual In-line Memory Module
NVM	Non-Volatile Memory
NVML	nVIDIA Management Library
NVRAM	Non-Volatile RAM
OpenCL	Open Computing Language
OS	Operating System
PAPI	Performance Application Programming Interface
PCE	Parametric Constrain Evaluation
PCI-e	Peripheral Component Interconnect Express
PCM	Phase-Change Material
PCRAM	Phase-Change RAM
PDE	Partial Differential Equation
RAID	Redundant Array of Inexpensive Disks

RAM	Random Access Memory
RAPL	Running Average Power Limit
RAS	Reliability, Availability and Serviceability
RBD	Random Balance Designs
RBM	Reduced Basis Models
ReRAM	Resistive RAM
RMA	Remote Memory Access
R&D	Research & Development
SATA	Serial Advanced Technology Attachment
SORM	Second Order Reliability Method
SPMD	Single Program, Multiple Data
SPEC	Standard Performance Evaluation Corporation
SQL	Structured Query Language
SSD	Solid State Disk
ST-MRAM	Spin-Torque Magnetoresistive RAM
STTRAM	Spin Torque-Transfer RAM
SW	Software
TAU	Tuning and Analysis Utilities
TBB	Task Building Blocks
TCO	Total Cost of Ownership
TVS	Through Silicon Via
VVUQ	Verification, Validation and Uncertainty Quantification
UQ	Uncertainty Quantification

1. Executive Summary

HPC is a strategic instrument to advance scientific excellence and industry competitiveness. The technological evolution and the increasing computational demand will lead to a new generation of computers composed of millions of heterogeneous cores which will provide extreme performances, in the range of Exascale, in 2020. Such innovative architectures will lead to outstanding technological breakthrough possibilities both in computations and software challenges.

EESI 1 federated the European community and built a preliminary European cartography, vision and roadmap on HPC technology and software challenges.

Now, EESI 2 goes one step further towards implementation, by establishing a structure to gather the European community, by providing periodically cartography, roadmaps and recommendations in defining and following up concrete impacts of R&D projects, detecting disruptive technologies, addressing cross cutting issues and developing gap analysis methodology towards an Exascale roadmap implementation.

This Document reports the first year of activity of EESI 2 WP5 “Cross cutting issues Work Groups”. The objective of this WP is to create and manage five working groups (WG) of experts on cross cutting issues for:

- 1) Data management and exploration;
- 2) Uncertainties (UQ/V&V);
- 3) Power & Performance;
- 4) Resilience;
- 5) Disruptive technologies

Cross cutting issues address themes transversal to the different activities from applications to technologies so the activity in WP5 is synergic to the activity in WP3 “Applications” and WP4 “Enabling Technologies”.

“Cross Cutting issues” is a new WP which was not present in EESI 1, so almost all the topics are new, apart “Resilience” and, in some aspects, “Power and performance”.

This Deliverable reports the work done by WP5 in the first year of activity. The five WGs organized the activity with groups of experts, leaders in the specific scientific context of the WG. The activity proceeded first focusing better the state of the art of the topics addressed, then continued to better understand the evolutions in the domains and trying to identify a gap analysis and some recommendations for approaching the Exascale goal. Not all the WGs were able to produce a gap analysis at this stage, as the activity was new, so the gap analysis activity will be better refined in the next reporting period.

The main recommendations identified in the different WGs are reported below, with the intent to advocate actions to prepare European software initiatives for the emergence of exascale computing.

- Data management and exploration

Set up actions to address end-to-end techniques for efficient disruptive I/O and data analysis, to describe the full life-cycle of data for a set of applications in order to produce highly parallel data 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.

Promote research in transformational algorithms to address fundamental challenges in extreme concurrency, asynchronous parallel data movement and access patterns, new alternative execution models, supporting asynchronous irregular applications and resilience, to enhance data analytics and computational methods in big data scientific applications.

Promote 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. This research should also promote and support the adoption of efficient metadata specification, management and interoperability in different scientific disciplines, as a key element to govern the scientific discovery process.

- **Uncertainties (UQ/V&V)**

It is important to adopt uncertainty analysis in academic and industrial studies. The use of the uncertainty analysis methodologies requires competence that is somewhat different from the ones required to develop a simulation code, and a key issue is that of training.

Investment is also required in numerical methods. In order to deploy uncertainty analysis on highly CPU-consuming codes, two strategies should be followed: improving adaptive designs of experiments and progressing on surrogate models. Furthermore, traditional uncertainty analysis deals mostly with parameter uncertainty, a huge progress for the validation of scientific codes would be achieved by also taking into account model errors.

The software tools are very important for facilitating the uncertainty analysis dissemination in the numerical simulation community.

Investment on tools and middleware taking into account the problem of resilience to failures is important to make more robust uncertainty tools and therefore to facilitate their wider usage.

Last, modern multiphysics computations involve multiple levels of parallelism (domain decomposition, code coupling, multiscale, etc.). Support developing tools that ensure these different levels of parallelism should be combined with the ones related to the design of experiments for efficient parallelisation of the ensemble.

- **Power & Performance**

Support the development of standard interfaces for power monitoring and power management at all levels of the system architecture. This would need to involve industry and academia. This joined effort will have several outcomes, which could include extensions to performance monitoring standards, such as the Performance Application Programming Interface (PAPI), or the creation of a set of best practices on how to operate systems in an energy efficient manner. This effort should also produce energy efficiency benchmarks to guide and monitor the improvements in energy efficiency.

Define major training and education initiative to prepare developers to face the power wall challenge by applying energy-aware programming techniques. A manual of tips and tricks for green programming would also be an extremely valuable resource for the HPC community.

We need more experts and professional HPC developers to support the wider community with the more efficient use of the expensive Peta and Exascale systems.

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 codes.

- **Resilience**

Improve checkpoint/restart performance by improving Multi-level checkpoint/restart (by minimizing the overhead of copying checkpoint images between the different storage levels), leveraging application and data properties (like memory access patterns, redundancy across multiple processes data structures) to enhance checkpointing asynchrony.

Improve fault tolerance protocols to increase system efficiency and execution recovery performance in presence of fail stop errors. This requires to understand how message logging can accelerate recovery state inconsistency, how to leverage partial restart to improve system efficiency, how to exploit new MPI concepts like neighbor collectives and RMA. More fundamentally more exploration is needed to refine the notion of global state consistency in the context of HPC executions and take advantage of it.

Investigate alternatives to checkpoint/restart. This covers improving fault tolerance approaches based on task-based programming/execution models, developing new concept of application level process migration, improving replication to reduce its overhead in resources.

Develop a fault aware software stack. This requires that software involved in the resilience (including applications, runtime, OS, etc.) should be fault aware and a notification/coordination infrastructure should guarantee relevant and consistent notifications/decisions/actions between these software layers.

Improve failure prediction and proactive actions. There are essentially two main research problems: 1) increase significantly the number of correctly predicted failures and 2) design failure prediction workflow to work with extremely large and growing system data sets (>1GB per day).

Improve resilient algorithms for fail stop errors and data corruption and their integration in the global resilience design. In particular the composability of resilience algorithms with other fault tolerance solutions should be explored.

- **Disruptive technologies**

Concerning the roadmap to exascale, the following recommendations are suggested to easily take-up disruptive technologies that may become available.

Recommendations related to the I/O and Memory disruption:

- a) analyse alternatives to parallel Filesystems, improve and revise them, together with their usage model;
- b) rewrite applications I/O functionalities to work at an higher level (Data container);
- c) promote tiered memory and I/O systems;
- d) re-write application to improve data locality.

Recommendations related to Cooling technologies and Facility management:

- a) evaluate and implement energy aware monitoring systems (better if embedded in the operating system), schedulers and applications;
- b) search for opportunities of joint venture with energy company, or heat re-use.

Recommendations concerning the Network infrastructure:

- a) develop networking system with the possibility to implement an adaptive topology, to enhance the routing capability (useful to avoid message congestion and fault tolerance)
- b) develop active network chip that can perform some data processing "on the fly" (data conversion, compression, elemental arithmetic operation);
- c) tests and validate direct end-to-end data exchange technology.

Recommendations concerning Data Transfer technologies

- a) promote early adoption of photonic technology;
- b) find synergy between HPC and BigData workload.

Recommendations concerning Semiconductor technology:

- a) study and evaluate fine grain resource management to mitigate extreme parallelism;
- b) investigate and promote new parallel paradigms dataflow inspired, leading to tiny "codelets" that can be more easily scheduled, dispatched and placed close to the data (to avoid data movement);
- c) develop intelligent scheduling functionalities to move execution threads close to the data.

2. List of Experts and Work Methodology

The objective of WP5, leaded by Giovanni Erbacci (CINECA) is to create and manage five working groups of experts on cross cutting issues:

- WG5.1 Data management and exploration (chair: Francois Bodin, CAPS) ;
- WG5.2 Uncertainties (UQ/V&V) (chair: Vincent Bergeaud, CEA);
- WG5.3 Power & Performance (chair: Simon McIntosh-Smith, Bristol University);
- WG5.4 Resilience (chair: Franck Cappello, INRIA);
- WG5.5 Disruptive Technologies (chair: Carlo Cavazzoni, CINECA).

After the EESI-2 Kick-off meeting, held in Paris on 18 September 2012, the activity in WP5 started soon. All the chairs and vice provided a more accurate definition of their WGs, a methodology for the work activity and started the enrolment of the experts for each of the five WGs.

Regular monthly teleconfernces have been organised by the WP5 chair with the chairs and vice chairs of the five WGs to better co-ordinate the whole activity in WP5. The relevant material has been uploaded in the EESI internal web site.

All the WP5 chairs and vice chairs attended the first annual EESI conference organised on May 28 and 29 2013 in Le Tremblay near Paris. The meeting was the occasion to present the first activity in each WG and discuss and refine better the activity with the participants of the other WPs and some external experts.

A total of 43 expert have been enrolled in the 5 WGs of WP5, during the first year of activity. As presented in Figure 1, the experts represent 9 different Countries, including Japan and US, coming from Academia, Research Institutions and Industries.

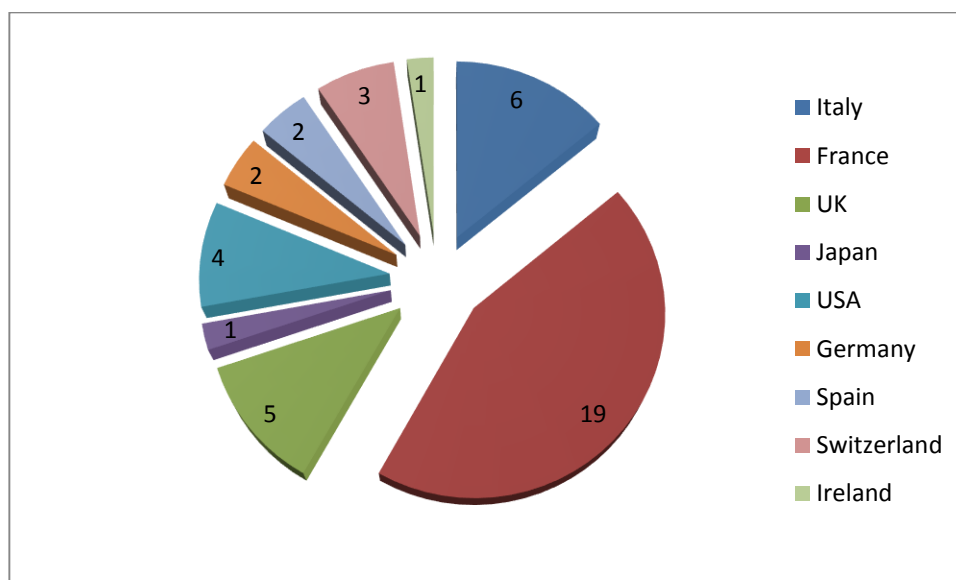


Figure 1: WP5 Experts distributed per Country (43 Experts from 9 Countries)

In the following, the list of the external experts and a brief description of the activity organisation is provided for each WG.

2.1 WG 5.1 Data Management and Exploration

WG 5.1 is chaired by Francois Bodin, a founder and the CTO of the French CAPS-Enterprise Company.

The following experts have been appointed to contribute to the WG:

- Jean-Michel Alimi, Observatoire de Paris, Meudon, France
- Gabriel Antoniu, INRIA, France
- Georges Hebrail, EDF, France
- Jacques-Charles Lafoucrière, CEA, France
- Malcolm Muggeridge, Xyratex, UK
- Kenji Ono, Riken, Japan
- Stéphane Requena, GENCI, France
- Alex Szalay, Johns Hopkins University, US
- Jean-Pierre Vilotte, Institut de Physique du Globe de Paris, France.

Once enrolled the external experts, the chair distributed a preliminary position document and a plan of the work. Then the WG5.1 activity proceeded in a electronic way by email communication and periodic teleconferences. Recently, the meeting in Le Tremblay was the occasion to refine the first results and the main recommendations for the data management and exploration topics.

2.2 WG 5.2 Uncertainties (UQ/V&V)

WG 5.2 is chaired by Vincent Bergeaud, Chef de laboratoire Génie Logiciel at CEA in France, and co-chaired by Alberto Pisanisi (EDF, France).

The following experts are contributing to the workgroup:

- Stefano Tarantola, JRC-ISPRA, Italy
- Christophe Prud'homme, University of Strasbourg, France
- Olivier Le Maître, LIMSI, Duke University, US
- Renaud Barate, EDF R&D, France
- Bertrand looss, EDF R&D, France
- Fabrice Gaudier, CEA, France

Once defined the list of the experts, WG5.2, organized a first workshop to investigate the topics of HPC and uncertainties. The workshop was organized in Paris on April 22 and 23, 2013. During the first day the presentations focused on numerical methods instead, on the second day, the software aspects were investigated. The agenda is reported the [Table 1](#) below.

Workshop on HPC and Uncertainties - Paris, 22-23 April 2013		
April 22 HPC and Uncertainties: Numerical methods		
9h30	Welcome	
10h	Introduction EESI2 WG 5.2	V. Bergeaud - A. Pisanisi
11h	Gaussian processes in uncertainty analysis	B. looss (EDF R&D)
12h	Lunch	
14h	Reduced basis methods and high performance computing: applications to non-linear multiphysics problems	C. Prud'homme (Univ. Strasbourg)
14h50	Spectral methods for Uncertainty Quantification	O. Le Maître (Duke University, LIMSI)
15h40	Break	
16h	Sensitivity analysis	S. Tarantola (JRC ISPRA)
16h50	Wrap up	
17h	adjourn	

April 23 HPC and Uncertainties: Software aspects		
9h	Design of experiments with the URANIE platform	V. Bergeaud, F.Gaudier (CEA)
9h50	Deployment of Design Of Experiments with OpenTurns software	R. Barate (EDF R&D)
10h40	Break	
11h	Applicative needs	A. Doering (Oxford University)
11h30	Work plan and conclusion	V. Bergeaud - A. Pasanisi
12h30	Adjourn	

Table 1: *Agenda of the Workshop on HPC and Uncertainties, Paris, 22-23 April 2013*

The slides that were presented during the workshop are available on the EESI website:

<http://www.eesi-project.eu/pages/menu/eesi-access.php?g=88>

The workshop was the occasion for better investigate the different topics of uncertainties and allowed to define a set of recommendations for exascale.

2.3 WG 5.3 Power & Performance

WG 5.3 is chaired by Simon McIntosh-Smith, head of the Microelectronics Research Group in the Department of Computer Science at the University of Bristol (UK). Thomas Ludwig), computer scientist at the Climate Computing Centre University Hamburg (DKRZ, Germany) is the co chair; WG5.3 in the first year of activity, appointed five experts in the different fields of micro electronics, computer architectures, energy efficiency, computer applications and performance:

- Alex Ramirez, Barcelona Supercomputing Centre, Spain;
- Matthias Müller, RWTH Aachen University, Germany;
- Jean-Marc Pierson, Laboratoire IRIT, Université Paul Sabatier, Toulouse, France;
- Laurent Lefevre, INRIA / University of Lyon, France;
- James Perry, EPCC, University of Edinburgh, UK.

The WG5.3 technical experts have held a number of conference calls to establish the scope of this working group. The interactions culminated in a face to face meeting organised to coincide with International SuperComputing (ISC) in Leipzig, Germany in June 2013. The majority of the technical information in this report was gathered from a pro-forma circulated amongst the experts at the end of this process, just after ISC. A final conference call was used to discuss the findings.

2.4 WG 5.4 Resilience

WG 5.4 is chaired by Franck Cappello, senior researcher at INRIA and Project Manager of Research on Resilience at the Extreme Scale at Argonne (US). WG 5.4 appointed at the beginning seven experts in resilience

- Luc Giraud, INRIA, France;
- Torsten Hoefler , ETH Zurich, Switzerland;
- Simon McIntosh-Smith, Bristol University, UK ;
- Christine Morin, INRIA, France;
- Bogdan Nicolae, IBM Research Lab, Dublin, Ireland;
- Pascale Rosse-laurent, BULL, France;
- Osman Unsal, Barcelona Supercomputing Centre, Spain;

After the engagement of the experts, the activity in the WG proceeded with the analysis of existing reports and the material produced in EESI 1. A WIKI has been set up for the WG members at

<https://collab.mcs.anl.gov/display/ESR/EESI2+Resilience+Working+Group> to share the documents and the analysis.

The experts worked electronically and different teleconferences have been organized to produce a gap analysis between existing reports and projection about the resilience challenge for exascale simulation. In addition, a set of recommendations based on this gap analysis have been produced. Since the EESI 2 project is in its first year and since the work of the WG started recently, the recommendations may differ from the ones that will be emitted by the WG at the end of the project.

2.5 WG 5.5 Disruptive Technologies

WG5.5 is chaired by Carlo Cavazzoni, head of the HPC Production Services Division at the CINECA supercomputing centre in Italy and co-chaired by Marie-Christine Sawley, Director of the Intel Exascale Lab in Paris. Facing exascale, disruptive technologies address challenge aspects in different fields like Semiconductor Technology, Packaging, Data transfer, Memory, Network, Cooling and Infrastructure, I/O Subsystem. To address the above issues the following main experts have been initially engaged:

- Shekhar Borkar, Director of the Intel Extreme-scale Technologies Lab, US
- Bruno Michel, IBM Research Laboratory, Zurich, Switzerland
- Patrick Demichel, HP, Lyon, France
- Piero Vicini, INFN National Institute of Nuclear Physics, Rome, Italy
- Giampietro Tecchiolli, Eurotech, Italy
- Malcolm Muggeridge, Xyratex, UK

As a first action of this WG, the expert have been interviewed by phone and they have been requested to send papers and reference of their activity on disruptive technologies, The material produced has served as the basis to prepare a WG workshop. The workshop took place in Milan, Italy on April the 15, 2013. The agenda of the workshop is reported in **Table 2**. The workshop was the occasion to better analyze and investigate the disruptive technologies in different fields and produce some initial roadmaps and recommendations.

Further experts will be engaged in the next period to address specific issues related to disruptive technologies.

EESI2 WG5.5 Disruptive Technologies Workshop Milan (Italy) April 15th, 2013	
11:00	<i>Introduction.</i> Carlo Cavazzoni (CINECA) Marie-Christine Sawley (Intel)
11:30	<i>Cooling and engineering: high efficiency solutions</i> Giampiero Tecchiolli (Eurotech)
12:15	<i>Lunch</i>
13:30	<i>I/O Technologies</i> Malcolm Muggeridge (Xyratex)
14:15	<i>Packaging and microfluidics</i> Bruno Michel (IBM)
15:00	<i>Silicon photonics</i> Patrick Demichel (HP)
15:45	<i>Coffee break</i>
16:00	<i>Semiconductor Technology: Near threshold voltage</i> Shekar Burkhar (Intel)
16:45	<i>Network technology: Adaptive free devices</i> Piero Vicini (INFN)
17:30	<i>Discussion on possible recommendations</i>
18:30	<i>Wrap-up</i>

Table 2: Agenda of the Workshop on Disruptive Technologies, Milan, 15 April 2013

3. WG 5.1 Data Management and Exploration

3.1 Introduction

This section proposes R&D actions/initiative on one of the major challenge of exascale applications: Data management. On the new generation HPC systems, the memory by core will decrease dramatically and at the same time data to be treated will increase dramatically too [1] [2] [3]. The recommendations aim at ensuring a coherent approach toward the evolution on management of I/O, such as big data transfer, storage, compression, massively parallel I/O, memory access, memory storage, etc.

Previous European Exascale Software Initiatives [4], [5], have explored the data issues from the technological point of views. For instance they identified critical topics such as “parallel file systems, disaster recovery mechanisms, mechanisms for end-to-end data integrity, data mining and visualization tools, data reduction techniques to carry out in-situ domain-specific data reduction and feature extraction, etc.”

Workgroup 5.1 has been addressing “*Data management and exploration*” in Exascale applications viewed as the organization of the scientific discovery workflow. This is illustrated in Figure 2. In this figure, data may follow any paths (blue arrows) in the ecosystem, each component having its own performance profile, quality of service and cost. For instance while HPC technology optimizes writing in parallel the data, data mining techniques favour reading. Choosing to use one or the other technology must be carefully planned according to a global view of the workflow.

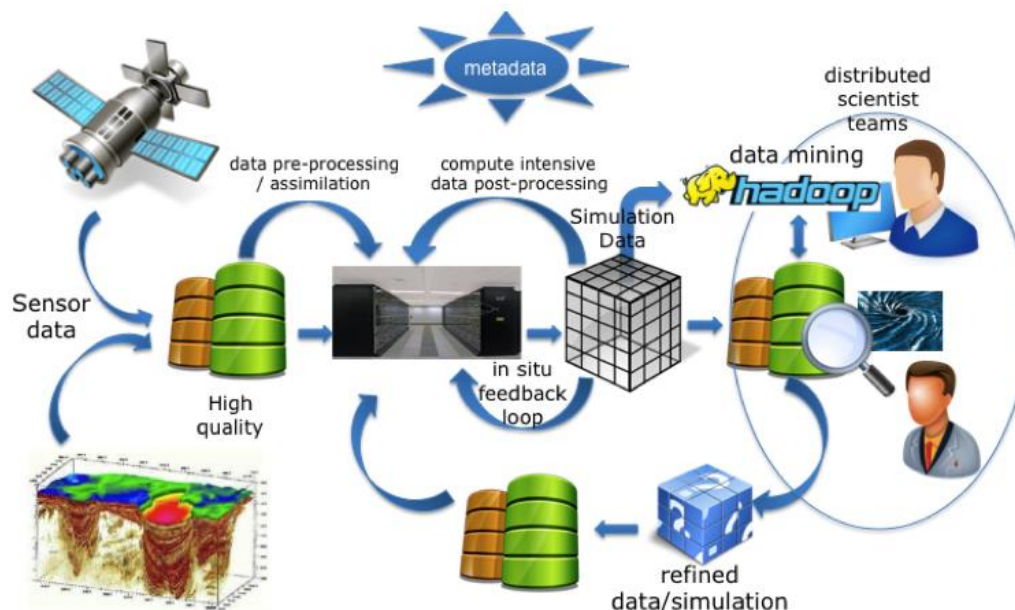


Figure 2: Complex work flow of Exascale applications

One important consideration in this work is the rising price of IO systems on one hand. On the other hand, as a deluge of data is to be expected, synergies between big data and traditional HPC techniques have to be well thought-out. Data types are also an important concern. For instance, data from sensors cannot be regenerated and must be stored safely while some data produced by simulation may be easier to re-compute when combine with in-situ data processing technique. Each data must be stored and organized to use the proper resources. As well metadata must be kept consistent all the way. This likely will strongly disrupt current practices.

In a nutshell, data management and I/O performance will strongly influence for the design of applications. However 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.

3.2 Challenges

When addressing end-to-end data life cycle many challenges arise due to combining the technology, the human resources and the ecosystem economy.

Designing an exascale applications 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.

Because of the expected deluge of data, new data analysis techniques must be designed. 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 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 favour 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 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 trade-off 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.

Maybe the biggest challenge of all, is to provide scientists with an ecosystem that is stable, intelligible and efficient. The exascale technology is very likely to have many handling hard constraints and a high operational cost (e.g. energy). Missing to understand the full consequences of technical choice on the complete workflow is likely to produce expensive use of resources and high probably of application development failure.

3.3 Impact on application development

This paragraph explores the potential data management issue and exploration impact on applications implementation and design.

A first trade-off to deal with is concept of “what data to output?”. For instance, it is frequent that adding more in-situ computation will negatively impact the efficiency of the simulation part of the applications. However, if this later provides for a faster and simpler analysis of the data, it will be worthwhile to pay the corresponding penalty. It is important to remember that human time, even in exascale environment remains the most expensive resources.

The data life cycle must be clearly understood to allow building an indexing and typology of the data that promote an efficient use of the different storage systems. The most reliable storage must only be

used in a cost-effective manner. For instance, it is necessary to distinguish the needs in pre- and post-processing so that the right technology could be used. Typically, three cases can be distinguished:

- 1) Post processing very large, out of memory data that requires powerful computing power (e.g. out of memory FFTs);
- 2) In-memory processing of mid-size chunk of data (e.g. can benefit of Hadoop technology);
- 3) Complex search with associative patterns over very large, out of memory data.

These techniques, to be fully exploited in an HPC context, will require disruptive practices.

Applications must allow optimizing the use of IO bandwidth thanks to interleaving compute and data transfers in a manner known/understood by the system. Indeed, contrary to computing resources, an application uses IO only at some execution point; this idle IO time can be exploited by another applications in order to make an efficient use of the IO sub-systems.

When designing the simulation, all numerical model trade-offs must be considered in order to minimize fault-tolerance needs, in/out-situ analysis, etc. For instance, it is probably better considering multiple middle scale simulation to build the full picture rather than a large atomic one. The right trade-off is in the end dictated by the "economy" of the exascale system.

As a consequence of the previous considerations, system and programming environment designers should provide to application developers efficient and standard APIs (or other methods), and corresponding best practices, to drive the hierarchies of storage to use and to describe more about the exploitation of the data.

3.4 Findings

Here is a summary of the findings of WG 5.1. These findings are consequence of the impact of data management and exploration in a complete workflow of an exascale application:

- 1) Both HPC and Database communities need to connect to design the technology and corresponding best practices;
- 2) There is a need for describing the technology deployment scenarios and the available options for organizing the data storage and processing flow. This aims at allowing system designers to understand off-line analysis, in-situ analysis, hybrid schemes, etc. The outcomes of these scenarios will be to identify big data technology and HPC technology synergies, identify workflow time-consuming / costly parts for a given application domain, to help to carefully examine the candidates of exascale computing platforms;
- 3) Exascale technology should also be available as many peta-scale systems in order to allow an adaptation of the computing and storage strategy according to the scientific objectives. Furthermore smaller size machine may be very convenient for data post-processing;
- 4) Data storage management must be flexible enough to accommodate use change of data (i.e. locality optimization);
- 5) Exascale technology is asking for new support people with technical profile that can bridge the gap between the "data-I/O" technology and the applications / scientific discovery process. There is a strong need in training of engineers in I/O systems.

Overall, this translates into the need to build an ecosystem where computing, storage, network resources uses/deployments (and corresponding business model) are carefully planned and stable overtime to allow an efficient local (e.g. scientist view) and global (e.g. computing centre operators) utilization. Support engineering teams, able to provide insights to scientists from the design phase to the implementation phase of the applications, will be a key component of this ecosystem.

3.5 Recommendations

The main recommendation of WG 5.1 is to set up actions 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:

- Support research in transformational algorithms to address fundamental challenges in extreme concurrency at the benefit of data analytics and computational methods for data intensive applications;
- Support highly parallel data workflow, encompassing I/O middleware and scientific data formats supporting high-level data objects and data access patterns, scientific database technologies and indexing methods;
- Push research in advanced data analytics algorithms and techniques to face the analysis of big data in different scientific applications;
- Support the adoption of efficient metadata specification, management and interoperability in different scientific disciplines;
- Specification scenarios for technology deployment and the available options for organizing the data storage and processing flow;
- Gathering Big data and HPC experts to identify best practices to be convey;
- Specifying curriculum for support engineering teams;
- Development of mini-apps to help conducting research in IO and data managements.

If these recommendations are implemented we expect the following outcomes:

- 1) Better designed applications and ability to innovate by cross fertilization of HPC and Big Data technology;
- 2) Best practices to be available to scientist and support teams (for instance via a Massive Open Online Course);
- 3) Coherent policy for managing exascale resources;
- 4) Evaluation of exascale platforms in regards of the full operational chain;
- 5) Afford the analysis challenges posed by the big data deluge in different scientific domains.

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

4.1 Introduction

Computer simulation is undoubtedly a fundamental question in modern science and engineering. Whatever is the purpose of the study, computer models help the analysts to forecast the behavior of the system under investigation in conditions which cannot be reproduced in physical experiments (e.g. accidental scenarios), or when physical experiments are theoretically possible but at a very high cost.

The need for simulating and forecasting gave indeed a dramatic momentum in the last decades to the growth of computers' power and vice versa. Since the very first large scale numerical experiments carried out in the 40's, the development of computers (and computer science) has gone pair-wise with the will of simulating more and more deeply, more and more precisely, physical, industrial, biological, economic systems.

A deep change in science and engineering has gone on in the last decades in which the role of the computer has been compared to the one of the steam engine in the first industrial revolution [22]. Together with formulating theories and carrying physical experiments, computer simulation has rapidly become a *third way to Science* [23] which allows solving problems which were absolutely unaffordable in a not so far past.

We believe in computer simulation as a major tool in daily engineers' work; simulation is a great tool for understanding, for forecasting, for guiding decision. We think that the possibility to simulate more and more complex phenomena, taking into account the effect of more and more input parameters must be seen as a chance, but, at the same time, we are aware of the fact that quantitative uncertainty assessment of results is a fundamental issue for assuring the credibility of computer model based studies, and represents a challenge too.

Besides technical and theoretical difficulties, maybe the most challenging point is, in industrial practice, to bridge the cultural gap between a traditional engineering deterministic viewpoint and the probabilistic and statistical approach which consider the result of a model as an "uncertain" variable.

Even if the fundamentals of these topics are rooted since decades in probabilistic and statistic literature, in the last years there has been a considerable rise of interest in industries and academia in the uncertainty quantification (UQ) of computer models' results.

A quick look at the recent bibliography can witness the variety of disciplinary fields involved: e.g. nuclear waste disposal, water quality modeling, avalanches forecasting, welding simulation, buildings performance simulation, galaxies formation, climate modeling, fires simulation etc.

In the last decade, in the frame of an ESREDA (European Safety, REliability and Data Association) project, CEA and EDF R&D settled a global methodology of uncertainty treatment that has been now accepted and improved by industrial and research institutions. As currently deployed in the industrial practice of engineering, the methodology essentially focuses on the so-called *parametric uncertainties*, i.e. the ones affecting the input parameters of a model, whatever it is: a complex numerical code which requires an approximated resolution or an analytical expression. It does not question explicitly uncertainties attached to the computer model itself, coming from the necessarily simplified modeling of the physical phenomenon under investigation, nor numerical uncertainties due to its practical implementation into a computer code.

The step forward is to develop and to spread in the engineering community an enhanced unified framework for model verification & validation and uncertainty quantification, what is commonly called *VVUQ*. This unified framework shall 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.

HPC and uncertainty quantification have a two-sided relationship. On the one hand, the ever increasing size of the computational data leads to increasing sources of uncertainties, due to the accumulation of numerical errors. On 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, as reported in this section, aimed at exploring these different aspects of the relationship between uncertainties and HPC. We will identify methodologies for

the analysis of these uncertainty sources, software tools related to uncertainty analysis and give guidelines for the evolutions required both in tools and in methodologies for exploitation of Exaflop machines.

4.2 Characterization of Uncertainty and Terminology

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. This process is now commonly called VVUQ (Verification, Validation and Uncertainty Quantification). 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 [24],[25].

4.3 Embedded uncertainty analysis methods

Embedded methods for uncertainty analysis fall in two main categories: adjunct methods and spectral methods. Here we introduce only the spectral method, and leave the analysis of adjunct methods for the future activity of the WG.

4.3.1 Spectral methods

Spectral methods are based on the principle illustrated by Figure 3:

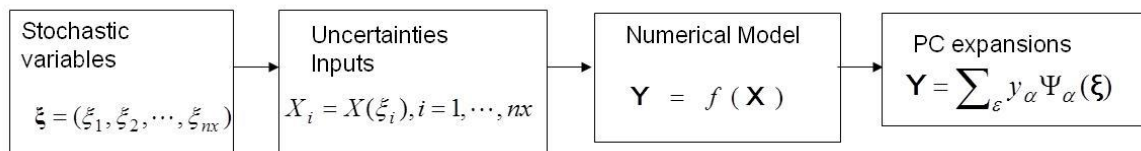


Figure 3: *Principle of spectral expansions*

Uncertain input parameters are written as functions of stochastic variables. Propagating these functions via the numerical model, one obtains output variables which are themselves functions of the stochastic variables. The uncertainty analysis consists in defining the spectral decomposition of the output variables y . To do so, two major strategies arise from this decomposition. The first one is a non intrusive method that is akin to the methods described in the 'DOE-based uncertainty analysis methods' paragraph. The second one is a Galerkin projection method. This method consists in using the orthogonality properties of the spectral decomposition in order to write a set of problems, each problem corresponding to one of the functional of the spectral decomposition basis. Except in the case of a fully linear model, the problems are coupled [26].

This method offers an accurate framework explicitly computing parametric uncertainty on every point of the computational domain. It was effectively demonstrated for hyperbolic systems as discussed in Olivier Lemaître presentation during the WG5.2 workshop in Paris on 22 and 23 April 2013.

4.4 DOE-based uncertainty analysis methods

In this chapter the methodologies and the tools for the uncertainty analysis methods based on Designs of Experiments (DOE) are presented.

4.4.1 Methodologies

DOE based methods consist in running the numerical model a number of times in order to span the range of variations of the uncertain variables. They have encountered a lot of success in industrial and research applications because they are non intrusive: they do not imply any modification in the numerical models themselves, consisting in smart design of the numerical experiments which are realized. They however require a very significant computational power, the number of points in the DOE depending on the smoothness of the outputs with respect to the uncertain inputs and on the type of quantity of interest being considered.

- Uncertainty analysis and model calibration methods

Figure 4 gives an overview of the uncertainty analysis methodology. In this framework, the DOE is realized in step C [25]. This is the stage at which computational power is required, since the numerical model is executed several times (typically from 10's to 1000's times).

Uncertainty analysis methods enable to give uncertainty measurement/rankings for various quantities of interests: variances, complete pdfs, distribution tails, etc.

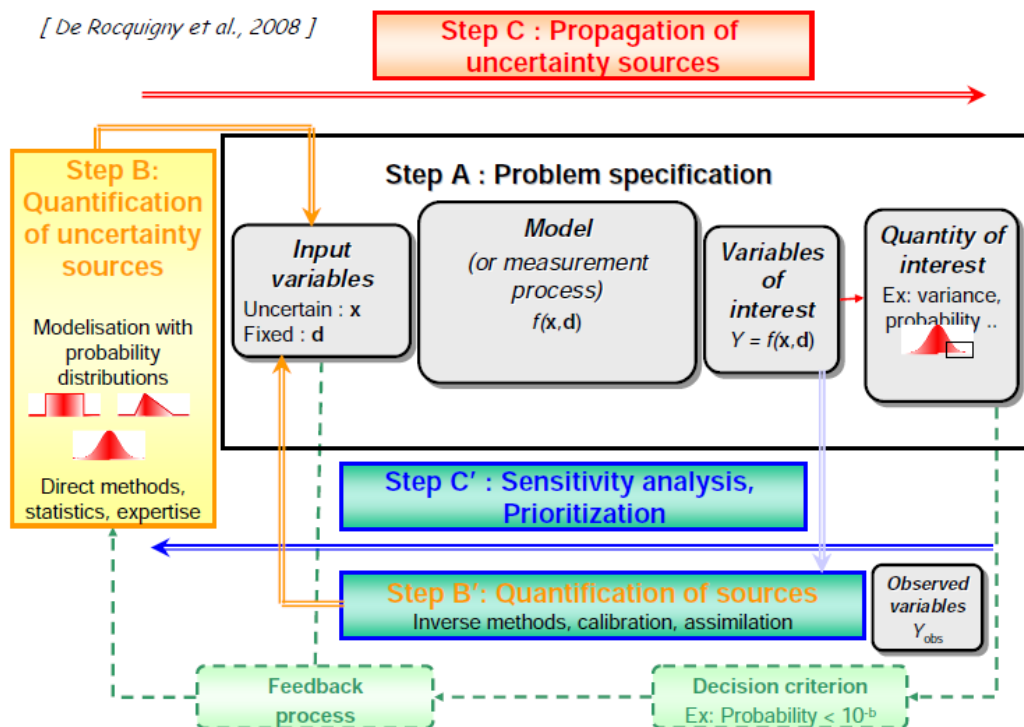


Figure 4: Uncertainty analysis methodology

The sensitivity analysis stage (step C' in Figure 4) will be performed differently according to the cost of the numerical model. Generally speaking, it is an iterative method: for complex/costly models, it is interesting to perform a screening stage in order to identify parameters whose uncertainty has little or no impact on the output uncertainty. It is then possible to simplify the DOE, considering a parametric space with smaller dimension.

Figure 5 gives an overview of methods commonly used for sensitivity analysis. In his talk during the WG 5.2 workshop, Stefano Tarantola gave an overview about the improvements achieved by recently developed approaches providing either more cost-efficient DOEs or more accurate strategies:

Using radial design based screening methods instead of classical Morris method in order to save computations (in this way both Sobol' sensitivity analysis and screening analysis can be performed using the same design and therefore the same set of model runs [27]).

In cases in which selection of points is not possible, recent techniques have been developed to retrieve sensitivity indices: scatter-plot smoothing offers a possibility to retrieve first-order indices, moment-based methods offer strategies independent on the number of considered factors, EASI methods In [28] Plischke propose an accuracy equivalent to that of RBD methods [29] with no constraint on the design.

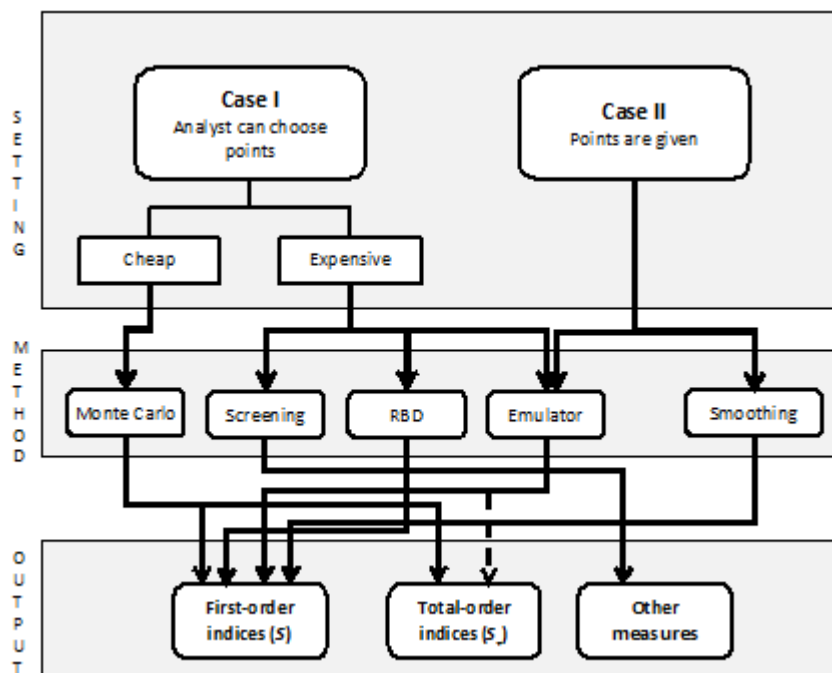


Figure 5: Common methods for sensitivity analysis

- Metamodels for HPC codes

As was seen in the previous paragraph, when dealing with DOE based methods, computational burden can exceed significantly the available computational resources.

The use of metamodels (aka surrogate models or emulators) enables to replace the execution of the numerical model by a much faster model. The construction of the metamodel can be made starting from a DOE but also using a given sample if necessary, but it typically requires a smaller number of points than the full uncertainty analysis. Also, the metamodel can be reused for various purposes (sensitivity analysis, ranking, optimisation, etc.)

Various metamodels techniques exist (polynomials, smoothing functions, radial basis functions, gaussian processes, neural networks, NISP), which have different properties in terms of physical interpretation, ability or limitations to deal with large number of parameters and strong non linearity. These different techniques employ various DOEs (factorial, Orthogonal Array, Latin Hypercube, Monte Carlo, D-optimal). The improvement of the DOEs to retrieve the same amount of information with reduced number of computations is a very active subject of research, which is of great importance for the usability of uncertainty analysis techniques on computationally intensive software.

In addition to these 'classical' metamodels, reduced basis methods offer a distinct approach. These methods consist in creating a numerical model which is cheaper than the original one, but which contains a full spatial representation of the solutions (as opposed to the metamodels discussed before which only describe the relationship between a small number of input parameters and a few global output variables). During the workshop held in Paris, Christophe Prudhomme gave an overview of his work on reduced basis models, including the tool FEEL++ and its ability to implement Reduced Basis Models for FEM models, both non linear and linear, with respect to the input parameters.

4.4.2 Tools

The realization of the DOE-based uncertainty analysis methods follows a pattern that is largely independent from the numerical models which are analysed (see Figure 3). 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 metamodels
- Computation of output statistical indicators

A number of tools have emerged, and many generic tools include some of the aspects of this procedure. Here, we will focus on the tools that have been presented in the frame of the WG 5.2 workshop:

- DAKOTA

This part is excerpted from <http://dakota.sandia.gov/about.html>.

DAKOTA is defined as a *Multilevel Parallel Object-Oriented Framework for Design Optimization, Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis*.

Written in C++, the DAKOTA (Design Analysis Kit for Optimization and Terascale Applications) toolkit provides a flexible, extensible interface between analysis codes and iterative systems analysis methods. DAKOTA contains algorithms for:

- optimization with gradient and nongradient-based methods
- uncertainty quantification with sampling, reliability, stochastic expansion, and epistemic methods
- parameter estimation with nonlinear least squares methods
- sensitivity/variance analysis with design of experiments and parameter study methods.

These capabilities may be used on their own or as components within advanced strategies such as hybrid optimization, surrogate-based optimization, mixed integer nonlinear programming, or optimization under uncertainty.

For a comprehensive overview of Dakota, see:

http://dakota.sandia.gov/papers/DAKOTA_Overview_Jan2010.pdf

- URANIE

Uranie is the Open Source platform developed at CEA/DEN dedicated to the study of propagation uncertainties, sensitivity analysis or model calibration in an integrated environment. It is based on Root (Version v5.32), an object-oriented software multi-platform developed at CERN for particle physics concerns, more exactly for data analysis generated by LHC (Large Hadron Collider) (see <http://root.cern.ch/> for more information). Uranie integrates a large amount of features enabled by Root and especially, a C++ interpreter, SQL databases access, visualisation tools and statistical analysis.

URANIE DOE distribution mechanism enables the user to leave the analysis script untouched regardless of the architecture on which it runs. It gives the possibility to mix together several levels of MPI-based parallelism: the numerical models used in the DOE can be serial codes, MPI-based parallel codes or simulations coupled via the SALOME framework [SALOME13].

The URANIE framework works by analyzing the environment variables in order to define a number of cores available for computation. The available cores are the used in order to distribute the simulation points according to the cores required for each computation (1 for serial codes, more for parallel computations or coupled computations).

The intrinsically parallel nature of the distribution of computations calls for excellent performances in terms of scaling. However, the parallel performance is limited by the I/O pattern of the codes. The simultaneous execution of hundreds or thousands of simultaneous instances of the same code can lead to heavy loads for the I/O system, which can result in poor performance.

Another track for improvement is the placement of processes on a processor. When using simulations with SALOME framework, processes which encapsulate the code services are launched without MPI,

therefore with no indication on the placement of the process on the processor. Therefore, processes often compete for CPU on the same core, leading to inefficient behaviour.

- OpenTURNS

OpenTURNS is a open source software under LGPL and FDL licenses for the code source and its documentation respectively, specifically designed for non-intrusive uncertainty quantification.

Running under the Windows and Linux environment, Open TURNS is a C++ library proposing a Python textual interface. It can be linked to any code communicating through input/output files (thanks to generic wrapping files) or to any Python-written functions. It also proposes standard interface for complex wrappings (distributed wrappers, binary data).

Gradients of the external code are taken into account when available and otherwise can be approximated automatically by finite differences schemes. In addition to its more than 40 continuous/discrete univariate/multivariate distributions, Open TURNS proposes several dependence models based on copulas (independent, empirical, Clayton, Frank, Normal, Gumbel, Sklar copulas). It offers a great variety of definitions of a multivariate distribution: list of univariate marginals and the copula, linear combination of probability density functions or random variables.

The propagation step is covered through numerous simulation algorithms. Open TURNS implements the innovative Generalized Nataf transformation and the Rosenblatt one for the FORM/SORM methods.

For the ranking analysis, Open TURNS implements the Sobol indices, and the usual statistical correlation coefficients.

Open TURNS is innovative by its input data model, based on the multivariate cumulative distribution function (cdf), which enables the usual sampling approach (statistical manipulation of large data set) but also the analytical approach: if possible, the exact final cdf is determined (thanks to characteristic functions implemented for each distribution, the Poisson summation formula, the Cauchy integral formula, etc.); furthermore, different sophisticated mechanisms are proposed: aggregation of copulas, composition of functions from \mathbb{R}^n into \mathbb{R}^p , extraction of copula and marginals from any distribution.

Open TURNS implements some up-to-date efficient sampling algorithms: it uses the Mersenne Twister Algorithm to generate uniform random variables, the Ziggurat method for normal variables, the Sequential Rejection Method for binomial variables and the Tsang & Marsaglia method for Gamma variables. The exact Kolmogorov statistics is evaluated with the Marsaglia Method and the Non Central Student and Non Central chi-squared distribution with the Benton Krishnamoorthy method.

Open TURNS is also the repository of some recent results of PhD researches carried at EDF R&D: sparse PCE based on the LARS method, or ADS Sampling (Adaptive Directional Stratification).

Difficulties faced when using OpenTURNS in the HPC context come from the variety of combinations that can be addressed in a DOE context. Indeed, the questions arising are the following:

- Use of a cluster (homogeneous, centralized) / a grid (heterogeneous, decentralized)
- Communication protocol with the cluster
- Which batch / grid manager
- Can we install softwares on the cluster
- Global / local (by node) filesystem
- Execution of OpenTURNS script on the client workstation / on the cluster
- Which middleware for the distribution on the cluster
- Size of input and output files of the solver code.

The varieties of contexts in which the platforms are used make it difficult to design generic solutions. Two compromises have been found to face the variety of problems:

- Using a distributed Python function (solution included since the version 1.1). OpenTURNS must run on a computation node and the distribution of the computations is made with SSH connections.
- Using SALOME distribution mechanism in order to perform the DOEs.
- OpenTURNS provides a module for using SALOME distribution mechanisms. Therefore, OpenTURNS uses the CORBA-based mechanism provided by SALOME for distributing computations. It uses a simple python mechanism for wrapping the numerical model.

4.5 First recommendations for exascale

4.5.1 Diffusion of tools and practices

As this document shows, 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.

4.5.2 Progresses in numerical analysis

As was shown in previous sections, 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.

Adaptive 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 hundreds 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, adaptive 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 computationally 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 of a metamodel representing 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 use of 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.

4.5.3 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.

5. WG 5.3 Power & Performance

5.1 Introduction

In the quest to achieve Exascale systems in the 2020 timeframe, energy efficiency has become one of the primary challenges. DARPA's 2008 comprehensive review of the technological challenges facing Exascale systems [30] was the first to identify energy as one of the primary, if not the primary, barrier to Exascale machines. Subsequent reports from the IESP and EESI-1 confirm this view [31], [32]. Early theoretical 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.

5.2 The remaining key energy efficiency and power management challenges to achieve Exascale systems

In WG 5.3, each expert was asked to describe what he or she believed are the critical challenges to Exascale remaining in the area of energy efficiency and power management. Each challenge was rated as critical, important or nice to have. Where experts identified closely related challenges, these have been combined.

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. A proposed extension is total-power usage effectiveness (see [33]). This definition cannot be applied without the ability to distinguish between different categories of power consumers in systems.

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) (see [34]) 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.

5.3 Current state of the art

Hardware energy monitoring: the latest hardware from vendors such as Intel, IBM, Nvidia et al now include quite comprehensive counters for energy-related metrics, such as energy consumption broken down per component, temperature etc. These counters need to continue to be expanded so that anything consuming more than, for example, 1% of the power in a node, has a hardware counter sampling it at an appropriate resolution, which can then be read by software under user control. The expanded set of counters should include node-level power supplies, memories, NICs etc. A system should then have a way to combine per-node energy information in a hierarchical fashion to produce a system-wide view of the energy consumption of a parallel job. Some examples of recently available energy counters are Intel's Running Average Power Limit (RAPL), AMD's Application Power Management, IBM's PAPI and NVIDIA's Management Library (NVML). At the time of writing, no on-chip energy counting capability has been made available to the user from other major manufacturers. Therefore a better understanding to access to these counters may be a key element in the analysis of applications to cope with the power wall.

Performance analysis tools: Scalable performance analysis tools such as Paraver¹, Scalasca², or Vampir³ supported through projects such as Mont-Blanc, DEEP, and CRESTA, represent the state of the art in HPC application analysis. There is still room for improvement in interfacing them to the power monitoring support in the systems, however the lack of a standard API is a significant barrier

¹ <http://www.bsc.es/computer-sciences/performance-tools/paraver>

² <http://www.scalasca.org>

³ <http://www.vampir.eu>

(as it is the lack of a standard API for performance counters, only alleviated in part by the current PAPI interface).

Power and Energy system profiling: the ability to use the new hardware counters for energy and temperature etc. need to be usable by today's software profiling tools. For example, the Score-P⁴ project provides a highly scalable measurement infrastructure and easy-to-use tool suite for profiling, event tracing, and online analysis of HPC applications. It has been created in the [German BMBF project SILC](#) and the [US DOE project PRIMA](#) and will be maintained and enhanced in a number of follow-up projects such as [LMAC](#) and [HOPSA](#). Score-P can now plug-in to the hardware energy counters in modern processors and make this information available in a standardized manner to a range of profiling and analysis tools, including Periscope, Scalasca, Vampir, and Tau. Intel's vtune profiler can also show energy and temperature measurements for a running application, and tools from the embedded vendors, such as ARM and Imagination Technologies, are quite sophisticated in their energy use measurement and reporting. Despite all this, there is a significant lack of publicly available information about how power is used in current HPC systems. While it is likely that vendors have this information, it is not disseminated. Research projects such as the PRACE prototyping work packages [18], or Mont-Blanc are generating a great deal of information about how energy is used in a system, and how it relates to other factors such as cooling, applications, etc.

Standard API for accessing energy information: in order for applications to be able to auto-tune themselves for optimal energy efficiency, they will need to be able to access information about their energy consumption in a standard manner at run-time. Today all the energy-related hardware counters are presented in proprietary fashion by each vendor. A standard API for accessing such information will enable applications and tools to adapt in real-time to each system. Today's examples of auto-tuning have been very successful, but this approach has largely focused on performance as the primary goal, with improved energy efficiency a fortunate side effect measured after the fact. Early results indicate that if energy efficiency information can be an input to an auto-tuning framework, larger energy efficiency gains can be made compared to auto-tuning for performance alone.

Performance and operating states in latter CPUs: the ability to use the processor power/performance states (P-states) and processor operating states (C-states) in future HPC applications is a key factor to saving energy. These mechanisms allow a processor to switch between different supported operating frequencies and voltages to modulate power consumption. The Advanced Configuration and Power Interface (ACPI) specification defines the CPU power management states (P-states), nevertheless their use is not common across all the manufacturers. Apart from CPU P-states there exists CPU C-states or power management states with the ability to turn off unused components and attain major energy savings. Different levels of C-states are defined; at higher C-states levels more components are shut down to save energy, incurring slower wake up times to recover into a normal operational state. An optimised configuration according to the current system workload will enable energy savings in the execution of applications. Recent research has shown that the tracing and profiling of these states help developers to understand the execution behaviour of their applications.

Modeling power and energy consumption: A better understanding of the power consumption in applications is another key factor to improve scientific applications. In this sense, power/energy models are an important issue to know when, where and how our applications consume energy. Current models are quite simplistic and rarely take a whole system view of energy consumption. They also tend to be system specific.

INRIA-Illinois-ANL Joint Laboratory for Petascale Computing. This US-based laboratory focuses on software challenges found in complex high-performance computers. The Joint Laboratory is based at the [University of Illinois at Urbana-Champaign](#) and includes researchers from the French national computer science institute called [INRIA](#), Illinois' [Center for Extreme-Scale Computation](#), and the [National Center for Supercomputing Applications](#). Much of the Joint Laboratory's work will focus on

⁴ <http://www.vi-hps.org/projects/score-p/>

algorithms and software that will run on [Blue Waters](http://jointlab.ncsa.illinois.edu/) and other petascale computers. Link: <http://jointlab.ncsa.illinois.edu/>

Mont-Blanc: The Mont-Blanc project deserves special mention as it is specifically focused on addressing the breadth of energy challenges for Exascale systems. Launched in October 2011 and based at the Barcelona Supercomputing Centre, Mont Blanc aims to develop a new type of system architecture that will be able to deliver Exascale performance while using 15-30 times less energy than current technology. It intends to achieve this by leveraging European expertise in energy efficient processor technology from the embedded and mobile markets, where European companies are world-leading. The hardware comprises multicore ARM processors with integrated OpenCL accelerators and Ethernet NICs, with high-density packaging. ARM processors currently dominate in mobile and embedded applications, where power efficiency has always been a priority, and it is hoped that they will lead to more energy efficient HPC systems.

HPC accelerators: many-core processors from vendors such as Intel, Nvidia and AMD have been demonstrating significant energy efficiency gains over traditional CPUs alone. However, modifying applications to use these very parallel architectures efficiently is a major challenge, beyond most software developers. Tools, application frameworks and software libraries that make it easier for more developers to tap these benefits could have a major positive impact on the energy efficiency of HPC.

Application specific systems: The EU-wide Human Brain Project is taking a very different approach to energy efficiency, intending to build very specialised hardware to solve one particular problem. The overall aim is to be able to simulate a human brain in as much detail as possible, eventually even simulating an entire brain, however the power demands make this extremely difficult on current computers. The project estimates that simulating a single neuron in software consumes 14 orders of magnitude more energy than an actual biological neuron requires. By building special purpose hardware that is optimised for simulating neurons, with memory built into the same chips as compute cores, researchers hope to be able to reduce this gap to only 5 or 6 orders of magnitude. Application-specific systems such as this are another potential avenue to address the energy efficiency challenges of Exascale systems.

Energy efficiency benchmarks: there are a few benchmarks available that address energy efficiency. The Linpack benchmark used to create the TOP500 and Green500 list has been extended to include power consumption. However, the metric is rather simple (MFlops/Watt) and the run rules for how to measure the power consumption lack precision. The SPECpower_ssj2008 benchmark was specifically created to measure energy efficiency, has very detailed and precise run rules, but is focused on Java workloads. SPEC OMP2012 is an application benchmark with scientific applications using OpenMP that has been extended with an energy efficiency metric and detailed run rules for energy measurement. However, it is limited to shared memory systems.

5.4 Gap analysis for each challenge

For each challenge previously identified, the group of experts was asked to provide a short gap analysis, including in their analysis: a) what is the goal for this challenge? b) what recent progress has been made towards addressing this challenge (in the last 1-2 years)? and c) what is the remaining gap to meeting this challenge, and how far do we have left to go?

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. Other research has also shown more benefit if the CPU C-states are traced: specifically, a proper study of discrepancies between C-states and task/performance traces can detect power sinks in the applications in order to relieve them. 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 an 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.

5.5 Recommended specific actions from WG5.3

The group of experts was asked to suggest any WG5.3-related actions relevant to “software for extreme scale computing”. Potential actions could include education, training (how can we attract and retain new talent?); research programs, including type of funding tools (NoE, IP, CS-CSA, etc.), budget and agenda; creation of a task force (max 6 months duration); center of excellence; useful tools; and any other relevant ideas. Suggestions from WG5.3’s team of experts included:

There is an urgent⁵ need for standard interfaces for power monitoring and power management at all levels of the system architecture. This would need 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)⁶. 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 centres 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.

⁵ It will take 2-3 years after this interface is defined until it actually becomes available in systems, and it will easily take 5 years until it is widely adopted in HPC sites.

⁶ <http://icl.cs.utk.edu/papi/>

6. WG 5.4 Resilience

6.1 Introduction

This section reports the activity done in WG 5.4, from the group of experts, on the topics of resilience. 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.

The working group has mainly based his gap analysis and recommendations on the following available documents: the [IESP](#) road map [35], the [EESI](#) 1 report [36], the report of the [ICIS](#) workshop (2013) and a recent report from [DoE](#) [37].

Members of the working groups also considered other publications, like papers published in conferences and journal to perform the gap analysis and establish their recommendations. These documents are cited in the following subsections.

In the following of Section 6, the EESI1 recommendations are first recalled, then the gap analysis and recommendations are presented for each of these eight important aspects:

- 1) Reliability, Availability Serviceability system
- 2) Runtime
- 3) High Performance Checkpointing
- 4) Multilevel Checkpointing
- 5) Advanced fault tolerance protocols
- 6) MPI3 and one sided communications
- 7) Failure prediction
- 8) Resilient numerical algorithms

6.2 EESI1 recommendations

Establish a Fault model for HPC system at Exascale:

- Start: now, Duration: 7 years, HR: 6 PM/year per system

Extend the applicability of checkpoint-restart:

- New FT protocol: Start: now, Duration: 7 years, 24PM/year
- Diskless checkpoint: Start: now, Duration: 4 years, 24PM/year

Failure avoidance: Develop tools for root cause finding:

- Start now, Duration 7 years, 24PM/year Fault & Failure prediction + proactive migration: Start now, Duration 7 years, 48PM/year

Non-transparent approach:

- API to allow description of application needs (critical data sets, redundant computations etc.), Start: now, duration 4 years, 24 PM/year
- Adapt and test 4 key applications for the API, Start: now, duration 4 years, 24PM/year Language and new paradigm for fault tolerance:

Develop new FT models based on non-volatile memory (task based, transactions, etc.)

- Start: now, Duration: 7 years, 48 PM/year Cross-layer fault consistency system:

System itself:

- Start: now, Duration: 4 years, 24 PM/year

Adapting all layers to use the system:

- Start:now, Duration: 4 years, 56 PM/year (5 layers: hardware, OS, runtime, application, job manager)

6.3 Reliability, Availability Serviceability system

6.3.1 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.

For cache and memory multiple hardware mechanisms offer error correction or protection: ECC cache, Memory address parity, Redundant Bit Steering, Memory scrubbing, Memory mirroring, Memory DIMM sparing, Memory rank sparing, Memory Enhanced Single Device Data Correction (SDDC+1) or Memory Enhanced Double Device Data Correction (DDDC+1) provides protection against memory soft errors, transient faults, stuck-bit, or up to DRAM device hard failure. To complete intra node data resilience researches have done on non volatile memory integration at node level.

For Internals links such as inter sockets links (ex QPI links for Intel) or socket to memory links, multiple mechanisms allow using link with downgraded capabilities to minimize fatal failure and allow some correcting actions or packet retry: protocol CRC protection, self-healing, clock failover, packet retry. Some platform also integrate some redundant processor-to- I/O PCI bus links

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.

6.3.2 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. Machine Check Architecture recovery is one example of implementation of intel hardware errors reporting. Machine Check Architecture (MCA) refers to a mechanism in which the CPU reports hardware errors to the operating system. Next generation of MCA will be extended to allow 'Corrupt Data Containment' (also called as data poisoning). New MCA architecture is managed through BIOS and firmware and further extends the uptime when certain uncorrected errors are detected. A similar concept in IBM servers is referred to as first failure data capture.

Even if software is not a main root cause of failures on hpc systems it is important to minimize software faults impact. At OS level solutions based on virtualization are studied to decrease the severity of operating system software faults.

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 platforms have a critical need of fault aware software stack.**

6.3.3 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.**

6.3.4 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.**

6.4 Runtime

Although runtime has been identified as a critical issue for Exascale resilience by several recent reports, among them the International Exascale Software Project Roadmap, 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. We base this expectation on the following: (i) the trend of using off-the-shelf components expected to continue for Exascale computing, (ii) multi- and many-core processors will continue to dominate the off-the-shelf computing market (iii) the modified Moore Law which stipulates that the number of cores on a Chip Multiprocessor (CMP) will double every 18 months with each new node technology will continue to hold until at least the Exascale timeframe.

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. In these systems, a task is fired only when all its inputs are ready, the programmer annotates task directionality information as well as task inputs and outputs; therefore the state to be checkpointed could be minimized since the runtime has all the state that the task produces. Moreover, both checkpointing and recovery are asynchronous since independent tasks that are not affected by the error could continue to execute. As a final benefit, since all task inputs and outputs are known, it might be possible to recover even from long error latencies since a detected error could be traced to its source by walking the task dependencies in reverse.

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. Note that this propagation of events is a relatively recent phenomenon; a couple of years ago; even such fundamental information such as on-chip thermal sensor data was shielded from the runtime and was handled directly by the hardware. In recent years, we have seen thermal, power dissipation and other reliability events exposed to the runtime; such examples include the energy hardware performance counters introduced by Intel in their Sandy Bridge architecture, and the ability to signal not only unrecoverable errors but also corrected errors as well, again in the Sandy Bridge. We expect that we will see other functionality to be exposed beyond just reporting reliability related hardware information to the runtime, such as giving the runtime the option to decide how the hardware should utilize a particular local failure recovery policy. Currently this decision is taken by the hardware automatically. Examples of these hardware-baked error recovery policies include the microarchitectural checkpoint/restart mechanism, called instruction replay technology by Intel, in the Poulson processor to recover from soft errors, or the Intel cache safe technology for Montecito processor which transparently remaps a permanent error in a cache line to a spare cache line; a last example is the inclusion of an extra core for reliability purposes in the IBM BlueGene/Q processor. 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.

6.5 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.** Reducing checkpoint sizes has traditionally been attempted using incremental approaches and compression/deduplication techniques for each process individually. More recently, collective techniques that leverage redundancy across multiple processes have shown dramatic reduction of overall checkpointing data compared to individual techniques. **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** (e.g. work with memory regions instead of files) while leveraging locality as much as possible but still keep the checkpoints resilient. This involves creating redundancy (through erasure coding or replication) or exploiting already existing redundancy (e.g. by identifying it through deduplication) and then place it in such way as to minimize the impact of failures on the ability to recover checkpoints.

Another direction where significant potential has been recently shown is to complement checkpoint-restart with other weaker resilience techniques that can absorb a part of the failures, effectively lowering the failure rate for which checkpoint-restart is required, which in turn means less frequent checkpointing and thus lower overhead. The key in this context is to understand how expensive such complementary techniques are and how successful they are in absorbing failures in order to pay off. One such promising technique is proactive response to faults based on failure prediction, e.g. migration of processes suspected to fail in the near future to safer nodes. 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.

6.6 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 require 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. Another issue with checkpointing on remote file system is that it could be a significant source of execution failures. Users have reported cases where executions were stopped and ultimately crashed because the application was not able to perform checkpointing successfully.

Multilevel checkpointing was presented in the IESP and EESI1 reports. Since their publication, progress has been made in this domain to include more storage levels. The 2 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. If checkpointing on remote file system fails then this should not make the whole application fail. Since multiple copies of the checkpoint are available (multi-level checkpointing) and provide redundancy then there is no reason why the failure of one level would lead to the failure of the full 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 (see Figure 6, extracted from Rob. Schreiber talk for the 30 years of parallel computing at Argonne National Laboratory) 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.

New memory on the horizon

- Spin-Torque-Transfer RAM (STTRAM)
 - Grandis (54nm, acquired by Samsung)
- Phase-Change RAM (PCRAM)
 - Samsung (20nm, diode, up to 8Gb)
 - Micron and Nokia – In phones now
- Resistive RAM (ReRAM)
 - Panasonic (180nm process, 4-layer xpoint)
 - Unity Semi (64MB, acquired by Rambus)



Figure 6: New non volatile memory technologies

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.

Figure 7 (extracted from Rob. Schreiber talk for the 30 years of parallel computing at Argonne National Laboratory) shows a potential architecture of an Exascale computer node. In this design, computer

nodes are equipped with hybrid memory technologies: DRAM and NVRAM. Both are addressable (the NVRAM is not used here for a block device).

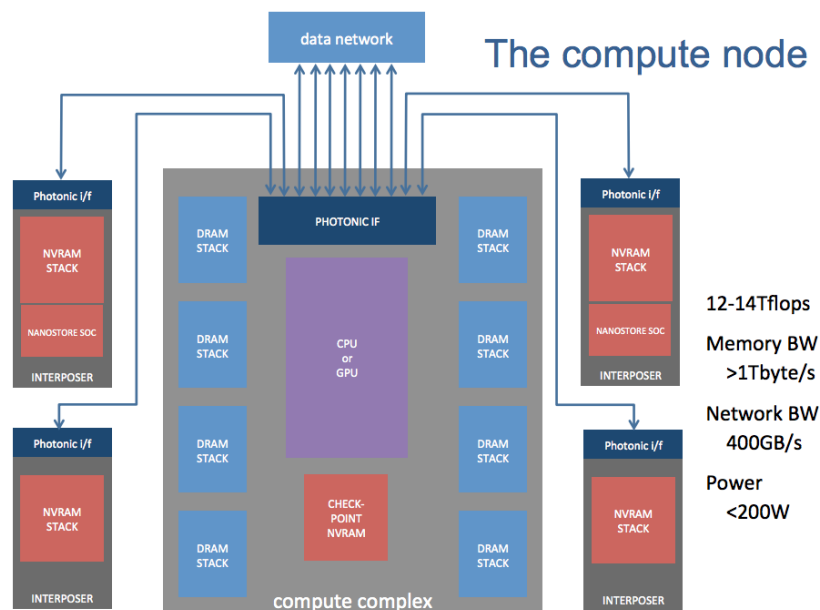


Figure 7: Potential architecture of an Exascale computer node

More research is needed to understand how to make the best usage of future non-volatile memory to for fault tolerance.

6.7 Advanced fault tolerant protocols

Fault tolerant protocols play a critical role in capturing and restoring a consistent state of a parallel execution. Recent progresses in this domain concern hierarchical protocols combining coordinated checkpointing with some form of message logging. The benefit of using such hybrid protocol is avoiding global restart when only a small fraction of the execution processes fails, which represents the large majority of failure cases. 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. Most of existing protocols need to log all communication events (reception) even the ones inside each cluster. This event recording is typically increasing the communication latency and makes these protocols impractical. Research has focused on how to reduce drastically the overhead of event recording and how to completely avoid it. Drastic reduction of event recording can be obtained by storing event logs on remote clusters volatile memory. Avoiding completely event recording assumes some property on the communication patterns (Send-determinism). Until recently avoiding event recording implied a centralized recovery procedure. This problem has been solved recently with the notion of SPMD determinism.

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.

Recent results on performance modeling of hierarchical protocol suggest that the best way of taking advantage of partial restart is to schedule other jobs on nodes hosting the non-restarting processes while the failed processes are restarted and until they recover the state just before the failure. At that

point all processes of the initial jobs (the non restarting processes) are rescheduled on their initial nodes. This approach assumes a fast checkpoint/restart procedure for the non-restarting processes and the jobs scheduled on the nodes hosting the initial execution. **While this has been demonstrated theoretically, this approach needs more research on the experiment side.**

It has been observed several time that message logging could accelerate recovery significantly. The explanation is simple: message logging allows a restarting process 1) receiving all incoming message without waiting and 2) to skip message emissions during the recovery phase (because these messages have already been received by the non restarting processes). It has been observed recently that the less messages are logged the smaller is the recovery acceleration. So to maximize recovery acceleration, one could be tempted to construct clusters in hierarchical fault tolerance protocols to maximize message logging. However this goal is clearly the opposite of the one that motivated hierarchical protocols: reducing the amount of log messages. **Since both properties are desirable (fast recovery and limited message logging), new clustering algorithms need to be designed to target user defined recovery speed/message logging trade-offs.**

6.8 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.

Neighborhood collectives allow the user to specify a data exchange pattern declaratively which is then automatically transformed into a collective operation when called. The specification is performed at a fixed point in time and later 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 such operations 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.**

6.9 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%.** There are several research opportunities in that domain: develop failure precursor detectors and develop better predictive algorithms.

Nevertheless, all failure prediction results have a major weakness: they perform prediction from event/failure logs and predict what is next on the log. Since researchers have access to what should be predicted, even without knowing it, they use this information to improve their prediction algorithms. **So 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. To give an example, the HELO event clustering tool that clusters events in different groups according to their types was generating 10s of clusters for system logs available publically in 2010 (LANL logs, BlueGene/L logs, etc.). For Blue Waters, HELO generates thousands of clusters, that two order of magnitude higher than before. This illustrates the difficulty of online failure predictions on real large system: if the very first stage of prediction (event clustering) struggle to handle the massive volume of information generated large systems, then how the other stage can generate accurate predictions?

6.10 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.

7. WG 5.5 Disruptive technologies

7.1 Introduction

The global trends in system architectures in the last years have been mostly following incremental improvement. In order to keep the same pace for the performance of the machines, the proposed architectures offer to exploit ever increasing levels of parallelism. In doing so, the architecture design puts higher stress on software, communication, system size and power consumption. This WG focus on the search of disruptive candidate technology/components that have good potential to create a discontinuity on the current architectural trends while reducing the demands on other component of the HPC environment, especially regarding system density and efficiency. The activity analysis done in the WG regards the main different aspect/component of an HPC system: Semiconductor Technology, Packaging, Data transfer, Memory, Network, Cooling and I/O.

7.2 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. On one hand, very large scale integration used in manufacturing processes is a mature technology that allows very low end-user costs by relying on high production volumes. On the other hand, large scale integration is rapidly approaching some physical limitation of silicon semiconductors, namely: the integration scale (a transistor cannot have less than few silicon atoms) and power dissipation. Therefore, 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.

In the WG 5.5 expert's vision (i.e. Shekhar Burkar, Intel), 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. The efficiency of CMOS transistor against the supply voltage peaks close to the transition between conducting/non conducting states of the transistor itself. This gives the possibility 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 (typically a ring), to form a cluster (16 blocks), see Figure 8, finally 16 cluster can be connected together into a single chip, with a global shared non-coherent address space, see Figure 9.

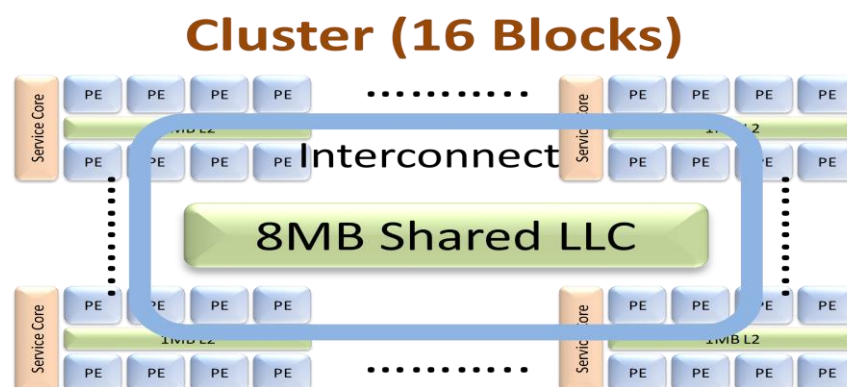


Figure 8: Cluster of core blocks of the NTV chip (courtesy of S. Burkar)

Processor Chip (16 Clusters)

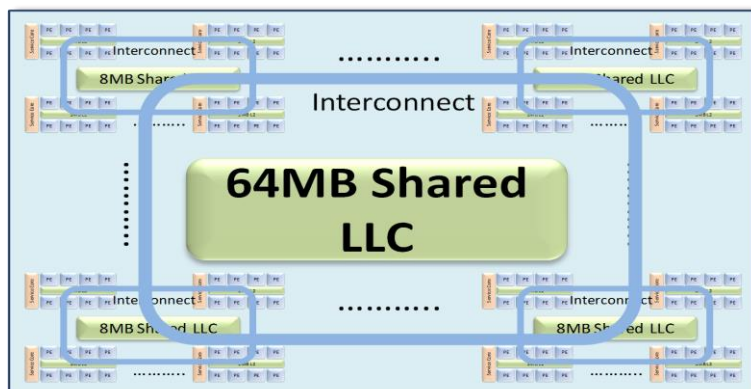


Figure 9: Hierarchical structure of NTV chip (courtesy of S. Burkar)

The main characteristics of NTV chips are reported in Table 3. 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. Indeed considering data movement, most inefficiency come at system level (cabinet and multi cabinet level) as shown on Figure 10, right side.

Technology	7nm, 2018
Die area	500 mm ²
XE/die	2048
Frequency	4.2 GHz@Vdd, 600 MHz@50% Vdd
TFLOPs	17.2 @Vdd, 2.5 @50% Vdd
Power*	600 W@Vdd, 37 W@50% Vdd
E Efficiency*	34 pJ/F@Vdd, 15 pJ/F@50% Vdd
Memory B/F	39 mB/F@Vdd, 268 mB/F@50%Vdd

Table 3: Main characteristics of NTV chips

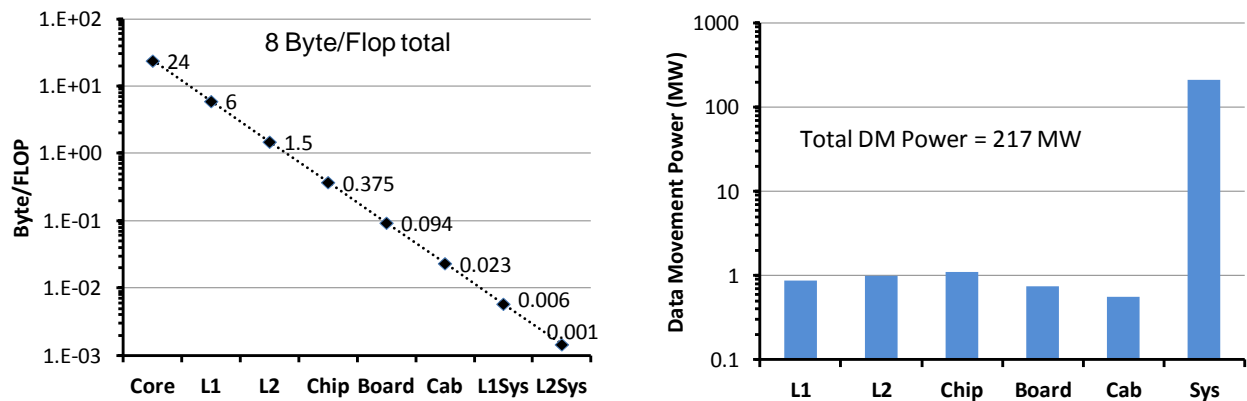


Figure 10: Typical power dissipated by moving data across different layer of the memory hierarchy

To drastically reduce this dominance one can use an “intelligent” tapering approach, which is, on the other hands, inversely proportional to the performances, see Figure 11 right side.

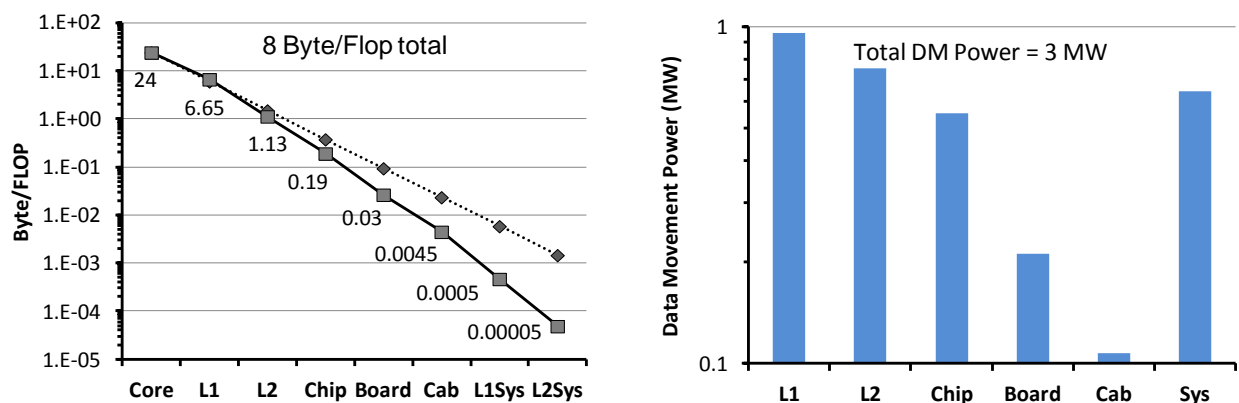


Figure 11: Estimated power dissipated by data movement using “intelligent” tapering techniques to exchange data among the outer layers of the memory hierarchy

The above architectural change requires from the applications point of view to be more and more data local.

NTV chips with respect to the exascale roadmap prediction will force an increase of parallelism by at least a factor four. The challenges imposed on software by NTV technology can be summarized in:

1. Extreme parallelism (1000X due to Exa, additional 4X due to NTV)
2. Data locality—reduce data movement
3. Intelligent scheduling—move thread to data if necessary
4. Fine grain resource management (objective function)
5. Applications and algorithms incorporate paradigm change

Impact on programming and execution models need to be considered as well, and a possible scenario can be the following:

1. **Event driven tasks (EDT)**
 - a. Dataflow inspired, tiny codelets (self contained)
 - b. Non blocking, no preemption
2. **Programming model:**
 - a. Separation of concerns: Domain specification & HW mapping
 - b. Express data locality with hierarchical tiling
 - c. Global, shared, non-coherent address space
 - d. Optimization and auto generation of EDTs (HW specific)

3. **Execution model:**

- a. Dynamic, event-driven scheduling, non-blocking
- b. Dynamic decision to move computation to data
- c. Observation based adaption (self-awareness)
- d. Implemented in the runtime environment

4. **Separation of concerns:**

- a. User application, control, and resource management.

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, by developing a revolutionary architecture, in which the data movement will be more costly than computations. As a consequence, we, as a community, need to prepare to a number of software challenges, summarized as follows:

- a major refactoring and a rethink of algorithms and applications;
- programming models to harness extreme concurrency;
- an introspective, self-aware, execution model;
- and last, but not least, resiliency to provide system reliability.

7.3 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.

In the discussion about packaging with the experts invited in the WG (mainly Bruno Michel from IBM) it has been envisioned that the evolution of HPC architectures are going through three main paradigm changes:

- **Paradigm Change 1: From Cold Air Cooling to Hot Water Energy Re-Use:**
 - Green Datacenter Drivers and Energy Trends
 - Aquasar Zero Emission Datacenters
 - SuperMUC.
 - From Hardware Cost to Total Cost of Ownership
- **Paradigm Change 2: From Performance to Efficiency,**
 - From Maximal Performance per Chip to Performance per Joule
 - Focus on Energy and Exergy
 - Efficiency of Computer vs. Efficiency of Biological Brains
 - Integration of Photonics
- **Paradigm Change 3: From Areal Device Size Scaling to Volumetric Density Scaling**
 - The “Missing” Link between Density and Efficiency
 - Interlayer Cooling and Electrochemical Chip Power Supply
 - Link between Allometric Scaling and Rent’s Rule
 - Towards Five-Dimensional Scaling

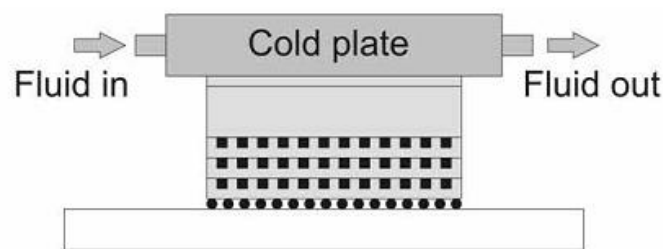
Common to all these changes there is the possibility to design a new packaging concept around “pervasive” water cooling, with the liquid entering directly inside the chip.

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.

Semiconductor technology and packaging since the first computers have shifted their main focus to performance and downsizing (on 2D). This is not sustainable in the long term, and further downsizing

of components is possible only having system-level efficiency as main goal and not pure processor performance. The analogy with biological system (brain) is useful to support that volumetric density and efficiency are strictly correlated. Moreover, high system density can help to mitigate communication of data appear, one of the main bottleneck for supercomputers, 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 (see Figure 12) appear to be the first candidate to be used to start stacking chips. One can start stacking cool components and hot components, or even two hot components if the hot spots are located in different region (see Figure 13).



Microchannel back-side heat removal

Figure 12: *Microchannel based cooling system*

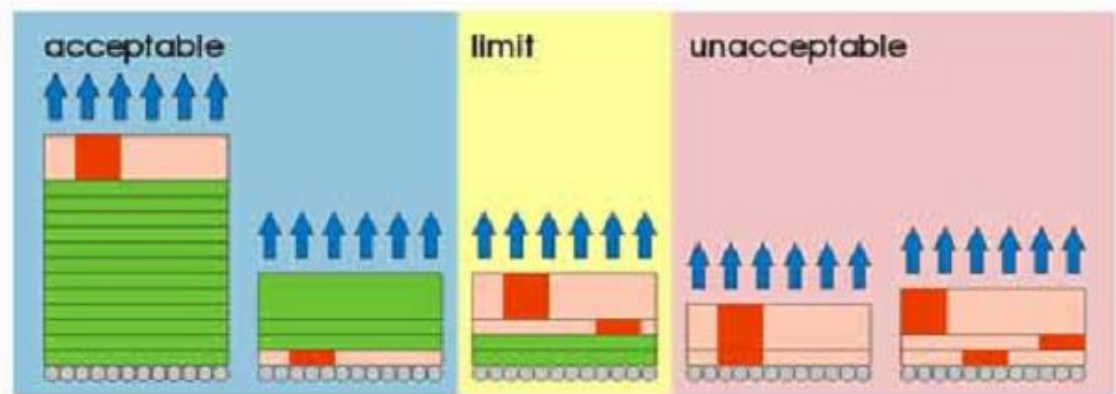


Figure 13: *Different physical layout of 3D stacking with microchannel cooling.*

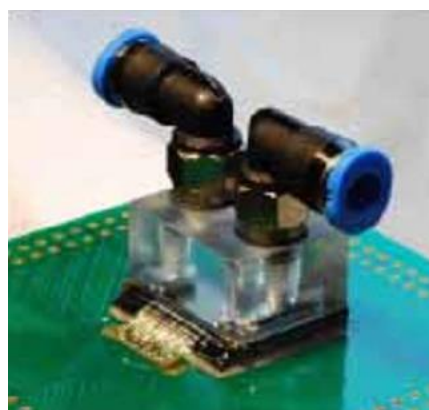


Figure 14: *Lab test sample of a microchannel cooled chip*

The Hybrid Memory Cube (see Figure 14), first attempt to put this concept into practice, is a project sponsored by a number of vendors [38]. The main characteristic of this chip are:

- 4 to 8 memory layers
- Vertical memory cells
- Comparison with DDR3
- 15x performance
- 70% less energy per bit
- 9x smaller form factor

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 fact that size matters so much for computers evolution is evident if we consider that only one millionth of the volume of today's server is occupied by transistors. In practice the vast majority of the volume of a computer today is occupied by power supply, and air cooling. Moreover the large majority of energy is wasted for communications between components, in particular processor and main memory.

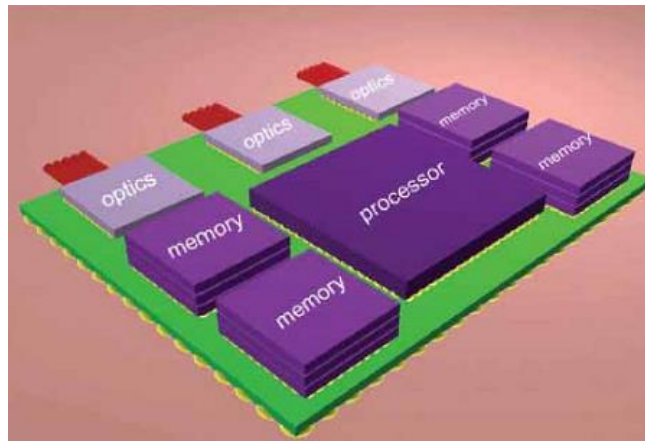


Figure 15: *Single die design with processor, optical interconnect and 3D memory*

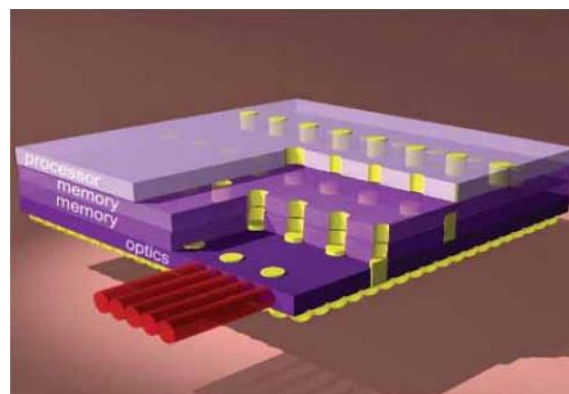


Figure 16: *3D packaging of optical interconnect, memory and processor*

Going 3D and filling up space with transistors displaces the main problem from the supply of energy to the components and the removal of heat. In this respect we can take inspiration from nature. Making a comparison with biology, one discovers that nature solves the problem of power supply and removal of heat by using hierarchical structures. The same is true for other systems like dwelling and cities (hierarchical network of streets, water, etc.), so it is a possible solution for computer infrastructures too. Using the biological analogy a dense volumetric scaling (called allometric scaling) and hierarchical fluid and communication structures are used to build large, efficient, tightly integrated processor clusters with integrated main memory. This is made possible by interlayer cooling of chip stacks and

hierarchical transport systems for coolant fluids. The ultimate density and efficiency 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 (more or less like blood in biology).

Ultimately, working to find solutions to improve density will also implicitly help efficiency, the two characteristics as anticipated above are correlated by natural laws.

All this put together could allow building a peta-scale computer 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), see Figure 15
- 7 years interlayer cooled chip stacks (at or post exascale), see **Erreur ! Source du renvoi introuvable.**
- 10+ years for electrochemical power supply (post exascale)

7.4 Data transfer

A microprocessor needs to move data to be processed between different logical units and from the inside and outside. In today microprocessors moving data is obtained via the application of an electrical potential bias to a semiconductor in a conducting state, that means perturbing electrons, and in the end (due to inelastic scattering) producing heat and wasting energy. In fact the energy required to move data around below a certain scale of integration is higher than the energy required to perform computations on the data itself. A critical factor then is to find new way 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.

During the discussion with the experts of the WG, there was a general consensus about an enormous expansion in the quantity of data being produced, transmitted at all level, and all converging into datacenter, data growing faster than the Moore's Law. Patrick Demichel suggests that the focus has to be to project and design datacenter infrastructure to support the data movement and capacity, then his opinion is that we need a data-centric high performance computing, and new disruptive technology to support it. This ultimately will induce a paradigm shift, from computational to data exploration knowledge based research.

Since it is clear, in the above perspective, that data movement is crucial, it is of fundamental importance to reduce the energy used by transmission, increase bandwidth and reduce latency. In this respect Photons are the natural candidates to substitute Electrons to do this job even for short distance (inside chip).

In principle one can 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.

First of all good laser source with low power consumption need to be developed in order to feed the wave guides that carry data –hence the photons- around, much like we need a power supply for the

electrons. Then (ring) modulator and resonator to code and decode information into the light being transmitted have to be integrated into silicon chip. Finally thermal stability and ring tuning issues have to be solved (see Figure 17).

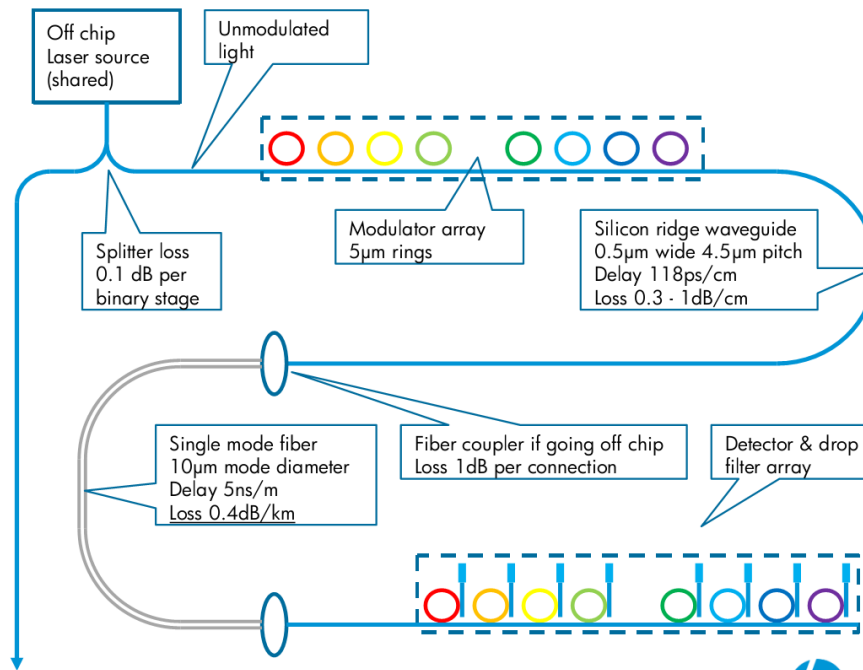


Figure 17: Draft schema of photonic data transport circuit

To have all in place to be implemented in an end-user (exascale, or post exascale) system it will require other 10 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 (see Figure 18).

System-level architecture to large-scale integration

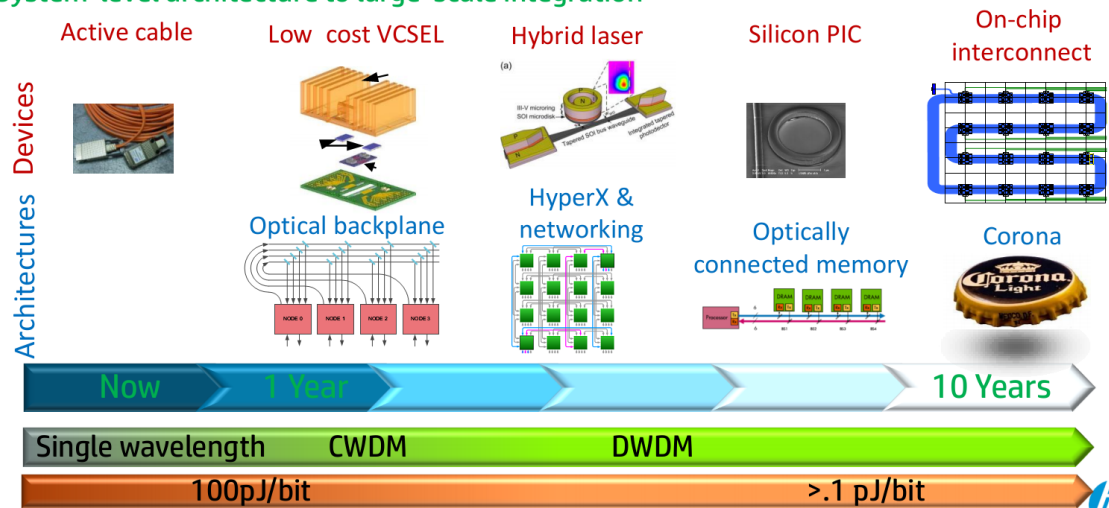


Figure 18: Estimated roadmap of photonic technology

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. Patrick presents a number of simulations to evaluate the efficiency and the performance of different implementation of optical switches on different workload (see Figure 19).

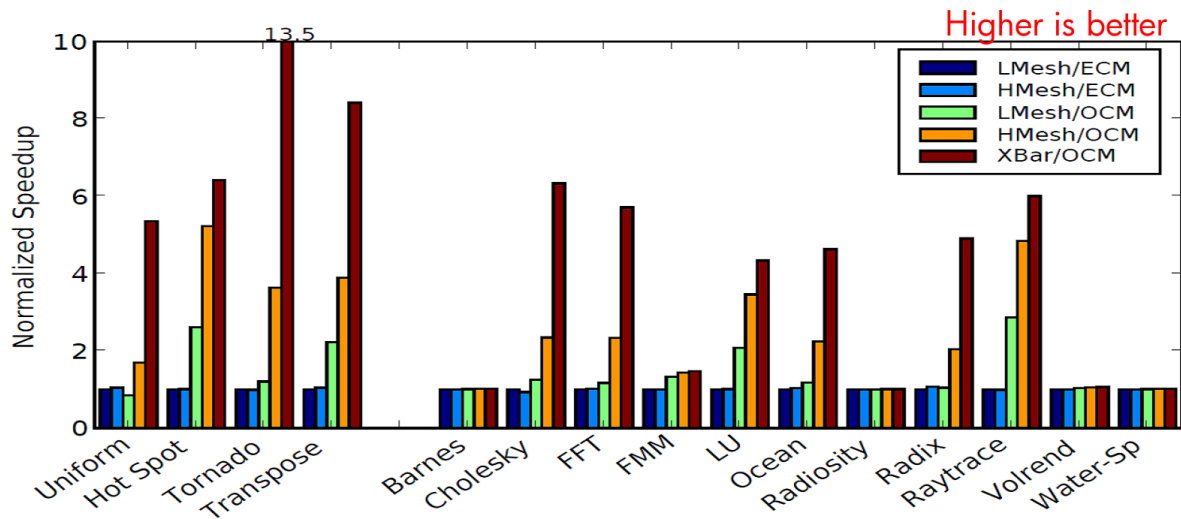


Figure 19: Simulated performances of different numerical kernels, for different switch infrastructure

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 (see Figure 20) and will allow the setup of an exascale system with a switch-based interconnect.

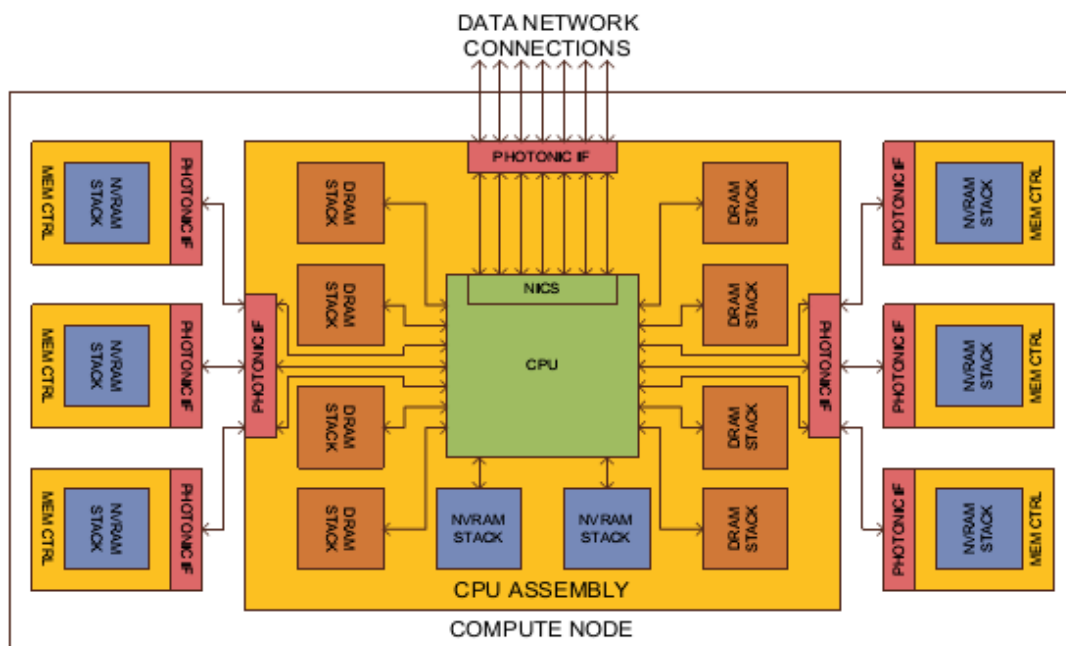


Figure 20: Optical interconnection internal to the node of an exascale system

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 or fat tree. 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.

7.5 Memory

Main system memory in today computers is implemented with DRAM, where keeping data alive requires being powered regardless of the fact that data are changed or not. This is not an ideal condition, since the system is wasting energy even when no change of state is performed. 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, as discussed about the packaging, 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 cheap (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-phonic technology as discussed in the data transfer section (see Figure 21).

Year	Peak Performance	number of optical channels	Optics Power Consumption	Optics Cost
2008	1PF	48,000 (@ 5Gb/s)	50mW/Gb/s (50pJ/bit)	\$10,000 per Tb/s
2012	10PF	2×10^6 (@ 10Gb/s)	25mW/Gb/s	\$1,100 per Tb/s
2016	100PF	4×10^7 (@ 14-25 Gb/s)	5mW/Gb/s	\$170 per Tb/s
2020	1000PF (1EF)	8×10^8 (@ 25 Gb/s)	1mW/Gb/s	\$25 per Tb/s

Figure 21: Main characteristics of optical interconnect (projected for 2016 and 2020)

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*, which are not new (memristor have been envisaged and studied, theoretically, already back in the '70s) but till now they have not been successfully productised and manufactured. On the other hands memristor based memory device seems to have very good characteristics:

- High density $4F^2$, stackable
- Low cost
- Low power : pJ/bit
- High speed : ns
- High endurance : $>> 10^{10}$
- High retention : 10+ years
- Reconfigurable architectures
- Multiple optimized variants
- Long term roadmap : post Moore
- Can do logic « better transistors »
- Can do neuristor

Moreover they allow to positively address issues related to power, performance, architecture flexibility, fault tolerance, programmability, capacity and cost. They then seem essential for exascale systems, and memristor technology trends are such that:

- Scaling down to less than 10 nm width per cell
 - ~ 32 Gbyte/cm²/layer by 2018

- Scaling up to multiple (≥ 8) layers on chip
 - ~ 0.25 Tbyte/cm²/chip by 2018
- Truly nonvolatile – many, many years
- Random Access
- Fast cell write and erase (\sim nanosec)
- Low energy cell write and erase (\sim picoJ)
- Good to excellent endurance ($> 10^{10}$ cycles)
 - Still counting – goal is to exceed 10^{18} cycles

These characteristics make the memristor based memory chips 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 Backcomb, main objective of the project are:

- new distributed computer architectures that address exascale resilience, energy, and performance requirements
- replace mechanical-disk-based data-stores with energy-efficient non-volatile memories
- explore opportunities for NVM memory, from plug-compatible replacement (like the NV DIMM)
- radical, new data -centric compute hierarchy (nanostores)
- place low power compute cores close to the data store
- reduce number of levels in the memory hierarchy
- Adapt existing software systems to exploit this new capabilities.

7.6 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). The main aspects related to networking, in terms of technology and architecture, as well as in terms of software and application impacts, will be better addressed in the next period of the WG5.5 activity.

7.7 I/O subsystem

I/O subsystem of high performance computers are still deployed using spinning disks, with their mechanical limitation (spinning speed cannot grow above a certain regime, above which the vibration cannot be controlled), and like for the DRAM they consume energy even if their state is not changed. Solid state technology appears to be a possible alternative, but 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.

During the discussion among the expert within the WG, Malcolm contributed to the discussion with his personal viewpoint: IO System Challenges, Solid state technologies, Disk as an Archive, Middleware for Exascale and High bandwidth optical interfaces.

The challenges for the I/O subsystem can be well illustrated starting from few fundamental use cases: dump and read the content of the nodes memory. Considering the main characteristics of a system in the exascale era, roughly they will be:

- 10^8 cores – each ~ 10 GF/sec, each ~ 1 G RAM
- 1,000 cores / node, 0.5 TB RAM / node (10 TF / node)
- 100K cluster nodes, 50 PB RAM / cluster
- Mem BW 5 TB/sec - System network BW 200 GB/sec

- I/O: 300 TB / sec, one node 3 GB / sec
- File system > 1 EB
- I/O nodes likely not more than 1,000 – 300 GB/sec

And some technology revolutions are around the corner:

- File system clients will have ~10,000 cores
- Architectures will be heterogeneous
- Flash and/or PCM storage leads to tiered storage
- Anti revolution – disks will only be a bit faster than today

Now, if we consider the fundamental use case: checkpoint/restart, even if we can imagine to be able to write data (maybe asynchronously), a restart imply application jump starting reading large files from all nodes. This is killing if nothing happen on the architecture of I/O subsystem. In fact today in system with roughly 10 PF we have:

- handled by large storage systems – 1TB/sec
- several billions of files

At 100 PF we can think to manage this use case with the following infrastructure:

- Flash cache approach – 10 TB/sec
- Flash takes the bursts / Disks more continuously used
- Takes ~ 20,000 disks (0.5MW / lots of heat / lots of failed drives)
- Probably a metadata server becomes a scalability limit

But at 1EF we have the gap and the paradigm appears to break: 100K drives 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 striped 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.

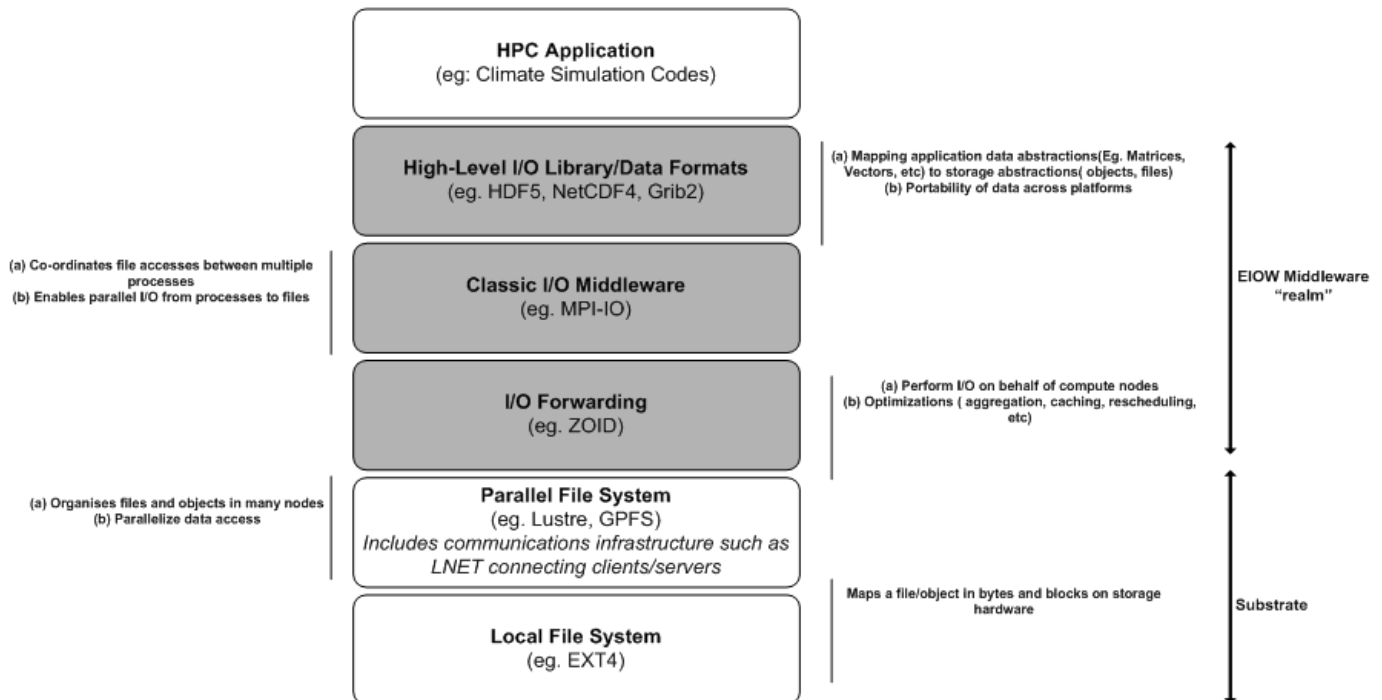


Figure 22: I/O "middleware" stack for a typical HPC application

Analyzing a typical application stack one can observe that each layer has its own semantic and may re-implement the same optimization strategy. These layer specific approaches drastically degrade performance and prevent scalability.

An interesting project that tries to go beyond this approach is represented by Exascale I/O Workgroup Middleware ("EIOW"), see Figure 22, based on the following principles:

- Let HPC application experts explain requirements for next generation storage
- Architect, design, implement an open source set of exa-scale I/O middleware
- Approach
 - Gather requirements: Europe 02/12, US 04/12, Japan 05/12
 - Design the architecture – breakout groups Barcelona 09/12
 - Analyze architecture for completeness
 - Begin implementation

Already 40 organizations around the world are participating to this initiative. It is an Open effort (<http://www.eiow.org/>) and will move in the direction of IETF (<http://www.ietf.org/>) style controlled openness. EIOW aims to be a ubiquitous middleware:

- An agreed, eventually standardized API for applications / management is targeted to be uniform
- We hope to be an implementation of choice for researchers to study, amend, influence and change
- Such research projects have started to spring up
- A storage access API allowing storage vendors to bolt it onto their favorite data object and metadata stores.

The requirement gathering phase as produced the following list of wishful characteristics the EIOW middleware should provide.

EIOW should provide guided mechanisms for tuning performance and parallelism. This can be implemented with "indicators" associated to different I/O operations to be used at application level. Examples of useful indicator to be implemented and associated with data to be stored are life cycle indicators so to avoid movement, and keep data in cache or memory, suitable stripe width indicators

and levels of integrity indicators. All this will assist the storage system with read-ahead decisions and indicate when transactions need to become persistent.

EIOW should implement a new I/O library with collective I/O provisions, support for high performance/scalable I/O features (Zero Copy I/O, 2 Phase I/O), support for various data layout abstractions (RAID Layouts, De-duplicated Layouts, Compressed layouts, Encrypted layouts, Memory only layouts).

EIOW should provide distributed containers for different type of data and I/O functionalities. Containers holds data for a family of related applications (HDF5 containers, POSIX containers, NetCDF containers), providing a isolated abstractions from the rest of the system. Moreover they can be check-pointed and taken offline independently, migrated separately from other containers. Container “Merge” and “Split” rules can be separately defined.

EIOW should provide system diagnostics, analytics and simulation functionalities (see Figure 23). This implies the availability of “Always On” telemetry data (Very structured unlike Lustre logs). Then telemetry data can be used to “Anomaly Detect”, and provide “Root Cause Analysis”. Simulation functionalities should include: “Operation Log Edition” to provide “What if” scenarios in workloads, simulation drivers for providing a framework to model storage devices/hardware, core simulation engine based on open frameworks. Then machine learning functionality should be implemented to provide adaptive inputs to I/O subsystem based on Telemetry data and Simulation outputs.

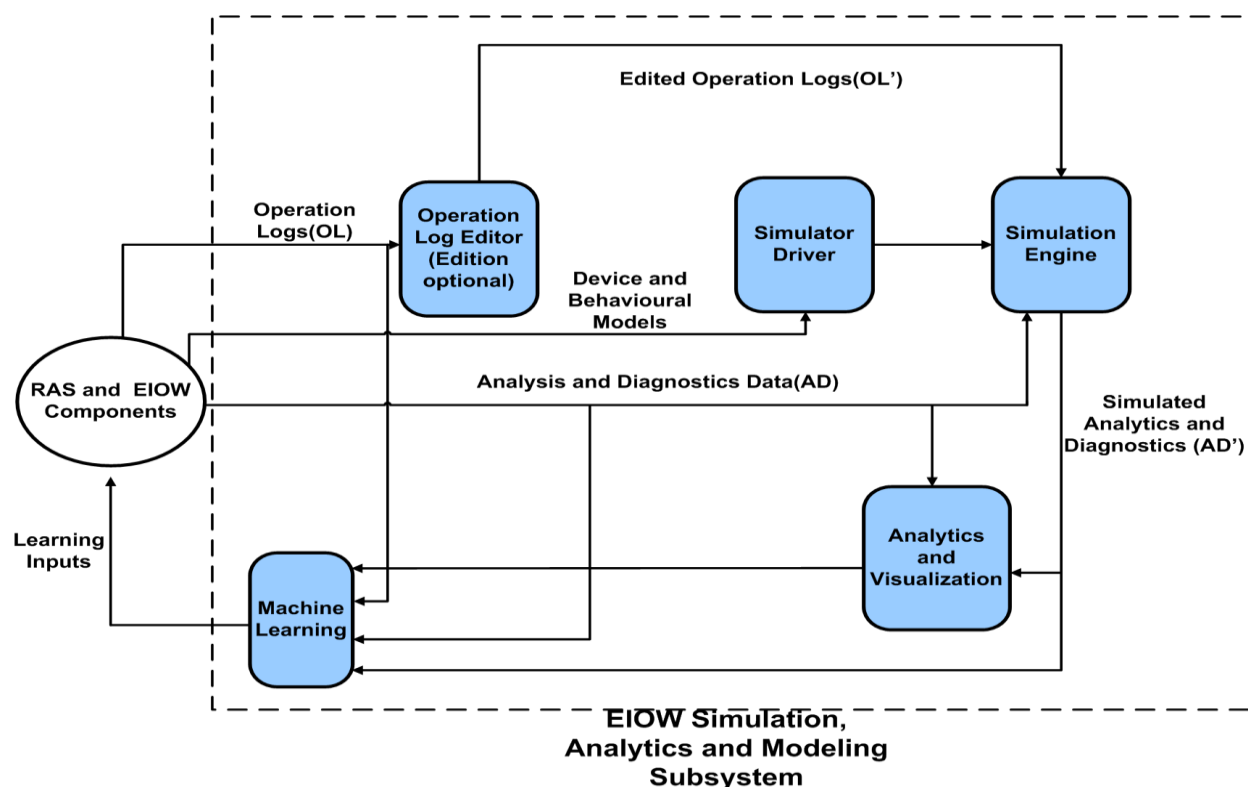


Figure 23: System Diagnostics/Analytics/Simulation , big picture architecture

EIOW middleware architecture, driven by the necessity to mitigate the exascale I/O gap, once completed will be disruptive in the way application developer and user consider the problems related to the I/O. Much like what happens with profiler assisted code optimization, one can profile and evaluate different I/O strategy and tune its own I/O needs to the low end infrastructure (see Figure 24).

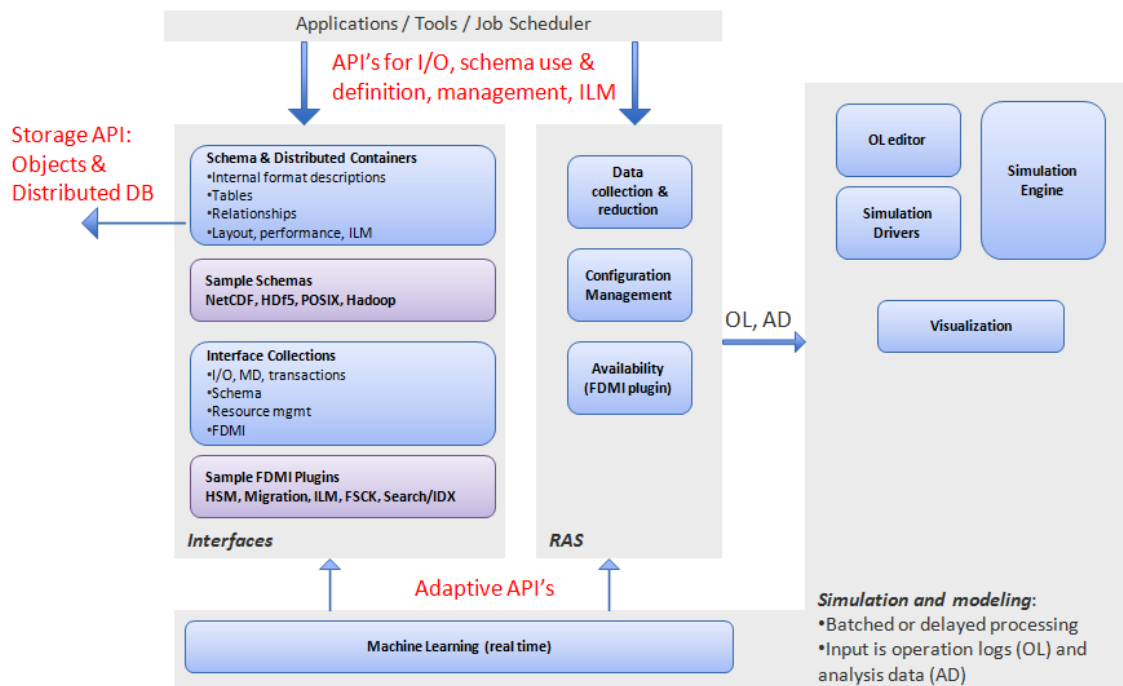


Figure 24: ELOW High Level Architecture Summary

Internally to the WG was discussed the role that optical interconnects will play in the architecture of the I/O subsystem.

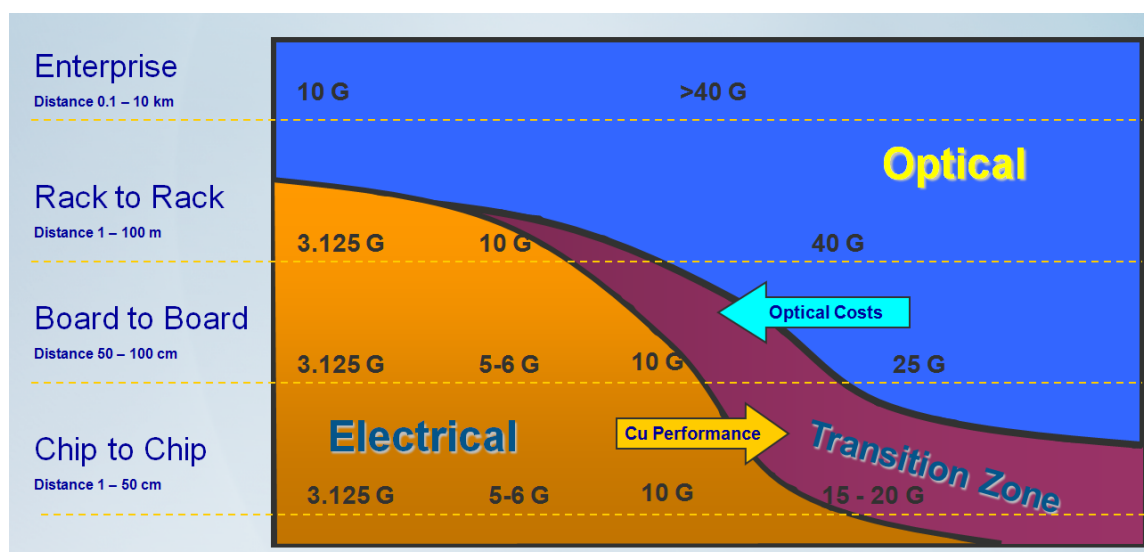


Figure 25: Performance, distance and cost phase diagram for different physical transport media (Electrical vs Optical)

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 scale. Limit of copper are: Crosstalk, Reflections, Electro-magnetic interference, Dielectric Loss / "Skin effect", Signal skew. Then also for the I/O subsystem optical connections 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 (see Figure 25).

7.8 Cooling

It is clear that the computers and 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.

During the WG workshop Giampietro Tecchioli shared his vision that liquid cooling will no more be an option but a must for HPC infrastructures, 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 come 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 an exascale system within the power and cost constraints.

7.9 Next Steps

During the discussion it appeared that the two main concerns about the building of an exascale system are I/O and in general data movement on one hand and efficiency of the architecture on the other hand. Disruptive technologies appearing in these two fields may allow dramatic redesign in system architecture (for example with non volatile memory or optical interconnect to substitute PCB) and in new application paradigms.

In the second year of activities of the WP5.5 it is planned to update all components of an HPC infrastructure with a focus on I/O, data and efficiency in order to express the requirements that this will entail on the software (OS, software stack, libraries and applications).

8. Bibliography

- [1] "Scientific Discovery at the Exascale", Report from the DOE ASCR 2011 "Workshop on Exascale Data Management, Analysis, and Visualization", February 2011.
- [2] "Open Problems in Network-aware Data Management in Exa-scale Computing and Terabit Networking Era", M. Balman and S. Byna, NDM'11, November 14, 2011, USA.
- [3] Scientific Data Management and I/O – Challenges for Exascale, Alok Choudhary, International Workshop on Peta-Scale Computing WPSE 2010, Feb 18, 2010, Kyoto.
- [4] "Strategic Research Agenda", European Technology Platform for High Performance Computing, April 2013
- [5] "Deliverable D5.6, Final report on roadmap and recommendations development", European Exascale Software Initiative, January 2012.
- [6] "Building a Large Scale Climate Data System in Support of HPC Environment", Feiyi Wang et al., published in IEEE 7th International Conference on Next Generation Web Service Practices, October 19-21, 2011. Salamanca, Spain.
- [7] Visualization and Data Analysis at the Exascale, James Ahrens et al, A White Paper for the National Nuclear Security Administration (NNSA) Accelerated Strategic Computing (ASC) Exascale Environment Planning Process, <https://asc.llnl.gov/exascale/exascale-vdaWG.pdf>
- [8] Advanced Scientific Computing Research (ASCR) <http://science.energy.gov/ascr/news-and-resources/program-documents/>
- [9] HPSS in the Extreme Scale Era, report to DOE Office on HPSS in 2008-2022, July 15, 2009, <http://www.nersc.gov/assets/HPSC-Requirements-for-Science/HPSSExtremeScaleFINALpublic.pdf>
- [10] Challenges facing HPC and the associated R&D priorities: a roadmap for HPC research in Europe, PlanetHPC, 2013.
- [11] Data-intensive research The RCUK Data Policy, Mark Thorley, Franco-British bilateral Workshop Big Data in Science, 6th and 7th Nov 2012.
- [12] Synergistic Challenges in Data-Intensive Science and Exascale Computing, DOE ASCAC Data Subcommittee Report March 2013.
- [13] EUDAT, Towards a pan-European Collaborative Data Infrastructure, Damien Lecarpentier, CSC-IT Center for Science, Finland E-Infrastructure workshop, Kajaani, 26 April 2013
- [14] Alex Szalay, Turning Large Simulations into Numerical Laboratories, BDEC 2013 (<http://www.exascale.org/bdec/>)
- [15] EIOW – exa-scale IO workgroup, Status Update Q3, 2012.
- [16] David A. Bader Opportunities and Challenges in Massive Data-Intensive Computing, <http://lyra.berkeley.edu/CDIConf/pdfs/DavidBader.pdf>
- [17] EUDAT (<http://www.eudat.eu>): pan-European Collaborative Data Infrastructure
- [18] PRACE (<http://www.prace-project.eu/>): Partnership for Advanced Computing in Europe
- [19] SCIDB (<http://www.scidb.org/>) open source software for managing scientific data
- [20] EIOW (<http://www.eiow.org>)
- [21] Fast Forward (<https://asc.llnl.gov/fastforward/>)
- [22] Schweber, S., Wachter, M. Complex Systems, Modelling and Simulation. Stud. Hist. Phil. Mod. Phys. 31, 583-609 (2000)

- [23] Heymann, M. Understanding and misunderstanding computer simulation: The case of atmospheric and climate science - An introduction. *Stud. Hist. Phil. Mod. Phys.* 41, 193-200 (2010)
- [24] A. Dienstfrey, R.F. Boisvert, eds., *Uncertainty Quantification in Scientific Computing*, IFIP Advances in Information and Communication, Vol. 377, Springer, 2012.
- [25] E. de Rocquigny, N. Devictor, S. Tarantola, eds., *Uncertainty in Industrial Practice: A Guide to Quantitative Uncertainty Management*, Wiley, 2008.
- [26] O.P. Lemaître, O.M. Knio, *Spectral Methods for Uncertainty Quantification*, Springer, 2010
- [27] A. Saltelli, K. Chan, E. M. Scott, *Sensitivity Analysis*, Wiley, 2000.
- [28] E. Plischke. An effective algorithm for computing global sensitivity indices (EASI). *Reliability Engineering&System Safety*, 95(4):354{360, 2010.
- [29] S. Tarantola, D. Gatelli, and T. A. Mara. Random balance designs for the estimation of First order global sensitivity indices. *Reliability Engineering&System Safety*, 91:717{727, 2006
- [30] P. M. Kogge and et al, *ExaScale Computing Study: Technology Challenges in Achieving Exascale Systems*, DARPA Information Processing Techniques Office, Washington, DC, pp. 278, September 28, 2008.
- [31] Dongarra, J., Beckman, P. et al. The International Exascale Software Roadmap. *The International Journal of High Performance Computer Applications*, Volume 25, Number 1, 2011, ISSN 1094-3420.
- [32] J.Y Berthou et al. Deliverable D5.6 Final report on roadmap and recommendations development, Dec. 2011. <http://www.eesi-project.eu/pages/menu/eesi-1/publications/working-group-reports.php>.
- [33] Michael K. Patterson et al. "TUE, a New Energy-Efficiency Metric Applied at ORNL's Jaguar", *Proceedings 28th International Supercomputing Conference, ISC 2013, Leipzig, Germany, June 16-20, 2013*.
- [34] C. Hsu and W. Feng. A power-aware run-time system for high-performance computing. In *Proceedings of the 2005 ACM/IEEE conference on Supercomputing*. IEEE Computer Society Washington, DC, USA, 2005.
- [35] IESP roadmap, www.exascale.org/mediawiki/images/2/20/IESP-roadmap.pdf
- [36] EESI-1 D5.4 report www.eesi-project.eu/modules/download_pictures/dlc.php?file=99&id=1349445649&sid=17
- [37] US DoE, Fault Management Workshop, Final Report August 13, 2012.
- [38] Hybrid Memory Cube Consortium <http://www.hybridmemorycube.org/>

9. Annexes

9.1 List of experts in WGs

EESI 2 - WG 5.1 Data Management and Exploration				
Name	Organisation	e-mail	Country	Area of Expertise
Francois Bodin (chair)	CAPS Enterprise	francois.bodin@caps-entreprise.com	France	Data Management, GPU
Giovanni Erbacci (vice chair)	CINECA	g.erbacci@ceneqa.it	Italy	HPC infrastructures
Jean-Michel Alimi,	Observatoire de Paris	jean-michel.alimi@obspm.fr	France	Big Data, Numerical Cosmology
Gabriel Antoniu	INRIA	Gabriel.Antoniou@inria.fr	France	Cloud Service, Storage systems: BlobSeer
Georges Hebrail	EDF	hebrail.georges@enst.fr	France	Data Mining
Jacques-Charles Lafoucrière	CEA	jacques-charles.lafoucriere@cea.fr	France	File system: Lustre
Malcolm Muggeridge	Xyratex	malcolm_muggeridge@xyratex.com	UK	Cloud Storage
Kenji Ono	Riken	keno@riken.jp	Japan	Data reduction for large scale datasets
Stéphane Requena	GENCI	stephane.requena@genci.fr	France	Big Data
Alex Szalay	Johns Hopkins University	szalay@jhu.edu	USA	Large scalable Databases, Numerical modeling of Galaxy.
Jean-Pierre Vilotte	IPG	vilotte@ipgp.frlto	France	Computational Physics, Seismology

EESI 2 - WG 5.2 Uncertainties (UQ/V&V)				
Name	Organisation	e-mail	Country	Area of Expertise
Vincent Bergeaud (chair)	CEA	vincent.bergeaud@cea.fr	France	Uncertainty quantification, Uranie
Alberto Pasanisi (vice chair)	EDF	alberto.pasanisi@edf.fr	Italy	Uncertain Analysis, Bayesian Decision Theory, Numerical Simulations
Stefano Tarantola	JRC-ISPRA	stefano.tarantola@jrc.it	Italy	Statistic, EU policies.
Christophe Prud'homme	University of Strasbourg	cprudhomme@math.unistra.fr	France	Applied Mathematics, Computer sciences,
Olivier Le Maître	LIMSI, Duke University	olm@limsi.fr	USA	Uncertain Propagation, Chaos Theory, CFD, stochastic nonlinear problems
Renaud Barate	EDF R&D	renaud.barate@ensta.fr	France	Automatic Design, Uncertainty analysis tools
Bertrand Looss	EDF R&D	biooss@yahoo.fr	France	Monte Carlo methods, Environmental modeling, Geostatistics
Fabrice Gaudier	CEA	fabrice.gaudier@cea.fr	France	Data Mining, Modeling of Uncertainty, Nuclear Energy, URANIE Framework

EESI 2 - WG 5.3 Power & Performance				
Name	Organisation	e-mail	Country	Area of Expertise
Simon McIntosh-Smith (chair)	Bristol University	simonm@compsci.bristol.ac.uk	UK	Microelectronics, HPC, MIC, Machine Learning
Thomas Ludwig (vice chair)	DKRZ	ludwig@dkrz.de	Germany	High volume data storage, Energy efficiency, Performance analysis, Parallel computing, Meteo climate
Alex Ramirez	BSC	alex.ramirez@bsc.es	Spain	Energy efficient hardware, , Computer architectures, ARM based HPC
Matthias Müller	RWTH, Aachen University	mueller@informatik.rwth-aachen.de	Germany	Energy efficient HPC, Scientific Computing, Performance tools
Jean-Marc Pierson	Paul Sabatier University	pierson@irit.fr	France	Energy aware HPC
Laurent Lefevre	Lyon University, INRIA	laurent.lefevre@ens-lyon.fr laurent.lefevre@inria.fr	France	Energy efficient Computing and Networking
James Perry	EPCC, University of Edinburgh	j.perry@epcc.ed.ac.uk	UK	Accelerator technologies, Embedded systems, Code optimisation

EESI 2 - WG 5.4 Resilience				
Name	Organisation	e-mail	Country	Area of Expertise
Franck Cappello (chair)	INRIA, Argonne Natl. Lab.	fci@lri.fr	France,	Resilience, Application workloads, Extreme-scale computers
Luc Giraud	INRIA	luc.giraud@inria.fr	France	Parallel Algorithms, HPC simulations, Petaflop scalability
Torsten Hoefer	ETH Zurich	htor@inf.ethz.ch	Switzerland	Optimization of Parallel algorithms, Performance Modelling and Tuning, Large Scale Parallel Architectures, Resilience
Simon McIntosh Smith	Bristol University	simonm@cs.bris.ac.uk	UK	Microelectronics, HPC, MIC, Machine Learning, Parallel Architectures
Christine Morin	INRIA	christine.morin@inria.fr	France	Distributed operating systems, Fault tolerance, Autonomic Computing,
Bogdan Nicolae	IBM Research Dublin	bogdan.nicolae@ie.ibm.com	Ireland	Scalable Storage Techniques, Exascale Architectures, Data Resilience
Pascale Rossé-Lauren	BULL	PASCALE.ROSSE-LAURENT@bull.net	France	Applications, Compilers, Programming Environments
Osman Unsal	BSC	osman.unsal@bsc.es	Spain	Runtime Environments, HPC, Transactional Memory, Fault Tolerance
George Bosilca	UTK	bosilca@cs.utk.edu	USA	HPC, Accelerators, Common Communication Infrastructure, HARNESS

EESI 2 - WG 5.5 Disruptive Technologies				
Name	Organisation	e-mail	Country	Area of Expertise
Carlo Cavazzoni (chair)	CINECA	c.cavazzoni@ Cineca.it	Italy	HPC, MIC, GPU, Numerical Simulations, Energy efficient computing
Marie-Christine Sawley (vice chair)	Intel	marie-christine.sawley@intel.com	Switzerland	HPC, Massive parallelism, Energy efficient computing
Shekhar Borkar	Intel	Shekhar.borkar@intel.com	USA	Semiconductor Technology: near threshold voltage
Bruno Michel	IBM	bmi@zurich.ibm.com	Switzerland	Packaging: microfluidics
Patrick Demichel	HP	patrick.demichel@hp.com	France	Memory: memristor, 3D-stack memory, NVRAM. Data Transfer: Silicon Photonics
Piero Vicini	INFN	piero.vicini@roma1.infn.it	Italy	Network Technology: Adapter free device
Giampiero Tecchiolli	Eurotech	giampietro.tecchiolli@eurotech.com	Italy	Cooling & Engineering: High density high efficiency solutions
Malcolm Muggeridge	Xyratech	malcolm_muggeridge@xyratex.com	UK	I/O sub system