SEVENTH FRAMEWORK PROGRAMME
Research Infrastructures

INFRA-2010-2.3.1 – First Implementation Phase of the European High Performance Computing (HPC) service PRACE

PRACE-1IP

PRACE First Implementation Project

Grant Agreement Number: RI-261557

D9.1.2
Exascale Technology Assessment Report

Final

Version:	1.0
Author(s):	Jonathan Follows, STFC Daresbury Laboratory
Date:	31/05/2012

Project and Deliverable Information Sheet

PRACE Project	**Project Ref. №: RI-261557**	
	Project Title: PRACE First Implementation Project	
	Project Web Site: http://www.prace-project.eu	
	Deliverable ID: D9.1.2	
	Deliverable Nature: Report	
	Deliverable Level: PU	**Contractual Date of Delivery:** 31/05/2012
		Actual Date of Delivery: 31/05/2012
	EC Project Officer: Thomas Reibe	

* - The dissemination level are indicated as follows: **PU** – Public, **PP** – Restricted to other 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).

Document Control Sheet

Document	**Title:** Exascale Technology Assessment Report	
	ID: D9.1.2	
	Version: 1.0	**Status:** Draft
	Available at: http://www.prace-project.eu	
	Software Tool: Microsoft Word 2003	
	File(s): D9.1.2.docx	
Authorship	**Written by:**	Jonathan Follows, STFC Daresbury Laboratory Torsten Wilde, LRZ Jeff Poznanovic, CSCS Carlo Cavazzoni, Cineca Gilles Civario, ICHEC Eric Boyer, CINES
	Contributors:	Simon McIntosh-Smith, Bristol University Rich Vuduc, Georgia Tech Iris Christadler, LRZ Jeff Vetter, ORNL Aad van der Steen, NWO Giampietro Techiolli, Eurotech Jim Cownie, Intel Franck Cappello, INRIA
	Reviewed by:	Dietmar Erwin, FZJ; Aad van der Steen, NWO
	Approved by:	MB/TB

Document Status Sheet

Version	Date	Status	Comments
0.1	11/May/2012	Draft	
0.2		Draft	
0.5		Draft	
0.8	24/May/2012		
1.0	24/May/2012	Final version	

Document Keywords

Keywords:	PRACE, HPC, Research Infrastructure

Disclaimer

This deliverable has been prepared by the responsible Work Package of the Project in accordance with the Consortium Agreement and the Grant Agreement n° RI-261557 . It solely reflects the opinion of the parties to such agreements on a collective basis in the context of the Project and to the extent foreseen in such agreements. Please note that even though all participants to the Project are members of PRACE AISBL, this deliverable has not been approved by the Council of PRACE AISBL and therefore does not emanate from it nor should it be considered to reflect PRACE AISBL's individual opinion.

Copyright notices

Table of Contents

Project and Deliverable Information Sheet i

Document Control Sheet i

Document Status Sheet ii

Document Keywords ii

Table of Contents iii

List of Figures iii

References and Applicable Documents iii

List of Acronyms and Abbreviations iv

Executive Summary 1

1 Introduction 2

2 Major hardware trends affecting Exascale developments and their potential impact on software 3

3 Application and hardware co-design 4

4 Emerging technologies in the path to Exascale 7

5 An update on memory technologies 9

6 Evolution and perspective of topology-based interconnect 10

7 Overcoming the barriers to Exascale through innovation 12

8 From fault tolerance to resilience 14

Conclusion 17

List of Figures

Figure 1: Power and Area tradeoffs 4

Figure 2: Matrix multiplication machine versus 3D FFT machine in comparison to Nvidia Echelon 5

Figure 3: Matrix multiplication and 3D FFT performance in relationship to number of nodes ... 6

References and Applicable Documents

[1] http://www.prace-project.eu

[2] Daresbury Workshop Presentations: http://www.stfc.ac.uk/CSE/sas/disco/39060.aspx

[3] Warren Buffet, according to http://c2.com/cgi/wiki?QuotesOnTheoryVsPractice

[4] Blackcomb Project. Webpage: http://ft.ornl.gov/trac/blackcomb

[5] Dong Li, Jeffrey Vetter, et al. "Identifying Opportunities for Byte-Addressable Non-Volatile Memory in Extreme-Scale Scientific Applications". In *Proceedings of the International Parallel and Distributed Processing Symposium (IPDPS)*, 2012.

[6] Vancouver Project. Webpage: http://ft.ornl.gov/trac/vancouver

[7] K. Spafford, et al. The Tradeoffs of Fused Memory Hierarchies in Heterogeneous Architectures, The ACM International Conference on Computing Frontiers (CF'12), 2012. (to appear)

[8] Kyle Spafford, et al. Maestro: Data Orchestration and Tuning for OpenCL Devices. Euro-Par (2) 2010: 275-286.

[9] A. Danalis, el al. The Scalable Heterogeneous Computing (SHOC) Benchmark Suite. *Proceedings of the Third Workshop on General-Purpose Computation on Graphics Processors (GPGPU 2010)*, March 2010.

[10] TAU Performance Analysis. Web: http://www.cs.uoregon.edu/Research/tau

[11] N. Farooqui, et al. A Framework for Dynamically Instrumenting GPU Compute Applications within GPU Ocelot, Proceedings of Fourth Workshop on General-Purpose Computation on Graphics Processing Units, March 2011.

[12] HyVM project. Webpage: http://www.cercs.gatech.edu/projects/HyVM

[13] Technical Report of the INRIA-Illinois Joint Laboratory on PetaScale Computing TR-JLPC-09-01 “Toward Exascale Resilience”: http://jointlab.ncsa.illinois.edu/pubs/Toward_Exascale_Resilience.pdf

[14] International Exascale Software Project Roadmap 1.1: http://www.exascale.org/mediawiki/images/a/a8/IESP-roadmap-1.1.pdf

List of Acronyms and Abbreviations

<Below is an extensive the List of Acronyms used in previous deliverables. Please add additional ones specific to this deliverable and delete unrelated ones. >

AMFT	Advanced Multilevel Fault Tolerance
AMD	Advanced Micro Devices
APGAS	Asynchronous PGAS (language)
API	Application Programming Interface
ASCR	Office of Advanced Scientific Computing Research
ASIC	Application-Specific Integrated Circuit
BSC	Barcelona Supercomputing Center (Spain)
CAF	Co-Array Fortran
ccNUMA	cache coherent NUMA
CEA	Commissariat à l’Energie Atomique (represented in PRACE by GENCI, France)
CINECA	Consorzio Interuniversitario, the largest Italian computing centre (Italy)
CINES	Centre Informatique National de l’Enseignement Supérieur (represented in PRACE by GENCI, France)
CPU	Central Processing Unit
CSC	Finnish IT Centre for Science (Finland)
CSCS	The Swiss National Supercomputing Centre (represented in PRACE by ETHZ, Switzerland)
CUDA	Compute Unified Device Architecture (NVIDIA)
DARPA	Defense Advanced Research Projects Agency
DDR	Double Data Rate
DIMM	Dual Inline Memory Module
DOE	Department of Energy
DP	Double Precision, usually 64-bit floating point numbers
DRAM	Dynamic Random Access Memory
EC	European Community
EESI	European Exascale Software Initiative

EPCC	Edinburgh Parallel Computing Centre (represented in PRACE by EPSRC, United Kingdom)
EPSRC	The Engineering and Physical Sciences Research Council (United Kingdom)
ETHZ	Eidgenössische Technische Hochschule Zuerich, ETH Zurich (Switzerland)
FFT	Fast Fourier Transform
FHPCA	FPGA HPC Alliance
FP	Floating-Point
FPGA	Field Programmable Gate Array
FPU	Floating-Point Unit
FZJ	Forschungszentrum Jülich (Germany)
GB	Giga (= 2^{30} ~ 10^9) Bytes (= 8 bits), also GByte
Gb/s	Giga (= 10^9) bits per second, also Gb/s
GB/s	Giga (= 10^9) Bytes (= 8 bits) per second, also GByte/s
GCS	Gauss Centre for Supercomputing (Germany)
GDDR	Graphic Double Data Rate memory
GÉANT	Collaboration between National Research and Education Networks to build a multi-gigabit pan-European network, managed by DANTE. GÉANT2 is the follow-up as of 2004.
GENCI	Grand Equipement National de Calcul Intensif (France)
GFlop/s	Giga (= 10^9) Floating point operations (usually in 64-bit, i.e. DP) per second, also GF/s
GHz	Giga (= 10^9) Hertz, frequency =10^9 periods or clock cycles per second
GigE	Gigabit Ethernet, also GbE
GNU	GNU's not Unix, a free OS
GPGPU	General Purpose GPU
GPU	Graphic Processing Unit
HBA	Host Bus Adapter
HCA	Host Channel Adapter
HDD	Hard Disk Drive
HE	High Efficiency
HMPP	Hybrid Multi-core Parallel Programming (CAPS enterprise)
HP	Hewlett-Packard
HPC	High Performance Computing; Computing at a high performance level at any given time; often used synonym with Supercomputing
HPCC	HPC Challenge benchmark, http://icl.cs.utk.edu/hpcc/
HPCS	High Productivity Computing System (a DARPA program)
HPL	High Performance LINPACK
HT	HyperTransport channel (AMD)
IB	InfiniBand
IBA	IB Architecture
IBM	Formerly known as International Business Machines
ICHEC	The Irish Centre for High-End Computing
IDRIS	Institut du Développement et des Ressources en Informatique Scientifique (represented in PRACE by GENCI, France)
IEEE	Institute of Electrical and Electronic Engineers
IESP	International Exascale Software Project
IMB	Intel MPI Benchmark
I/O	Input/Output
IOR	Interleaved Or Random

IPMI	Intelligent Platform Management Interface
ISC	International Supercomputing Conference; European equivalent to the US based SC0x conference. Held annually in Germany.
JSC	Jülich Supercomputing Centre (FZJ, Germany)
KB	Kilo (= 2^{10} ~10^{3}) Bytes (= 8 bits), also KByte
KTH	Kungliga Tekniska Högskolan (represented in PRACE by SNIC, Sweden)
LINPACK	Software library for Linear Algebra
LLNL	Laurence Livermore National Laboratory, Livermore, California (USA)
LRZ	Leibniz Supercomputing Centre (Garching, Germany)
MB	Mega (= 2^{20} ~ 10^{6}) Bytes (= 8 bits), also MByte
MB/s	Mega (= 10^{6}) Bytes (= 8 bits) per second, also MByte/s
MFlop/s	Mega (= 10^{6}) Floating point operations (usually in 64-bit, i.e. DP) per second, also MF/s
MHz	Mega (= 10^{6}) Hertz, frequency =10^{6} periods or clock cycles per second
MKL	Math Kernel Library (Intel)
Mop/s	Mega (= 10^{6}) operations per second (usually integer or logic operations)
MPI	Message Passing Interface
MPP	Massively Parallel Processing (or Processor)
MRAM	Magnetoresistive RAM
MTTF	Mean Time To Failure
NCF	Netherlands Computing Facilities (Netherlands)
NoC	Network-on-a-Chip
NIC	Network Interface Controller
NUMA	Non-Uniform Memory Access or Architecture
NVM	Non-volatile memory
NVRAM	Non-volatile RAM
OpenCL	Open Computing Language
OpenGL	Open Graphic Library
Open MP	Open Multi-Processing
OS	Operating System
PC-RAM	Phase-Change RAM (also PCRAM or PRAM)
PCIe	Peripheral Component Interconnect express, also PCI-Express
PCI-X	Peripheral Component Interconnect eXtended
PGAS	Partitioned Global Address Space
PRACE	Partnership for Advanced Computing in Europe; Project Acronym
PSNC	Poznan Supercomputing and Networking Centre (Poland)
QDR	Quad Data Rate
RAM	Random Access Memory
RDMA	Remote Data Memory Access
RISC	Reduce Instruction Set Computer
SARA	Stichting Academisch Rekencentrum Amsterdam (Netherlands)
SAS	Serial Attached SCSI
SATA	Serial Advanced Technology Attachment (bus)
SDK	Software Development Kit
SGI	Silicon Graphics, Inc.
SIMD	Single Instruction Multiple Data
SNIC	Swedish National Infrastructure for Computing (Sweden)
SP	Single Precision, usually 32-bit floating point numbers
SSD	Solid State Disk or Drive

STFC	Science and Technology Facilities Council (represented in PRACE by EPSRC, United Kingdom)
STRATOS	PRACE advisory group for STRAtegic TechnOlogieS
STT	Spin-Torque-Transfer
TB	Tera (= 240 ~ 1012) Bytes (= 8 bits), also TByte
TCO	Total Cost of Ownership. Includes the costs (personnel, power, cooling, maintenance, ...) in addition to the purchase cost of a system.
TDP	Thermal Design Power
TFlop/s	Tera (= 10^{12}) Floating-point operations (usually in 64-bit, i.e. DP) per second, also TF/s
Tier-0	Denotes the apex of a conceptual pyramid of HPC systems. In this context the Supercomputing Research Infrastructure would host the Tier-0 systems; national or topical HPC centres would constitute Tier-1

Executive Summary

This document gives an overview of technology trends, which are likely to lead to products with applicability to high performance computing in the 2015-2018 period. These insights can be used as input to future PRACE Tier-0 procurements, although of course they are applicable to anyone interested in thinking about the sorts of HPC systems which will be available in this time.

An earlier internal PRACE deliverable (D9.1.1, Multi-petascale Technology Assessment Report) was produced in 2011, and was a synthesis of input from a wide range of industry specialists. It was conducted under confidentiality agreements, and therefore the final publication has been restricted to a PRACE audience. It identified 14 areas in which technology developments would be required and seen in the 2012-2015 period:

1. Energy
2. Memory
3. Interconnect
4. Cooling
5. Operating system
6. Hardware reliability
7. Application reliability
8. Management
9. Processor
10. Packaging
11. Storage
12. File system
13. Archive
14. Application

This current work and this resulting document have been based entirely on public discussions of likely trends and directions, avoiding information restricted to PRACE, and therefore it can be published openly by PRACE. It concentrates on five identified themes and their impact on Exascale systems design and development likely to be seen in 2015 and beyond. The major themes identified are:

1. Data and memory hierarchy – 3D stacking of memory, silicon photonics, deepening memory hierarchies
2. Fault tolerance – possible at a price of increased hardware and running cost
3. Energy efficiency – the need for software optimisation
4. Architecture – heterogeneity, type of cores, interconnect
5. Scale – number of cores and massive parallelism

These five areas incorporate all the 14 areas explored in our original report, but necessarily is in more general terms about trends and directions rather than imminent products.

1 Introduction

PRACE WP9 invited subject matter experts and selected experts active in research and development in both academia and industry to present their views on technology developments at a workshop held in STFC Daresbury Laboratory, UK, in April 2012.

The contents of the deliverable are:

- A report written by PRACE WP9 members which discusses the presentations and draws conclusions from these;
- The presentations delivered during the workshop [2].

The report firstly summarises and identifies the five major technology areas for which major changes are to be required for future Exascale systems, and then reports on more specific trends and directions, mapping these back to the five major areas identified initially. The report is written by the PRACE WP9 members identified in the front of this document and is the opinion of these members and PRACE WP9 in general.

2 Major hardware trends affecting Exascale developments and their potential impact on software

Presenter: Simon McIntosh-Smith (Bristol University) [2]

The major trends identified in the introduction are set in the context of increasing transistor counts on processor chips, further advances in fabrication technologies but no increase in processor clock speeds and the limit on performance being imposed by the power which chips can consume.

Memory stacking will deliver greater bandwidth and energy efficiency, but whatever memory technologies are used we will see that *moving* data within future Exascale systems will come to dominate the total power bill, and we need to move to a paradigm in which we view the compute performance of such systems as essentially free with the major constraint being that we have to reduce the cost of moving data around Exascale systems. One major technology implementation, on which significant research and development work is being performed, is to move data around systems entirely through the use of optical connections – *silicon photonics* – in which optical signals are generated, transmitted and detected directly. This technology is likely to need an external "power supply" in the form of a laser injector, in much the same way as today's electrical circuits are driven by a direct current power supply.

The implications on code design are significant when we consider the likely trends in which microprocessor performance is increasing at 55% annually whereas memory bandwidth only increases at less than 30% annually, meaning that every four years the balance between the two changes by a factor of two.

Fault tolerance computing is taken for granted today, but as processor lithography shrinks the sizes of the transistors and the numbers of transistors increases the likelihood of faults developing in components will increase. The view is that it **will** be possible to construct future reliable Exascale system hardware, but the price may make such systems unrealistic for most potential users – both the cost of the hardware itself and the running cost of such fault-tolerant systems. In such a context, there is a significant role for fault-tolerant **software**, certainly including operating system and middleware software but potentially including application fault-tolerant awareness.

Hardware improvements will deliver increasing energy efficiency but not sufficient to meet the requirements of future Exascale systems alone. Significant energy efficiency improvements will need to be delivered through software optimisation – optimisation for energy efficiency rather than just for performance that we have been used to.

Scaling considerations will need to cover increasing numbers of processor cores, which on Exascale systems will not be homogeneous. So increasing multi-level parallelism in multiple levels will lead to increasing complexity of programming as well as increasingly complex hardware designs.

3 Application and hardware co-design

Presenter: Rich Vuduc (Georgia Tech) [2]

Topic Categories: 1) Data and Memory Hierarchy, 3) Energy Efficiency and 4) Architecture

Algorithms and computer architecture especially CPU architecture influence each other. “Well, it may be alright in practice, but it will never work in theory.” (Warren Buffet, [3])

Current CPU architectures only cover a very small efficiency space for scientific problems. They excel in on area (matrix multiply) but are very inefficient for others (3D FFT).

CPU design can be seen as a notional problem for fixed computation, die area (transistors) and power budget. With these boundary conditions a set of formulas can be used to calculate the best layout to run a set of algorithms so that they complete in minimum time. **Figure 1** shows two possible tradeoffs that can be explored. One is more bandwidth versus higher frequency (power) and the other is more cores versus more cache (transistor area). Other considerations are fast memory, which sits on-chip versus slow memory, which currently is DRAM and the possibility of hiding latency if there is a lot of concurrency.

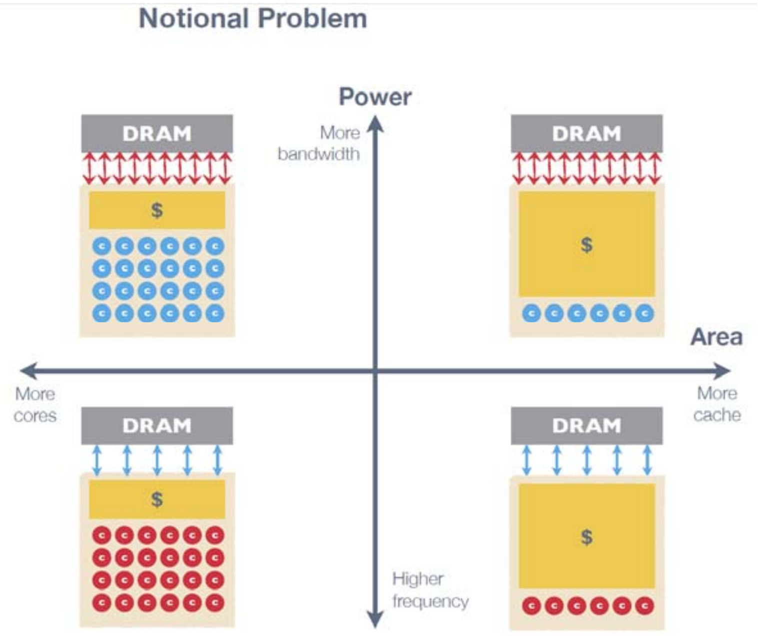

Figure 1: Power and Area tradeoffs

Projections for 2018:

Plotting memory bandwidth and cache size shows that current CPU and GPU designs are good for matrix multiplication but very inefficient for 3D FFTs. It will be more important to tune memory bandwidth and not the cache because the cache sweet spot seems to be 64MB both for 3D FFT and matrix multiplication.

The upcoming Echelon design by NVIDIA expected in 2017 will feature an increased cache size and higher memory bandwidth then current GPUs and CPUs. It will have a more balanced performance for 3D FFT versus matrix multiplication. Machine balance B can be defined as Flops/Memory Bandwidth and different ratios are important for different scientific problems.

For 3D FFT a lot of the system power goes into the memory subsystem opposite to matrix multiplication where the most power is spent.

Figure 2 shows the difference between an optimal Matrix Multiply machine and a FFT machine. Relative to Echelon the special matrix multiplication machine is 5 times faster for matrix multiplication but only 0.9x as fast for 3D FFT. On the other hand the special 3D FFT machine is 28 times faster for 3D FFT but only 0.14 times as fast for matrix multiplication. Another interesting point is that the transistor distribution of the chip is very similar but the node power and system power budget looks very different. For example 3D FFT will use a lot of power for memory and network whereas the matrix multiplication spends most of it on computational power.

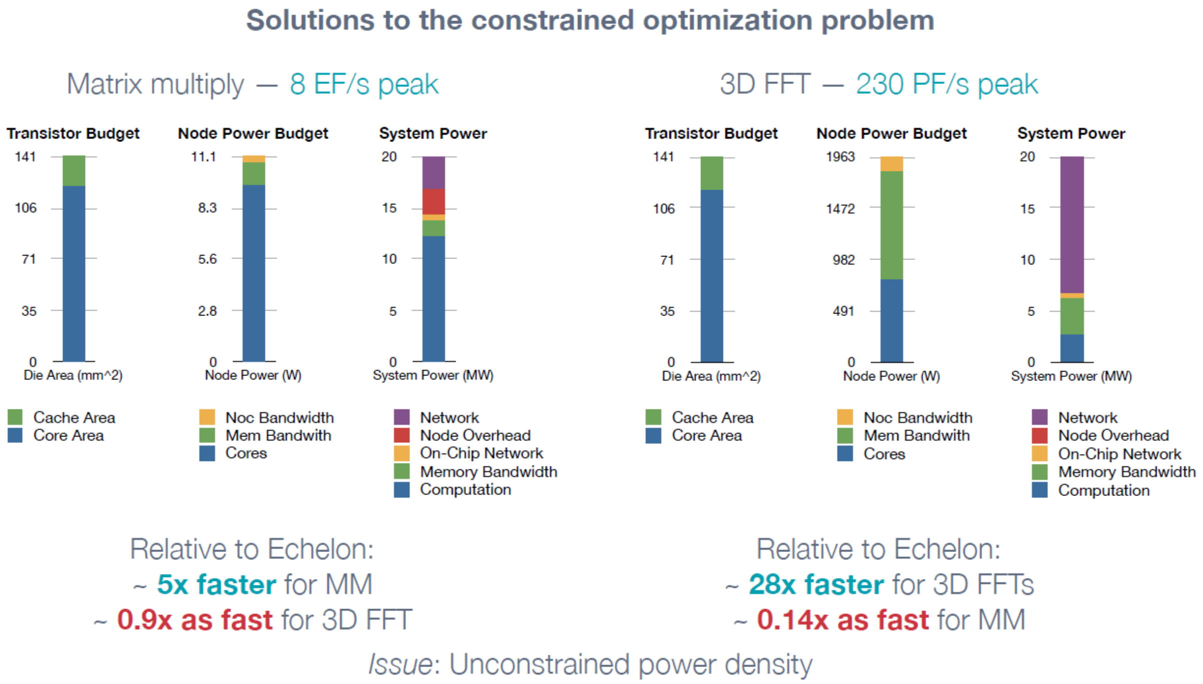

Figure 2: Matrix multiplication machine versus 3D FFT machine in comparison to Nvidia Echelon

In addition to the power allocation differences the optimal number of nodes for a future system varies depending on the problem. **Figure 3** shows that for an optimal FFT machine the required number of nodes is much smaller than for the matrix multiplication one.

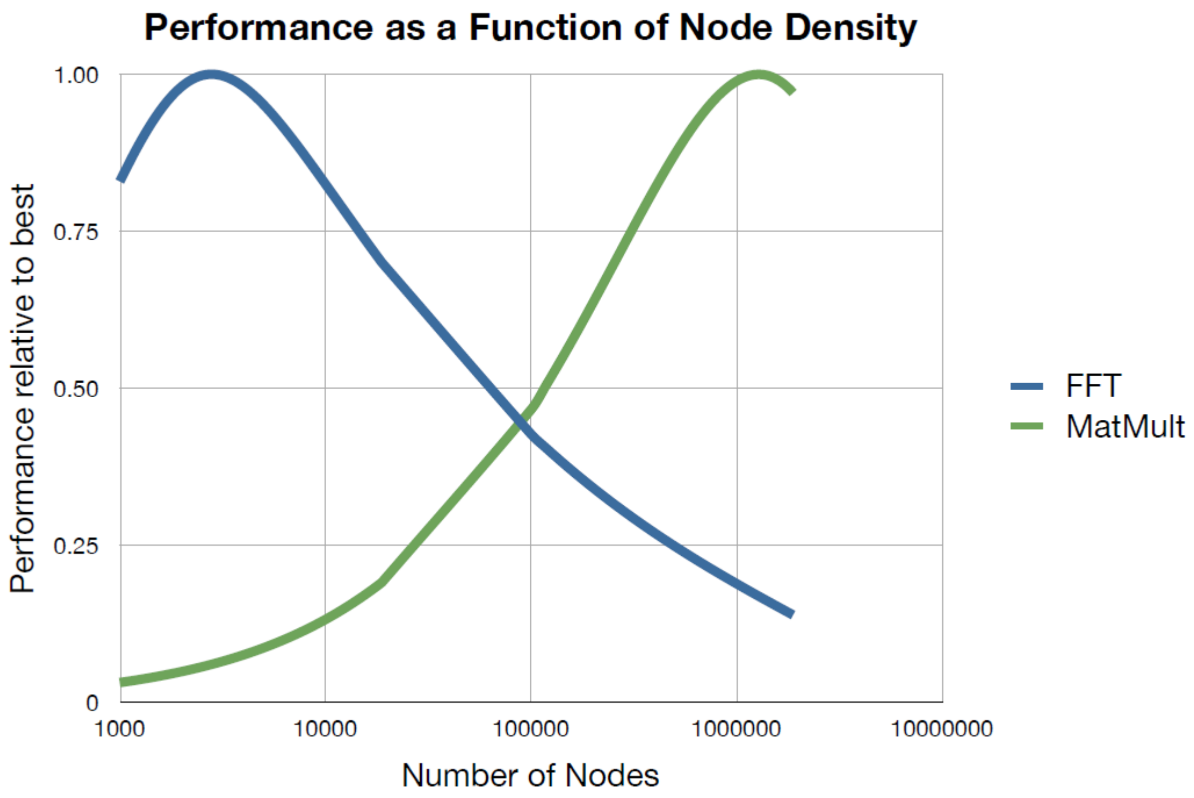

Figure 3: Matrix multiplication and 3D FFT performance in relationship to number of nodes

The question is if we are even looking in the right “space” of designs for Exascale. Because it is clear that more diverse and even specialized systems are possible. We could be at a point where it makes more sense to move forward by using a specific scientific application domain as co-design vehicle for a highly optimized special purpose HPC system. In any case memory bandwidth and concurrency are very important for future systems.

4 Emerging technologies in the path to Exascale

Presenter: Jeff Vetter (ORNL) [2]

Topic Categories: 1) Data and Memory Hierarchy, 3) Energy Efficiency and 4) Architecture

NVRAM

Key research questions for Exascale systems are: assessment of technologies in terms of their availability and reliability, the role of the technologies in terms of their usage in Exascale systems as replacement of main memory or disks or both, evaluation of hybrid DRAM-NVRAM technologies, and possible implications for system software development. One active area of research is evaluation of NVRAM for scientific applications [5]. Two key features of NVRAM are:

- their energy efficiency in stand-by mode but
- higher latencies and energy usage for write operations.

By collecting memory access traces of selected applications and measuring read and write ratios, object sizes and memory references it could be possible that most of the patterns and behaviors in the targeted applications map well to NVRAM characteristics, with minor changes. Hence, with minor changes these applications can exploit a hybrid DRAM and NVRAM node and at the same time may offer additional benefits for co-design approaches.

Byte-addressable NVRAM has interesting implications for the software stack, specifically programming models and compiler support, and potential impact on the memory hierarchies.

Heterogeneous Systems

A U.S. DOE-funded project called *Vancouver—Designing a Next-Generation Software Infrastructure for Productive Heterogeneous Exascale Computing* [6] builds on the Keeneland system, a U.S. NSF computing resource based on a hybrid CPU-GPU architecture, which has a surprisingly high rate of adoption—97 projects and over 200 users. Even though there are many success stories with heterogeneous computing, there are still many challenges that need to be addressed in order to make the technology accessible to a wider user base. The Vancouver project has tried to address some of these challenges by focusing on various levels of the software stack.

One of the major limitations with accelerator computing has been the lack of a unified host-device memory—without this, applications must overcome the slow PCIe link. AMD addressed this with their Llano A-Series APU, and experimental results demonstrate how there are many significant performance and programmability trade-offs associated with this unified memory approach [7].

Another challenge is that heterogeneous applications are being forced to use complex programming models. Directive-based programming models can simplify the challenges related to hybrid accelerator programming. A NOAA weather model showed that their PGI Accelerator directive implementation is competitive in performance to their hand-written CUDA implementation, at the same time being significantly easier to code and implement. Work on numerical libraries to target heterogeneous systems (e.g. MAGMA) is an active area of research, as are runtime systems for accelerators that orchestrate data movement with minimal application input [8].

Hybrid accelerator-based systems also cause challenges in understanding application performance. The Vancouver project has developed benchmarks that provide quantitative performance information about important data movement and kernel operations for many heterogeneous architectures [9]. Additionally, the project incorporates work on tools like TAU [10] that integrate heterogeneous performance analysis results into a single view, which allows for a more holistic view of a heterogeneous application's performance.

Finally, portability is as another challenge for heterogeneous systems. As an example solution, a dynamic compilation framework called Ocelot [11] allows retargeting of CUDA code to other architectures such as AMD GPUs and multicore x86 processors. Also, the HyVM (Hybrid Virtual Machine) project [12] attempts to tackle this problem by looking at virtualization and task scheduling of various many-core systems.

5 An update on memory technologies

Presenter; Aad van der Steen (HPC Research/NWO) [2]

Topic Categories: 1) Data and Memory Hierarchy, 3) Energy Efficiency

The presentation is an update of the presentation given to PRACE WP9 in October 2010.

Current DDR3 memory will be superseded by DDR4 in 2013/14, but this is an incremental technology development which will significant increase the potential memory bandwidth but at a cost of increased latency. 3-D stacked memory is also likely to be implemented in 2013 and this will significant increase memory bandwidths. But memory continues to be a major bottleneck to performance on current systems and the mismatch between processor and memory capabilities will only increase in the near future.

By 2015-2018 a different memory technology will be required, which needs to improve bandwidth and latency characteristics, but also needs to be inexpensive, durable, reliable, small and consume less power than today’s technology.

As we identified in 2010, this almost certainly requires non-volatile memory which does not consume power when storing data, as well as requiring low power when reading and modifying data.

The first step towards this will be *commercialisation* of memory technology, and it was observed that Z-RAM (identified as potentially interesting in 2010) is a viable technology for which the patent owners are not making any product.

In other words, it is not sufficient to prove that a new memory technology works, it has also to be viable to produce and to use.

A roadmap for memory technology going towards 2018 could be:

1. 3-D stacking of DDR3, then DDR4 memory in 2013 and 2014
2. Phase change memory to replace SSD but not DRAM (it is too slow for the latter use) starting in 2013
3. Spin Torque Transfer Magnetic RAM as DRAM replacement in 2014-2015
4. Memristor as DRAM replacement in 2015-2016
5. Racetrack memory – if developed commercial from the existing IBM prototype – 2015 and onwards
6. Graphene memory after 2016 depending on further research

6 Evolution and perspective of topology-based interconnect

Presenter: Giampietro Tecchiolli (Eurotech) [2]

Topic Categories: 3) Energy Efficiency and 4) Architecture

Giampietro Tecchiolli from EUROTECH analysed the impact of interconnect components on the evolution of HPC systems along the roadmap to Exascale system. Interconnects have their main impact on the scalability of the system, but they rise also important issues on Energy Efficiency and Fault tolerance. Interconnection technologies inside HPC systems can be divided in three main categories: chip, board and system wide interconnections; each having different problem and different technology solutions. The first category regards mainly architectural aspects and chip design and is less relevant if we focus on system integration and scalability issues.

On Board Interconnects

These are used to connect different chips inside a given board (Printed Circuits Board, PCB), like memory chips or PCI bus to CPU. They are mainly realized using copper and the quality of the signal they can carry depends very much on the material used as a support of the board (this is clear if we think that different materials have different dielectric constants and can behave like small capacitors that interfere with the propagation of the signal). Typical lengths are of the order of few centimetres up to one meter, and should have no problem in carrying signals up to 30Gb/s. Today typical frequencies are well below 10Gb/s, therefore the current technology guarantees room for improvement for next generation architectures looking at the Exascale roadmap. Soldering chips and soldering technology may impact the performance of these interconnects as well. The copper base on-board interconnections may be limited by power constraints, since as the bandwidth increases the total power loss (all signal power not delivered to the receiver) increases exponentially. For typical copper strips used in PCB carrying a signal at 20GHz the total power loss could be as high as the 60dB/meter. This power constraint motivates the research for alternative ways to connect chips at board level. From this point of view the most attractive alternative is represented by optical interconnects based on silicon photonic technology. Many different possible solutions are investigated in research labs (MIT, Sandia, IBM, Intel, etc.), but unfortunately they are not yet mature for production at an industrial level. Another alternative is represented by the possibility to connect chips with copper wires embedded in strip cables not printed on the board; this reduces the capacitance of the wire itself allowing a smaller total power loss. It seems highly probable that in the next five year we will stay with copper base on board interconnects.

System interconnects

Today and in the future all supercomputers are and will be built using a large number of nodes (running a shared memory operating system) bound together using a system wide (usually ad-hoc) interconnect. The system interconnect is therefore the main component impacting the scalability of a given system. The more performing it is (in terms of bandwidth and latency), the more scalable is the system. The network has also important impact on power consumption, fault tolerance and data hierarchy. Two main alternative topologies are available today: Fat Tree and N dimensional Torus. Fat Tree topology is typical of x86 Linux based cluster while N dimensional torus is mainly used in specialist HPC systems such as IBM’s Blue Gene, Cray’s XE6 and Fujitsu’s K computer. A Fat Tree topology has the advantage of allowing a fully interconnected machine with a full bisection bandwidth. On the other hand, in a Fat Tree topology the number of switches and cables scale more than linearly with the number of nodes. It has been estimated that for an implementation of a state of the art

Fat Tree topology using InfiniBand switches, to connect a machine with 11664 nodes (maximum dimension using the largest switches available) will require 100km of optical cables, 648 Level 1 36-port switches and 18 Level 2 648-port switches. This behaviour makes the Fat Tree topology a solution that can hardly be adopted on machines with million of nodes in the Exascale roadmap. It is also important to remark that the market is dominated by a single vendor (Mellanox) and pricing (with respect of the total cost of the machine) could become an issue.

With InfiniBand it is possible to build a different topology (n-dimensional Torus, or constellation) but in this case the routing protocols are, for the time being, not very effective. More research and engineering effort has to be spent in order to improve performance, resilience and reliability. Nevertheless 3D torus topology using InfiniBand has some attractive features, and there are projects working on it today (RED-SKY, Direct 3D, Switch level implementation).

N dimensional Torus topologies do not allow a full bisection bandwidth system but the number of network components (cables, adapter, network chips) scale linearly with the system size, and its implementation is sustainable in the Exascale roadmap. Connections, apart from those used for the torus closure, are between neighbouring nodes, allowing a reduced number of cables and the usage of copper instead of optical cables, thanks to the short length of the cables themselves. Using topological transformations the long connection required to close the torus can be configured to be short in length too. So that, for the same hypothetical machine using Fat Tree topology in which 100km of cable is required, only 0.7km of cable is required in the deployment of an alternative 3D Torus topology. Torus networks can built on ASIC or FPGA network processors, the main difference is that ASIC could allow a lower latency than FPGA, with no difference in bandwidth between the two. Torus link speeds could be as high as 120Gb/s in three years' time, and this is aligned with Exascale roadmap (considering a total node bandwidth of 6*120 = 720Gb/s).

Torus network based on FPGA or ASIC have re-routing capability around failures and this is of fundamental importance for large Exascale implementations. Many Torus networks have also the possibility to reconfigure the links and the topology in order to implement closed sub-torus networks within the system network (for example, Blue Gene). This is a good option since it allows an optimal use of the system when it is used by more than a single application (as it is often the case).

7 Overcoming the barriers to Exascale through innovation

Presenter: Jim Cownie (Intel) [2]

Topic Categories: All

Intel as a company is very much business driven and focuses primarily its R&D on the technology the market tends to. But the good news is that Intel considers that the current and foreseeable trends for both the consumer and the mass market are converging. Therefore we can be certain that Intel's commitment to reach the Exascale milestone on time is full and genuine since the technologies required to achieve this goal will directly benefit to the mass market, where Intel's main business is.

The main technological barriers and challenges Intel focuses its efforts on are

Data and memory hierarchy

The memory issues for reaching the Exascale target can be addressed with:

- Some new memory technologies, where pages might become smaller to reduce the power consumption of data fetching / refreshing
- Some new / rearranged memory hierarchy with even more data locality, maybe at the expense of cache coherency at some stage
- A minimisation of the data movements along the hierarchy
- Some new memory / chip packaging at the hardware level, with 3D stacking of both memory and CPUs, here again to promote data locality, reduce the data movement and keep a good bandwidth with a sufficient number of pins in a reduced footprint.

Fault tolerance – Programming tools

The increase in total system parts and the possible decrease or not substantial increase of components MTTF means that for an Exascale machine, the mean time to interrupt would become smaller than the time needed for a checkpoint using today's paradigms. This makes it very important to investigate new alternatives.

Energy efficiency

This is the main barrier which constrains and drives all the other research domains. Some current research domains include:

- 3D packaging and stacking of memory with processors
- Extreme voltage scaling to remain on the hardware's peak power effectiveness
- Software support of data locality maximisation

Architecture

Intel is committed towards the MIC architecture and generally speaking the unavoidable shift towards heterogeneous "cores". However, since this was not the topic of the presentation, it was not developed further.

Scale

Hardware developments alone won't permit us to reach the Exascale target on the expected timeframe – and this is a major statement for a hardware provider such as Intel. For the first time in the HPC history, software will be at least as much important as hardware in moving forward. Significant efforts in developing and proposing innovative tools and languages to address this issue, and hardware providers such as Intel are willing and active participants in this effort. All the projections of what an Exascale machine will look like show that the current programming model won't be sufficient. With many hundreds of cores per processors, and many hundreds of chips, the fully MPI model won't work anymore while the shared memory model won't be addressable with OpenMP. One needs a new programming paradigm for effectively addressing the shared memory parallelism while allowing the distributed memory parallelism to be addressed as well in parallel. Intel's proposed solutions are based on Cilk, a new programming language defined as a small parallel extension to C and C++. This language permits the developer to easily express parallelism at the code level, without having to think about the implementation and hardware details. In addition, Intel develops some development tools to support the language, such as profilers and debuggers.

Many of the places where this software research is done is actually located in Europe. Four Intel Exascale labs are based in Germany (with JSC – Julich), Spain (with BSC – Barcelona), France (with GENCI and CEA – Saclay) and Belgium (Leuven). All included, 35 Intel research labs are located in Europe for a total of several hundreds of researchers understanding and addressing the many issues the technology is facing today and for tomorrow. Furthermore, Intel is very keen to explore whatever new collaboration we could propose to even better address those challenges.

8 From fault tolerance to resilience

This contribution includes parts from the "Technical Report of the INRIA-Illinois Joint Laboratory on PetaScale Computing TR-JLPC-09-01" [13] and the "IESP roadmap" [14]. No presenter was able to present during the workshop, but this chapter is a synthesis of the presentation provided by INRIA PRACE members.

Context

Over the past few years resilience has became a major issue for HPC systems, for current large Petascale systems and future Exascale ones. These systems will typically comprise half a million to several millions of CPU cores running up to a billion threads. From the current knowledge and observations of existing large systems, it is anticipated that Exascale systems will experience various kinds of faults many times per day. It is also anticipated that the current approach for resilience, which relies on automatic or application level checkpoint-restart, will not work because the time for checkpointing and restarting will exceed the mean time to failure of a full system.

This set of projections leaves the community of fault tolerance for HPC systems with a difficult challenge: finding new approaches, possibility radically disruptive, to run applications until their normal termination despite the essentially unstable nature of Exascale systems. Yet, the community has only five to six years to solve the problem.

Issues in Exascale systems

There is a broad consensus in the community about the fact that Exascale systems will be hit by errors/faults much more frequently than Petascale systems. There are two main reasons behind this belief:

(1) An Exascale system will be composed of many more components than Petascale systems and (2) the mean time to failure (MTTF) of each of these components will not improve enough to compensate for (1).

As previously mentioned, current projections of Exascale systems will comprise millions of CPU cores and may have to run up to billions of threads. If we look to the past ten years, the performance increase of the supercomputers in the Top500 resulted from an increase in CPU clock frequency, an increase in the number of transistors per chip and an increase in the number of sockets in a machine. Clock frequency has flattened in the last few years, so that the increase in the number of sockets can be expected to accelerate.

As a consequence, the number of components in Exascale system will be much higher than the one of Petascale systems (100,000 is the order of magnitude of the number of sockets we may see in Exascale systems).

Moreover, the reliability of individual components is not likely to improve significantly in the near future. The lifetimes of consumer products provide no incentive for manufacturers to change the existing reliability levels of components, which are typically a few years. Indeed, the reliability of the components in HPC systems has not improved and may have degraded in the last 10 years. HPC vendors have compensated for this by adding more hardware redundancy and error checking in their systems. Even if there is not yet a consensus on this aspect, there is a suspicion that software errors will dominate in Exascale system. The rationale behind this belief is that (1) the software stack running on every node of a parallel computer is already very complex (current estimations of the number of code lines in such software is several millions) and (2) this software stack has not been designed or tested with high availability and resilience in mind.

As a consequence most of the software parts: (a) are not restartable or replaceable without impacting the other software parts, (b) do not integrate enough fault-error detectors, and (c) have not been tested, validated or formally verified at the (much more expensive) level used for critical software. The community has translated these projections and suspicions into the following statement: faults/errors/failures will not be rare events anymore and should be considered as normal events. In other words Exascale systems will need to resist a continuous stream of faults/errors/failures.

The community, based on its past experiences and the observations of the current largest systems, envisions the following major issues:

1*) Some faults will not be detected (silent errors). Both hardware and software silent errors are likely to happen.*

2) Detected but uncorrectable transient errors may represent a large fraction of errors. The assumption is that with the increase of the integration level, the phenomena causing transient errors will have a much wider impact on the affected components, despite the redundancy mechanisms added by the manufacturers.

3) Correctable errors will increase the hardware jitter due to the background error recovery activities (e.g., memory scrubbing & error handling) that will become significant.

4) Long running jobs may be hit by hardware & software faults (of multiple types) several times before completion.

5) Current designs and practices of global, synchronized Checkpoint-Restart (on remote file system) will not work anymore.

The community concurs that research for more reliability and robustness is critical at every layer between the hardware and the end-user. Considering the existing technologies, the state of the art in research and the forecasted faults/errors characteristics of Exascale systems, new resilience paradigms are required.

From Fault Tolerance to Resilience

Essentially, users of HPC systems want to be able to submit long-running jobs and have them run to completion in a timely manner. This demand is even more stringent for users of top-level supercomputers because these systems are acquired to run jobs that cannot complete in a timely manner on smaller systems. However several obstacles make this demand difficult to achieve even for today’s supercomputers. Because of their scale and complexity, current supercomputers have frequent failures and can run for only a few days before some part of the system requires rebooting.

While techniques for fault tolerance and continuous and tightly-coupled operation exist and are used in some specialized systems, these techniques have not been scaled to the level required for supercomputing and are extremely expensive. The cheaper alternative of maintaining a safe state on stable disk storage does not work well for large, tightly coupled applications and results in the loss of significant compute work whenever a failure occurs.

The current response to faults in existing systems consists in restarting the execution of the application and the pieces of its software environment that have been affected by faults. To avoid restarting from the beginning, users may checkpoint the execution of their applications periodically and restart them from a safe checkpoint after faults have occurred. Note that in some situations, several pieces of the software environment have to be restarted as well. However, checkpointing and restarting has a cost: it takes time and energy.

Some projections estimate that, with the current technique, the time to checkpoint and restart may exceed the mean time to interrupt of top supercomputers before 2015.

This not only means that a computation will make little progress; it also means that fault-handling protocols have to handle multiple errors -- current solutions are often designed to handle single errors. Moreover, the current approach for fault tolerance is to apply the same technique (checkpoint-restart) to all types of faults (permanent node crash, detected transient errors, network errors, file system failures) and for the whole duration of the execution.

However, not all faults require the general and expensive checkpoint-restart approach. As an example, detected transient hardware faults (soft errors) may be managed in a more efficient way. If we observe the situation more closely, we see that none of the higher layers of the software stack have been specifically designed to cope with faults.

Only a few software components, such as some MPI libraries, have been partially retrofitted to tolerate some faults. Moreover, there is no communication and coordination between software layers and software components within every layer for fault detection and management. An example of this lack is the MPI environment itself: even if some MPI libraries have been adapted to tolerate failures, their associated runtime environment is not fault tolerant, and requires a restart from scratch at every fault.

Another example is the lack of coordination regarding fault detection and management between an application and the libraries used by the application. As a consequence, even if applications themselves were designed to resist to faults, most parts of their software environment would not let the execution survive the faults.

Since Exascale supercomputers, which are expected by 2018, will exhibit much more complexity and many more faults than today’s supercomputers, one can gauge the challenge that the community in HPC is facing: it is not only adapting or optimizing well known and proven techniques but it is also making the full software stack fault tolerant and/or fault aware and ensuring that fault detection and management is consistent across the whole software stack.

The IESP (International ExaScale Software Project) roadmap has identified application resiliency as key priority for X-stack (extreme-scale/Exascale software stack).

AMFT addressing challenges of Exascale application resiliency

AMFT – Advance multilevel fault tolerance PRACE prototype targets fault recovery as key aspect of Exascale computing resiliency. As the evolution of the networks and the bandwidth of the parallel file systems will not scale as needed, it will be impossible to checkpoint a full system image at appropriate frequency (for dealing with a low expected Mean Time To Interrupt). One solution consists in implementing application-based checkpoint/restart and to use in a smart way the different levels of storage hierarchies available on HPC systems.

The FTI middleware (Fault Tolerance Interface) co-developed by the INRIA-Illinois joint laboratory on Petascale computing and Tokyo Institute of Technology will be used as the multilevel checkpoint middleware.

The objectives of this prototype are to assess on different hardware platforms the interesting potential of FTI and AMFT on new profiles of applications coming from the PRACE benchmarks, or the newly EUABS (European Unified Applications Benchmark Suite) or applications proposed by community codes from 1IP-7.2 or 2IP-WP8.

Conclusion

It is clear that energy considerations drive the development of Exascale systems, and the challenge for 2015-2018 will be to find commercial implementations of technologies which form part of active research projects as described in this document. Systems which emerge in this period will provide more computing resources than can actually be used – more processor cores and more floating point operations than any application will actually be able to use. This sounds negative, so what this means in a more positive light is that computing resources in Exascale systems will be “free” and the limits on performance of Exascale applications will be imposed primarily by reasons related to power consumption and the movement of data in and around the systems.

This will drive changes in memory hardware, something significantly more radical than another version of DRAM is going to be required in which energy consumption to store data is radically reduced yet with performance (bandwidth AND latency) at least as good as we see today.

Increasing complexity of systems will significantly reduce the mean time between failure, and we will see two divergent paths: firstly one in which systems and operating system design “covers up” the underlying failures and continues today’s paradigm in which the programmer can assume a completely reliable system, and a second path in which fault tolerance will be exposed to the programmer who will have to take explicit action to handle faults when they occur. The divergence stems from the fact that the former option of a “fault free” system will be significantly more expensive to buy and to run than on which exposes faults when they occur, and means a more likely path of the cheaper option for the majority of future HPC systems.

Energy efficiency is going to drive the requirement to optimise software for energy use – so programmers will need tools to enable them to be aware of the energy savings which can result from appropriate reengineering of software.

Exascale performance is not going to be possible to homogeneous systems, and so high levels of performance are only going to be possible with a combination of heterogeneous hardware and software. MPI will not be replaced in the next few years, but it will need to be accompanied by an increasingly complex toolbox in which maximum performance will only be possible by deploying multiple tools in parallel.

Scaling of systems used to be about building larger and larger systems, comprising faster and faster components. The hardware vendors are no longer promising this, and indeed there is emerging unanimity across the processor vendors, systems suppliers and programming community that software is the key to future Exascale performance, and without significant investment in potentially major and radical software re-engineering the step change to Exascale will not be possible by hardware alone. The good news is that this consensus is also accompanied by a reality check that – with the appropriate investment by all parties – the future radical step change in supercomputing science capability is indeed going to be possible. This document attempts to capture the key areas in which changes in technology will be managed and implemented on the road to Exascale and will be seen in multi-Petascale computer systems in a few years’ time.