# A Survey of State-of-the-art Hardware and Software

Delivered in Month 9– June2016

E-CAM
The European Centre of Excellence for
Software, Training and Consultancy
in Simulation and Modelling

Funded by the European Union under grant agreement 676531

# 1 Preamble

Deliverable 7.1 states: A joint report by STFC, FR-IDF, and ICHEC on: (a) Report on the hardware developments that will affect the scientific areas of interest to E-CAM and detailed feedback to the project software developers; (b) discussion of project software needs with hardware and software vendors, completion of survey of what is already available for particular hardware platforms; (c) detailed output from direct face-to-face session between the project end-users, developers and hardware vendors.

The following people contributed to the the present report: Daniel Borgis, Pierre-François Lavallée, Isabelle Dupays, Marie Flé, Liang Liang (FR-IDF), Leon Petit (STFC), Michael Lysaght (ICHEC) and Alan O'Cais (Jülich).

This deliverable is a mix of TASK 1 and TASK 2 of WP7 in the proposal, namely:

**TASK 1: Monitoring of hardware developments (coordinated by UK-HARTREE):** will follow and document hardware developments on the 3 to 5 year horizon; relate hardware developments with specific requirements for software development; feedback back to software developers on interfacing to hardware, specific programming concepts, or restrictions on hardware (e.g. memory, logics, NUMA, precision); document best practices on hardware restrictions.

**TASK 2: Survey of specific user communities and hardware centres (coordinated by FR-IDF):**. This concerns a broad survey on specific software needs, and provides input for the software development teams and hardware vendors. The survey will take into account future trends in algorithms, methods, parallelization techniques, needs for data storage and analysis. The survey will target the CECAM and PRACE user communities and a significant number of industrial companies.

# 2 Introduction

Since their wide-spread use in the 1980s, the efficiency of microprocessors has been linked intrinsically to the increase of their clock frequency and also the increase of the transistors' density. Microprocessor efficiency is due, in large part, to lithographic processes becoming finer and finer. This efficiency evolution is even acknowledged in the so-called Moore's law that states that the number of transistors is doubling every 18 months.

If this rule remains true, then the regular increases in the clock frequencies of the microprocessors should have stopped in 2004 due to size of the thermal dissipation induced by the high voltage necessary to control such frequencies. Since then, in order to continue to increase the power of the microprocessors, the informatics industry has taken advantage of further progress in lithography to integrate, at a constant frequency and reduced voltage, a larger number of computing units for the same surface. Today this can be as much as 16 units, or cores, per processor. This is the paradigm for modern supercomputers, composed for the top-ranked machines of tens of thousands of processors, and thus hundreds of thousands of cores, with ultrafast connection between the processors. However, taking advantage of all of the available computing capability requires an extensive parallelization effort on the codes. This is accomplished through parallel programming standards such as OpenMP and MPI. The distribution of computer tasks makes it crucial to manage finely the different levels of memory (cache L1, L2 and L3, RAM, hard-drive) on the different processors. Moreover, hybrid architectures have appeared in recent years, coupling CPU to Manycore or GPU accelerators with many execution units; many top-ranked machines are already based on these principles. These hybrid architectures have required additional coding paradigms such as using CUDA or OpenACC, and an even more subtle management of memory levels. Yet progress here also requires a major effort in terms of software architecture and data structure.

The first aim of this report will be to review the HPC situation in Europe in connection to E-CAM, in order to describe briefly the HPC machines that are available. We do not intend to provide an extensive list but proceed by vendors and type of machines. We will also comment on the state-of-the art software that are recommended to the E-CAM community for programming, profiling, debugging, and using efficient Input/output in order to take full advantage of those massively parallel architectures.

The tendency to even more massive parallelism will continue and probably accelerate over the coming years with the challenge of reaching the exascale ($10^{18}$ floating-point operations per second), that the vendors are now facing. The challenge of getting machines that are 100 times more efficient while only increasing energy consumption by a modest amount requires denser chip architectures, with either much more cores per processor (the so called manycore processors, such as the next generation of Xeon Phi named Knights Landing (KNL) proposed by INTEL), or hybrid architectures coupling CPU and GPU accelerator with ultrafast NVLink interconnections (e.g. the OpenPower consortium grouping IBM, NVIDIA among other companies). Those new platforms provided by the vendors are currently tested in the computing centers attached to E-CAM. They are expected to require a major effort in programming or reshaping the codes, and new programming standards, such as OpenMP4.x, are still under discussion and development.

| Platform | Conceptor | Architecture |
|---|---|---|
| Xeon | Intel | x86 |
| Xeon Phi | Intel | Many Integrated Core (MIC) |
| OpenPower | IBM + NVIDIA | Power CPU + GPU accelerator |
| Tesla | NVIDIA | General-purpose GPU (GPGPU) |

Table 1: List of platforms considered within E-CAM

This report is thus also aiming at a set of prospective recommendations proposed to the E-CAM software developers in order to prepare them for the next generation of HPC machines.

The exploitation of High Performance Computing (HPC) systems within E-CAM will focus mainly on the employment of well-known materials science and life science/biophysics codes as well as high performance software libraries and modules developed during E-CAM's lifetime as part of tasks within WPs 1-4 and WP 6.

Based on an initial review of early-stage E-CAM requirements (carried out by way of a "HPC Requirements Questionnaire" in M3 – hereafter referred to as HRQ), a number of community codes in the field of electronic structure and ab-initio molecular dynamics (e.g. CPMD, CP2K, QuantumEspresso, ABINIT, CRYSTAL, ...) or classical molecular dynamics (NAMD, GROMACS, LAMMPS, DLPOLY, AMBER, ...) have proven to be highly scalable on peta-scale systems. Moreover, they are already targeting emerging many-core architectures featuring as part of many of today's petascale systems. (It is expected that the current list of community codes of interest will grow over E-CAM's lifetime.)

Since E-CAM software libraries and modules will typically interface with highly scalable community codes, it is paramount that E-CAM software modules meet the requirements of scaling to similar levels (both "across" and "on" nodes). It is also vital that E-CAM software is fully prepared to exploit emerging many-core compute platforms that are already being targeted by large-scale community codes of interest to E-CAM. Such massively parallel many-core platforms are expected to feature heavily as part of near-future deep-petascale systems across Europe (as well as reflected in future exascale systems). To this end, the compute platforms that will be immediately targeted during the lifetime of E-CAM are those that are present or that emerge within the PRACE research infrastructure; they are listed in Table 1 and are described in more detail in Section 3.4.

Access to systems featuring these computing platforms for both development and production runs will be sought through PRACE as well as through the HPC centre members of E-CAM.

# 3 Hardware Platforms

## 3.1 Overview of current PRACE infrastructure

Supercomputers can be viewed as very large clusters of processors linked by ultrafast communications and they are indeed committed to the processors units that they are built of. There are only a few companies offering computing processing units (CPU) for HPC applications, mainly INTEL, IBM, AMD, ARM, NEC and Fujitsu. Intel has developed over the years its X86 Xeon processors, improving lately their architecture from Sandy Bridge, Ivy Bridge, to the present Haswell. They represent over 85% of the processors built in the Top500 machines. AMD has improved since 2003 its Opteron processor. IBM is the developer of the Power processor used in its Blue-Gene supercomputers (presently Power8.0). NEC with their SX series vector processors and Fujitsu with their SPARC series use their processors in their own HPC machines. On the other hand, NVIDIA has developed graphic processing units (GPU), initially for video game applications, and more recently as accelerators for computing applications; the latest product is the Tesla processor including a CPU and a GPU with the associated software environment.

Apart from NEC and Fujitsu, and IBM for the Blue Gene, all the HPC vendors (IBM itself, SGI, CRAY, Bull/Atos) propose a massive integration of Intel or AMD CPU's into massively parallel clusters. Some of them now offer hybrid architectures coupling a CPU to an accelerator, be it a Many Integrated Core (MIC) co-processor suc as Xeon Phi, or a GPU co-processor. This trend is reflected by the the top2 machine Tianhe-2 built by the National University of Defense and Technology in China (Intel Xeon + Xeon Phi), or the CRAY top3 TITAN supercomputer at Oakridge (AMD Opteron + NVIDIA K20). Although at a smaller scale, IBM and Bull have incorporated hybrid configurations in several of their European machines, e.g. for SuperMuc@LRZ and Curie@TGCC. The situation of the world Top-500 machines, in terms of vendors, technology, performance, can be found on the website www.top500.org

As far as the Partnership for Advanced Computing in Europe (PRACE) initiative is concerned, the 6 machines associated with the infrastructure (Tier-0) are listed below with their peak performance

- JUQUEEN, IBM Blue Gene/Q, 5,9 PFlop/s, Juelich Supercomputing Centre (JSC)

- Hazel Hen, Cray XC40, 7.4 Pflop/s, High Performance Computing Center Stuttgart (HLRS)
- SuperMUC, IBM iDataPlex, 6.8 PFlop/s, Leibniz Supercomputing Centre (LRZ) in Garching
- Fermi, IBM Blue Gene/Q, 2,1 PFlop/s , CINECA in Bologna
- Curie, BULL, 2 PFlop/s, Très Grand Centre de Calcul (TGCC) in Bruyères-le-Châtel
- MareNostrum III, IBM iDataPlex, 1,1 Pflop/s, Barcelona Computing Center (BSC)

See http://www.prace-ri.eu/prace-resources/. To go into more details of the hardware architectures, we go below by vendors taken from the above PRACE list, or among the national (Tier-1) machines connected in our E-CAM consortium (IDRIS, ICHEC, STFC). Our intention here is not to present an exhaustive list of all of the supercomputer vendors

## 3.2 Overview by vendors

### 3.2.1 IBM

IBM provides the Blue Gene series, Blue Gene/Q for the latest generation, based on the IBM PowerPC processor. Systems of this type are currently the most energy-efficient supercomputers according to the Green 500 List. For example **JUQUEEN** at JSC has an overall peak performance of 5.9 Petaflop. It consists of 28 racks; each rack comprises 1024 nodes of IBM PowerPC A2, 1.6 GHz, 16 cores per node (458 752 computing cores). The main memory amounts to 458 TB.

IBM builds also massively parallell cluster architectures using its iDataPlex systems, based on INTEL Xeon processors –including the option nowadays of coupling them or part of them to Xeon Phi co-processors . For example **SuperMuc**, the Tier-0 machine at the Leibniz supercomputer Center, consists in phase 1 of 18 Thin Node Islands based on Intel Sandy Bridge-EP processor technology, 6 Thin Node Islands based on Intel Haswell-EP processor technology and one Fat Node Island based on Intel Westmere-EX processor technology. Each Island contains more than 8,192 cores. All compute nodes within an individual Island are connected via a fully non-blocking Infiniband Fat-Tree network. In addition, SuperMIC, a cluster of 32 Intel Ivy Bridge-EP nodes each having two Intel Xeon Phi accelerator cards installed, is also part of the SuperMUC system. For parallel I/O, SuperMUC provides 10 PByte of storage with the GPFS file system from IBM. A similar configuration, with a full FAT-Tree network rather than island-based, is implemented on the IBM **MareNostrum** Tier-0 supercomputer at BSC.

IBM Blue Gene/Q supercomputers and IBM cluster architectures can be found in several Tier-1 computing centers connected to E-CAM (respectively, Turing and Ada at IDRIS in France, Blue Joule and Blue Wonder at the Hartree Center in Daresbury, and others).

It is important to note that IBM has sold recently all its X86 section to Lenovo, which will takeover the construction of the INTEL-based cluster architectures (see below). The Blue-Gene branch will be stopped too. IBM is now involved in the OpenPower foundation gathering among other companies IBM, NVIDIA and Mellanox to develop the Power 9.0 processor architecture with Nvidia Volta acceleration and Mellanox EDR Infiniband connections. This will most probably be one of the architectures chosen in the CORAL project in the United States in order to upscale two of the DOE (Department of Energy) supercomputers, in particular the top2 machine TITAN, to the 100-200 PFlops range in 2018. Several PRACE centers are presently in contact with IBM in order to test prototypes of OpenPower architectures for a panel of scientific applications (including atomic and molecular calculations).

### 3.2.2 LENOVO

As mentioned above, Lenovo has taken over the IBM x86 activity, and is offering versions of NextScale system that exploit the iDataPlex system of IBM. The phase 2 of SuperMuc is currently built under the Lenovo NextScale system. The next Tier-0 machine replacing Fermi at CINECA, Marconi, has been co-designed by Cineca on the Lenovo NeXtScale architecture. It is equipped with the new Intel processors (Broadwell and Xeon Phi KNL) and will reach a performance of about 20 PFlop/s, while maintaining a low energy consumption.

### 3.2.3 CRAY

CRAY provides the CRAY XC series, and currently the CRAY XC40 supercomputers. For example, The new Tier-0 **Hazel Hen** system at HLRS is powered by the latest Intel Xeon processor technologies and the CRAY Aries Interconnect technology leveraging the Dragonfly network topology. The installation encompasses 41 system cabinets hosting 7,712

compute notes with a total of 185,088 Intel Haswell E5-2680 v3 compute cores. Hazel Hen features 965 Terabyte of Main Memory and a total of 11 Petabyte of storage capacity spread over 32 additional cabinets hosting more than 8,300 disk drives which can be accessed with an input-/output rate of more than 350 Gigabyte per second. As of today **Hazel Hen** is listed $9^{th}$ among the Top-500 supercomputers.

The **Sisu** supercomputer installed at the CSC computing center, Finland, connected to our E-CAM consortium, is also a CRAY XC40 supercomputer, ranking $38^{th}$ among the Top-500 (https://research.csc.fi/sisu-supercomputer)

#### 3.2.4 BULL (Atos technologies)

The Bull company (acquired in 2014 by ATOS) provides the Bullx supercomputer clusters based on the Intel Xeon processors and co-processors such as Xeon Phi or NVIDIA GPU. It offers original solutions for interconnect (such as the BXI system), power, and cooling. The Tier-0 **CURIE** is a BULL x86 system based on a modular and balanced architecture of thin (5040 blades, each with 2 sockets based on the latest Intel SandyBridge processor), large (90 servers, each with 128 cores and 512 GB of memory) and hybrid (144 blades, each with 288 nVIDIA M2090 GPUs) nodes with more than 360 TB of distributed memory and 15 PB of shared disk. Altogether, CURIE delivers a peak performance of 2 Petaflop/s (2 million billion operations a second).

Bull is now proposing the Sequana supercomputers with major improvements in terms of energy consumption and cooling and in particular the Bull sequana X1210 blade composed of 3 compute nodes each powered by an Intel Xeon Phi x200 processor (code-named Knights Landing). This MIC architecture is currently tested in France by GENCI (Grand Equipement National de Calcul Intensif) for scientific applications, including an electronic structure code and a molecular dynamics code.

#### 3.2.5 SGI

SGI company proposes the ICE supercomputer series. For example the Pangea supercomputer of the Energy giant Total is a SGI ICE X supercomputer that delivers up to 6.7 petaflops with expanded storage of 26 petabytes. SGI says that it is the largest supercomputer used by industry and it was ranked among the top ten of the Top500 list in 2015. There is currently no SGI Tier-0 machine, but there are several of them at a Tier-1 level accessible by the E-CAM community, including the **Fionn** supercomputer at ICHEC.

Fionn (fionn.ichec.ie) is a heterogenous machine made up of four components: Thin, Hybrid, Fat and Service. The Thin component is an SGI ICE X system of 320 nodes or 7680 cores made of 2.4GHz Intel Ivy Bridge cores. Each node has 2x12 core processors, 64GB of RAM and is connected using FDR InfiniBand. This amounts to 20TB of RAM across the partition. The Hybrid partition contains 32 nodes. Each of these nodes has 2x10 core 2.2GHz Intel Ivy Bridge with 64GB of RAM. This partition has accelerators from Intel and NVIDIA. 16 nodes have 32 Intel Xeon Phi 5110P's while the other 16 have 32 NVIDIA K20X's. The Fat section is an SGI UV2000 where 1.7TB of RAM is accessible to 112 cores of Intel Sandy Bridge (14x8 cores processors) and 2 Intel Xeon Phi 5110P's.

#### 3.2.6 other vendors

Again it is beyond the scope of this report to cover all the supercomputer vendors and the review above is limited to those appearing in the PRACE infrastructure or in the national computing centers involved directly in E-CAM (IDRIS, ICHEC, STFC, CSC). Other major vendors (HP, Dell, Fujitsu, ShenWei with its Top1 machine TaihuLight since June 2016 ...) are indeed present at a European and worldwide level. The readers are referred to the Top-500 list, listed by vendors (http://www.top500.org/statistics/list/).

### 3.3 Accessing resources

PRACE, as well as in most Tier-1 computer centres typically allocate computer time allocation through specific calls for scientific proposals that are reviewed and selected by a scientific panel. This is appropriate for scientific data production with well established codes. For the code development anticipated in E-CAM, the best strategy is the preparatory access mode (http://www.prace-ri.eu/prace-preparatory-access/). Such a rapid and flexible procedure for test calculations exists in most computer centres.

## 3.4 Emerging multi-/many-core platforms

Each subsection will cover an emerging multi/many-core platform that E-CAM should be aware of.

### 3.4.1 Intel Xeon

**Brief description**

Xeon is the brand name for Intel's continually evolving set of x86-based server and workstation processors. It is the high performance version of Intel's mainstream processor, i.e. with the same features as their high-end desktop processor, such as large L3 cache and hyperthreading, but also a number of extra features enabled:

- supports ECC memory. ECC (Error Correction Code) RAM is popular in servers or other systems with high-value data, as it protects against data corruption by automatically detecting and correcting memory errors. (ECC RAM has an additional memory chip which acts as both error detection and correction for the other memory chips on standard RAM);
- works in multiple CPU systems. Many Xeons possess added on-chip logic to facilitate communication between CPUs so that they can share access to memory and coordinate workloads. Each CPU in such a configuration has it's own memory controller and a set of memory modules, plus its processing cores.

Since 2007, Intel's so-called "tic-toc" model, has seen the process technology shrink from 65 to 14 nm and the microarchitecture change from "Nehalem" to "Sandybridge" and "Haswell". The latest evolution in the Xeon line-up, E5-2600 v4, uses the 14 nm "Broadwell" microarchitecture, features up to 22 cores (44 threads) per socket, supports 4 channels of DDR4 memory and 32 lanes of PCI Express 3, and has 2.5MB of L3 cache included in each core. Apart from the E5 type, for each microstructure, Intel also produces the E3 for high-end desktops and single socket servers, and the E7 used in high-end servers with four to eight sockets.

As of November 2015, 6 of the 10 largest supercomputers rely on Xeon or Xeon/Xeon-Phi hybrids, including the worlds No. 2 system, the Tianhe-2 at the National Super Computer Center in Guangzhou, China. The system share worldwide of various Xeon E5 processor generations (Ivybridge, Haswell, Sandybridge), is over 80 %. Most of the PRACE machines (CURIE, HAZEL HEN, Mare Nostrum, SuperMUC) also rely heavily on Intel Xeon processors.

**Key recommendations**

Over the next few years, Intel plans to continue the Xeon series, increasing the number of cores and memory channels within a new microarchitecture code-named "Skylake". This continuous evolution guarantees that the overwhelming amount of code currently running on Xeon type machines as well as the E-CAM software being developed for the Xeon architecture will remain sustainable for the foreseeable future. At the same time, it will be straightforward to improve code performance by exploiting the increasing levels of parallelization. This is unlike the Xeon-Phi where often you need to rewrite your code, utilizing all the cores and hardware threads along with the core vector units in order to get good performance.

### 3.4.2 Intel Xeon Phi

**Brief description**

The Intel Xeon Phi is the brand name for Intel's 'Many Integrated Core (MIC)' platform. At the time of writing, the Intel Xeon Phi product line so far includes two generations of architecture. The first generation Intel Xeon Phi platform, known as Knights Corner (KNC) was released on the market in 2013 as a "co-processor" in the form factor of a PCIe card. The KNC platform typically has 60 low frequency ( 1 GHz) cores (in-order), with 4 hardware threads per core. Each of the cores has a 512-bit wide Vector Processing Unit (VPU) (a KNC VPU can process 32 single precision and 16 double precision operations per clock cycle). Each core has 512kB of private Last-Level Cache (LLC), and is connected to the other cores via an on-chip bi-directional ring interconnect. The KNC platform delivers over 2 teraflops of single precision and more than 1 teraflop of double precision peak performance.

The 2nd Generation Intel Xeon Phi platform, known as Knights Landing (KNL) is expected to be released on the market in Q2 of 2016 and will come in several form factors including, as an on-socket, self bootable chip. With up to 72 cores (with a similar clock frequency to KNC), the KNL platform is expected to deliver more than 6 teraflops of single precision and more than 3 teraflops of double precision peak performance. The Knights Landing chip is the first of Intel's Xeon family that will support AVX512 instructions and will have two AVX512 VPUs per core.

The self-bootable KNL platform will have 16 GB of MCDRAM high bandwidth memory on the package and DDR4 memory controllers to link out to a maximum of 384 GB of regular DRAM memory. The chip will have two PCI-Express

3.0 x16 ports and one x4 root port as well as a southbridge chipset to link to various I/O devices. The Omni-Path interconnect will not be on the chip itself, but will be on the package, with each PCI-Express x16 port having its own bi-directional, 100 Gb/sec ports. Each port is expected to deliver 25 GB/sec of bandwidth in both directions.

The KNL on-chip interconnect will be a 2D mesh (as opposed to the ring on KNC) and will operate in three modes: all-to-all, quadrant, and sub-NUMA. In all-to-all mode, addresses will be uniformly hashed across all distributed cache directories. This will be the most general mode with the easiest programming model, but it will offer lower performance than the other modes. In quadrant mode, the KNL chip can be divided into quarters and addresses will be hashed to directories in the same quadrant, providing lower latency and higher bandwidth for the cores running in each quadrant. In sub-NUMA mode, the operating system will expose all four quadrants as virtual NUMA clusters, which will make the KNL chip look like a four-socket Xeon server. This mode will provide the lowest latency, but applications have to be NUMA-aware and care will probably need to to be taken with regard to memory pinning to get maximum performance.

**Key Recommendations**

One of the main benefits of working with Intel Xeon Phi (co) processors is that they use common x86 languages, models, and familiar and standard development tools (e.g. C/C++, Fortran, MPI, OpenMP), so there is no need to learn new languages or tools and performance focused improvements are fully portable (but not necessarily performance portable).

It is expected that the KNL platform will offer significant performance gains over two-socket Intel Xeon Haswell and Intel Xeon Broadwell platforms, where much of this performance boost is expected to come as a result of the high bandwidth on package memory on the KNL platform. This high bandwidth MCDRAM memory will be of particular interest to E-CAM developers working with memory-bandwidth bound workloads. As a result, training on the effective utilization of the MCDRAM memory on KNL should be a key focus for the ESDWs.

In terms of compute-bound workloads, of which there are (and will be) many for ECAM, effective vectorisation (i.e, exploitation of the 512bit VPUs) will be paramount in order to exploit the Intel Xeon Phi to its full potential. This is much more so the case than for Intel Xeon platforms, due to the lower clock frequency per core on the Intel Xeon Phi. With this in mind, a key focus within the ESDWs should be placed on teaching and developing techniques for effective vectorisation, of which there has not been enough focus to date. As far as the community codes referred to in the introduction, many now (e.g, LAMMPS, NAMD, DL-POLY 4 and AMBER) have been ported and optimised for Intel Xeon Phi KNC to date, with much of the focus of optimisation work on exploiting the 512 bit VPUs and preparing for AVX512. The same level of focus will need to be applied to the efficient vectorisation of E-CAM developed software modules and libraries.

In summary, the Intel Xeon Phi has already garnered a huge amount of interest with the scientific computing community due to the comparable floating performance promises it offers relative to GPUs (certainly when considering double precision), but by way of familiar programming models and ecosystem rather than proprietary frameworks, such as CUDA, or lower-level frameworks, such as OpenCL. However, it should be kept in mind, that achieving optimal performance on the Intel Xeon Phi can be very challenging, and more often than not requires a significant refactoring of legacy code, reflecting a more “revolutionary over evolutionary” approach to code modernization.

### 3.4.3 NVIDIA Tesla

**Brief description**

As of 2012, Nvidia Teslas power some of the world’s fastest supercomputers, including Titan at Oak Ridge National Laboratory and Tianhe-1A, in Tianjin, China. With the advent of initiatives such as the OpenPower Foundation (see Section 3.4.4), it is very likely that the accelerating co-processing model required for GPGPUs is set to continue. While iterations of the architecture are already available in multiple sites across Europe, we focus here on the latest iteration, the Pascal P100 since this is the hardware that will appear in systems once they become available in Q4 2016.

The notable hardware improvements of the architecture of NVIDIA Pascal P100 are:

- In Pascal, an SM (streaming multiprocessor) consists of 64 CUDA cores. Maxwell packed 128, Kepler 192, Fermi 32 and Tesla only 8 CUDA cores into an SM; the GP100 SM is partitioned into two processing blocks, each having 32 single-precision CUDA Cores, an instruction buffer, a warp scheduler, 2 texture mapping units and 2 dispatch units.
- CUDA Compute Capability 6.0.
- High Bandwidth Memory 2 – some cards feature 16 GiB HBM2 in four stacks with a total of 4096bit bus with a memory bandwidth of 720 GB/s

- Unified memory – A memory architecture, where the CPU and GPU can access both main system memory and memory on the graphics card with the help of a technology called "Page Migration Engine".
- NVLink – A high-bandwidth bus between the CPU and GPU, and between multiple GPUs. Allows much higher transfer speeds than those achievable by using PCI Express; estimated to provide between 80 and 200 GB/s.
- 16-bit (FP16) floating-point operations (colloquially "half precision") can be executed at twice the rate of 32-bit floating-point operations ("single precision") and 64-bit floating-point operations (colloquially "double precision") executed at half the rate of 32-bit floating point operations.
- More registers - twice the amount of registers per CUDA core compared to Maxwell.

**Key Recommendations**

NVIDIA are continuously trying to decrease the programming effort for GPUs. Of the characteristics listed above, that of most significance is perhaps the "Unified Memory". CUDA 6 introduced Unified Memory, where managed memory is accessible to both the CPU and GPU using a single pointer. The CUDA system software automatically migrates data allocated in Unified Memory between GPU and CPU, so that it looks like CPU memory to code running on the CPU, and like GPU memory to code running on the GPU. Expanding on the benefits of CUDA 6 Unified Memory, Pascal GP100 adds support for large address spaces and page faulting capability, further simplify programming and sharing of memory between CPU and GPU.

Combined with the system-wide virtual address space, page faulting provides several benefits. First, page faulting means that the CUDA system software doesn't need to synchronize all managed memory allocations to the GPU before each kernel launch. If a kernel running on the GPU accesses a page that is not resident in its memory, it faults, allowing the page to be automatically migrated to the GPU memory on-demand. Alternatively, the page may be mapped into the GPU address space for access over the PCIe or NVLink interconnects (mapping on access can sometimes be faster than migration). Note that Unified Memory is system-wide: GPUs (and CPUs) can fault on and migrate memory pages either from CPU memory or from the memory of other GPUs in the system. With the new page fault mechanism, global data coherency is guaranteed with Unified Memory. This means that with GP100, the CPUs and GPUs can access Unified Memory allocations simultaneously. This was illegal on Kepler and Maxwell GPUs, because coherence could not be guaranteed if the CPU accessed a Unified Memory allocation while a GPU kernel was active. Note, as with any parallel application, *developers need to ensure correct synchronization* to avoid data hazards between processors.

Finally, on operating systems that support this feature, memory allocated with the default OS allocator (e.g. 'malloc' or 'new') can be accessed from both GPU code and CPU code using the same pointer. On these systems, Unified Memory is the default: there is no need to use a special allocator or for the creation of a special managed memory pool, greatly increasing programmability.

### 3.4.4 OpenPower

**Brief description**

The OpenPOWER Foundation (http://openpowerfoundation.org/) was founded in 2013 as an open technical membership organization that will enable data centres to rethink their approach to technology. Member companies are enabled to customize POWER CPU processors and system platforms for optimization and innovation for their business needs. These innovations include custom systems for large or warehouse scale data centres, workload acceleration through GPU, FPGA or advanced I/O, platform optimization for SW appliances, or advanced hardware technology exploitation. To date, the OpenPOWER foundation has over 150 members. Nvidia, Mellanox Technologies, and IBM are the key technology partners in the OpenPower Foundation, which propose an HPC hybrid computing platform to run traditional HPC and parallel analytics workloads alike and, importantly, offer an alternative to Intel's Xeon and Xeon Phi compute.

The POWER8 hardware characteristics are as follows. At the heart of the system is the 22nm SOI POWER8 processor. It is a 12-core processor, each core with 64 KB L1 data, 32 KB L1 instruction caches, 512 KB of SRAM L2 cache on a 64-byte wide bus, and 8 MB of L3 eDRAM cache. Thus, a POWER8 processor would have a total of 96 MB of L3 eDRAM cache. The chip can also utilize up to 128 MB of off-chip eDRAM L4 cache. The on-chip memory controllers can handle 1 TB of RAM and 230 GB/s sustained memory bandwidth. The cores are designed to handle clock rates between 2.5 and 4.5 GHz. Each core can issue ten instructions and dispatch eight during each cycle to 16 execution pipes: two fixed-point pipelines, two load/store pipelines, two load pipelines, four double-precision floating-point pipelines which can also act as eight single-precision floating-point pipelines, two VMX pipelines, one cryptographic pipeline, one decimal floating-point pipeline, one condition register pipeline, and one branch execution pipeline. Each core is eight-way hardware multithreaded and can be dynamically and automatically partitioned to have either

one, two, four or all eight threads active. POWER8 also provides added support for hardware transactional memory. In addition to the Integrated PCIe Gen3, the chip benefits from the Coherent Accelerator Processor Interface (CAPI) technology. CAPI is a direct link into the CPU and allows peripherals and coprocessors to communicate directly with the CPU, substantially bypassing the operating system and driver overheads. IBM has developed CAPI to be open to third-party vendors and even offers design enablement kits. CAPI can be used to attach accelerators, such as FPGAs, directly to the POWER8 CPU for significant workload-specific performance boosts.

The IBM Firestone machine is a heterogeneous architecture with a 2-socket server and two Nvidia dual-GPU Tesla K80 coprocessors embedded in the system. NVIDIA's high-speed PCIe replacement, NVLink is used to couple CPU and GPU. It is intended to allow faster, lower latency, and lower energy communication between processors and accelerators. This technology should dramatically improve (up to 80 GB/s bidirectional) the limited bandwidth of the PCIe interface of today's servers. With NVLink in place for CPU-GPU communications, this architecture will be able to offer unified memory support which should enable optimal performance for tasks that require frequent CPU/GPU interaction while simplifying the burden of developers to manage data movement.

This architecture will first evolve in mid-2016 with the Garrison architecture, an update of the firestone with a POWER8+ processor coupled with NVLink connexions to four NVIDIA Pascal GPU (up to 160 GB/s bidirectional). The unified memory will be enhanced with a page migration engine.

Then, a second evolution will lead to the machines targeted for the CORAL procurement, i.e. the SUMMIT and SIERRA machines. Both systems will be of similar design. Powering the systems will be a triumvirate of technologies: IBM POWER9 CPUs, NVIDIA Volta-based Tesla GPUs with NVLink2, and Mellanox EDR Infiniband for the system interconnect. NVLink 2.0 will introduce cache coherency, which will allow for further performance improvements and the ability to more readily execute programs in a heterogeneous manner.

**Key Recommendations**

Developing and refactoring applications to improve performance portability on accelerated architectures is not an easy task. Many different frameworks are available for this purpose (OpenMP, OpenACC, CUDA, OpenCL ...). Nevertheless, some general guidelines to create applications that explore performance portability and exploit untapped parallelism should be useful for E-CAM developers:

a. Use accelerated programming libraries whenever possible.

b. Prefer high-level compiler directives such as OpenMP/OpenACC over low-level frameworks such as CUDA or OpenCL.

c. Expose as much node-level parallelism as possible.

d. Rely on a suite of development tools to maximize parallelism.

Hybrid programming, mixing MPI (for inter-node communication) and OpenMP (for intra-node vectorisation, GPU offloading and CPU multi-threading) should be the preferred model of development for HPC OpenPOWER heterogeneous computing.

It should be noted the Firestone architecture is presently tested for applications in the E-CAM scope (in particular at Maison de la Simulation and IDRIS); the recommendations are at this stage rather generic for such hybrid architectures.

## 3.5 Emerging memory/storage technologies

**Brief description**

The memory/storage hierarchy can be classified by access times, persistence, and storage capacity. Some of the fastest, most expensive memory is static random access memory (SRAM), which is used for computer cache memory (L2), i.e. integrated on the processor die and providing access times as low as 10 nanoseconds. Dynamic random access memory (DRAM), unlike SRAM, is volatile and must be continually refreshed in order to maintain the data. Also DRAMs access time is around 60 nanoseconds, i.e. much slower than SRAM, but it is cheaper to manufacture and constitutes the main memory in most computers. Over the last 60 years, hard disk drives (HDDs) have seen their areal recording densities increase from 0.002 Mbit/in2 to larger than 800 Gbit/in2, including the change from longitudinal to perpendicular recording that took achievable densities beyond the 200 Gbit/in2 limit for longitudinal recording. Concurrently the cost per byte of storage has been exponentially decreasing. Eventually however even the perpendicular recording will reach a limit at densities of around 1.5 Tb/in2. To go beyond this, new technologies, such as heat-assisted magnetic recording (HAMR) have been proposed, which would allow writing on a much smaller scale. As of 2016, hard disks using HAMR are not on the market, but they are in an advanced state of development with demonstration drives produced by companies such as Seagate. The largest HHD these days have capacities of

up to 10 Tb, compared to for example DRAM modules that can have capacities up to 128 Gb (Samsung TSV DDR4 RDIMM). HHDs are much slower to access (larger latency by a factor of 105) but cost up to 100 times less than DRAM. Flash memory is increasingly bridging the gap between these two extremes. NAND Flash Memory is an electronic, solid-state, storage medium that can be electrically erased and rewritten. Its advantages are

- non-volatile storage of data (unlike DRAM memory which must be powered continuously to retain data)
- costs have dropped sufficiently to make new primary storage devices, solid-state drives (SSDs), possible for client systems and servers.
- allows the storing and retrieval of data more quickly than with conventional rotating disk storage (HDDs have access latencies in milliseconds, while SSDs operate in hundreds of microseconds.
- no moving parts

Recently Intel and Micron have announced a new mas-storage class non-volatile memory technology named 3D XPoint, which though slower than DRAM, will be cheaper to produce. It will be more expensive but with 100 times lower latency, and 1000 times more write cycles than NAND.

For more permanent storage solutions, magnetic tape storage, which packages data in cartridges and tapes is still going strong and remains one of the least expensive solutions for long-term archiving. Thus Sony in 2014 announced they successfully developed magnetic tape technology achieving an areal recording density of 23 Gbit/cm2, making it possible to record more than 185 Terabytes of data per data cartridge.

Cloud storage: Improvements in internet bandwidth and the falling cost of storage capacity means it's frequently more economical to outsource data to the cloud, rather than buying, maintaining and replacing hardware.

For more details, see Eleftheriou et al., IEEE Data Eng. Bull., 33(4), 4-13, IEEE, 2010

**Key recommendations**

With CPU capacity increasing exponentially due to improved architecture and number of cores, and the increasing areal recording densities leading to growing HDD storage capacities, the bottleneck in HPC lies increasingly with the latency in data access limiting the performance that could otherwise be achieved. The way forward is to work with tiered storage, using the most expensive storage solutions for applications that require faster access to the data. This implies using the fastest most expensive, solid-state storage in the highest tier, also referred to as Tier-0. Whilst early on Tier-0 consisted of RAM disk, NAND flash memory plays an increasingly important role. Tier-1 usually consists of high speed SAS HHDs. With respect to ECAM software tuning, access to Tier-0 machines such as the PRACE (CURIE. FERMI, SuperMUC) will be crucial.

## 3.6 Preparing E-CAM for Future European Exascale Systems

It is widely expected that the exascale systems of the future will be qualitatively different from current and past computer systems. They will be built using massive multi-core processors with hundreds of cores per chip, their performance will be driven by parallelism, constrained by energy, and with all of their parts, will be subject to frequent faults and failures [7].

While the focus of E-CAM is on developing software for current multi-petascale systems, this does not to mean that E-CAM should ignore the challenges that are already confronting software on the road to exascale. While there still may not be a general consensus on what an exascale machine in the future will look like, it is becoming increasingly likely that it will share some of the characteristics of the current No.1 systems in both the Top500 and Green500 lists. These are characteristics that E-CAM knows and can currently target. As to preparing for future exascale system characteristics that are not immediately clear or fully evaluated, E-CAM will need to engage more strongly with various European exascale projects and initiatives throughout its lifetime.

### 3.6.1 Report on the Extreme-Scale Demonstrators (ESDs) Workshop

With this challenge in mind, in M8, E-CAM participated at the EXDCI organized Extreme-Scale Demonstrators (ESDs) Workshop in Prague as part of the European HPC Summit Week. The ESDs will be HPC system prototypes built out of the technologies produced by the European HPC technology projects (and other technologies available in the market, if needed). The objective of the ESD initiative is to verify the preparedness of the technologies available for the delivery of complete and competitive European HPC systems on the road to exascale. The current Strategic Research Agenda (SRA) of ETP4HPC (available at: http://www.etp4hpc.eu/sra/).

The workshop was attended by representatives of FETHPC and other European HPC Technology projects, Centres of Excellence in Computing Applications and PRACE, which entailed all of European HPC Eco-system stakeholders, i.e. the developers and providers of HPC technology (including potential system integrators of the EsDs), application communities, infrastructure providers and a representative of the Commission in a discussion on the following topics in relation to the EsDs:

The ESD workshop in Prague was focused on the following topics:

a. the main technical EsD characteristics and performance targets;

b. defining a suite of challenging pre-exascale problems to validate the EsDs instances, together with examples of possible applications;

c. the options for funding and project implementation;

d. Identifying potential parties interested in participating to the EsD calls.

With regards to these topics, a, b, and d are of immediate relevance to E-CAM, and are discussed further in the key recommendations section below. During the ESD workshop, E-CAM was provided with the opportunity to discuss the centre's anticipated compute requirements as well as to gain insight into the possible design-space of ESD systems.

**Key Recommendations**

It is clear that the ESD initiative (and associated EXDCI workshops) represents an opportunity for E-CAM to feed requirements into next-generation extreme-scale platforms. While several possible hardware design characteristics of the future ESDs were discussed at the EXDCI workshop in Prague, the hardware and system designs of each will only be determined during the proposal phase of a future associated call for proposals. However, it has been noted that several of the already established HPC Technology Project system designs have included compute architectures that are not currently listed in Table 1. For example, many energy efficient system designs as investigated by current and previous HPC Technology Projects feature both ARM processors and FPGAs, which have not typically been targeted by the scientific computing community to date, and which are yet to be fully validated as viable platforms for floating point intensive HPC workloads (this is the case at the time of writing, but may change soon). It is the point of view of E-CAM that while it is currently too early for E-CAM developers to target these more novel platforms, they should still be considered during the lifetime of the centre and even more so, if these architectures feature as part of the ESD designs (and are suitable for workloads as represented by E-CAM)

With this view in mind, it is vital that E-CAM liaise closely with the ESD projects as well as the current FETHPC projects, not only so that it can prepare for emerging European prototype systems, but also to have a direct influence on the design of such systems. To ensure that this engagement continues and strengthens, a key recommendation is that E-CAM aims to have a presence and shape discussions at future EXDCI/ETP4HPC/PRACE joint workshops, with a particular focus on the ESD initiative.

### 3.6.2 The DEEP, DEEP-ER and Mont-Blanc projects

Concerning European initiatives to the exascale that the E-CAM community should be aware of, one cannot not mention also two other European projects: 1) the DEEP and DEEP-ER projects (http://www.deep-er.eu), developing an innovative architecture for heterogeneous HPC systems on the combination of a standard HPC Cluster and a tightly connected HPC Booster built of many- core processors, and presently evolving this architecture to address two significant Exascale computing challenges: highly scalable and efficient parallel I/O and system resiliency, 2) the Mont-Blanc project (http://www.montblanc-project.eu), aiming at designing a new type of computer architecture capable of setting future global HPC standards, built from energy efficient solutions used in embedded and mobile devices.

# 4 State-of-the-art Tools and Techniques

Here we highlight the state-of-the-art tools and techniques for coding on HPC architecture that the E-CAM community should be aware of. All those tools and techniques were reviewed in a very complete public report during the third implementation phase of PRACE in 2013; see http://www.prace-ri.eu/IMG/pdf/d7.2.1.pdf.

## 4.1 Programming Models

During the initial surveying of E-CAM requirements (by way of HRQ), the following programming tools were found to be relevant for the development of high performance software libraries and modules within E-CAM:

- For inter-node parallelization, MPI is most widely used and is still regarded as the parallel programming model of choice by much of the scientific computing community. Currently, E-CAM has little experience with PGAS approaches such as Fortran CoArrays, UPC, Chapel or X10.
- For intra-node multithreading, OpenMP appears to be the model of choice for all codes of interest.
- For exploitation of many-core "accelerator" platforms other than Intel Xeon Phi (e.g., GPGPU) CUDA, OpenACC and OpenCL are all of interest to E-CAM software development teams.

For the above programming tools, openness (open standard), sustainability and portability are considered very important to E-CAM, with productivity considered somewhat less important. In the rest of this section, we provide a brief overview of each of these programming models, their current state (in terms of version and latest features) as well as key recommendations for E-CAM at this early stage of the project.

### 4.1.1 OpenMP

**Brief description**

OpenMP (Open Multi-Processing, http://www.openmp.org) is an application programming interface (API) that supports multi-platform shared memory multiprocessing programming in C, C++, and Fortran. It consists of a set of compiler directives, library routines, and environment variables to specify high-level parallelism and influence run-time behavior. OpenMP uses a portable, scalable model that gives programmers a simple and flexible interface for developing parallel applications for platforms ranging from the standard desktop computer to the supercomputer. Over time, the OpenMP specification has evolved to add new important functionalities to target new architectures, including the Intel 'Many Integrated Core (MIC)' platform and the heterogeneous GPU accelerated platform. The latest OpenMP4.5 specification was published in November 2015. In this release, the directives extend the C, C++ and Fortran-based languages with single program multiple data (SPMD) constructs, tasking constructs, device constructs (for accelerators), worksharing constructs and synchronization constructs, and they provide support for sharing, mapping and privatizing data. Compilers which support the OpenMP API (complete list: http://openmp.org/wp/openmp-compilers/) often include a command-line option to that activates and allows interpretation of all OpenMP directives.

Initially, OpenMP targeted only SMP architecture (shared memory node). Developers can use directives to annotate their codes to create a team of threads and distribute work among them. It is the responsibility of the user to deal with the data-sharing attribute of variables, SHARED or PRIVATE, and to synchronize threads so that the semantics of the sequential code is preserved. In May 2008, OpenMP3.0 was released whose principal new feature is the concept of tasks and the task construct, which significantly broadens the scope of OpenMP beyond the parallel loop constructs. For example, link lists or recursive algorithms that used to raise very difficult issues, can now be simply and efficiently parallelized with tasking. Version 4.0 of the specification was released in July 2013 and version 4.5 in November 2015. These versions add or improve the following features:

- Support for accelerators - OpenMP 4.0 API provides mechanisms to describe regions of code where data and/or computation should be moved to another computing device (offloading). OpenMP 4.5 improves support for asynchronous device execution.
- Support for SIMD vectorization - OpenMP 4.0 API provides mechanisms to describe when multiple iterations of the loop can be executed concurrently using SIMD instructions and to describe how to create versions of functions that can be invoked across SIMD lanes. This functionality is of primary importance as it allows a user to manage vectorization without any proprietary directives.
- Thread affinity - OpenMP 4.0 API provides mechanisms to define where to execute OpenMP threads. The advantages for the user are improved locality, less false sharing and more memory bandwidth. In OpenMP 4.5, query functions for OpenMP thread affinity were added.
- Tasking extensions - OpenMP 4.0 API provides several extensions to its task-based parallelism support. Tasks can be grouped to support deep task synchronization and task groups can be aborted to reflect completion of cooperative tasking activities such as search. Task-to-task synchronization is now supported through the specification of task dependency. OpenMP 4.5 introduces support for nestable parallel loops that create OpenMP tasks.
- Many other features: Error handling, Support for Fortran 2003, User-defined reductions, ...

**Key Recommendations**

OpenMP is an open standard, portable across many types of architectures. Parallelization with OpenMP is relatively easy to implement, even when starting from a sequential program and can be done incrementally. Good scalability can be obtained, if the code is carefully parallelized. Efficient and portable vectorization can be achieved regardless of the targeted architecture (Xeon, KNL, Power8, ...). Moreover, heterogeneous architectures (i.e. OpenPOWER) can now be efficiently used (CPU and accelerator at the same time) within OpenMP. Debugging OpenMP application is not an easy task even with specialized tools and special attention must be payed to the definition of the data-sharing attribute of used variables in a construct. Finally, hybrid programming mixing MPI (for inter-node communication) and OpenMP (for intra-node vectorization, GPU offloading and CPU multi-threading) is of paramount importance and should be considered as the preferred model of development for the new HPC pre-exascale architectures to come.

### 4.1.2 MPI

**Brief description**

MPI is still the most widely employed parallel programming model within E-CAM and the wider European HPC community. The latest MPI 3 standard (recently updated to v3.1) was published in September 2012 [11] and is widely viewed as a major update, positioning MPI for the deep-petascale and future exascale era.

In brief, MPI 3.0 offers the following new features to MPI

- Non-blocking and neighbourhood collective operations
- Revamped remote memory access (RMA, a.k.a. "one-sided" operations)
- New Fortran 2008 bindings
- Richer external tool support
- Better support for large counts
- "Matched" probe support
- C const correctness
- Shared memory windows
- Non-blocking communicator creation / duplication
- Countless small grammar fixes, textual cleanups, and clarifications

To our knowledge, the majority of existing implementations of MPI now offer the full set of MPI 3.0 features.

The MPI Forum added support for one-sided communication (also known as remote memory access, or RMA) in version 2.0 of the MPI standard, to function alongside MPI's traditional two-sided and collective communication models. While MPI 2 was effective for a variety of applications and systems, it has lacked various communication and synchronization features, and its conservative memory model has limited its ability to efficiently utilize hardware capabilities, such as cache coherence.

The MPI 3 standard now adds a variety of new atomic operations, synchronization primitives, window types, and a new memory model that better exposes the capabilities of architectures with coherent memory subsystems. It is believed that these features will address issues in the MPI 2 model and greatly improve the performance potential of MPI RMA.

Other important new functionality includes non-blocking collective communication and better handling of situations arising out of process failures, e.g., an application may choose to be notified when an error occurs anywhere in the system and an application may ignore failures that do not impact its MPI requests. There is also the capability to rebuild a communicator when a process fails or allowing it to continue in a degraded state. MPI processes may also ignore failures that do not impact its MPI requests. An important new feature is that an application that does not use collective operations will not require collective recovery.

**Key Recommendations**

While MPI is widely employed by members of E-CAM and by the developers of many of the community codes mentioned in the Introduction, its exploitation has typically been restricted to MPI 1, with little evidence of modern MPI 3 exploitation to date. This will need to change quickly if E-CAM is to position its software modules for the efficient

exploitation of emerging extreme-scale systems, particularly with regard to next generation network fabrics. With better hardware and middleware support emerging for one-sided RMA communications (e.g., Intel OmniPath) as well as the increasing limitations on near-memory capacity, it is paramount that E-CAM develops MPI-based software that can leverage these hardware developments to achieve optimal performance of its software on extreme scale systems. In particular, E-CAM software should be designed with communication hiding algorithms as much possible. For example, with FFT methods central to many current and future workloads in E-CAM, modern designs of FFT workflows that exploit new non-blocking collective communications available in MPI 3 should be a key focus for E-CAM developers.

As well as this, many algorithms that would typically employ nearest-neighbour point-to-point communications are now better redesigned to exploit nearest neighbour collective routines that allow for the MPI implementation to optimize for the communication pattern more efficiently (with the added benefit of reduced code complexity, which will improve maintenance overhead in the long term). Finally, with increasing investigations into the benefits of using the MPI shared memory windows over the MPI-plus-OpenMP hybrid model, this newly evolving "MPI-plus-MPI" paradigm should be exploited where possible, both for the potential to decrease the memory footprint of E-CAM software as well as improve performance by way of shared memory interconnects (both considerations of which are particularly important when targeting Intel Xeon Phi-based sytems).

### 4.1.3 CUDA

**Brief description**

CUDA is a parallel computing platform and application programming interface (API) model created by NVIDIA. It allows software developers to use a CUDA-enabled graphics processing unit (GPU) for general purpose processing – an approach known as GPGPU. The CUDA platform is a software layer that gives direct access to the GPU's virtual instruction set and parallel computational elements, for the execution of compute kernels. The CUDA platform is designed to work with programming languages such as C, C++ and Fortran. This accessibility makes it easier for specialists in parallel programming to utilize GPU resources, as opposed to previous API solutions like Direct3D and OpenGL, which required advanced skills in graphics programming. Also, CUDA supports programming frameworks such as OpenACC (see Section 4.1.4) and OpenCL.

**Key Recommendations**

As discussed briefly in Section 3.4.3, each increase in the compute capability of hardware is related in a new version of CUDA, where these hardware features are integrated. The CUDA model itself is relatively low-level and NVIDIA increasingly pushes OpenACC (see Section 4.1.4) as the preferred approach to leveraging GPU co-processing. However, if low-level control and management is required, CUDA remains the only well-performing option (since development of the OpenCL standard has stalled).

### 4.1.4 OpenACC

**Brief description**

OpenACC (for Open Accelerators) is a programming standard for parallel computing developed by Cray, CAPS, Nvidia and PGI. OpenACC is a directive-based programming model designed to provide a simple yet powerful approach to accelerators without significant programming effort. Like in OpenMP, the programmer can annotate C, C++ and Fortran source code to identify the areas that should be accelerated using compiler directives and additional functions. Like OpenMP 4.0 and newer, code can be started on both the CPU and GPU. Version 2.5 of the specification was released in October 2015. Developers might mix OpenACC with CUDA C or CUDA Fortran, selecting whichever model performs best on a given operation. Furthermore, OpenACC interoperability improves programmer productivity by allowing the use of common GPU libraries, such as the cuBLAS, cuRAND, and cuFFT libraries that come standard with the NVIDIA CUDA Toolkit. In short, OpenACC's interoperability means that the developer can use whichever technology works best in a given situation.

**Key Recommendations**

Currently, there is a limited selection of compilers that support the standard, with the PGI compiler (owned by NVIDIA) being the most likely choice. OpenACC is an experimental feature of GCC 5.1 and may not meet the needs of general application development. Compared to GCC 5, the GCC 6 release series includes a much improved implementation of the OpenACC 2.0a specification.

OpenACC is being heavily promoted by NVIDIA as the more accessible programming model for GPU capabilities. With "Unified Memory" (where the same memory pointer can be used by CPU and GPU) and OS support for OS allocations

to be accessed by the GPU, the need to explicitly allocate and move data to and from GPU memory is removed. The OpenACC model is then a very attractive model since the lines of code necessary for implementing GPU support (boilerplate code) is greatly reduced, and the performance is comparable to that of CUDA code.

#### 4.1.5 FPGAs languages

**Brief description**

Field Programmable Gate Arrays (FPGAs) are semiconductor devices that are based around a matrix of configurable logic blocks connected via programmable interconnects. FPGAs can be reprogrammed to desired application or functionality requirements after manufacturing. FPGAs are predominantly programmed using hardware description languages (HDLs), i.e. computer languages that are used to describe digital circuits and operations textually. The two most common HDLs are Verilog and VHDL. These languages operate both at the behavioral and structural levels, but VHDL is somewhat more suitable for describing hardware at high levels of abstraction whilst Verilog is better at describing hardware down to gate level. VHDL borrows many of its features from ADA, whilst the syntactical roots of Verilog are in C.

The aim of high-level synthesis (HLS) tools is to raise the level of abstraction at which the user can write a program to compile it down to VHDL or Verilog. According to [Bacon et al. "FPGA programming for the masses, Comm. of the ACM, vol. 56, no. 4, pp. 56–63, 2013.], HLS languages and tools can be classified into five categories:

- HDL-like languages: Verilog has had an influence on the development of SystemVerilog, extending the language to include a greater number of C and object-oriented concepts such as classes, and interfaces. Bluespec SystemVerilog (BSV) based on guarded rules and syntax inherited from SystemVerilog, is a more recently developed HDL. The language is primarily for synthesizing the gate design, not for verification.
- OpenCL: Open Computing Language (OpenCL) is an open industry standard for highly parallel programming across heterogeneous computing platforms consisting of CPU, GPU, DSP or FPGA. A number of compiler frameworks have been proposed to convert OpenCL code to VHDL/Verilog (eg. Altera OpenCL framework). OpenCL defines a set of core functionalities that are supported by all devices, optional functionalities that may only be implemented on high-function devices, and an extension mechanism for some vendor specific features.
- C-based frameworks: Most commercial high-level synthesis efforts have focused on C-based design (whether ANSI C, C++ or SystemC), with toolsets that provided for translation of functionality defined at the behavioral level into logic at a lower level of abstraction.
- High-level language-based frameworks: includes frameworks that translate highlevel languages other than C into hardware. The languages in this category are highly abstract, usually object-oriented, and offer high-level features such as polymorphism and automatic memory management.
- Model-based frameworks: These frameworks provide an abstract way of designing complex control- and signal-processing systems. Matlabs HDL Coder for example generates portable, synthesizable Verilog and VHDL code from MATLAB functions, such as Simulink, which is a graphical programming environment for modelling and analyzing dynamic systems. Labview from National Instruments similarly has a graphical environment for designing FPGA applications, as does Gedae providing among others hardware/software integration for heterogeneous systems with FPGAs and processors. Furthermore a number of open source modeling frameworks exist such as Umbrello and Eclipse.

**Key Recommendations** The complexity of programming with HDLs is the main bottleneck in the exploitation of FPGAs. HLS will enable programmers to use a high-level language such as C++ to automatically generate optimized implementations of algorithms to program a FPGA, and will result in a much broader adoption of this programming paradigm in line with increasing FPGA capacity.

See also for details: Arcas-Abella et al. 10.1109/FPL.2014.6927484

### 4.2 Profilers and debuggers

#### 4.2.1 Performance Optimisation

The Performance Optimisation and Productivity (POP) Centre of Excellence in Computing Applications (https://pop-coe.eu/) provides performance optimisation and productivity services for academic and industrial code(s) in all domains. Since their services are free of charge and the tools used by POP (https://pop-coe.eu/partners/tools) are both mature and state-of-the-art, we highlight some of them here.

**Brief description of core POP Tools**

The Jülich Supercomputing Centre develops Scalasca (http://www.scalasca.org/) and associated tools for scalable performance analysis of large-scale parallel applications that are deployed on most of the largest HPC systems including Blue Gene/Q, Cray, Fujitsu, K Computer, Stampede (Intel Xeon Phi), Tianhe, Titan and others. The Score-P (http://www.score-p.org/) instrumentation and measurement infrastructure, developed by a community including RWTH Aachen University, supports runtime summarisation and event trace collection for applications written in C, C++ and Fortran using MPI, OpenMP, Pthreads, SHMEM, CUDA and OpenCL. Scalasca trace analysis characterises parallel execution inefficiencies beyond those captured by call-path profiles. It detects wait states in communications and synchronisations, such as the "Late Sender" where receive operations are blocked waiting for associated sends to be initiated, and the root cause of these, such as excess computation or imbalance. Also contributions to the critical path of execution are quantified to highlight call-paths that are the best candidates for optimisation.

The Barcelona Supercomputing centre develops another core tool, Paraver (http://www.bsc.es/computer-sciences/performance-tools/paraver), a trace-based performance analyser with great flexibility to explore and extract information. Paraver provides two main visualisations: timelines that graphically display the evolution of the application over time, and tables (profiles and histograms) that provide statistical information. These two complementary views allow easy identification of computational inefficiencies such as load balancing issues, serialisations that limit scalability, cache and memory impact on the performance, and regions with generally low efficiency.

**Key Recommendations**

The POP Centre of Excellence provides specialised performance analysis services to European scientists and is currently the main contact point for the (potential) users of their tools. They will provide a performance audit on user applications, typically on-site where possible, with a relatively short turn-around. The application form for POP services is available on their website (https://pop-coe.eu/request-service-form).

The Virtual Institute - High Productivity Supercomputing (VI-HPS)(http://www.vi-hps.org/) provides a more exhaustive list of available tools in their VIHPS tool guide (http://www.vi-hps.org/upload/material/general/ToolsGuide.pdf.

#### 4.2.2 HPC debugging tools (TotalView, DDT)

Every HPC programmer has encountered a bug in his parallel program at least once. Debugging is not always easy in parallel applications when specific problems occur such as deadlocks, concurrent access to memory, asynchronous events and a greater propensity for race conditions. Putting "print" calls everywhere in a program becomes tedious quickly. Parallel debuggers can help the user in tracking bugs. These tools offer very powerful functionalities such as visualisation of variables in the code on each thread or process and synchronisation of the threads or processes near the location of the problem. Last but not least, debuggers can be attached to all the individual processes simultaneously when the code is running.

We review here two HPC parallel debuggers, TotalView and DDT. These two tools are commercial and distributed with licenses by Rogue Wave Software for TotalView (http://www.roguewave.com/products-services/totalview) and by Allinea for DDT (http://www.allinea.com/products/ddt). These two debuggers are usually installed on most supercomputers.

**Brief description of Totalview**

Totaview supports multi-threaded and large-scale applications, using the following programming models: MPI, OpenMP, hybrid application (MPI+OpenMP, MPI+CUDA), C, C++ and Fortran. It includes a number of features:

- During the execution, there is the possibility of putting breakpoints on all processes or a subset of processes. It is also possible to run the application only on a subset of threads or processes.
- When a deadlock problem occurs, the "Halt" function allows the user to see where each process and thread are stopped. A backtrace view displays the status of every process and thread, which makes it possible to view thousands of processes at once, and helps to identify stray processes.
- A variable value in each thread and process can be seen simultaneously using the "Across Threads/Processes" functionality, allowing the programmer to pinpoint a synchronization problem, incorrect shared variables or values when sending or receiving messages.
- In some cases, when the code blocks after a long running time, it is useful to attach the parallel application to TotalView. This way avoids replaying the application for debugging.
- The "Memory Scape" product is integrated in TotalView and makes it possible to monitor the allocation or deallocation of the application.

- The "Replay Engine" product makes it possible to go in reverse through a program, from the point where it crashed to the point where the problems started. However, when using a high number of processes, this becomes very difficult to use.

**Brief description of DDT**

DDT supports multi-threaded and large-scale applications, using the following programming models: MPI, OpenMP, hybrid application (MPI+OpenMP, MPI+CUDA), C, C++ and Fortran. This debugger has been configured to be integrated with a batch environment. It is useful if you want to debug with resources that are not available in interactive mode. It offers the following features:

- A friendly way of searching or viewing all the functions and subroutines of your application using the "Project Files" tab. Source files can be loaded by clicking on the object name.
- Breakpoints can be used for the group, one process or several threads. Note that there are conditional breakpoints for both TotalView and DDT. If the bug appears in a loop at a high-step value, it is possible to stop the whole program when this value is reached for all the threads and processes.
- The parallel "Stacks" view can also make it possible to see where the processes and threads are in case of problem. The "Steps threads together" can be used inside OpenMP parallel regions to synchronise the threads in the current process. The "Cross-Process Comparison" and the "Cross-Thread Comparison" windows can also be used to compare expressions across the different processes and threads.
- The "Memory Debugging Options" is integrated in DDT. This functionality can intercept memory exhaustion or memory leaks and track an invalid memory access.

**Key Recommendations**

The two debuggers offer almost the same functionalities and the user can find TotalView or DDT on most supercomputers. They can help the user a great deal in tracking bugs in his parallel applications.

However, they are not miracle tools! In HPC, the problem is often difficult to locate, especially on hybrid codes where shared and distributed memory models are used. Parallel debuggers can refine the area on which to focus.

## 4.3 I/O management techniques

It should be noted first that a current need in the E-CAM community is to define common input-output formats and to favour the inter-operability between different codes. This is part of WP2 and is currently achieved using I/O formats such as YAML I/O, UPF and ESCDF (http://esl.cecam.org/Category:I/O).

In the E-CAM HPC community, however, as well as in other communities that are more advanced in that respect (such as climate modeling), the volume of data that are produced, read and stored, or that have to be shared, is continuously increasing. It will become even more pronounced on the road to the exascale. The use of high-level I/O libraries with standardized formats, and moreover parallel protocols bridging the performance gap between file system and compute system, is therefore highly recommended.

The I/O libraries that are relevant for HPC applications have been reviewed in the PRACE-3IP survey on tools and techniques (http://www.prace-ri.eu/IMG/pdf/d7.2.1.pdf). Namely:

- HDF5
- PNetCDF
- XIOS
- ADIOS
- SIONlib

Among those HDF5 and NetCDF (and its parallel version pNetCDF) are the most widely used in various communities and they are well interfaced with high-quality visualization tools such as VisIt and ParaView. XIOS is conceived as an I/O server which uses dedicated nodes for writing and post-processing the outputs. It was developed by the climate community.

All the libraries above are yet marginally used in atomic and molecular modeling codes. For example codes such as ABINIT and LAMMPS use the HDF5 library. The molecular dynamics package AMBER includes an output option for PnetCDF, but this option seems to be little used.

It should be noted that XIOS is presently developed further within the Energy-Oriented Centre of excellence (EoCoE) for computing application (www.eocoe.eu) in order to extend it from the climate modeling codes to other scientific applications, including those of E-CAM.

Within EoCoE should be noted also the Parallel Data Interface (PDI) initiative, that is intended to enable decoupling simulation codes from the low-level gory details of Input/Outputs. It has been designed so as to enable use of the best suited library for any machine without having to change a single line of code or worry about the portability of the library used. It is made of three parts. 1) A very simple application programming interface where the user code just has to provide an identifier and a pointer to the data to make it known to PDI. 2) A plugin system that enables the re-use of existing libraries to handle the data exposed by the code, e.g. HDF5 for I/O, or any other library, with no additional line of code. 3) A configuration file system to glue 1) and 2) together and specify what data to pass to what library.

**Key recommendations**

With the increasing need in massive data storage, the E-CAM community should consider the recent development of I/O libraries for their applications. An eye should be kept on various European initiatives such as XIOS and PDI, carried out within the EoCoE Centre of Excellence.

# 5 Conclusions

In this survey we have reviewed the massively parallel architectures that are available for the E-CAM community on the Tier-0 or Tier-1 supercomputers, as well as the new computing platforms that are emerging on those machines. We have provided a set of recommendations on the state-of-the art software that should be used by the E-CAM developers for programming, profiling, debugging, and using efficient Input/output in order to take full advantage of the present and next HPC machines.

The continuous evolution of the Xeon family by INTEL guarantees that the overwhelming amount of code currently running on Xeon type machines as well as the E-CAM software being developed for the Xeon architecture will remain sustainable for the foreseeable future. it will remain straightforward to improve the code performance by exploiting the increasing levels of parallelization using standards like OpenMP and MPI and the HPC development tools that have been recommended above. On the other hand, the route to the exascale at a sustainable energy consumption implies nowadays a major revolution (rather than evolution) in the computer architectures, with the advent of the hybrid, MIC or FPGA's platforms described in Table 1 and section 3.4.

As was noted in the section 3.6 devoted to future exascale architectures, several of the already established HPC Technology Project system designs have included compute architectures that are not currently listed in Table 1, featuring both ARM processors and FPGAs, which have not typically been targeted by the scientific computing community to date. While it is currently too early for E-CAM developers to target these more novel platforms, they should still be considered during the lifetime of the centre. A key recommendation is that E-CAM has a presence and shape discussions at future EXDCI/ETP4HPC/PRACE joint workshops, with a particular focus on the ESD (Extreme Scale Demonstrator) initiative.

Staying with the platforms targeted in this report and listed in table 1, the important message to be caught by the E-CAM community is that these new architectures rely on new coding paradigms (such as OpenACC), that we have reviewed in section 4.1), and require **a major effort in the software architecture of the codes and their data structure, to which the E-CAM developer community should be prepared and proactive in order to catch the exascale challenge**.

Such effort has started for a certain number of E-CAM community codes that have been ported and optimised for MIC architectures (Intel Xeon Phi KNC to date) or GPU-based hybrid architectures. The modules produced by E-CAM should at least reach the same standards.

In order to complement the present deliverable and to make sure that the E-CAM pool of developers is prepared for the exascale challenge for their applications, we propose to organise a CECAM HPC workshop in Q1-2017. This workshop will gather, as an audience, the pool of E-CAM developers coming from the various application WPs, and, as speakers, HPC vendors and HPC specialists presenting the emerging platforms and their programming paradigms.