Hardware developments V

E-CAM Deliverable 7.9
Deliverable Type: Report
Delivered in July, 2020

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

Project and Deliverable Information

Project Title	E-CAM: An e-infrastructure for software, training and discussion in simulation and modelling
Project Ref.	Grant Agreement 676531
Project Website	https://www.e-cam2020.eu
EC Project Officer	Juan Pelegrín
Deliverable ID	D7.9
Deliverable Nature	Report
Dissemination Level	Public
Contractual Date of Delivery	Project Month 54(31st March, 2020)
Actual Date of Delivery	6th July, 2020
Description of Deliverable	Update on "Hardware Developments IV" (Deliverable 7.7) which covers: - Report on hardware developments that will affect the scientific areas of interest to E-CAM and detailed feedback to the project software developers (STFC); - discussion of project software needs with hardware and software vendors, completion of survey of what is already available for particular hardware platforms (FR-IDF); and, - detailed output from direct face-to-face session between the project end-users, developers and hardware vendors (ICHEC).

Document Control Information

Document	Title:	Hardware developments V
	ID:	D7.9
	Version:	As of July, 2020
	Status:	Accepted by WP leader
	Available at:	https://www.e-cam2020.eu/deliverables
	Document history:	Internal Project Management Link
Review	Review Status:	Reviewed
Authorship	Written by:	Alan Ó Cais(JSC)
	Contributors:	Christopher Werner (ICHEC), Simon Wong (ICHEC), Padraig Ó Conbhuí (ICHEC), Alan Ó Cais (JSC), Jony Castagna (STFC), Godehard Sutmann (JSC)
	Reviewed by:	Luke Drury (NUID UCD) and Jony Castagna (STFC)
	Approved by:	Godehard Sutmann (JSC)

Document Keywords

Keywords:	E-CAM, HPC, Hardware, CECAM, Materials

6th July, 2020

***Disclaimer**:This deliverable has been prepared by the responsible Work Package of the Project in accordance with the Consortium Agreement and the Grant Agreement. 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.*

[1] a.ocais@fz-juelich.de

Contents

List of Figures

Executive Summary

The recommendations of this deliverable are targeted at the development community connected to E-CAM. We highlight the impact that immediate hardware developments are likely to have on the software applications of the E-CAM community. We also provide a set of recommendations to our user community with respect to how to prepare for, and deal with, these developments. Based on the perceived needs of our user community, this report is composed of 3 main parts, namely:

- *Hardware Developments* (Section 2) which provides a summarised update to Hardware Developments that we see as relevant to the E-CAM community in the 3-5 year horizon. In this section, roadmaps of HPC development, power considerations, hardware developments of particular significance are presented, as well as the current and expected availability of extreme-scale resources at the European level. Detailed feedback to E-CAM software developers is also given covering topics such as: using several development approaches such as CUDA, OpenACC, OpenMP, HIP and others to exploit the full power of GPUs, FPGAs and many-core architectures. It is fundamental to consider the several issues related the heterogeneity of exascale architectures, such as CPU-GPU bandwidth communication. The technologies described in this section are very likely to feature in any available European exascale technologies.
- *Software Needs* (Section 3) gives an update to the software needs of the project and what will be required of the software when targeting extreme-scale resources. In this section, an analysis of the ongoing "E-CAM Survey of Application Software" is provided. Feedback for hardware and software vendors is given, noting in particular that there is still a heavy reliance in the community on Fortran and support for this language will be essential on future systems for current workloads. A subset of possible programming paradigms are covered: C++17, Fortran 2018, several Open Standards (MPI/OpenMP/OpenACC/OpenCL) and runtime-system approaches (HPX/Kokkos/OmpSs/Charm++). For software developers, awareness of the latest standards and the status of their implementations are critical during application development as these new features exist to target the scalability challenges on modern systems. We provide a limited set of recommendations for people in E-CAM community who are embarking on a new software project: using MPI+OpenMP with their latest standards remains the safest bet with good performance and scalability achievable; you can prototype quickly using Python and leveraging the Python APIs to the libraries you need; using C++ with Python interfaces can help achieve maximal performance (and it helps that computer scientists seem to prefer C++ when creating innovative runtime systems).
- *Interactions between end-users, developers and vendors* (Section 4) describes the face-to-face discussions that have taken place between E-CAM developers, users and the hardware and software vendors. This section reflects the E-CAM Extreme-Scale State-of-the-Art Workshop held in Barcelona in June 2017, the Scoping Workshop *"Building the bridge between theories and software: SME as a boost for technology transfer in industrial simulative pipelines"* held in Genoa in 2018, as well as the interaction of the project with ETP4HPC, EuroHPC and EU hardware projects (in particular via the CoE coordination project FocusCoE). The European exascale roadmap, emerging hardware architectures relevant to exascale computing and exascale challenges in exploiting massive parallelism are reported in this section. Key outcome and recommendations include: E-CAM needs to develop niche activities to provide mutual benefit for both our industry partners and E-CAM's HPC oriented goals. We should provide a reliable software delivery mechanism between academic software developers and industrial partners. We also sought to increase the E-CAM community exposure to HPC technologies by organising a set of pilot projects covering High Throughput Computing (HTC), load-balancing and novel runtime technologies as well as a number of activities oriented towards deep learning.

  Discussions are also realised with hardware/software vendors through our participation in the development of the ETP4HPC Strategic Research Agenda and our participation in the EuroHPC Working Group on User Requirements.

We conclude this document by putting it's findings in the context of a scientific area of E-CAM that is gaining growing attention and industrial appeal: multiscale modelling, which has huge potential in the E-CAM community for the computational study of structure formation and processes at mesoscopic length and time scales. When addressing multiple scales there are significant methodological challenges, including algorithm design for horizontal coupling of scales (where all scales are simulated at once) and mapping techniques for vertical coupling (distinct, but coupled, simulations at different scales, with each scale posing different simulation challenges).

The coupling of scales is necessarily complex and frequently requires an adaptive workflow (either internally through libraries, or externally through additional application codes) orchestrating a simulation across different scales that is highly susceptible to scalability issues, in particular load-balancing problems. This complexity quickly exposes the advantage of using, and contributing to, middle-layer libraries since they can be leveraged across the workflow and help to minimise the effort of adapting to new architectures (which becomes a *tuning* effort as opposed to a development effort). Eurolab4HPC has already indicated that there are several initiatives moving in this direction, and

that developing for current and future high-end architectures, as found in HPC centres, will require such a shift:

> Manual optimization of the data layout, placement, and caching will become uneconomic and time consuming, and will, in any case, soon exceed the abilities of the best human programmers.

Nascent fields such as multiscale modelling which have scientific use cases that should map well to exascale resources would be wise to heed such advice.

1 Introduction

1.1 Scope of the deliverable

This deliverable is divided into 3 main parts, namely:

- Analysis of hardware developments expected in the 3-5 year horizon with feedback to software developers (Section 2);
- Survey of software and hardware currently used by the E-CAM community (and the level of parallelisation used). Analysis of software future needs and requirements with feedback to software developers, as well as hardware and software vendors (Section 3);
- Discussion of project software needs with hardware and software vendors, and detailed output from direct face-to-face session between the project end-users, developers and hardware vendors (Section 4).

1.1.1 Target audience

The recommendations of this deliverable are targeted at the development community connected to E-CAM. We highlight the impact that immediate hardware developments are likely to have on the software applications of the E-CAM community. We also provide a set of recommendations to our user community with respect to how to prepare for, and deal with, these developments.

In addition, our software survey, in particular, provides hardware and software vendors with valuable insight into the normal simulation software and practices of our user community.

1.2 Extent of the update to *Hardware Developments IV*

In E-CAM's *Hardware Developments IV* deliverable [1], we maintained a focus on the disruptive aspects of the hardware and software technologies that are most likely to impact the software development practices of the E-CAM community.

The current deliverable maintains a similar structure to the previous deliverable with updates to reflect the state of the art. Significant additions include

- the consideration of the "Eurolab4HPC Long-Term Vision on High-Performance Computing (2nd Edition)" in our analysis
- hardware updates from major vendors
- further details on the EuroHPC joint undertaking and the related hardware developments,
- discussions on how to deal with the I/O layer,
- the inclusion of the E-CAM Scoping Workshop entitled "Building the bridge between theories and software: SME as a boost for technology transfer in industrial simulative pipelines" (Section 4.2),
- the cultivation of machine learning expertise within the project, and the organisation of associated training events (Section 4.3),
- updates on relevant aspects of WP7 pilot projects.

Task 3 of WP7 involved interfacing the user-community with software developments and hardware vendors. We have provided 2 feedback sessions where the software developed within the project is presented to a combined audience of users and hardware vendors. The first was the Extreme-scale State-of-the-Art workshop (Section 4.1) held in RP2. The second of these is the Scoping Workshop mentioned above (Section 4.2), where the audience was primarily software as opposed to hardware vendors since E-CAM has recognised key role of SMEs that take academic concepts and prepare them for the marketplace.

2 Hardware Developments

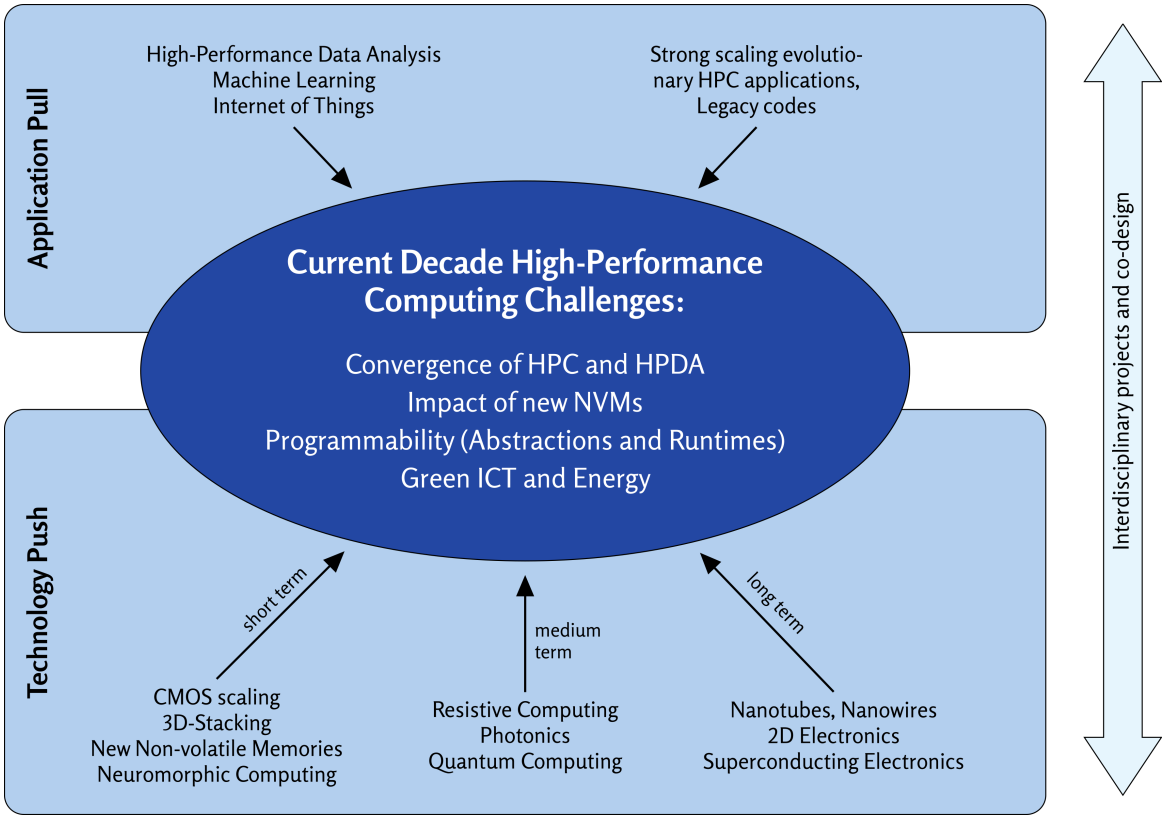

Figure 1: Factors that will influence the challenges in HPC over the coming decade (©Eurolab4HPC)

There are a number of different organisations and projects that are generating roadmaps about the current and future technologies that are (or may be in the future) available in the High Performance Computing (HPC) space. A summary table of which had been created by Eurolab-4-HPC for their report on the "Eurolab4HPC Long-Term Vision on High-Performance Computing (2nd Edition)", including summary input from projects such as PRACE, ETP4HPC, HiPEAC, BDVA and others.

The second edition Eurolab4HPC Vision was released in January 2020 but a summary graphic they created highlights the factors that influence this document (reproduced in Fig.1). In this section we will focus on a small subset of the content of these roadmaps (primarily from Eurolab4HPC and ETP4HPC) that are most likely to impact the target community of E-CAM in the 3-5 year horizon and likely beyond.

2.1 Influential factors

There are a number of economical factors that will have a strong influence on aspects of the hardware that will be realised in exascale machines. We discuss two here: the practical consideration of the power budget required to run such a resource and the market-led influence of deep learning on *commodity* hardware that can be leveraged.

2.1.1 General outlook

The current trend of HPC scaling based around floating point operations is expected to continue well beyond Exascale computers, however, the landscapes in which HPC is operating are changing. This is due to 3 main trends.

Firstly is the emergence of High Performance Data Analytics (HPDA) complementing simulation in scientific discovery and future HPC applications.

The second trend is the continued emergence of cloud computing and warehouse-scale computers, which aim at cost-effective data manipulation at previously unprecedented scales, which has led to fundamental breakthroughs as a result. Different algorithms are needed in data and cloud computing centres compared to traditional HPC applications and may not require the same computer structures.

The third trend arises from Deep Neural Networks (DNN) for back propagation learning of complex patterns. GPU accelerators are seen as very effective for this field, given support for 16-bit floating point and tensor processing units.

The overall trend and target is to develop systems that adapt more quickly to changing environments, more complex systems and this is expected to continue over the next decade.

2.1.2 The impact of power considerations

For the last decade, power and thermal management has been of high importance. The entire market focus has moved from achieving better performance through single-thread optimizations, e.g., speculative execution, towards simpler architectures that achieve better performance per watt, provided that vast parallelism exists. The HPC community, particularly at the higher end, focuses on the flops/watt metric since the running cost of high-end HPC systems are so significant. It is the potential power requirements of exa-scale systems that are the limiting factor (given currently available technologies). The practical outcome of this is the rise of accelerating co-processors and many-core systems. In the following sections we will mention such technologies in the context of relevant international exa-scale projects. The problem with the current two-pronged advance is that it is not always easy to develop parallel programs for these technologies and, moreover, those parallel programs are not always performance portable between each technology (particularly since FPGAs use a data-flow programming model), meaning that each time the architecture (or indeed the vendor) changes, the code may have to be rewritten. While there may be open standards available for each technology, each product currently has different preferred standards which are championed by the individual vendors (and therefore the best performing).

In general, we see a clear trend towards more complex systems, which is expected to continue over the next decade. These developments will significantly increase software complexity, demanding more and more intelligence across the programming environment, including compiler, run-time and tool intelligence driven by appropriate programming models. Manual optimisation of the data layout, placement, and caching will become uneconomic and time consuming, and will, in any case, most likely soon exceed the abilities of the best human programmers.

2.1.3 The impact of Deep Learning

Traditional machine learning uses handwritten feature extraction and modality-specific machine learning algorithms to label images or recognize voices. However, this method has several drawbacks in both time-to-solution and accuracy. Today's advanced deep neural networks use algorithms, big data, and the computational power of the GPU (and other technologies) to change this dynamic. Machines are now able to learn at a speed, accuracy, and scale that are driving true artificial intelligence and AI Computing.

Deep learning is used in the research community and in industry to help solve many big data problems such as computer vision, speech recognition, and natural language processing. Practical examples include:

- Vehicle, pedestrian and landmark identification for driver assistance
- Image recognition
- Speech recognition and translation
- Natural language processing
- Life sciences

The influence of Deep Learning on the market is significant with all major vendors of processors including accelerators having to accommodate workloads that are different from conventional HPC, e.g. support for massively parallel lower precision 32-bit floating point arithmetic, and integer arithmetic. Increasingly, we are seeing new processor designs that adopts a modular, chiplets approach where execution units specialised for different workloads can be integrated on the same chip.

2.1.4 Intellectual Property origin

In modern high technology economies access to Intellectual Property (IP) is crucial. When IP is concentrated in a region, such as it is with computing technologies, the potential exists that access to this IP may be precarious, or simply expensive. It is economically important for large regions to have access to multiple sources of IP (for competitive pricing) and to try to generate their own critical IP (to guarantee access). The consequences of this can be seen in China with the development of their own line of x86 compatible CPUs, in Japan with their investment in ARM technologies and the EU with the European Processor Initiative (EPI).

2.2 Notable Hardware Developments

In previous iterations of this deliverable (see [2, 3, 4]), we have in this section provided lists of available hardware and platforms that are, or will be, accessible to E-CAM developers. In the current iteration we no longer try to be exhaustive but focus more on particular developments that may be of significant potential impact to our user community.

We continue to add increasing importance to Section 3 which attempts to identify software solutions that may help insulate E-CAM developers from an increasingly diverse computing environment over which they do not ultimately have a lot of control.

2.2.1 European Processor Initiative

The EPI is a project currently implemented under the first stage of the Framework Partnership Agreement signed by the Consortium with the European Commission, whose aim is to design and implement a roadmap for a new family of low-power European processors for extreme scale computing, High Performance Data Analysis (HPDA) and a range of emerging applications.

The initial focus will be placed on integration of technologies to allow the creation of a General Purpose Processor (GPP), over multiple generations with increasing levels of IP generation. The first generation of the GPP, codenamed "Rhea", will have a design that incorporates Arm, RISC-V, high bandwidth memory, and other technologies in a multi-tile package. It had been scheduled for release in 2021, intended to be used in the first European exascale prototype systems. The second generation architecture, code-named "Cronos", is scheduled for 2022-2023 and expected to be included into European production exascale systems.

The ARM tiles will act as the host processor for other auxiliary processors, which includes Multi-Purpose Processing Array (MPPA) technology (generally used to speed up embedded computing functions), embedded FPGA (that can be used as a reconfigurable accelerator or to implement dynamic hardware modifications), and a custom HPC accelerator architecture that will be developed by EPI.

The accelerator stream will develop and demonstrate fully European processor IPs based on the RISC-V Instruction Set Architecture, providing power efficient and high throughput accelerator tiles within the General Purpose Processor (GPP) chip. Using RISC-V allows leveraging open source resources at hardware architecture level and software level, as well as ensuring independence from non-European patented computing technologies.

2.2.2 Intel

Intel is attempting to create serious competition for (the currently dominant) NVIDIA in the accelerator space with the recently announced Intel X^e architecture, which will be used in new integrated and discrete Intel GPUs from 2020. The strong selling point is that they can provide an accelerated computing ecosystem to rival CUDA that will be needed for exascale computing.

The X^e architecture encompasses 3 flavours: X^e-LP (Low Power), X^e-HP (High Power) and X^e-HPC (High Performance Computing). While the first two are more graphics/data centre focused, X^e-HPC is the most relevant here with its emphasis on raw throughput. X^e contains both CPU-style SIMD (Single Instruction, Multiple Data) units as well as GPU-like SIMT (Single Instruction, Multiple Threads) units, that can be combined on the same card to maximise performance for particular workloads, e.g. kernels can perform some serial work on the SIMD units of the GPU without having to move these back to the CPU. For the X^e-HPC design at least, support for double precision floating point computations will be extensive.

See Section 3.2.5 for information on the programming model intended for leveraging these developments.

Moving data in and out of these X^e-HPC compute units will involve some direct connections between neighbouring units and the use of a new, scalable memory fabric called XeMF that connects to HBM (High Bandwidth Memory) channels. There is also a special high-speed cache for the compute units to call upon.

Intel announced in late 2019 that the first X^e-HPC GPU will be a design called Ponte Vecchio (PVC). The technology is expected to be the workhorse of the upcoming Aurora A21 exascale system to be installed in Argonne National Laboratory.

2.2.3 AMD

While Intel has dominated the HPC processors market for the last decade or so, there is renewed competition from AMD in 2019 with the release of its second generation (Zen 2) EPYC processor called "Rome". It represented the first time that AMD has released a HPC oriented processor with smaller, 7nm transistors ahead of its biggest rival. AMD Rome CPUs have since been deployed on large European supercomputers such as the 26-petaflop Hawk system at HLRS (Stuttgart, Germany) and an upcoming 6.4-petaflop supercomputer at CSC (Finland) to be installed in 2020.

Apart from CPUs, AMD has also announced aggressive plans to compete with Intel and NVIDIA in the accelerator space. It has announced in early 2020 a GPU architecture called CDNA, which is more compute focused compared to the company's gaming GPUs. AMD plans to couple the CPU and GPUs together, using its Infinity architecture, into one unified data model so data movement will not have to be explicitly managed by the programmer.

Rome will be followed by the third generation (Zen 3) “Milan” processor, expected late 2020 and the Zen 4 based “Genoa” processor, expected in 2022. These AMD EPYC CPUs and GPUs will power 2 upcoming exascale systems in the U.S. – the 1.5-exaflop Frontier supercomputer at Oak Ridge National Laboratory in 2021 and the 2-exaflop El Capitan supercomputer at Lawrence Livermore National Laboratory (LLNL) that serves the National Nuclear Security Administration (NNSA) in 2023.

2.2.4 NVIDIA

The HPC accelerator space has been dominated by NVIDIA GPUs for many years. While both Intel and AMD are planning competition in this area of the HPC market, NVIDIA benefits from an established hardware and software ecosystem, including rich support for applications and the popular CUDA programming framework optimised for its GPUs. NVIDIA GPUs are also widely used for AI and deep learning workflows.

NVidia has just released the new GPU architecture, Ampere, which contains a higher mix of CUDA cores (for compute workloads), Tensor Cores (for AI and Deep Learning workloads, and now even in full FP32) and RT Cores (a relatively new feature aimed at ray tracing and related workloads). The new card allows more than 1.5x speedup, compared to previous Volta generation, on most of the Molecular Dynamic solvers like LAMMPS, NAMD and GROMACS widely used in the E-CAM community. It is based on a TSMC 7nm N7 technology and provides 1.5TByte of memory bandwith, i.e. 1.7x the previous generation which explains the speedup mentioned, since most of the solvers are memory bandwidth bound. On the half precision side, it achieves 78 TFLOPS in FP16 which could enable new algorithms based on mixed and reduced precision.

ATOS has just announced that the first supercomputer with A100 GPU card will be available in the next Q2 of 2020 at Julich Supercomputer Centre.

2.2.5 ARM

ARM has a different business model compared to Intel and AMD in the server and HPC domain, by developing technology that is licensed to semiconductor companies, i.e. other companies can license complete ARM microarchitectures or obtains an architectural license to new CPU microarchitectures from scratch, while still conforming to the ARM model. The latter has seen the emergence of new, power-efficient processors that have seen increasing adoption in HPC. For example, the ARM-based Cavium ThunderX2 chips from Marvell are used in supercomputing projects at Sandia National Laboratory as well as in CEA, France. Notably, Fujitsu’s A64FX ARM CPU will power Japan’s next pre-exascale "Fugaku" supercomputer (the successor to its "K" supercomputer) which will be installed in RIKEN in 2021. As highlighted above, the European Processor Initiative is also planned to adopt ARM technologies in its roadmap, which will use ARM’s "Zeus" cores as its general-purpose processor.

Recently ARM announced a partnership with NVidia to quickly build GPU-accelerated Arm-based servers. Not only NVidia, but the full Arm ecosystem partners — including Ampere, Fujitsu and Marvell — will ensure NVidia GPUs also work seamlessly with Arm-based processors.

2.2.6 EuroEXA

Field Programmable Gate Arrays (FPGAs) are semiconductor devices that are based around a matrix of configurable logic blocks (CLBs) connected via programmable interconnects. FPGAs can be reprogrammed to desired application or functionality requirements after manufacturing. This feature distinguishes FPGAs from Application Specific Integrated Circuits (ASICs), which are custom manufactured for specific design tasks.

EuroEXA is an EU Funded 20 Million Euro for an ARM+FPGA Exascale Project which will lead Europe towards exascale, together with ExaNeSt, EcoScale and ExaNoDe projects, scaling peak performance to 400 PFLOP in a peak system power envelope of 30MW; over four times the performance at four times the energy efficiency of today’s HPC platforms.

2.2.7 Complex I/O layers

The recognition of "Big Data" in scientific applications and the complexity of new storage technologies are likely to have a number of impacts in the wider scientific community when it comes to I/O. This is summarised well within "Eurolab4HPC Long-Term Vision on High-Performance Computing (2nd Edition)":

> ... the POSIX I/O interface to cold data (e.g. Lustre, GPFS, or the limited Hadoop DFS) remains largely agnostic to the I/O optimization opportunities that array-centric computing presents. This simplistic view of I/O for array-centric analyses is challenged by the dramatic changes in hardware and the diverse application and analytics needs in today’s large-scale computers: On the hardware front, new storage devices

> such as SSDs and NVMe not only are much faster than traditional devices, but also provide a different performance profile between sequential and random I/Os. On the application front, scientific applications have become much more versatile in both access patterns and requirements for data collection, analysis, inference and curation. Both call for more flexible and efficient I/O abstractions for manipulating scientific data.
>
> ...
>
> The rapid adoption of new storage media, has shifted the perception on using third-party I/O libraries for data storage and access. A 2014 user survey of the NERSC scientific facility shows that the HDF5 and NetCDF file format libraries are among the most widely used libraries, and have at least as many users as popular numerical libraries, such as BLAS, ScaLAPACK, MKL and fftw [16]. In 2018, 17 out of the 22 projects in the U.S. Department of Energy Exascale Computing Project (ECP) portfolio were using HDF5. These file format libraries present a semantically richer, array-centric data access interface to scientific applications. Using such file format libraries allows applications to navigate through a hierarchical organization of datasets, explore rich metadata about each dataset, choose subsets of an array, perform strided accesses, transform data, make value-based accesses, and automatically compress sparse datasets to save space.
>
> Many research efforts seek to bring richer data management functionality in file format libraries and largely explore three complementary avenues: (1) extending the functionality of file format libraries, (2) developing connectors and (3) automatically migrating data.

Given this analysis, for developers to avail of hardware developments at the I/O level, it would seem prudent for applications to adopt an I/O layer built upon an established and maintained file format library. At this time, it is clear HDF5 should probably be the base layer of whichever library is chosen.

2.2.8 Memory Hierarchy

It is expected that future supercomputers are going to fall into two main categories of memory hierarchy;

- Deep Memory; low speed non-volatile memories may lead to additional levels to close the gap between mass storage and memory.
- Shallow Memory; where main memory may be merged with storage for smaller systems, so the cache, main memory and mass storage might be merged to a single level, providing the system with an improved performance system-wide. It would enable applications to use more non-uniform/highly random data access.

The implications of this are that merging main memory and mass storage allows the quicker start of applications, crash recovery, and a reduction in energy consumption. Additional work on security and privacy needs to be incorporated into the design features however.

2.2.9 Feedback for software developers

Extreme computing and AI are currently global issues and key components of global economies and global security. It is not surprising then to see competition increase in such a huge (and growing) market. The unfortunate end result for a scientific developer is that there is no developer ecosystem that can leverage all of this diverse hardware.

Several approaches have been developed to exploit the full power of accelerators: from parallel computing platform and application programming interface specific for NVIDIA GPUs, like CUDA 10.0, to the latest version of OpenMP 5.0 which contains directives to offload computational work from the CPU to the GPU. While CUDA currently is likely to achieve best performance from an NVIDIA device, OpenMP allows for better portability of the code across different architectures. Finally, the OpenACC open standard is an intermediate between the two, more similar to OpenMP than CUDA, but allowing better usage of the GPU. Developers are strongly advised to look into these language paradigms.

Moreover, it is fundamental to consider there the several issues linked to hybrid architectures. In the case of a GPU we have CPU-GPU and GPU-GPU bandwidth communication, direct access through Unified Virtual Addressing and the presence of new APIs for programming (such as Tensor Core multiplications specifically designed for deep learning algorithms).

At this stage, accelerator programming is quite mainstream and there are many training courses available online, see for example the NVIDIA education site for material related to CUDA and OpenACC. Material for OpenMP is more limited, but as an increasing number of compilers begin to support the OpenMP 5.0 standard, we expect the amount of such material to grow (see this presentation on performance of the Clang OpenMP 4.5 implementation on NVIDIA GPUs for a status report as of 2016).

FPGAs use a dataflow based programming model, a programming paradigm that models a program as a directed graph of the data flowing between operations. Despite their high efficiency in performance and power consumption, FPGAs are known for being difficult to program. In the EuroEXA project one of the partners is Maxeler who provide hardware systems that combine conventional CPUs with high performance Dataflow Engines (DFEs) built around Field Programmable Gate Array (FPGA) chip technology. The application design process using Maxeler technologies is described very well by EoCoE in a June 2018 webinar on FPGA computing.

In general, the current FPGA environment is similar to the GPU space some years ago in that HPC is not the primary market but could be a significant one. There is no common standard being championed to improve their programmability, so choosing one or the other vendor locks the user into a particular development platform (to some extent).

Overall, the hardware is growing in diversity and unfortunately there is no API that is going to allow you to program for all of them. Each hardware combination is likely to have different capabilities, and different performance characteristics and bottlenecks. As a result, an optimisation for one combination of hardware components may not be possible on another, or may indeed be a pessimization. It's for this reason that we focus on viable "middle-men" in Section 3.

With respect to I/O, we must note that the hardware complexity at this level is also going to increase significantly and will become a burden if developers attempt to engage with this directly. The advice would be to avoid creating an issue of this topic through the use of a file-format library in your I/O layer. In particular, it would be advisable that your I/O layer is built (either directly or indirectly) upon HDF5 whose adoption is widespread and is undergoing significant and continuous development.

2.3 Extreme-scale resources at the European level

2.3.1 EuroHPC

As part of the EU HPC policy, a multi-government agreement (20 as of June 2018) has been signed to acquire and deploy, by 2022/2023, a pan-European integrated exascale supercomputing infrastructure: EuroHPC. It will address three urgent needs:

- to procure and deploy in Europe in competitive timeframes a world-class pre-exascale HPC infrastructure;
- to make it available to public and private users for developing leading scientific and industrial applications;
- to support the timely development of the next generation European HPC technologies and their integration into exascale systems in competitive timeframes with respect to our world competitors.

With a total budget of approximately EUR 1 billion, the Joint Undertaking will function until 2026. The Joint Undertaking aims by 2023 to deploy at least one exascale system based on European technology and in June 2019 selected the Hosting Entities for Precursors to Exascale Supercomputers. The decision was taken in the meeting of the Governing Board of the EuroHPC Joint Undertaking, in Luxembourg, on June 7 2019, where three different hosting sites for pre-exascale systems were selected: Bologna (Italy), Barcelona (Spain) and Kajaani (Finland). The call for Hosting Entities for Exascale Supercomputers will also soon be announced.

E-CAM is participating in the EuroHPC Working Group on User Requirements, submitting a position paper to the Group in June 2018 and attending an associated workshop.

2.3.2 Current PRACE Resources

As far as the Partnership for Advanced Computing in Europe (PRACE) initiative is concerned, the complete list of available resources are shown in Figure 2.

Access to PRACE resources can be obtained by application to the PRACE calls.

Moreover, the Distributed European Computing Initiative (DECI) is designed for projects requiring access to resources not currently available in the PI's own country but where those projects do not require resources on the very largest (Tier-0) European Supercomputers or very large allocations of CPU. To obtain resources from the DECI program, applications should be made via the DECI calls.

2.3.3 Feedback for software developers

While it is still too early to know the hardware technology content of the EuroHPC, the technologies described already are very likely to feature in the precursors. Practically speaking one must then consider the software implementation implications, something that is covered in some detail in Section 3.2.

		HAWK	JOLIOT-CURIE AMD	JOLIOT-CURIE SKL	JOLIOT-CURIE KNL	JUWELS	SuperMUC NG	MARCONI 100	MARCONI SKL	Mare Nostrum4	Piz Daint
System Type		HPE	Bull Sequana X1000	Bull Sequana X1000	Bull Sequana X1000	Bull Sequana X1000	Lenovo	Lenovo System NeXtScale	Lenovo System NeXtScale	Lenovo	Hybrid Cray xC50
Compute	Processor Type	AMD EPYC Rome	AMD EPYC Rome @ 2.6GHz	Intel Xeon Platinum 8168 @ 2.7 GHz	Intel (Knights Landing) Xeon Phi 7250 @ 1.4 GHz	Intel Xeon Platinum 8168 @ 2.7 GHz	Intel Skylake Xeon Platinum 8174 @ 3.1 GHz	IBM POWER9 AC922 @ 3.1 GHz	Intel Xeon Platinum 8160 @ 2.1 GHz	Intel Xeon Platinum 8160 @ 2.1 GHz	Intel Xeon E5-2690 v3 @ 2.6 GHz (12 cores)
Compute	Total # nodes	5,000	2,292	1,656	828	2,511	6,480	980	3188	3,456	5,704
Compute	Total # cores	640,000	293,376	79,488	56,304	120,528	311,040	31,360	153,024	165,888	68,448
Compute	# accelerators	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.	4 per node	n.a.	n.a.	1 per node
Compute	Type of accelerator	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.	NVIDEA Volta V100 GPUs, NvLink 2.0, 16 GB	n.a.	n.a.	NVIDEA Tesla P100 16GB
Memory	Memory/ Node	128 GB	256 GB	192 GB	96 GB DDR4 + 16 GB MCDRAM	96 GB	96 GB (144 nodes with 768 GB)	256 GB	96 GB DDR4 + 16 GB MCDRAM	96 GB (200 nodes with 384 GB)	64 GB
Network	Network Type	Infinibad HDR	Infiniband HDR	Bull Exascale	BULL BXI	InfiniBand EDR	Intel Omni-Path Architecture	Intel Omni-Path Architecture	Intel Omni-Path Architecture	Intel Omni-Path Architecture	Cray Aries
Network	Connectivity	9D Enhanced Hypercube	Dragonfly +	Fat Tree	Fat Tree	Fat Tree	Fat Tree	Fat Tree	Fat Tree	Fat Tree	Dragonfly

Figure 2: List of PRACE Resources (as of 2020) with associated hardware characteristics.

3 Software Needs

While the original scope of this section was intended to collect information from the E-CAM community that could be relayed back to hardware and software vendors, it is clear that the hardware developments described in Section 2 will greatly impact the software development practices of the E-CAM development community. For this reason, we go beyond this original scope and in addition we highlight the language standards, runtime environments, workflows and software tools that can help E-CAM developers to deliver high quality, resilient software for current and next generation machines.

3.1 Software needs as collected by the E-CAM Software Survey

The E-CAM Survey of Application Software was created to collect basic information about the software, libraries and applications that are frequently used by people in the extended Centre Européen de Calcul Atomique et Moléculaire (CECAM) community, with a view to trying to develop a support infrastructure for heavily-used software within E-CAM. It includes questions about the application; scientific area; parallelization techniques; the users familiarity with, and capabilities of, the application; the sizes of users typical resource requirements; and finally whether they are a user or developer of the application.

The survey was circulated to our complete set of mailing lists (including industrialists), was distributed to all Extended Software Development Workshop (ESDW) participants in 2017 and remains available on the E-CAM webpage. To date, we have received 160 responses with 67 distinct application codes listed (154 are included in the analysis here). We are conscious that the responses related to the survey may not sufficient to be representative of the entire community since many of the responses are received following direct requests during events held by E-CAM.

Fig.3 shows that more than 54% of them simply *use* a open-source community code, while about 20% of them are open-source code developers with about 14% of them using a commercial code.

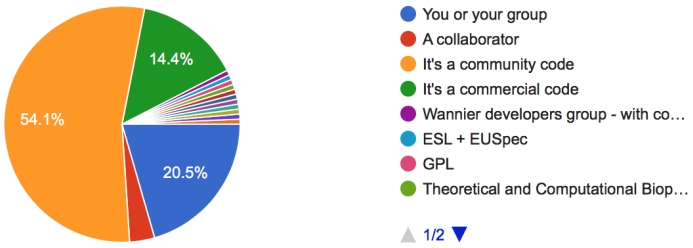

Figure 3: Question: Who develops the software?

Fig.4 shows that ~75% of the software of the survey is written in Fortran, with about ~36% in C/C++ and a further 24% using Python (or Python bindings).

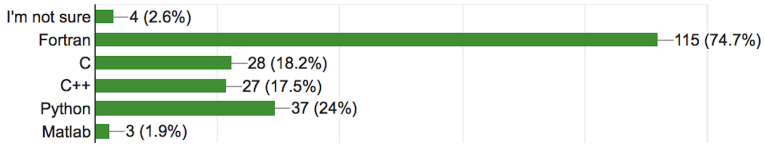

Figure 4: Question: The Programming Language used by the software.

Fig.5 shows that three quarter of the respondees use a code with an MPI implementation and half with an OpenMP implementation. About 40% of them are under Hybrid MPI/OpenMP mode. GPU capabilities are available to 45% of the applications.

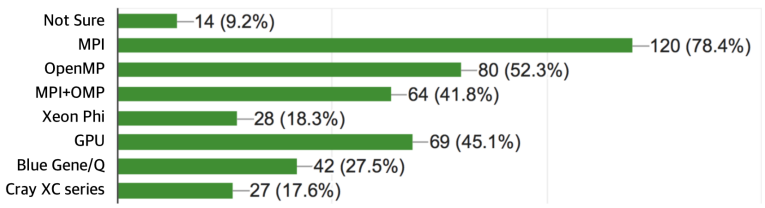

Figure 5: Supported (parallel) software capabilities

Fig.6 shows that 89% of respondees get their publication-quality results through the use of resources at HPC centres.

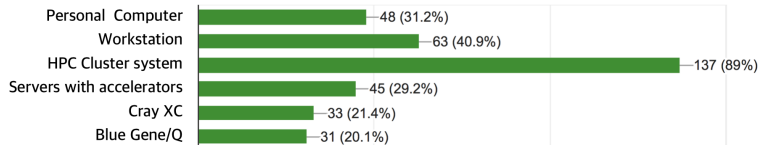

Figure 6: Most common hardware used for generating publication-quality results

Fig.7 shows that almost half them use less than 128 cores for simulation and less than one quarter of them use 1024 cores or beyond. This clearly indicates that only a small subset of respondees have requirements that potentially extend to extreme-scale resources (although the potential use case for High Throughput Computing was not included in the questionnaire and will be added in the next iteration).

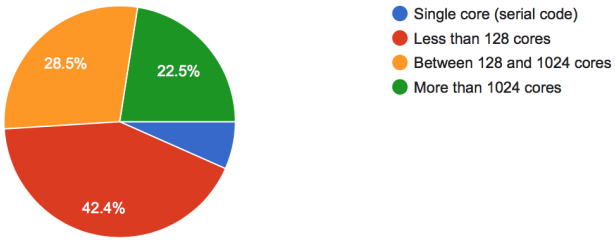

Figure 7: Max number of cores used per run

Fig.8 shows that more than 80% of them get their jobs finished within 24 hours (or have restart capabilities in the application).

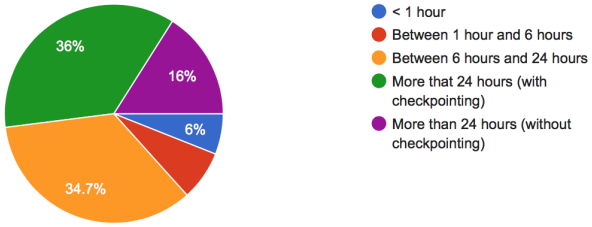

Figure 8: Max number of hours used per run

While the survey is still not large enough to draw strong conclusions, we can extract some trends with respect to the **softwares needs** of the E-CAM community:

- Most of the users use open-source community codes and don't necessarily have the capacity to maintain/develop them. The needs for commercial codes represent only a small portion of the community.
- Since the majority of the codes included are written with Fortran, any contributions that our community make to these codes will necessarily have to be written in Fortran. However, the development and use of Python bindings becomes increasingly popular. Teaching the community how to use (and create) efficient Python bindings and using Python for non-performance-critical components of their workflow should be very helpful for the community (and also ease any future transition to other programming models).
- MPI, OpenMP, GPU (CUDA/OpenACC) are the main techniques currently used to enable code parallelisation. Training in these techniques, in particular targeting "accelerator" technologies, are needed.
- About 90% of them use HPC centres for producing science. Guiding them to get access to PRACE resources should be very helpful *but* the scalability of their resource requirements will need to be demonstrated. Given that *ensemble* calculations are very common across E-CAM, the inherent scalability of such workflows need to be emphasised to the community.

3.1.1 Feedback for hardware and software vendors

As the survey stands (and the results should be considered limited despite 160 responses), the indications are that the E-CAM community relies heavily on a diverse set of community-developed applications. The most commonly occuring, at present, are GROMACS, QuantumESPRESSO, Yambo, CP2K, CPMD, LAMMPS, SIESTA, OpenMM, DL_POLY, DL_MESO and ESPRESSO++. Only one commercial applications have occurred more than 10 times: the VASP code.

There is still a heavy reliance in the community on Fortran, meaning that support for this language will be essential on future systems if people are to be expected to migrate their current workloads there.

The typical production use cases observed in the community do not necessarily require extreme scalability. This is not to say, however, that the community does not require extreme-scale resources. Vendors should be wary that ensemble calculations are an important part of the workflow across the E-CAM community and that this has potential cost implications for resources suitable to this case (for example, this may significantly impact the design of cost-effective network topologies for E-CAM workloads).

There are many cases where E-CAM developers are interfacing at a relatively abstract level with community codes. Therefore, the appetite for novel technologies is typically reliant on an implementation (and sufficient documentation) within these codes. This is both an opportunity and a burden: a subset of community applications ported to novel technologies may bring large user communities to such resources *but* such applications are so feature-rich (which usually implies monolithic) that this may be exceedingly difficult to implement satisfactorily in practice.

3.2 Programming Paradigms

3.2.1 C++17

C++17 is the most recent revision of the ISO/IEC standard for the C++ programming language, published in December 2017.

The previous C++ versions show very limited parallel processing capabilities when using multi/many core architectures. This situation changes with the C++17, in which the parallelised version of Standard Template Library is included. The STL is a software library for C++ programming which has 4 components: Algorithms, Containers, Functors and Iterators. *"Parallel STL advances the evolution of C++, adding vectorization and parallelization capabilities without resorting to nonstandard or proprietary extensions, and leading to code modernization and the development of new applications on modern architectures."*[5]

A multi-threading programming model for C++ is supported since C++11.

It has been noted by the creator of C++, Bjarne Stroustrup, that the continued introduction of expert-level features into C++ may drive away newcomers and he has helped write a set of guidelines to help people adopt modern C++.

E-CAM is very aware of the parallelism potential of C++17 and hosted a dedicated event to promote the use of C++17 within E-CAM in Barcelona in the first quarter of 2017.

3.2.2 Fortran 2018

Fortran 2018 is a significant revision of Fortran 2008 (which was when Fortran became a Partioned Global Address Space (PGAS) language with the introduction of coarrays). The revisions mostly target additional parallelisation features and increased interoperability with C and was released on November 28, 2018.

Most Fortran-based software E-CAM sees in practice is implemented in Fortran 95 and there appears to be little awareness of the parallel features of the latest Fortran standards. E-CAM has considered organising a workshop that addresses this lack of awareness (similar to the "Software Engineering and Parallel Programming in Modern Fortran" held at the Cranfield University), though initial demand for such an initiative has not been strong within the community. PRACE do offer similar courses, such as Advanced Fortran Programming at E-CAM partner CSC, and we currently direct interested parties to these courses.

It should be noted that low awareness of modern features may be due to the fact that compiler support for the latest Fortran standards is limited. This is most likely due to the fact that Fortran is not widely used outside of the scientific research (limiting its commercial scope), however this will hopefully change with the LLVM Flang Fortran compiler which uses Fortran 2018 as it's reference for implementation.

3.2.3 The (potential) role of Python

Given that it is an interpreted language (i.e., it is only compiled at runtime), Python is usually not discussed much in the HPC space since there is limited scope for control over many factors that influence performance. Where we are observing a lot of growth is where applications are being written in languages like C++ under the hood but are intended to be primarily used via their Python interfaces (as is common in many deep learning frameworks). Examples of this in the E-CAM space would be ESPResSo++, which was one of the applications featured in E-CAM, and PyLAMMPS, the Python interface to LAMMPS (which is preferred as an interface by a significant number of users and developers leveraging LAMMPS).

We also would like to make developers aware of highly sophisticated Python libraries that are extremely useful for HTC. One of these libraries, Dask, is leveraged by E-CAM within it's own HTC library jobqueue_features which targets MPI workloads.

One should also be aware of the possibilities provided by `numba`, which is an open source JIT (*just in time*) compiler that translates a subset of Python and NumPy code into fast machine code. `numba` even supports CUDA GPU programming by directly compiling a restricted subset of Python code into CUDA kernels and device functions following the CUDA execution model.

This is a valuable, and user friendly, development model that allows users to leverage Python for fast prototyping while maintaining the potential for high performance application codes.

As of January 1st, 2020, Python 2 is no longer being maintained. For any Python 2 users, the official porting guide has advice for running Python 2 code in Python 3.

3.2.4 Open Standards

We describe here some of the open standards that are most likely to be leveraged on next generation HPC resources.

MPI

Now more than 25 years old, Message Passing Interface (MPI) is still with us and remains the de facto standard for internode communication (though it is not the only option, alternatives such as GASNet exist). MPI-3.1 was approved by the MPI Forum on June 4, 2015. It was mainly an errata release for MPI 3.0 which included some important enhancements to MPI:

- Nonblocking collectives
- Sparse and scalable irregular collectives
- Enhancements to one-sided communication (very important for extreme scalability)
- Shared memory extensions (on clusters of SMP nodes)
- Fortran interface

Maintaining awareness of the scope of past and future updates to the MPI standard is important since it is the latest features that target the latest architectural developments.

OpenMP

OpenMP is also 20 years old and remains the most portable option for on-node workloads. The standard has introduced new features to deal with increasing node-level heterogeneity (device offloading, such as for the GPU, in particular) and varied workloads (task level parallelism).

From GCC 6.1, OpenMP 4.5 is fully supported for C and C++ (with Fortran support coming in the GCC 7 series). The level of OpenMP support among other compilers varies significantly.

There is partial support for OpenMP 5.0 (released in 2018) from GCC 9.1, the Cray compilers from CCE 9.0 (via Clang/LLVM) and from version 19.1 of the Intel compilers. OpenMP 4.5 is fully supported by Clang/LLVM with the features of OpenMP 5.0 under active and heavy development.

OpenACC

OpenACC (for open accelerators) is a programming standard for parallel computing developed by Cray, CAPS, Nvidia and PGI. The standard is designed to simplify parallel programming of heterogeneous CPU/GPU systems. Since the paradigm is very similar to the latest OpenMP specs, a future merger into OpenMP is not unlikely. Moreover, the latest PGI compiler implements latest C++17 features and allows to offload workload to NVidia GPUs in pure C++ language, as well as mixing with OpenACC and CUDA directives.

It should be noted that CUDA (with the nvcc compiler) is still the most commonly used (and highest performing) library for programming NVIDIA GPUs.

OpenCL

Open Computing Language (OpenCL) is a framework for writing programs that execute across heterogeneous platforms consisting of central processing units (CPUs), graphics processing units (GPUs), digital signal processors (DSPs), field-programmable gate arrays (FPGAs) and other processors or hardware accelerators. If you want to be portable beyond Nvidia and AMD GPUs the only option is OpenCL. Intel used to support the GPUs with OpenMP target directive, but recently dropped this.

OpenCL 2.2 brings the OpenCL C++ kernel language into the core specification for significantly enhanced parallel programming productivity. OpenCL's bad reputation is at least in part due to its verbose C host API. Its C++ host API on the other hand is quite readable.

When releasing OpenCL version 2.2, the Khronos Group announced that OpenCL would be merging into Vulkan (which targets high-performance realtime 3D graphics applications) in the future, leaving some uncertainty as to how this may affect the HPC space.

What is interesting to note is the progress of SYCL (also by the Khronos Group, SYCL 1.2.1 revision 5 is now the latest release as of April 18, 2019, is based on OpenCL 1.2) which is a royalty-free, cross-platform abstraction layer that builds on the underlying concepts, portability and efficiency of OpenCL that enables code for heterogeneous processors to be written in a "single-source" style using completely standard C++. SYCL includes templates and generic lambda functions to enable higher-level application software to be cleanly coded with optimized acceleration of kernel code across the extensive range of shipped OpenCL 1.2 implementations. Developers program at a higher level than OpenCL C or C++, but always have access to lower-level code through seamless integration with OpenCL, C/C++ libraries, and frameworks such as OpenCV or OpenMP.

HIP

Connected to the announcement of the Frontier exascale system, AMD has decided to create a new solution for programming their GPU: Heterogeneous-compute Interface for Portability, or HIP, is a new C++ runtime API which allows developers to create portable applications that can run on AMD and other GPU's.

Robustness and performance portability testing are still ongoing, but it offers a very similar interface to the CUDA language allowing to easily the conversion of a CUDA code into HIP.

3.2.5 Vendor Specific Libraries

Intel

The new specification, oneAPI supports direct programming and API programming, and will deliver a unified language and libraries that offer full native code performance across a range of hardware, including CPUs, GPUs, FPGAs and AI accelerators. At the core of the oneAPI specification is DPC++, an open, cross-architecture language built upon the ISO C++ and Khronos SYCL standards. oneAPI provides libraries for compute and data intensive domains. They include deep learning, scientific computing, video analytics, and media processing.

It is interesting to note what the Eurolab4HPC Long-Term Vision on High-Performance Computing (2nd Edition) had to say about oneAPI:

> oneAPI is based on the SyCL approach, using c++ templates to have a 3-way compilation, allowing for automatic host, accelerator and interface code generation. It is advancing in the direction of programmer-guided parallelization, but it is still leaving too many details to the programmer, like manually defining the interfaces between the host and the accelerator. In this way, the programmer is required to indicate the parameters that will be need to be copied to the accelerator. This is still low level programming, compared to higher level programming models like OpenMP, where the interface definition between host and accelerator code is made by the compiler, given the shared/private/firstprivate/map clauses using in the compiler directives.

AMD

The resurgence of AMD has led to the development of numerical libraries that are specifically tuned for the AMD EPYC processor family. These AMD Optimising CPU Libraries (AOCL) have a simple interface to take advantage of the latest hardware innovations. The tuned implementations of industry standard math libraries enable fast development of scientific and high-performance computing projects. These libraries include;

- AMD BLIS

- AMD FFTW
- AMD LibM
- AMD libFLAME
- AMD ScaLAPACK

The most recent, AMD ScaLAPACK is integrated with optimised versions of BLIS and libFLAME libraries that enables high performance dense linear algebra operations and easy linking.

BLIS is a portable software framework for instantiating high-performance BLAS-like dense linear algebra libraries. The framework has been designed to isolate essential kernels of computation that when optimised enable the optimised implementations of most is its commonly used and computationally intensive operations. Select kernels have been optimised for the AMD EPYC processor family, and have been carried out for single and double precision routines.

On the GPU side, AMD has introduced the ROCm platform, the first open-source HPC/Hyperscale-class platform for GPU computing that's also programming-language independent. Built for scale, supporting multi-GPU computing in and out of server-node communication through RDMA.

Connected to the announcement of the Frontier exascale system, AMD has decided to create a new solution for programming their GPU: Heterogeneous-compute Interface for Portability, or HIP, is a new C++ runtime API which allows developers to create portable applications that can run on AMD and other GPU's and falls under the umbrella of ROCm.

Robustness and performance portability testing are still ongoing, but it offers a very similar interface to the CUDA language allowing to easily the conversion of a CUDA code into HIP.

ARM

Until recently, the Arm developer/application ecosystem, e.g. OS support, and importantly for HPC tools and libraries required for performance, has been of concern for HPC users. But it has since garnered wide support from OEMs and industry standard partners. Specifically for HPC, there are open source libraries to optimise performance for ARM-based processors, which are also integrated in ARM Allinea Studio that encompasses C/Fortran compilers, performance libraries, as well as "ARM Forge" (consisting of a debugger, profiler and performance analyser).

3.2.6 Runtime System Approaches

As noted already, programming paradigm standards are moving forward to adapt to the technologies that we see in the market place. The complexity of the hardware infrastructure (as outline in Section 2) necessarily brings complexity to the implementation of the programming standards.

There are number of programming models that leverage runtime systems under development. They promise to abstract away hardware during the development process, with the proviso that tuning at runtime may be required. Our experience to date with these systems is limited so we simply provide a list of four such systems here (which is certainly not exhaustive) in no particular order:

- HPX, a C++ Standard Library for concurrency and parallelism. The goal of the HPX project is to create a high quality, freely available, open source implementation of ParalleX concepts for conventional and future systems by building a modular and standards conforming runtime system for SMP and distributed application environments. (Most recent release: v1.4.1, February 2020)
- Kokkos implements a programming model in C++ for writing performance portable applications targeting all major HPC platforms. For that purpose it provides abstractions for both parallel execution of code and data management. Kokkos is designed to target complex node architectures with N-level memory hierarchies and multiple types of execution resources. It currently can use OpenMP, Pthreads and CUDA as backend programming models. (Most recent release: v3.1, April 2020)
- OmpSs is an effort to integrate features from the StarSs programming model developed at Barcelona Supercomputing Centre (BSC) into a single programming model. In particular, the objective is to extend OpenMP with new directives to support asynchronous parallelism and heterogeneity (devices like GPUs). However, it can also be understood as new directives extending other accelerator based APIs like CUDA or OpenCL. The OmpSs environment is built on top of BSCs Mercurium compiler and Nanos++ runtime system. OmpSs-2 is the second generation of the OmpSs programming model (Most recent release: v19.06, June 2019 for OmpSs; 2019.11.2 November 2019 for OmpSs-2)

- Charm++ is an adaptive C/C++ runtime system providing speed, scalability and dynamic load balancing. Programs written using this system will run unchanged on MIMD machines with or without a shared memory. It provides high-level mechanisms and strategies to facilitate the task of developing even highly complex parallel applications. In particular we highlight the recent release of charm4py which builds on top of Charm++ but provides an Python API, with its recent major release containing API changes, many new features, bug fixes and performance improvements. (Most recent release: v6.10.0, December 2019 for Charm++; v1.0, September 2019 for charm4py)

3.2.7 Open Science

To quote from a recent article on the subject [6]:

> Open science refers to the movement of making any research artefact available to the public. This ranges from the disclosure of software source code ("open source") over the actual data itself ("open data") and the material used to analyse the data (such as analysis scripts, "open material") to the manuscripts reporting on the study results ("open access")... While open science is becoming generally accepted as a norm in other scientific disciplines, in software engineering, we are still struggling in adapting open science to the particularities of our discipline, rendering progress in our scientific community cumbersome.

In that article, they lobby for the creation of a set of guidelines to help scientists understand and implement Open Science since openness will hopefully enable better science faster.

Open source when you develop an application is only a part of open science, but it is a key part when we consider research codes. Developing software in the open, on resources such as GitLab and GitHub, help enable open science as well as open source since they also document the development and decision-making processes in research code development, as well the interactions of the users with research codes (through *Issues* and *Pull Requests* for example).

3.2.8 Feedback for software developers

Awareness of the latest standards and the status of their implementations are critical at all times during application development. The adoption of new features from standards are likely to have large impact on the scalability of application codes precisely because it is very likely that these features exist to target the scalability challenges on modern systems. Nevertheless, you should be aware that there can be very long gaps between the publication of a standard and the implementation in compilers (which is frequently also biased by who is pushing which standard and why: Intel pushed OpenMP because of their Xeon Phi line, NVIDIA who own PGI pushes OpenACC because of their GPUs, AMD pushed OpenCL for their own GPUs to compete with CUDA). The likelihood of there being a single common (set of) standard(s) that performs well on all architectures is not high in the immediate future. For typical developers that we see in E-CAM, MPI+OpenMP remains the safest bet and is likely to perform well, as long as the latest standards are used (and you can find a suitable compiler that implements them).

More disruptive software technologies (such as GASNet) are more likely to gain traction if they are used by popular abstraction layers (which could be PGAS languages, runtime systems or even domain specific languages) "under the hood". This would make the transition to new paradigms an implementation issue for the abstraction layer. Indeed, given the expected complexity of next generation machines, new programming paradigms that help remove the performance workload from the shoulders of scientific software developers will gain increasing importance.

As you may have noticed in the previous discussion, the computer scientists developing these abstractions are working mostly in C++ (SYCL, Kokkos, Charm++,...), and the implementation of new standards in compilers is also seen first for C++ (for Clang even proposed additions to the standard are usually implemented). From a practical perspective this has some clear implications: if you want to access the latest software technologies then you had better consider C++ for your application. This may appear harsh given that the Fortran standard has obvious capabilities in this space, but it is a current reality that cannot be ignored. Also, given the likelihood that the majority of researchers will eventually transition to industry it is perhaps prudent to ensure they have programming expertise in a language that is heavily used in the commercial space. Finally, the eco-system surrounding C++ (IDEs, testing frameworks, libraries,...) is much richer because of it's use in industry and computer science.

New programming frameworks are exciting and have new features but their implementation cannot be taken lightly: a lot of development work (and fault tolerance) is required. In E-CAM we have already held a Kokkos workshop and are likely to hold another in the near future. It is a model of particular interest in large part because it has had significant adoption and a 10-year funding horizon.

Taking all of the above into consideration, if you are a serious developer for exascale resources you should probably focus on C++. If you are, however, starting out with an application we would distil the discussion into the following advice: productivity is the most important part of your development process, prototype your application using Python

leveraging the Python APIs to the libraries you need; write tests as you go; and, when you start doing computationally intensive work, use C++ with Python interfaces to allow you to squeeze out maximal performance using the latest software technologies.

A final word related to open science, taking a proactive approach to open science in your research will help to enable a Declaration on Research Assessment (DORA) type research evaluation looking at the full range of research outputs (including software) and not just journal articles. It is all very well calling for better software engineering, but if this work is not adequately recognised in career reviews and grant applications it will not happen at the required scale. The recommendations in this report are hard work but are of great value to the community, we should all be pushing for DORA-type evaluations of researcher output to properly incentivise and reward the hard work recommended in this report.

4 Interactions between end-users, developers and vendors

4.1 Outcomes and key recommendations from the *E-CAM Extreme-Scale State-of-the-Art Workshop*

This section has been produced primarily based on some key input from the E-CAM Extreme-Scale State-of-the-Art Workshop that took place on 6-7 July 2017 in Barcelona (of which more expansive detail is provided in Appendix A). The event represented a gathering of key E-CAM personnel (code developers, application users) as well as representatives from other European projects (e.g. PRACE, ETP4HPC, Center of Excellence (CoE)s including POP, NoMAD, EoCoE and MAX) and the E-CAM user community. There were discussion sessions on both days which includes representatives of *hardware vendors* (a term which we loosely interpret to include EU-funded hardware development projects which clearly include industrial partners). *Software development* was represented by the presence of 5 CoEs, PRACE and developers of high profile applications in the E-CAM space.

One of the main objectives of the workshop was to inform, as well as instigating discussions with, E-CAM developers and end-users about some of the key HPC issues and challenges in the field of molecular and atomistic simulations, with particular emphasis on forward-looking exascale topics. Some of the main issues and topics included the following, and are discussed in more details in the sections as indicated in parentheses:

- The European exascale roadmap (Section 4.4)
- Emerging hardware architectures relevant to exascale computing (Section A.1).
- Exascale challenges in exploiting massive parallelism (Section A.2).

The discussion sessions of the meetings were intense and probing. Repeatedly raised was the potential conflict of interest between the E-CAM pilot projects (where industrial partners are interested in software developments that facilitate new science, including the development of new algorithms) and the funding agency desire to focus software developments exclusively on HPC goals. It was recognised that E-CAM needs to develop niche activities that can provide mutual benefit overlapping area between these two objectives.

To this end, E-CAM has highlighted the possibility to provide a reliable software delivery mechanism between academic software developers and industrial partners. In addition, it has sought to influence the workflows of the scientific developer community so that it aligns more with the expectations of software in the industrial space (in terms of documentation, usability and testing).

Increasing the E-CAM community exposure to HPC technologies is also a key goal that is to be enhanced by a set of HPC pilot projects (see Section 4.6) and the introduction of transversal components to our ESDW events (exposing people to new technologies and best practices).

Finally, to ensure that E-CAM can achieve it's own internal goals, a number of additional steps were considered:

- Deeper collaboration with projects such as EoCoE, POP, PRACE to collaborate on scaling and optimisation work for codes with potential for exascale systems,
- Leverage transversal ESDW workshops to create more synergy between the Work Package (WP)s and bring the E-CAM developers to the same level in terms of parallel code development skills,
- Increase the visibility and impact of the modules developed by E-CAM community by publishing pilot project webpages and advertising them via Twitter and other means.

4.2 E-CAM Scoping Workshop: *"Building the bridge between theories and software: SME as a boost for technology transfer in industrial simulative pipelines"*

E-CAM Scoping Workshop held a scoping workshop, with 28 participants, dedicated to software vendor SMEs in May 2018 in IIT (Genova) entitled *"Building the bridge between theories and software: SME as a boost for technology transfer in industrial simulative pipelines"*.

In this E-CAM workshop, we addressed in particular the following questions:

1. which is the most appropriate propelling element of innovation, top-level academic science or industrial needs of accelerating R&D towards novel and cheaper products?

   Traditionally, the approach to technology is conceiving technology as a corollary of scientific research. However, there is compelling evidence that for several mid-term projects an industry-requirements-driven approach is largely feasible if not best suited. A tightly connected topic regards on how to match and synchronise curiosity driven research with industrial needs and how to manage the resulting, possibly academic/industrial mixed, intellectual property.

2. This led us to consider the further question, can this 'engineering' or 'politechnique' approach to science/technology transfer be the way to boost the technological SMEs European tissue?

Talks from academia and industry were interleaved in the meeting. This allowed to complement and contrast experiences and knowledge coming from academia and industry.

Industrial requirements, expressed in particular by scientists from the drug discovery field, emerged quite clearly and consisted in the need of fast (albeit approximate) methods for analysing ligands. This analysis should include both thermodynamic and kinetic properties with very high efficiency (overnight production of results). It was also stated that pharma companies are quite sceptical about the real effectiveness of machine learning when applied to docking or thermodynamics prediction.

The role of the SME also emerged quite clearly. It was underlined that even though fundamental methodologies (such the creation of an interaction field) comes from academia, commercial products need to re-elaborate and recast methods to be effective. This observation created a certain consensus among the audience that while the idea is coming from academia, the innovation and the application to the real world problem often comes from the SME, whose success is tightly linked to its ability to address real industrial needs.

The role of software developers in SMEs and academia was also discussed. Different points of view emerged in the discussion. For some participants, software engineering is a fundamental component to any success in the computational field either commercial or academic software. For others, even for SME it is more important to hire personnel with a strong background in chemistry or physics, than in software developments.

There was also a consensus that EU funded Centers of Excellence for Computing Applications can provide an opportunity to enhance the expertise and scope of software vendors SMEs.

The following needs were highlighted by the academia and the industry participants.

- Academia:
    1. include in the research group software engineers able to properly code methods or ensure access to sustained consultancy and assistance in this area.;
    2. identify means to ensure a certain degree of time continuity in the code development and maintenance;
    3. identify and assist to assess the potential for technology transfer of in-house activities.
- Industry:
    1. increase the dialog between industry and academy;
    2. seed academia with real world problems whose solution industry could benefit also in economic terms.

There was unanimous consensus that SMEs are playing a significant role in bridging the two worlds. However, this can only work with a proper "feedback-cycle": academic talks with SME/Industry and vice-versa in a non-interrupted loop to foster high level science that on the long term could be applied to industrial problems. Opportunities to create and sustain this cycle were indicated as still insufficient and absolutely crucial.

The EU Centres of Excellence might provide an environment to systematize, host and enhance in-house software developments and foster technology transfer. However, in the discussion industrial participants, and in particular software vendors SMEs, raised the point that activities that are, or can be perceived, as establishing EU funded Centers of Excellence as competitors of SMEs should be avoided. The promotion of collaborative efforts was also indicated as a necessity.

A complete workshop report is also available.

4.3 E-CAM and the adoption of Machine Learning within the community

One of the E-CAM programmers has become a certified NVIDIA Ambassador due to having a regular collaboration with NVIDIA, not only through the NVIDIA Deep Learning Institute (DLI) for dissemination and tutorials, but also for optimization in porting software to multi-GPU as well as Deep Learning applications applied mainly to computer vision industrial problems. NVIDIA DLI Ambassador status was achieved in October 2018 and gaining NVIDIA Ambassador experience in Deep Learning has opened exciting opportunities in so-called *Naive Science*: the idea to use neural networks for replacing traditional computational science solvers. A Neural Network can be trained using real or simulated data and then used to predict new properties of molecules or fluid dynamic behaviour in different systems. This will speed up the simulation by a couple of orders of magnitude as well as avoiding complex modeling based on the use of ad hoc parameters that are often difficult to determine.

E-CAM participated in a course on Machine Learning in Scientific Computing in Nierstein (Germany) which gave an extensive introduction to the principal methods used in several scientific fields, ranging from Computer Vision to Quantum Dynamics simulations. This covered most of the E-CAM soft matter community interest and was very useful to give an understanding of the possible applications and potential future projects within E-CAM. Moreover, the course had a strong mathematical formalism adapt to give and understanding for usage as well as development of new algorithms specifically tuned for our community. Across the different approaches, Deep Learning has been briefly presented, but well supported by examples in Quantum Chemistry and Biology. Other applications, from Bayesian derived Classical Force Fields to clustering of Biological Networks, sparsely covered the wide range of our Working Packages interests. Finally, the afternoon hands-on sessions, mainly focused on the usage of Python and TensorFlow for Deep Learning programming, gave a basic introduction for building Neural Networks adapted to any type of scientific application and were ideal for a beginner in this field.

E-CAM also organised the PRACE & E-CAM Tutorial on Machine Learning and Simulations, a 4-day school in March 2020 which focused on providing the participants with a concise introduction to key machine and deep learning (ML & DL) concepts, and their practical applications with relevant examples in the domain of molecular dynamics, rare-event sampling and electronic structure calculations. ML is increasingly being used to make sense of the enormous amount of data generated every day by MD and electronic structure simulations running on supercomputers. This can be used to obtain mechanistic understanding in terms of low-dimensional models that capture the crucial features of the processes under study, or assist in the identification of relevant order parameters that can be used in the context of rare-event sampling. ML is also being used to train neural network based potentials from electronic strucutre which can then be used on MD engines such as LAMMPS allowing orders of magnitude increase in the dimensionality and time scales that can be explored with electronic structure accuracy. The first half of thw school covered the fundamentals of ML and DL, with the second half will be dedicated to relevant examples of how these techniques are applied in the domains of molecular dynamics and electronic structure.

4.4 Interacting with the European exascale roadmap

The European HPC Technology Platform, ETP4HPC, is an industry-led think-tank comprising of European HPC technology stakeholders: technology vendors, research centres and end-users. The main objective of ETP4HPC is to define research priorities and action plans in the area of HPC technology provision (i.e. the provision of supercomputing systems). It has been responsible for the production and maintenance of the European HPC Technology Strategic Research Agenda (SRA), a document that serves as a mechanism to provide contextual guidance to European researchers and businesses as well as to guide EU priorities for research in the Horizon 2020 HPC programme, i.e. it represents a roadmap for the achievement of European exascale capabilities. This roadmap is now being defined within the context of the EuroHPC objectives (see Section 2.3.1 for more details).

We have had numerous discussions of the E-CAM community software needs through our exchanges with ETP4HPC during the course of our contributions to the SRA. The particular contribution from our discussion related to the software needs for exascale computing within the ETP4HPC SRA report is shown in the paragraphs below:

> E-CAM has not committed itself to a single set of applications or use cases that can represented in such a manner, it is instead driven by the needs of the industrial pilot projects within the project (as well as the wider community). Taking into consideration the CECAM community and the industrial collaborations targeted by E-CAM, probably the largest exa-scale challenge is ensuring that the skillsets of the application developers from academia and industry are sufficiently up to date and are aligned with programming best practices. This means that they are at least competent in the latest relevant language specification (Fortran 2015, C++17,...) and aware of additional tools and libraries that are necessary (or useful) for application development at the exa-scale. For application users, this means that they have sufficient knowledge of architecture, software installation and typical supercomputing environment to build, test and run application software optimised for the target.
>
> While quite specific "key applications" are under continuous support by other CoEs, this is not the current model of E-CAM. E-CAM is more likely to support and develop a software installation framework (such as EasyBuild) that simplifies building the (increasingly non-trivial) software stack of a particular application in a reliable, reproducible and portable way. Industry has already shown significant interest in this and E-CAM is particularly interested in extending the capabilities of EasyBuild to Extreme-scale Demonstrators (EsD) architectures, performance analysis workflows and to new scientific software packages. Such an effort could easily be viewed as transversal since such developments could be leveraged by any other CoE.

One important focus of the SRA is the development of the Precursors to Exascale Supercomputers that are vehicles to optimise and synergise the effectiveness of the entire HPC H2020 programme (see Figure 9 for a list of European HPC technology projects involved, including E-CAM), through the integration of R&D outcomes into fully integrated HPC

system prototypes.

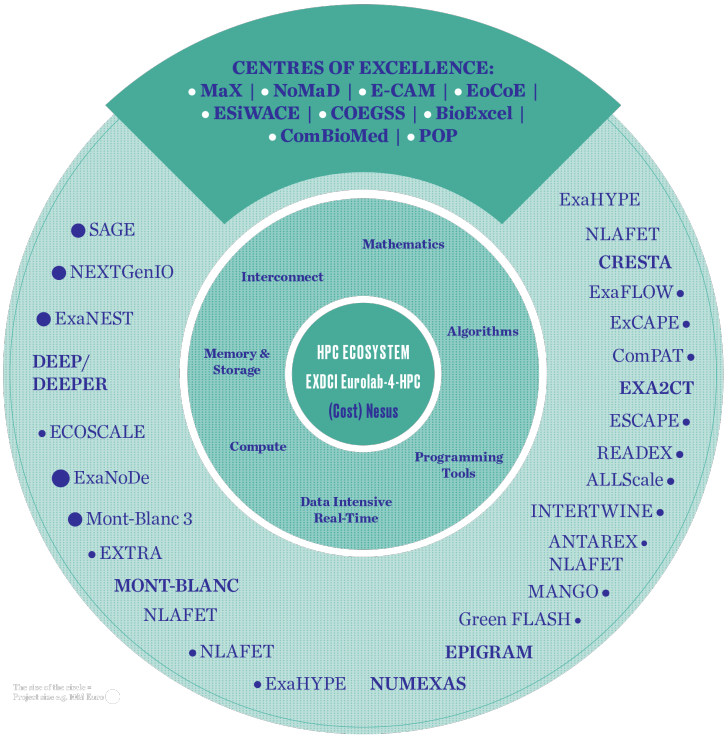

Figure 9: The European HPC Technology Projects within the European HPC Eco-system.

The work in developing these will fill critical gaps in the H2020 programme, including the following activities:

- Bring technologies from FET-HPC projects closer to commercialisation.
- Combined results from targeted R&D efforts into a complete system (European HPC technology ecosystem).
- Provide the missing link between the three HPC pillars: technology providers, user communities (e.g. E-CAM) and infrastructure

As one of the CoEs, E-CAM should aim to provide insight and input into the requirements of future exascale systems based on lessons learnt from activities within E-CAM (e.g. software development and relevant performance optimisation and scaling work). This would entail further knowledge and understanding within E-CAM on exploiting current multi-petaflop infrastructures, what future exascale architectures may look like, as well as interaction and close collaboration between E-CAM and other projects (i.e. the projects shown in Figure 9); these are also covered in subsequent sections of this paper.

4.4.1 FocusCoE

As part of the second round of CoE funding there was a coordination and support action for CoEs. FocusCoE was funded to contribute to the success of the EU HPC Ecosystem and the EuroHPC Initiative by supporting the EU HPC CoEs to more effectively fulfil their role within the ecosystem and initiative: ensuring that extreme scale applications result in tangible benefits for addressing scientific, industrial or societal challenges.

Part of this coordination effort is assisting the CoEs to interact with the EuroHPC hardware-related projects which the vendors themselves are engaged with. As such, we see FocusCoE as a concerted communication channel with ETP4HPC and EuroHPC for E-CAM and are extremely supportive of initiatives such as the HPC CoE Council where these topics are discussed.

4.5 Other discussions with hardware and software vendors

E-CAM people participated the 2017's Teratec forum and discussed with software and hardware vendors. NVidia presented the top ten HPC applications accelerated by GPU, among them, Gaussian, Gromacs, NAMD, VASP, AMBER are CECAM's community codes. This shows the extent of the efforts carried out by the community.

Bull presented to us the prototype of Exascale-focused ARM-based Sequana X1310 supercomputer. And the Mixed Intel Xeon-Xeon Phi based Sequana X1000 machine will be available at Spring 2018 in French TGCC center with up

to 20 Petaflops. We've also discussed with researchers from the CEA-UVSQ-Intel joint Exascale Computing Research Lab. We're very interested in their exascale efforts for one of the CECAM's community code, Abinit. They presented their work via poster at the forum. They used the MAQAO and CERE tools to analyse the Abinit code performance and use the MPC framework to facilitate the paralleling programming. In collaboration with the Abinit developers, they run directly this code on Intel specific architecture by developing an *hardware abstraction layer*. This may provide a good example, or possible orientation, for the other important community codes which have a real need to go to exascale.

E-CAM also had a booth at the PASC18 conference and presented a paper there on "Task-Based Parallelization of Replica Exchange Transition Interface Sampling in OpenPathSampling". Having a booth allowed us the opportunity to engage in discussion with software vendors (in particular) to understand where potential collaborations may lie.

4.6 WP7 Pilot Projects

To engage at a practical level with the challenges we have exposed (both in this Section and Section 3), E-CAM has two ongoing pilot projects that filter these challenges in two particular directions:

- An HTC project, jobqueue_features, that recognises the role of ensemble calculations in the E-CAM community and provides an adaptive framework that efficiently maps it to extreme-scale resources. This project was initially carried out in collaboration with Partnership for Advanced Computing in Europe (PRACE) and implemented as a Python library.
- A Load-balancing Library (ALL) which aims to provide an easy way to include dynamic domain-based load balancing into both particle and mesh based simulation codes. The library is developed in the Simulation Laboratory Molecular Systems of the Juelich Supercomputing Centre at Forschungszentrum Juelich.

In the past we have also looked at deploying HPX within an E-CAM application (detailed in Section 4.6.3 below) however, going forward, we recommend evaluating the KOKKOS framework when considering the porting of an E-CAM application to such a layer.

4.6.1 HTC in E-CAM

Within the E-CAM CoE target community there are a large number of use cases where the same molecular dynamics application can be run many thousands of times with varying inputs. The goal of the mini-project will be to allow scientific community to leverage large-scale resources to efficiently manage this form of high-throughput computing.

A collaborative project with PRACE was created to address this by preparing a library for optimising molecular dynamics simulations in terms of task scheduling for an initial particular use case of OpenPathSampling using the Dask framework. The project was approved after review within PRACE and an initial version of the library was released in Q1 2019. A follow-on project in collaboration with PRACE is due to complete in August 2020.

A number of additional use cases for HTC within E-CAM across all WPs have been identified. It has also been shown to be of interest to our industrial contacts during the DPD Scoping Workshop, where it was reported that "The main role exascale could play is in rapid screening of candidate systems for formulated product companies. This requires accurate models, efficient computation and appropriate workflows to be available". The full report is available here.

In 2018 we have organised a transversal ESDW to help address HTC use cases. The second part of the HTC workshop took place in July 2019 and focused exclusively on the E-CAM library.

4.6.2 A Load-balancing Library

At Jülich Supercomputing Centre (JSC), a library for dynamically redistributing work in particle- or mesh-based simulation codes is being developed. Currently it is developed under the name "A Load-balancing Library" (ALL) which contains a number of methods (e.g., tensor product decomposition, staggered mesh, topological mesh or Voronoi based methods) which iteratively modify the computational domain geometries to re-balance the computational load on each processor. Somewhat more expensive methods (based on d-dimensional work distribution functions) distribute the work within one library call and are suited for, e.g., static load balancing or initial partitioning of the simulation domain. The library is based on C++ template programming model to deal with different data types that describe domain boundaries and domain based work. In order to encapsulate the data, ALL uses a class in which the required data and the computed results of a load-balancing routine are saved and can be accessed. It is an ongoing project that will be reported in more detail in the remaining WP7 deliverables.

An ESDW focusing on ALL was part of the 2018/2019 ESDW program, where developers from 8 community codes came together to discuss collaborative opportunities, and included efforts to initiate integration of ALL into their codes. A second part of the ESDW was held in June 2019, with hands-on foreseen for the integration and testing of the library in part of the codes. A third part is planned for a later stage of the current project phase.

4.6.3 Leveraging HPX within E-CAM

HPX is a C++ runtime system for parallelism and concurrency. It aims to implement the C++ standard for parallelisms and concurrency as well as expanding and improving the functionality of the standard. Whereas the C++ standard only handles parallelism within a node, HPX has functionality for multi-node parallelism as well.

HPX handles parallel constructs using light weight *tasks*, instead of heavier OS threads. HPX runs one thread per core, each of these have their own task scheduler which is responsible for running the tasks, once a task has finished another task is run without switching the OS thread, this allows HPX to efficiently support a large number of tasks. One example where these tasks are used is to implement an extended version of std::future from the standard C++ library. The HPX version has more functionality than the one from the C++ standard with among other things the ability to invoke a second operation once then future has completed through the use of `hpx::future::then`. The HPX version also comes with the ability to more easily compose futures, waiting for or adding an action to perform when some, any or a given number of them have completed.

Distributed computing in HPX is handled by giving the programmer the ability to call functions and create objects on any node participating in running the program. In HPX functions that can be applied at remote locations are referred to as *actions*, these can be regular functions or member functions. These actions need to be defined using a set of macros before they are callable remotely. Lets say we have a function `some_global_function`, to be able to call it remotely we need to create the macro `HPX_PLAIN_ACTION(app::some_global_function, some_global_action)` which now gives us `some_global_action` which can be called on a remote location. Classes which can be created remotely are referred to as *components*, like the functions these need to be defined using a special macro to enable it to be created on remote locations, any member functions that can be called remotely also needs to be declared with a macro similar to the regular remote functions.

The default configuration is for HPX to run one *locale* per compute node. Effectively one process per node and then within this process shared memory parallelism is used, similar to running MPI communication between nodes and OpenMP within the node. Data transfers between the hosts are handled by the HPX *parcel transport layer* [7] which will move data around to the places where it is needed. Any data structures moved need to have methods to serialize and deserialize defined for them so that the data can be serialized before it is moved and then remade back into an object at the receiver.

HPX also supports distributed containers, i.e. a C++ vector where the elements are distributed over multiple hosts. Most of the parallel algorithms from the C++ STL are also implemented in HPX, additionally HPX adds the ability to use these on the distributed containers it implements.

ESPResSo++ and HPX

The ESPResSo++ code is built as a library to be called from python. The code itself is parallelized using MPI and is treating every core as an individual MPI rank. Both of these design characteristics cause some difficulties if one wants to port the code to use HPX.

Firstly, HPX needs its own runtime active for any HPX functionality to work. The regular way presented in the HPX documentation involves a specialized HPX main function that is called from the regular main function and the runtime is only active in the HPX main function. For libraries where the main function is not controllable by the library this becomes tricky to work with.

Secondly, since the code is originally designed to only utilize MPI, turning it into a hybrid parallel code is not straight forward, especially if one wants to exploit task based parallelism. For a simple version one could simply identify hot loops in the code and apply shared memory parallelism to them and still do MPI like communication between the hosts. But in that case simply applying OpenMP directives to the loops would likely be a better first step as it would involve far less code modification.

To get any real benefits from implementing the program with HPX would likely require a larger rewrite of the codebase with HPX and shared memory task based parallelism in mind. Without direct and deeper involvement from the ESPResSo++ development team this is not feasible.

HPX as a whole

HPX shows promise as a distributed computing framework. The ability to remotely create objects and call functions enables different programming paradigms from regular MPI type parallelism.

Implementing the C++ standard for parallelism and concurrency and providing extensions to it also provides a great starting point. In the future, one could in theory take a C++ code parallelized with what the standard provides and quickly convert it to a distributed program by utilizing HPX.

However, HPX also has some issues. The reliance on macros to register components and actions makes implementing them complex and requires significant bookkeeping. The programmer needs to remember to which macros each function/object names need to be passed and with what names to call them afterwards. Any errors made when calling these macros end up with long, unintelligible compiler errors.

The documentation for HPX is also at times sparse and possibly out of date, the project is still under development and things presented a few years ago are not necessarily accurate anymore. The best place to look for how certain functions and features are to be used is often the unit tests for that particular feature.

4.7 Feedback for software developers

E-CAM is attempting to act as a point of contact for the community to the wider EU HPC ecosystem, distilling information and highlighting projects and programs that may be of specific interest or potential impact to the community.

E-CAM can also act as the connection to industry, and continues to engage in particular with software vendors, looking at what model for software adoption or commercialisation would be mutually beneficial for industrial and academic partners (given the constraints of both). Furthermore, as pointed out in the E-CAM Scoping Workshop dedicated to software vendor SMEs, E-CAM can play a crucial role in improving software vendor SMEs competitiveness through the co-development of software to be included in their work-flows.

E-CAM has identified hardware abstraction layers as a potential key technology for the wider application space. The specific E-CAM software developments focus on aspects that are cross-sectional to the E-CAM community, targeting critical improvements to the efficiency of typical user workloads while remaining hardware independent. One example of which is the E-CAM HTC effort, which targets the ensemble workloads common to our users while providing capabilities that are not available with other solutions (such as interactive supercomputing and direct access to the memory space of executed tasks). The other example is the load-balancing library, ALL, which is implemented abstractly and provides a collection of methods that allow the user to experiment with multiple load-balancing algorithms implemented by the library.

5 Conclusions

This deliverable is the final in a series that has grown organically in scope over the course of the project. As the series comes to an end, we can now reflect on the relevance of *hardware developments* to the E-CAM community. The scope of this relevance is very dependent on whether we are in a *user* or *developer* context.

For the application user communities, they clearly would like to see the latest hardware supported by their chosen application, but they typically do not contribute to implementing this so have no targeted interest in hardware developments outside of the compute resources they have access to. For the developer communities, novel technologies are a huge development project and the human resources to implement support for these are sometimes difficult to come by. E-CAM sees middle-layer libraries (such as Kokkos for accelerators, HDF5 for I/O, ALL for load-balancing) as a good way to relieve these burdens somewhat going forward. As a community, we would suggest that the E-CAM community adopt proven libraries that fulfil such roles rather than embark on complex development projects with narrow target architectures. At this point, middle-layer technologies seem mature enough to do this.

A scientific area of E-CAM that is gaining growing attention and which has industrial appeal is multiscale modelling, which potentially has high impact for overcoming computational limits in simulations of systems presenting several time- and length-scales and couple different types of algorithms. It should allow new insight into phenomena and processes of scientific and industrial interest, e.g. computational study of structure formation and processes at mesoscopic length and time scales. It aims to capture details that relate to emergent material properties at the macroscopic scales. While a number of quantum mechanical and classical molecular dynamics simulation engines have been developed to study processes that take place at microscopic scales (such as CHARMM, GROMACS, LAMMPS, etc.), simulation engines that can easily address novel multiscale treatments are severely lacking. In addition, many coarse-graining maps (which are a basic tool of multiscale treatments) are created a priori using conceptual skills as much as technological ones. Taken together, these computational and scientific hurdles make the application of most of the available multiscale modelling technologies far from straightforward for non-experts. This greatly limits the exploitation potential of these new tools, and thus the breakthrough that they may represent in various parts of the materials research community and industry.

Most scientific groups that develop, implement, validate, and apply multiscale modelling approaches generally lack the human resources for the development of easy-to-use and versatile code for various platforms as well as the user base that is required for thorough validation. This reflects both the novelty of the field and the diversity of the approaches that are currently proposed and tested, and results in a situation where newcomers with an interest in particular techniques have to rely on the information found in scientific publications that generally provide neither sufficient insight to judge their applicability for other systems nor contain all the information needed to get started.

When addressing multiple scales there are significant methodological challenges, including algorithm design for horizontal coupling of scales (where all scales are simulated at once) and mapping techniques for vertical coupling (distinct, but coupled, simulations at different scales, with each scale posing different simulation challenges). In vertical coupling, there are the additional impacts of the coarsening that is typically carried out when moving between scales which requires the scientific validation of the resulting model. The current lack of scalable computational tools and coupling frameworks for these approaches makes it obvious that there is a high need for their development in order to make HPC attractive for industrial users, dealing with real world complex systems, presenting complex levels of length- and time-scale coupling but with high societal impact, e.g. in chemical- or pharmacological-industry or materials design.

The coupling of scales is necessarily complex and frequently requires an adaptive workflow (either internally through libraries, or externally through additional application codes) orchestrating a simulation across different scales that is highly susceptible to scalability issues, in particular load-balancing problems. This complexity quickly exposes the advantage of using, and contributing to, middle-layer libraries since they can be leveraged across the workflow and help to minimise the effort of adapting to new architectures.

Eurolab4HPC has already indicated that there are several initiatives moving in this direction, and that developing for current and future high-end architectures, as found in HPC centres, will require such a shift:

> Manual optimization of the data layout, placement, and caching will become uneconomic and time consuming, and will, in any case, soon exceed the abilities of the best human programmers.

Nascent fields such as multiscale modelling which have scientific use cases that should map well to exascale resources would be wise to heed such advice.

5.1 Open Science

A final word related to Open Science, E-CAM has continuously encouraged the adoption of best practices within the E-CAM community and advocates for a proactive approach to open science in the research domain. This will help to

facilitate a Declaration on Research Assessment (DORA) type research evaluation looking at the full range of research outputs (including software) and not just journal articles. It is all very well calling for better software engineering, but if this work is not adequately recognised in career reviews and grant applications it will not happen at the required scale. The recommendations in this report are hard work but are of great value to the community, we should all be pushing for DORA-type evaluations of researcher output to properly incentivise and reward the hard work recommended in this report.

A E-CAM Extreme-Scale State-of-the-Art Workshop, 6-7 July 2017 Barcelona

This appendix is a summary of the content and significance of the E-CAM Extreme-Scale State-of-the-Art Workshop which was held from from the 6th to the 7th of July 2017 in Barcelona. Due to the rapidly progress in the HPC hardware and funding environment over the project lifetime, some of the impact of the outcomes and discussions wane over time. Nevertheless, the significance of discussion at the time remains and we chose to include it here.

This appendix also provides a detailed account and record of the event itself.

A.1 Emerging hardware architectures relevant to exascale computing (in 2017)

The European Commission supports a number of projects in developing and testing innovative architectures for next generation supercomputers, aimed at tackling some of the biggest challenges in achieving exascale computing. They often involve co-design involving HPC technologists, hardware vendors and code developer/end-user communities in order to develop prototype systems. Some of these projects include:

- The DEEP (Dynamic Exascale Entry Platform) projects (DEEP, DEEP-ER and DEEP-EST)
- The Mont-Blanc projects (Mont-Blanc 1, 2 and 3)
- The PRACE PCP (Pre-Commercial Procurement) initiative

We discuss further the first two of these since they were of explicitly discussed during the workshop. E-CAM is also now represented in the PRACE PCP (Pre-Commercial Procurement) initiative.

A.1.1 The DEEP projects – overview of the architecture

The DEEP projects have been developing a modular approach to extreme-scale computing via an innovative "cluster booster architecture". In the initial DEEP and DEEP-ER projects, they have successfully integrated a standard, Infiniband cluster using Intel Xeon nodes (Cluster Nodes) that is connected to a novel, highly scalable Booster consisting of Intel Xeon Phi co-processors (Booster Nodes) connected via an underlying EXTOLL high-performance 3D torus network (see Figure 10).

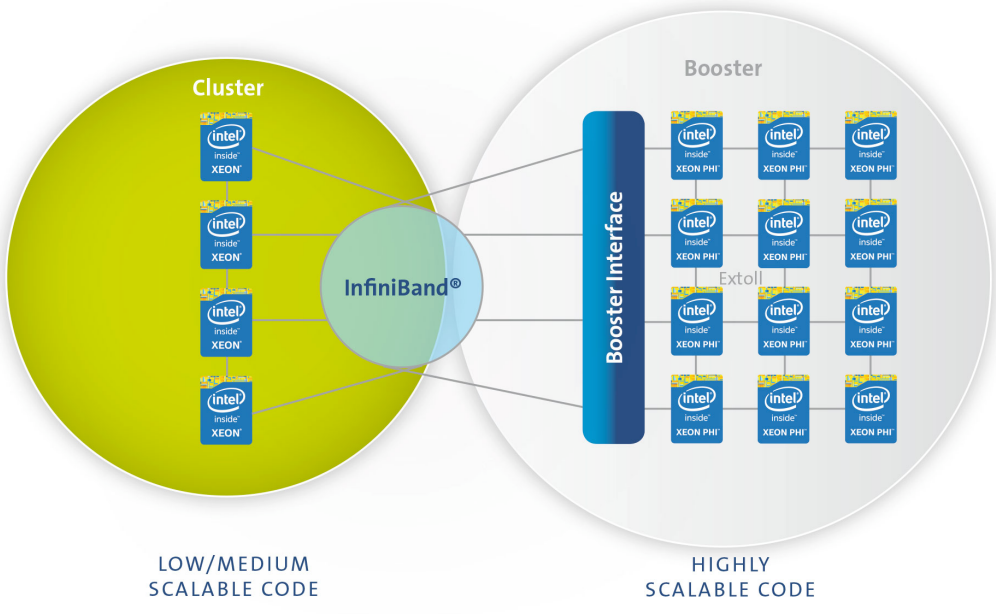

Figure 10: The DEEP hardware architecture.

This architecture enables high throughput and scalability on the Booster component, aimed at highly scalable code execution that supports parallelism via proven and standards-based programming models such as MPI and OmpSs. Parts of the code with more limited scalability (e.g. owing to complex control flow or data dependencies) are executed on the Cluster component, with transparent bridging of both interconnects to ensure high-speed data transfer. Innovative aspects:

- on-node memory, network attached memory.

- Novel fabric, uses ASIC-based network controller designed and produced by EXTOLL.
- Self booting Xeon Phi nodes.

Special optimisations:

- Resilience
- I/O: SIONlib, Exascale10

The project makes heavy use of co-design with applications selected from the beginning of the project. In the next phase, the DEEP-EST project will generalise the cluster-boosters approach by extending the modular concepts of the current system. Apart from cluster and booster (Intel Xeon Phi) components, additional components will be developed with architectures adapted for data-intensive and data analysis applications.

A.1.2 The Mont-Blanc projects

The Mont-Blanc projects focus on developing prototype HPC systems based on energy-efficient embedded technologies. It has developed several ARM-based computational platforms, leveraging some of the technologies that have emerged from the highly-competitive low-power mobile computing market, along with the requisite software stack to facilitate development and execution of complex parallel codes.

During the initial two phases of the project, which begun in 2011, it has examined a range of embedded and mobile processor architectures. The activities have included the development of small-scale, mini-clusters built using a variety of node architectures, e.g. an Nvidia Tegra K1 SoC with ARM Cortex processors and Kepler GPU, or a Samsung Exynos 5 SoC combined with Cortex processors and the ARM Mali T-604 GPU. The first official prototype for the project has since been built and operational since 2015. This system consists of 1,080 compute cards where each card or node consists of a Samsung Exynos 5 Dual SoC, powered by ARM Cortex-A15 processors (2x cores) and an ARM Mali-T604 GPU along with 4GB of LPDDR3-1600 memory, 64 GB of local storage on microSD and 10 GbE networking bridged via USB 3.0.

The project is currently in its third phase, coordinated by Bull, the Atos brand for technology products and software. It adopts a co-design approach to ensure that the hardware and system innovations can be translated into HPC application performance and energy efficiency. Here, a new demonstrator system, named Dibona, is being built based on the 64-bit ThunderX2 processors from Cavium® that implements the ARM v8 instruction set. The full system will eventually be composed of 48 compute nodes with 96 ThunderX2 CPUs or 3,000 cores in total.

Apart from hardware developments, the project has also invested significant efforts in developing a software ecosystem for ARM-based architectures in HPC. These include the porting and extension of performance tools to work on ARM-based platforms (e.g. Score-P, Scalasca, and BSC's performance analysis toolset including Extrae, Paraver and Dimemas). The Mont-Blanc project has also contributed to the OmpSs parallel programming model to support ARMv8 64-bit processors, further enabling it to exploit multiple cluster ARM-based nodes along with transparent application check-pointing for fault tolerance. Furthermore the project also developed a novel power measurement infrastructure in order to determine power efficiency that is key to exascale. This takes high-frequency measurements of power consumption at the granularity of a single compute node, and up to the entire prototype system.

A.2 Exascale challenges in exploiting massive parallelism

This section is a summary of various talks presented at the workshop that are related to the discussion of exascale challenges in exploiting massive parallelism. The presentations include those from application developers, code developers and users within E-CAM and beyond, and representatives from other CoEs.

A.2.1 "Large scale electronic structure calculations with CP2K" - Jurg Hutter

CP2K is a popular, freely available software package written in Fortran and used for atomistic and molecular simulations of solid state, liquid, molecular, periodic, material, crystal, and biological systems. It provides a framework for different modelling methods such as DFT and wave-function methods (two methods that are highlighted in the talk) as well as many others. The talk focused on some of the large-scale (towards exascale) challenges and contributions from the CP2K development team. The talk introduced the audience to the use of sparse matrix libraries for KS-DFT simulations for large scale computations. By using a blocked compressed sparse row format, followed by load balancing using randomized indices, Cannon's algorithm can then be applied, with computations optionally offloaded to accelerators, to achieve better scaling (up to $\frac{1}{\sqrt{N}}$, which could be a bottleneck for up to 1 million processes). Some of the results shown included the enhanced performance of a mini-app (using the LIBCUSMM library) on the NVidia Pascal architecture relative to its previous Kepler architecture, with up 20-50% increase in flops. In another example dominated by matrix multiplications, the performance figures (Fig.11) showed comparable performance between a

GPU node (on the Daint system) compared to a (*less expensive*) KNL node. More recent developments try to address the bottleneck of Cannon's method, including the use of remote memory access to replace point-to-point communication, and using a 2.5-dimensional variant of Canon's algorithm at the expense of higher memory usage. These have resulted in performance gains of 1.2-1.4x on large scale systems of up to 50,000 cores and 4,000 GPUs. Application of these methods have enabled researchers to carry out large-scale energy calculations on large systems, e.g. 1 million atoms of a STM virus, albeit only for a few femtoseconds.

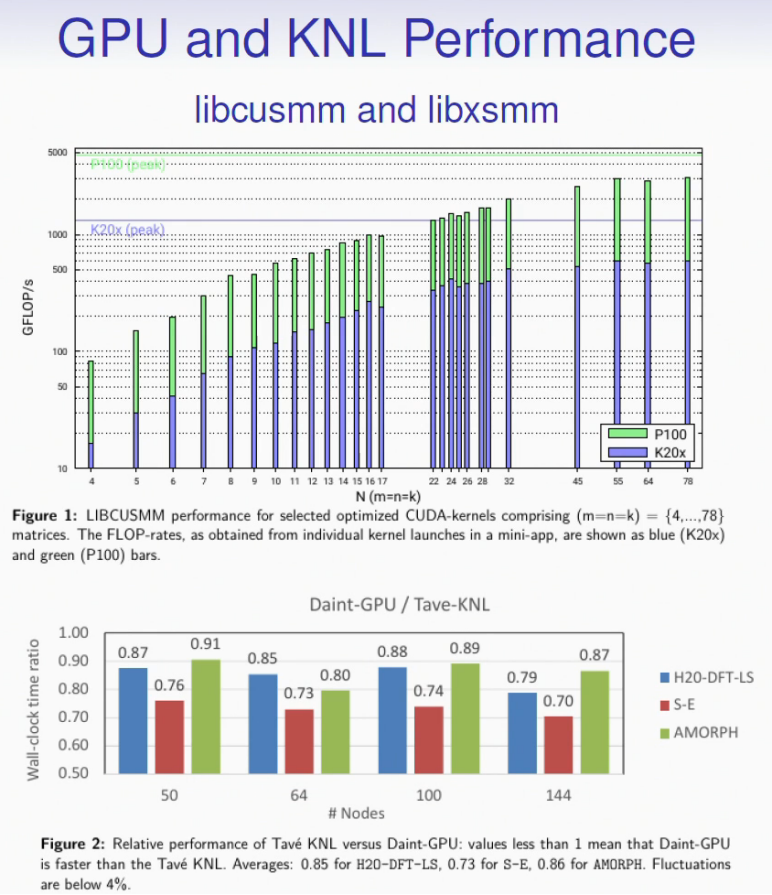

Figure 11: CP2K Presentation: GPU and KNL Performance

For wavefunction methods, the talk introduced a number of Resolution of Identity (RI) approaches based on Gaussian and plane waves (GPW), including RI-dRPA, RI-MP2, RI-G0W0. The switch to new algorithms using sparse tensor representations and overlap metric have allowed scaling to thousands of nodes for large systems.

In future, this DBCSR library can be generalised for sparse tensor algebra, meaning that tensor contractions can be mapped to matrix multiplications. This tensor library is still in development at the time of this report.

A.2.2 "Scalability of Path Sampling Simulations" – David Swenson

Path sampling methods, such as transition path sampling and transition interface sampling, are Monte Carlo simulations in the space of trajectories. They provide powerful tools for studying topics such as chemical reactions, crystallization, binding processes, and conformational changes in biomolecules. Path sampling software can leverage the high-performance capabilities of existing tools for molecular dynamics simulation and analysis, as well as producing additional scalability by running multiple trajectories in parallel. As a part of E-CAM, the OpenPathSampling (OPS) software package has been extended to support widely-used and highly-scalable molecular dynamics engines (e.g. OpenMM, LAMPPS).

The talk gave an overview of using OPS to monitor the trajectories space in order to handle rare events. This is carried out using a Python library that interface with the simulation engine directly (for those with Python APIs such as OpenMM, LAMMPS) or indirectly (e.g. file-based interaction for Gromacs). In terms of performance, the overhead from OPS was shown to be less than the variance due to Gromac's performance tuning. In realistic simulations, the cost of the underlying dynamics dominates OPS's performance, i.e. its scalability (towards exascale) is comparable to that of the underlying engine itself such as Gromacs. Finally, additional functionalities are being developed for the library, including support for running simultaneous trajectories (see Section 4.6).

A.2.3 "PaPIM: A Code for (Quantum) Time Correlation Functions" – Momir Malis

The Phase Integration Method (PIM) is a novel approximate quantum dynamical technique developed for computing systems time dependent observables. PIM employs an algorithm in which the exact sampling of the quantum thermal Wigner density is combined with a linearized approximation of the quantum time propagators represented in the path integral formalism that reduces the evolution to classical dynamics.

The quantities of interest can then be computed by combining classical molecular dynamics algorithms with an original generalised Monte Carlo scheme for sampling the initial conditions and averaging them over all sampled points in phase space. Because all trajectories are mutually independent, the algorithm can be efficiently parallelised. The work within E-CAM have resulted in novel Fortran90 PaPIM code parallelised using MPI, or PaPIM. While PaPIM has shown good scalability up to several thousand cores, with other parts of the code also having further potential for parallelisation. These include computing the polymer chains in parallel, parallel approaches for individual potential energy calculations (already being examined by other groups) and the decoupling of classical trajectory propagation. Finally, since PIM requires a multitude of input parameters, a large-scale parallel search for optimal sets of PIM sampling parameters will be useful and are planned as future work for the project.

A.2.4 "Porting DL_MESO_DPD on GPUs" – Jony Castagna

DL_MESO is a general purpose mesoscale simulation package developed by Michael Seaton written in Fortran90 and C++ which supports both Lattice Boltzmann Equation (LBE) and Dissipative Particle Dynamics (DPD) methods. The talk gave an overview of some of the challenges in porting the relevant Fortran90 code to Nvidia GPUs.

One of the main problems first encountered was the random memory access pattern, caused by particle locations being stored in a contiguous manner, which can be fixed by re-organising the cell-linked array to ensure coalescent memory access. The GPU port, when executed on a Tesla P100 GPU, showed weak scaling of up to 4x speed-up over an Intel Ivy Bridge 12-core processor (See Fig.12); this is some way off the theoretical ideal speed-up of 12× due to clustering of particles and hence load imbalance amongst threads (which is to be addressed in future work).

One key recommendation from this talk, which may be applicable for GPU porting work in general, is to maintain two versions of the code, one codebase for the GPU while the other in plain Fortran/C, and that new physics should only be introduced into the latter before porting work to the GPU.

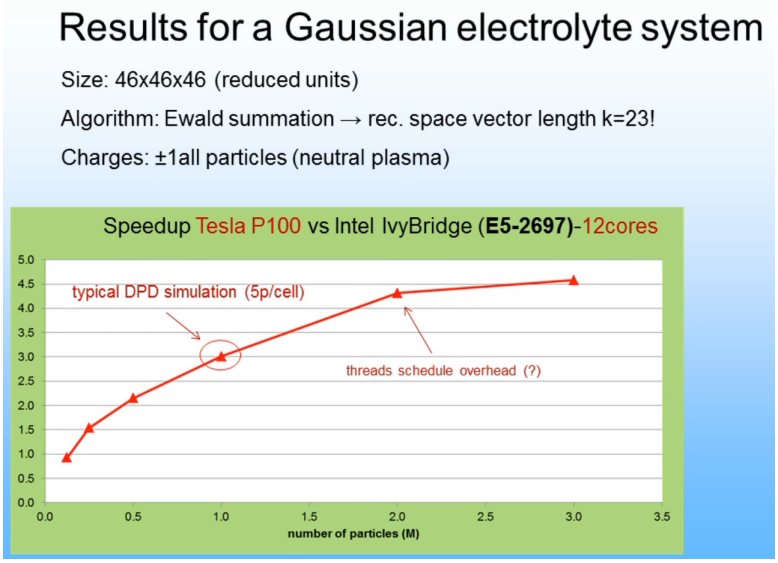

Figure 12: DL_MESO: GPU porting results

A.2.5 "MP2C: Multiple-particle collision dynamics on massively parallel computers" – Godehard Sutmann

The talk focuses the scalable implementation of MPC or MP2C for massively parallel computers. MP2C is a grid-free method for mesoscale simulation of hydrodynamic media mainly based on the concept of particles, bridging the gap between microscopic simulations on the atomistic level and macroscopic calculations on the continuum level by solving the Navier-Stokes equations in different types of discretisation. With a small set of paramters (particle density, scattering angle, mean free path of a particle), it is possible to reproduce hydrodynamic behaviour and map collective interactions to a local particle based algorithm.

The talk covered the hybrid domain partitioning methods used by MP2C to distribute work to the processors, including typical communication patterns and load balancing issues. This enables a coupling between molecular dynamics (MD) simulations and a mesoscopic solvent, taking into account hydrodynamic interactions between solutes. Both MPC and MD computations have demonstrated strong scaling on the JUGENE system. It then went on to become one of the first codes to demonstrate scalability across the whole JUQUEEN system (458k processor cores compared to 262k cores on the JUGENE, see Fig. 13) MP2C has also been adopted as a benchmark for parallel I/O using the SionLib I/O library, which have been able to achieve I/O reads of >100 GB/s and writes of 60 GB/s.

The final part of the talk discussed in more details regarding load balancing issues, caused by highly heterogeneous particle distribution, meaning inefficient use of the HPC resources (i.e. some cores will be busy while others left idle). The solutions include further distribution of workloads at the expense of more data traffic and communication, and adjustment of the volume of domains at the expense of more complex geometries and communication patterns. These approaches to address load imbalance play a pivotal role in the code's scalability.

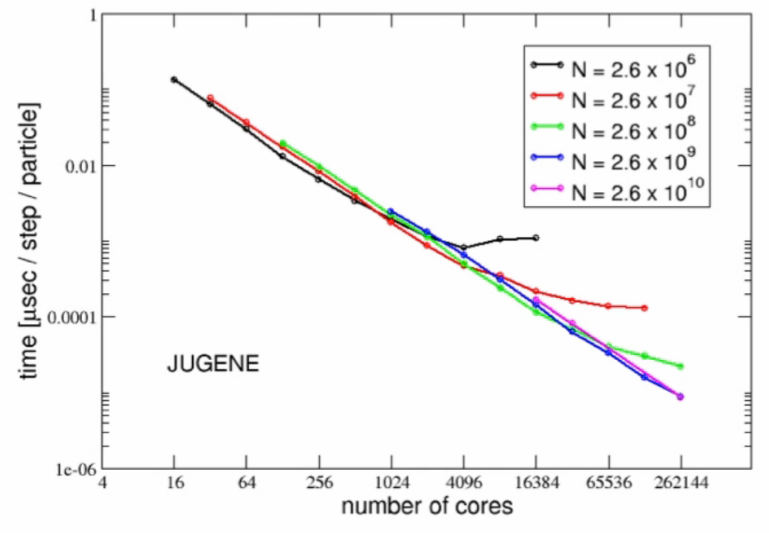

Figure 13: MP2C scaling on JUGENE

A.2.6 "POP – understanding applications and how to prepare for exascale" – Jesus Labarta

The POP CoE projects aims to apply common analysis methodologies to a large base of customer codes to increase the awareness of fundamental issues that limit the scalability and performance of HPC applications. The project offers assessment of application performance and helps customers of the POP service to refactor applications in directions that maximize the performance and scaling revenue at minimum cost.

Due to increasingly complexity and variability of novel HPC architectures, there seems to be a growing divergence between the ways systems behave, and the way developers expect them to behave. Hence the promoted vision of POP is to decouple the underlying hardware system from the application (where the science happens) via different software layers (e.g. programming model layer and runtime layer, Fig.14). An example of activities already being undertaken at the programming model layer includes the StarSs paradigm (encompassing OmpSs that have developed features being incorporated into OpenMP). This is also discussed in Section 3.2.6.

The POP CoE offers three levels of services: application performance audit, application performance plan and proof-of-concept software demonstrators. Its target customers include code developers (looking to assess performance and behaviour), application users (looking for potential improvements for specific conditions/set-ups), infrastructure operators (looking for information for allocation purposes or training of support staff) and vendors (looking to carry out benchmarking or system design guidance). Based on a hierarchy of fundamental performance factors (Fig. 15), the presentation provided a number of case studies where POP has improved the efficiency and scalability of applications it has evaluated. In total, it has analysed well over 50 codes.

A.2.7 "Exascale challenges: the view from MaX" – Carlo Cavazzoni

For the MaX CoE (HPC for materials research and innovation), the goal of enabling the exascale transition involves the modernisation of community codes, making them ready to exploit future exascale system for material science simulations. The presentation initially focused on some of the exascale challenges. With Dennard scaling – constant power density despite transistors getting smaller – compute core frequency and performance no longer follow Moore's Law. A key issue for compute performance is how it can be increased with the least amount of power. In order to achieve

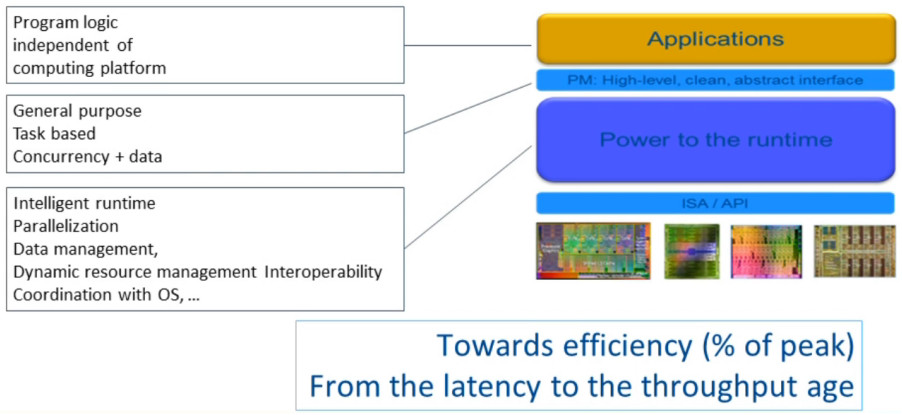

Figure 14: POP: Different software layers

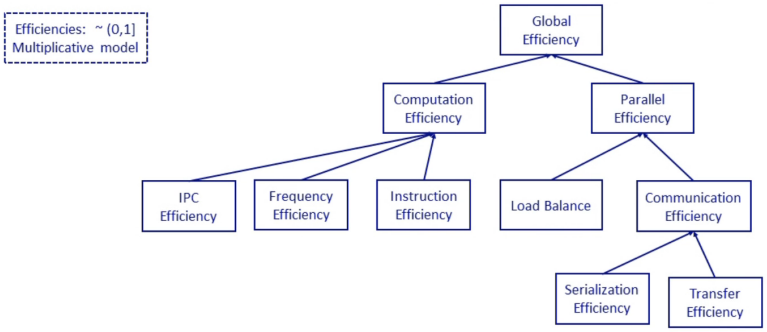

Figure 15: Fundamental performance factors in evaluating performance

this, chips that have been designed for maximum performance for all possible workloads (i.e. conventional CPUs) are increasingly being supplemented by new chips designed for maximum performance in a small set of workloads (e.g. many-core technologies).

It was shown that future exascale systems (with current technologies) will necessarily have 10^9 Floating Point Units (FPUs). This may be configured as 10^5 FPUs in 10^4 nodes, or 10^4 FPUs in 10^5 nodes. In either case, the use of MPI to distribute code to 10^4 or 10^5 nodes may not the main issue, but how applications make use of the 10^4 or 10^5 FPUs, i.e. many-core technologies, within single nodes. Furthermore, exascale is not only about scalability and flops performance, but moving data in and out of 10^9 FPUs poses tremendous challenges based on current technology. In addition, effective exploitation of exascale will have to contend with heterogeneous architectures and deep memory hierarchies compared to today, and considerably more attention paid to workloads and workflows.

As an example of the CoE's activities on applications, some of the core modules from the Quantum Espresso software package have been factored out and transformed into "mini-apps", self-contained ready-for-deployment mini applications that can be used to guide development of future hardware. The advantage in developing these mini-apps, although time and effort consuming, is that it makes it much easier for designers of future hardware to study key aspects of scientific applications without having to deal with typically very large codebases. MaX is also actively engaged in profiling efforts (in collaboration with POP), and the implementation of novel OpenMP and MPI features into some of its key applications.

A.2.8 "Exascale challenges: the view from NoMAD" – Claudia Draxl

The NOMAD CoE presentation focused on some of the big data challenges for materials research towards the exascale era, where the amount of data produced on workstations, compute clusters and supercomputers is growing exponentially but most are thrown away. An overview of materials data and their structure was shown (Fig.16), and an example was provided where the calculation for a single thermoelectric figure (at level IV) involves the generation of large amounts of intermediate data (at levels I thru III). In order to allow researchers to better exploit such data, the NoMAD repository is being established to host, organise and share materials data. It accepts input and output files from all major codes, and contains over 5 million entries at the time of this report. This has facilitated, for example, studies in DFT calculations of solids, where results from different electronic calculation software can be compared in order to assess data quality.

Apart from the repository, a NoMAD Encyclopedia has been developed that provides an easy way for researchers to

Level	Properties	Methods	Size
I	Atomic positions and nuclear charges, properties of free atoms, symmetry, temperature, pressure	Input: definition of material *gene*	10 kB - 10 MB
II	Total energy, electron density, potential, wavefunctions, atomic forces, optimized geometry, elastic constants, etc.	Density-functional theory (DFT) and *ab initio* molecular dynamics (MD)	10 MB - 10 TB
III	Excitation energies, dielectric screening, matrix elements of Coulomb interaction, etc. optical spectra, electrical conductivity, phonon spectra, thermal conductivity, etc.	Many-body perturbation theory (MBPT), DF perturbation theory, *ab initio* MD	1 GB - 1 TB
IV	Efficiency of solar cell, thermoelectric figure of merit, turn-over frequency of catalyst, etc. as a function of temperature and pressure	Modeling, output derived from levels I-III *phenotype*	10 kB - 1 MB

Figure 16: Overview of materials data and their structure

explore properties of given materials that have been previously computed, complemented by visualisation tools. Going further, the NoMAD Analytics Toolkit allows researchers to study correlations and structure in the data repository, enabling scientists and engineers to identify materials for potential use in novel products. In summary, the approach towards exascale is very much data-oriented, i.e. how to improve on ways to handle and better exploit the exponential increase in data quantity and complexity.

A.2.9 "Exascale challenges: the view from EoCoE" – Matthieu Haefele

EoCoE is an energy oriented Centre of Excellence with four main thematic pillars: meteorology, materials, water and fusion all supported by a transversal group focused on advanced maths linear algebra and HPC tools and techniques. Some of their contributions towards exascale include design of numerical schemes, software package improvements and some efforts on porting applications to new architectures. They also provide support for performance evaluations for the different communities.

With regards to exascale software stack improvements, the CoE has in-house expertise across different areas such as numerical method improvements, linear algebra packages (e.g. parallel sparse direct solvers, hybrid solvers, including promotion of European linear algebra technologies), parallel I/O (e.g. FTI, XIOS, SIONlib, PDI). On application performance, EoCoE has defined 28 performance metrics (Fig.17), and automated and reproducible methods to collect them using JUBE.

	Metric name	03/01/2016
	Test-case	case1
Golbal	Total Time (s)	43.2
	Time IO (s)	0.3
	Time MPI (s)	12.4
	Memory vs Compute Bound	1.1
	Load Imbalance MPI	24.8
IO	IO Volume (MB)	35.8
	Calls (nb)	384000
	Throughput (MB/s)	105.0
	Individual IO Access (kB)	0.1
MPI	P2P Calls (nb)	0
	P2P Calls (s)	0.0
	P2P Message Size (kB)	0.0
	Collective Calls (nb)	2721
	Collective Calls (s)	0.1
	Collective Message Size (kB)	908.4
	Synchro / Wait MPI (s)	11.7
	Ratio Synchro / Wait MPI	94.8
Node	Time OpenMP (s)	0.0
	Ratio OpenMP	0.0
	Time Synchro / Wait OpenMP	0.0
	Ratio Synchro / Wait OpenMP	0.0
Mem	Memory Footprint (B)	66 mB
	Cache Usage Intensity	N.A.
Core	IPC	N.A.
	Runtime without vectorisation (s)	46.5
	Vectorisation eff ciency	1.1
	Runtime without FMA (s)	44.6
	FMA eff ciency	1.0

Figure 17: EoCoE Performance metrics table

In collaboration with POP, EoCoE holds performance evaluation workshops that brings both code developers and HPC experts together. During the workshop, both groups work together to carry out the performance analyses, generate the performance metrics and producing the performance report. This allows the code developers to gain knowledge on performance tools and the capability to re-run the performance evaluation. Through this type of approach, it has

been shown that the resulting performance improvements have saved over 40 million core hours for one study alone (2.5× speed-up), while another case where a 12× speed-up has paved the way for simulations of larger systems.

However, the author (not necessarily the view across EoCoE) has raised some concerns about the "mini-app" route to exascale that are being undertaken by some. The argument is that while the mini-app itself may be factored out of an application and adapted for exascale systems, re-integration of these mini-app back into the applications can often be problematic (e.g. the physics may need to change when executed on new architectures, breaking the original relationship to the code factored out into the mini-app). The proposed approach, then, is to concentrate resources on a small number of applications, centred on key scientific projects, and start already the port of relevant parts of these applications (holistically) on new architectures.

References

Acronyms Used

HPC High Performance Computing
CECAM Centre Européen de Calcul Atomique et Moléculaire
HTC High Throughput Computing
HPDA High Performance Data Analysis
PRACE Partnership for Advanced Computing in Europe
ESDW Extended Software Development Workshop
WP Work Package
EsD Extreme-scale Demonstrators
MPI Message Passing Interface
PGAS Partioned Global Address Space
CoE Center of Excellence
FPUs Floating Point Units
IP Intellectual Property
EPI European Processor Initiative
GPP General Purpose Processor
ALL A Load-balancing Library
JSC Jülich Supercomputing Centre

URLs referenced

Page ii

https://www.e-cam2020.eu ... https://www.e-cam2020.eu
https://www.e-cam2020.eu/deliverables ... https://www.e-cam2020.eu/deliverables
Internal Project Management Link ... https://redmine.e-cam2020.eu/issues/48
a.ocais@fz-juelich.de ... mailto:a.ocais@fz-juelich.de
http://creativecommons.org/licenses/by/4.0 ... http://creativecommons.org/licenses/by/4.0

Page iii

E-CAM Extreme-Scale State-of-the-Art Workshop ... https://www.cecam.org/workshop-2-1512.html

Page 3

Eurolab4HPC Long-Term Vision on High-Performance Computing (2nd Edition) ... https://www.eurolab4hpc.eu/media/public/vision/Eurolab-Vision_2.pdf

Page 4

Eurolab4HPC Long-Term Vision on High-Performance Computing (2nd Edition) ... https://www.eurolab4hpc.eu/media/public/vision/Eurolab-Vision_2.pdf

Page 6

RISC-V ... https://en.wikipedia.org/wiki/RISC-V
Intel X^e architecture ... https://itpeernetwork.intel.com/intel-xe-compute/#gs.ev5wh0
Ponte Vecchio ... https://www.anandtech.com/show/15119/intels-xe-for-hpc-ponte-vecchio-with-chiplets-e
Aurora A21 exascale system ... https://www.anl.gov/article/us-department-of-energy-and-intel-to-deliver-f

Page 7

Ampere ... https://devblogs.nvidia.com/nvidia-ampere-architecture-in-depth/
first supercomputer with A100 GPU card ... https://atos.net/en/2020/press-release/general-press-releases_2020_05_14/atos-launches-first-supercomputer-equipped-with-nvidia-a100-gpu
partnership ... https://nvidianews.nvidia.com/news/nvidia-and-tech-leaders-team-to-build-gpu-accelerat
FPGAs ... https://www.xilinx.com/products/silicon-devices/fpga/what-is-an-fpga.html
ARM+FPGA Exascale Project ... https://www.hpcwire.com/2017/09/07/eu-funds-20-million-euro-armfpga-exasc
ExaNeSt ... http://www.exanest.eu/
EcoScale ... http://www.ecoscale.eu/
ExaNoDe ... http://www.exanode.eu/
Eurolab4HPC Long-Term Vision on High-Performance Computing (2nd Edition) ... https://www.eurolab4hpc.eu/media/public/vision/Eurolab-Vision_2.pdf

Page 8

CUDA 10.0 ... http://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html
OpenMP 5.0 ... https://www.openmp.org/wp-content/uploads/OpenMP-API-Specification-5.0.pdf
OpenACC open standard ... https://www.openacc.org/specification

Unified Virtual Addressing ... https://devblogs.nvidia.com/parallelforall/beyond-gpu-memory-limits-unified
Tensor Core ... https://devblogs.nvidia.com/parallelforall/cuda-9-features-revealed
NVIDIA education site ... https://developer.nvidia.com/cuda-education
presentation on performance of the Clang OpenMP 4.5 implementation on NVIDIA GPUs ... http://on-demand.gputechconf.com/gtc/2016/presentation/s6510-jeff-larkin-targeting-gpus-openmp.pdf

Page 9

EoCoE ... https://www.eocoe.eu/
June 2018 webinar on FPGA computing ... https://youtu.be/ptN67kwLpBM
EU HPC policy ... https://ec.europa.eu/digital-single-market/en/policies/high-performance-computing
EuroHPC ... https://ec.europa.eu/digital-single-market/en/eurohpc-joint-undertaking
Hosting Entities for Precursors to Exascale Supercomputers ... https://eurohpc-ju.europa.eu/participate.html
Hosting Entities for Exascale Supercomputers ... https://eurohpc-ju.europa.eu/participate.html
PRACE calls ... http://www.prace-ri.eu/call-announcements/
DECI calls ... http://www.prace-ri.eu/deci-13-call/

Page 11

E-CAM Survey of Application Software ... https://docs.google.com/forms/d/e/1FAIpQLSc9i-vb8VNLm-KqC7Wy4i6d1m viewform?c=0&w=1

Page 13

ISO/IEC standard for the C++ programming language ... https://www.iso.org/standard/68564.html
Standard Template Library ... https://en.wikipedia.org/wiki/Standard_Template_Library
multi-threading programming model for C++ ... https://en.wikipedia.org/wiki/C%2B%2B11#Multithreading_memory_model
continued introduction of expert-level features into C++ may drive away newcomers ... https://www.theregister.co.uk/2018/06/18/bjarne_stroustrup_c_plus_plus/
guidelines to help people adopt modern C++ ... https://github.com/isocpp/CppCoreGuidelines/blob/master/CppCoreGuidelines.md
Fortran 2018 ... http://fortranwiki.org/fortran/show/Fortran+2018
coarray ... https://en.wikipedia.org/wiki/Coarray_Fortran
Software Engineering and Parallel Programming in Modern Fortran ... https://www.cranfield.ac.uk/courses/short/aerospace/software-engineering-and-parellel-programming-in-modern-fortan
Advanced Fortran Programming ... https://events.prace-ri.eu/event/852/
compiler support for the latest Fortran standards ... http://fortranwiki.org/fortran/show/Compiler+Support+for+Modern+Fortran
LLVM Flang Fortran compiler ... https://fosdem.org/2020/schedule/event/llvm_flang/attachments/slides/3839/export/events/attachments/llvm_flang/slides/3839/flang_llvm_frontend.pdf

Page 14

Dask ... https://dask.org/
jobqueue_features ... https://github.com/E-CAM/jobqueue_features
official porting guide ... https://docs.python.org/3/howto/pyporting.html
GASNet ... https://gasnet.lbl.gov/
MPI-3.1 ... http://mpi-forum.org/docs/mpi-3.1/mpi31-report.pdf
scope of past and future updates to the MPI standard ... https://www.lrz.de/services/compute/courses/x_lecturenotes/Parallel_Programming_Languages_Workshop/MPI.pdf
level of OpenMP support among other compilers ... http://www.openmp.org/resources/openmp-compilers/
features of OpenMP 5.0 under active and heavy development ... https://clang.llvm.org/docs/OpenMPSupport.html#openmp-implementation-details
OpenACC ... https://www.openacc.org/
PGI compiler ... https://www.pgroup.com/index.htm

Page 15

nvcc compiler ... http://docs.nvidia.com/cuda/cuda-compiler-driver-nvcc/index.html
Vulkan ... https://en.wikipedia.org/wiki/Vulkan_(API)
SYCL ... https://www.khronos.org/sycl/
Frontier ... https://www.amd.com/en/products/frontier
HIP ... https://gpuopen.com/compute-product/hip-convert-cuda-to-portable-c-code/
oneAPI ... (https://www.oneapi.com/
Eurolab4HPC Long-Term Vision on High-Performance Computing (2nd Edition) ... https://www.eurolab4hpc.eu/media/public/vision/Eurolab-Vision_2.pdf

Page 16

ROCm ... https://rocm-documentation.readthedocs.io/en/latest/
Frontier ... https://www.amd.com/en/products/frontier
HIP ... https://gpuopen.com/compute-product/hip-convert-cuda-to-portable-c-code/
HPX ... https://github.com/STEllAR-GROUP/hpx
ParalleX ... http://stellar.cct.lsu.edu/pubs/icpp09.pdf
Kokkos ... https://github.com/kokkos/kokkos
OmpSs ... https://pm.bsc.es/ompss
OmpSs-2 ... https://pm.bsc.es/ompss-2

Page 17

Charm++ ... http://charm.cs.illinois.edu/research/charm
charm4py ... https://charm4py.readthedocs.io/en/latest/
Kokkos workshop ... http://training.e-cam2020.eu/datasets/5bbb52d4e4b0d62afd1161c4

Page 18

DORA ... https://sfdora.org/

Page 19

E-CAM Extreme-Scale State-of-the-Art Workshop ... https://www.cecam.org/workshop-2-1512.html
"Building the bridge between theories and software: SME as a boost for technology transfer in industrial simulative pipelines" ... https://www.cecam.org/workshop-details/174

Page 20

complete workshop report ... https://www.e-cam2020.eu/wp-content/uploads/2019/05/E-CAM-SCOW-IIT-Report_web.pdf

Page 21

Machine Learning in Scientific Computing ... https://www.cecam.org/workshop-details/164
PRACE & E-CAM Tutorial on Machine Learning and Simulations ... https://www.e-cam2020.eu/event/prace-e-cam-tutorial-in-simulation-and-machine-learning/?instance_id=83
European HPC Technology Strategic Research Agenda (SRA) ... http://www.etp4hpc.eu/en/news/18-strategic-resear.html
EasyBuild ... http://easybuild.readthedocs.io/en/latest/
Precursors to Exascale Supercomputers ... https://eurohpc-ju.europa.eu/participate.html#inline-nav-1

Page 23

PASC18 ... https://pasc18.pasc-conference.org/
jobqueue_features ... https://github.com/E-CAM/jobqueue_features
OpenPathSampling ... http://openpathsampling.org/latest/
Dask ... https://github.com/dask/dask
DPD Scoping Workshop ... https://www.e-cam2020.eu/event/scoping-workshop-dissipative-particle-dynamics
here ... https://www.e-cam2020.eu/wp-content/uploads/2018/09/SCOW_Dissipative-particle-dynamics_REPORT.pdf
transversal ESDW ... https://www.e-cam2020.eu/event/extended-software-development-workshop-intelligent-?instance_id=57
second part of the HTC workshop ... https://www.e-cam2020.eu/event/4424

Page 24

HPX ... http://stellar-group.org/libraries/hpx/

Page 27

DORA ... https://sfdora.org/

Page 28

E-CAM Extreme-Scale State-of-the-Art Workshop ... https://www.cecam.org/workshop-2-1512.html
E-CAM Extreme-Scale State-of-the-Art Workshop ... https://www.cecam.org/workshop-2-1512.html

Page 30

DBCSR library ... https://dbcsr.cp2k.org/

Page 32

POP CoE ... https://pop-coe.eu/
MaX CoE ... http://www.max-centre.eu/

Page 33

NOMAD CoE ... https://nomad-coe.eu/

Page 34
EoCoE . . . http://www.eocoe.eu/

Citations

[1] A. O. Cais, J. Castagna, and G. Sutmann, "D7.7: Hardware developments iv," Jun. 2019. [Online]. Available: https://doi.org/10.5281/zenodo.3256137

[2] D. Borgis, L. Liang, L. Petit, M. Lysaght, and A. O. Cais, "Hardware Developments I - A Survey of State-of-the- art Hardware and Software," Jul. 2016. [Online]. Available: https://doi.org/10.5281/zenodo.929533

[3] L. Liang, A. O'Cais, J. Castagna, S. Wong, and G. Sanchez, "Hardware developments II," Oct. 2017. [Online]. Available: https://doi.org/10.5281/zenodo.1207613

[4] A. O'Cais, L. Liang, and J. Castagna, "Hardware developments iii," Jul. 2018. [Online]. Available: https://doi.org/10.5281/zenodo.1304088

[5] R. Friedman, "C++ Parallel STL Introduced in Intel Parallel Studio XE 2018 Beta," May 2017. [Online]. Available: https://insidehpc.com/2017/05/parallel-stl/

[6] D. M. Fernández, D. Graziotin, S. Wagner, and H. Seibold, "Open science in software engineering," *CoRR*, vol. abs/1904.06499, 2019. [Online]. Available: http://arxiv.org/abs/1904.06499

[7] T. Heller, P. Diehl, Z. Byerly, J. Biddiscombe, and H. Kaiser, "Hpx–an open source c++ standard library for parallelism and concurrency," 2017. [Online]. Available: http://stellar.cct.lsu.edu/pubs/heller_et_al_opensuco_2017.pdf