1 The DEEP-EST project

Estela Suarez(1), Anke Kreuzer(1), Norbert Eicker(1,2), Thomas Lippert(1,3)

(1) Jülich Supercomputing Centre, Forschungszentrum Jülich GmbH, Leo Brandt Strasse, 52428 Jülich, Germany

(2) Fakultät für Mathematik und Naturwissenschaften, Bergische Universität Wuppertal, Gaußstraße 20, 42119 Wuppertal, Germany

(3) Goethe-Universität Frankfurt, Frankfurt Institute for Advanced Studies (FIAS). Ruth-Moufang-Straße 1, 60438 Frankfurt am Main, Germany

e.suarez@fz-juelich.de

1.1 Introduction

DEEP-EST (standing for *Dynamical Exascale Entry Platform – Extreme Scale Technologies*) is a research and development project funded by the European Commission through the Horizon 2020 framework program. It has run for 45 months from June 2017 until March 2021 under the coordination of Forschungszentrum Jülich (FZJ) and with the participation of 16 institutions from 9 European countries (see Figure 1.1).

UNIVERSITY OF ICELAND
Norwegian University of Life Sciences
ASTRON
epcc
KU LEUVEN
JÜLICH
lrz
ParTec
MEGWARE
Fraunhofer ITWM
intel
EXTOLL
Barcelona Supercomputing Center
CERN
NCSA

Forschungszentrum Jülich GmbH *(coord.)*
ParTec Cluster Competence Centre GmbH
Intel Deutschland GmbH
Bayerische Akademie der Wissenschaften
Barcelona Supercomputing Center
Megware Computer Vertrieb und Service GmbH
Ruprecht-Karls-Universität Heidelberg
EXTOLL GmbH
The University of Edinburgh
Fraunhofer Gesellschaft zur Förderung der Angewandten Forschungs e.V.
Katholieke Universiteit Leuven
Stichting Astron, Netherlands Institute For Radio Astronomy
Association National Centre For Supercomputing Applications
Norges Miljo-Og Biovitenskaplige Universitet
Haskoli Islands
European Organisation for Nuclear Research

Figure 1.1: The DEEP-EST consortium

DEEP-EST is, after its predecessors DEEP and DEEP-ER, the third member in the DEEP project series[1] funded by the European Commission and driven by a stringent codesign spirit, in which hardware-, software-, and application developers work tightly together to develop holistic solutions to today's HPC challenges. A new breed of HPC systems is needed to support the computation and data processing requirements of both traditional high performance computing (HPC) and emerging high performance data analytics (HPDA) workloads. To do so, DEEP-EST implements the **Modular Supercomputing Architecture (MSA)**, a novel system-level design to integrate heterogeneous resources and match the requirements of a wide spectrum of application fields, ranging from computationally intensive high-scaling simulation codes to data-intensive artificial intelligence workflows.

1.2 Modular Supercomputing Architecture (MSA)

The Cluster-Booster concept first implemented by DEEP broke with the traditional system architecture approach (based on replicating many identical compute nodes, possibly integrating heterogeneous processing resources within each node) by integrating heterogeneous computing resources in a modular way at the system level[2]. More precisely, it connected a standard HPC cluster based on general-purpose processors with a cluster of many-core processors or accelerators (the "booster") by way of a highly efficient and high-speed network. No constraints are put on the combination of Cluster and Booster nodes that an application may select, and resources might be reserved dynamically.

The Modular Supercomputer Architecture (MSA)[3] introduced in DEEP EST takes the Cluster-Booster architecture to the next step generalizing the concept to fulfil the requirements of a wider variety of applications from HPC and HPDA domains (see Figure 1.2). In the MSA several modules – each one tuned to best match the needs of a certain class of algorithms – are connected to each other at the system level to create a single heterogeneous system. Each module is a parallel, clustered system of potentially large size. A federated network connects the module-specific interconnects,

[1] www.deep-projects.eu

[2] N. Eicker, Th. Lippert, Th. Moschny, and E. Suarez, *The DEEP Project - An alternative approach to heterogeneous cluster-computing in the many-core era*, Concurrency and computation: Practice and Experience, Vol. 28, p. 2394–-2411 (2016), doi = 10.1002/cpe.3562. http://juser.fz-juelich.de/record/203150/files/concurrency-paper.pdf

[3] E. Suarez, N. Eicker, Th. Lippert, "*Modular Supercomputing Architecture: from idea to production*", Chapter 9 in Contemporary High Performance Computing: from Petascale toward Exascale, Volume 3, pp 223-251, Ed. Jeffrey S. Vetter, CRC Press. (2019) https://juser.fz-juelich.de/record/862856

while an optimised resource manager enables assembling arbitrary combinations of these resources according to the application workload requirements. This has two important effects: Firstly, each application can run on a near-optimal combination of resources and achieve excellent performance. Secondly, all the resources can be put to good use by combining the set of applications in a complementary way, increasing throughput and efficiency of use for the whole system.

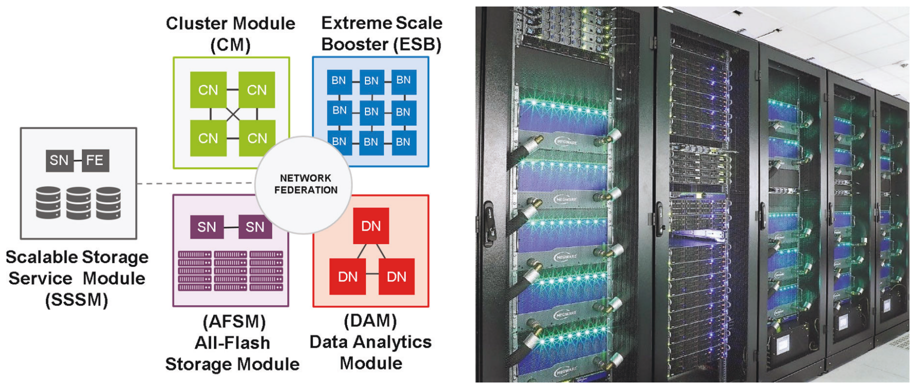
Cluster Module (CM)
Extreme Scale Booster (ESB)
CN
BN
SN
FE
NETWORK FEDERATION
DN
Scalable Storage Service Module (SSSM)
(AFSM) All-Flash Storage Module
(DAM) Data Analytics Module

Figure 1.2: The DEEP-EST prototype. *Left*: Architecture scheme. *Right*: picture at JSC's computer room

1.3 Hardware

The DEEP-EST hardware prototype (Figure 1.2) has been defined in close co-design cooperation between applications, system software and system component architects. It includes three computing modules and two storage modules. The modules are connected through a high-speed network and, most importantly, operated with a uniform system software and programming environment. This enables applications to be distributed over several modules, running each part of its code on the best-suited hardware.

The computational core of the DEEP-EST system is given by the general purpose **Cluster Module (CM)**, the **Extreme Scale Booster (ESB)**, and the **Data Analytics Module (DAM)**. Their main characteristics are given in Table 1.1. The following subsections give an overview on the different components. More details can be found in the DEEP-EST Wiki[4].

[4] https://deeptrac.zam.kfa-juelich.de:8443/trac/wiki/Public/User_Guide/System_overview

DEEP-EST system	CM	ESB	DAM
Usage and design target	Applications and code parts requiring high single-thread performance and a modest amount of memory, which typically show moderate scalability. General purpose performance and energy efficiency are essential for the CM.	Compute intensive applications and code parts with regular control and data structures, showing high parallel scalability. Energy efficiency, balanced architecture, packaging and hardware scalability are also important aspects in the ESB design.	Data-intensive analytics and machine learning applications and code parts requiring large memory capacity, data streaming, bit- or small datatype processing. Flexibility, non-volatile memory and different acceleration capabilities are key features of the DAM.
Node count	50	75	16
CPU type **CPU codename** **Cores @frequency**	Intel Xeon 6146 Skylake 12 @3.2 GHz	Intel Xeon 4215 Cascade Lake 8 @2.5 GHz	Intel Xeon 8260M Cascade Lake 24 @2.4 GHz
Accelerators per node	n.a.	1× NVIDIA V100 GPU	1× NVIDIA V100 GPU 1× Intel Stratix10 FGPA
DDR4 capacity **HBM capacity** **NVMe** **Node max. mem BW**	192 GB n.a. n.a. 256 GB/s	48 GB 32 GB (GPU) n.a. 900 GB/s (GPU)	384GB+32GB(FPGA) 32 GB (GPU) 3 TB Intel Optane 900 GB/s (GPU)
Storage	1x 512 GB NVMe SSD	1x 512 GB NVMe SSD	2x 1.5 TB NVMe SSD
Network technology **Network Topology**	EDR-IB (100 Gb/s) Fat-tree	EDR-IB (100 Gb/s) Tree	EDR-IB (100 Gb/s) Ethernet (40 Gb/s) Tree
Power /node **Cooling**	500 W warm-water	500 W warm-water	1600 W air
Integration	1× Rack MEGWARE SlideSX-LC ColdCon	3× Rack MEGWARE SlideSX-LC ColdCon	1× Rack MEGWARE

Table 1.1: Main hardware features of the DEEP-EST modular prototype, in its final configuration[5].

[5] During the project lifetime the DAM and one ESB partition featured an EXTOLL Fabri3 interconnect. For better user-experience and easier long-term maintenance in mind, after the end of the DEEP-

1.3.1 The Cluster Module (CM)

The CM is a general purpose HPC cluster with 50 nodes, each with two Intel® Xeon® Scalable (“Skylake” generation) Gold CPUs, 192 GB RAM and one 400 GB NVMe SSD. The nodes are interconnected by an InfiniBand EDR fabric with 100 Gbit/s bandwidth. The CPUs have relatively few cores (12 each) with a high clock frequency of 3.2 GHz. This module provides reliable performance and universal applicability with high single-thread performance, supporting highly complex and dynamic control flow HPC workloads. That means that application (parts) that do not fit well to the other more specialized modules (ESB or DAM) should be executed on the CM.

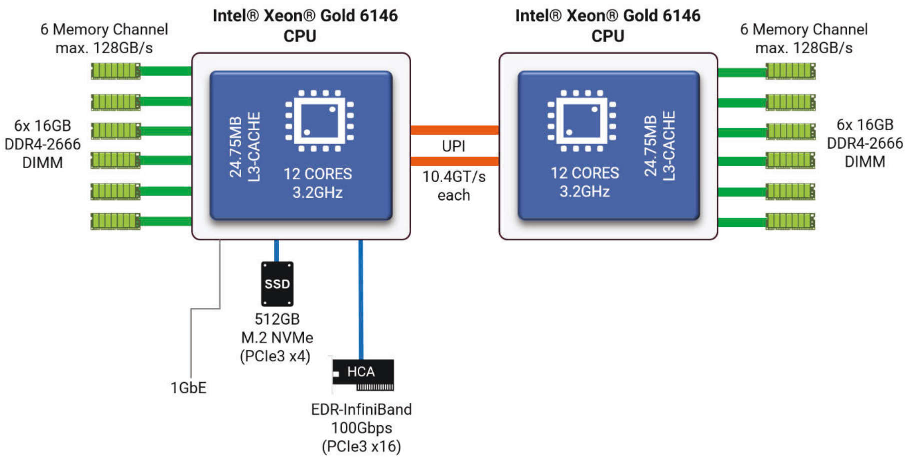
6 Memory Channel max. 128GB/s
Intel® Xeon® Gold 6146 CPU
Intel® Xeon® Gold 6146 CPU
6 Memory Channel max. 128GB/s
6x 16GB DDR4-2666 DIMM
24.75MB L3-CACHE
12 CORES 3.2GHz
UPI
10.4GT/s each
12 CORES 3.2GHz
24.75MB L3-CACHE
6x 16GB DDR4-2666 DIMM
SSD
512GB M.2 NVMe (PCIe3 x4)
HCA
1GbE
EDR-InfiniBand 100Gbps (PCIe3 x16)

Figure 1.3: Architecture of the CM node

1.3.2 The Extreme Scale Booster (ESB)

The ESB targets the needs of highly scalable (parts of) applications and workloads and is the largest module in the DEEP-EST prototype. With its 75 nodes, each containing one Intel Xeon Scalable (“Cascade Lake” generation) Silver CPU, one NVIDIA® Tesla® V100 GPU, 48 GB of RAM, and one 512 GB SSD, the ESB is a highly scalable system, and provides very high computational throughput for applications with very wide parallelism and suitable control structures. Achieving high energy efficiency is a key objective of the system integration. The ESB nodes are thus designed for GPGPU

EST project it was decided to reconfigure the prototype with a uniform, InfiniBand-only interconnect across all modules. Since this book targets future users of this MSA platform, this final configuration is the one described in this table and taken into account across the full volume.

centric computing, with the vast majority of compute operations running on the high-end GPGPU, and the Xeon CPU controlling I/O and network communication (including MPI), as well as managing the GPGPU device attached via a 16-lane PCIe generation 3 link. It targets highly scalable (data, thread and task parallel) HPC applications (or parts thereof) and workloads. The nodes are interconnected via InfiniBand EDR providing a 100 Gbit/s bandwidth.

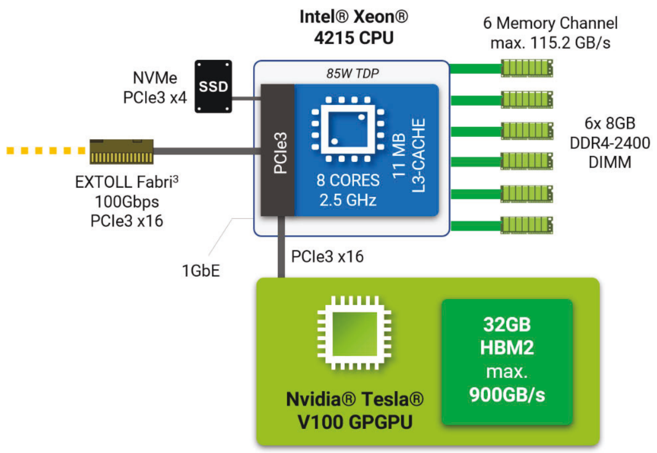

Figure 1.4: Architecture of the ESB node

1.3.3 The Data Analytics Module (DAM)

The DAM is a specifically designed cluster for high-performance data analytics (HPDA) and artificial intelligence (AI) workloads. It is composed of 16 nodes, each with two Intel Xeon Scalable ("Cascade Lake" generation) Platinum CPUs, one NVIDIA Tesla V100 GPU, one Intel® Stratix® 10 FPGA and 384 GB RAM plus 3 TB of Intel® Optane Persistent Memory. The module uses two interconnects in parallel: 100 Gbit/s InfiniBand and 40 Gbit/s Ethernet. In contrast to the ESB, each DAM nodes provides a duo of high-end CPUs, plus one high-end GPGPU and FPGA accelerator each. It is designed for applications that share the computation load between CPUs and accelerators, require large main memory, or can profit from using an FPGA.

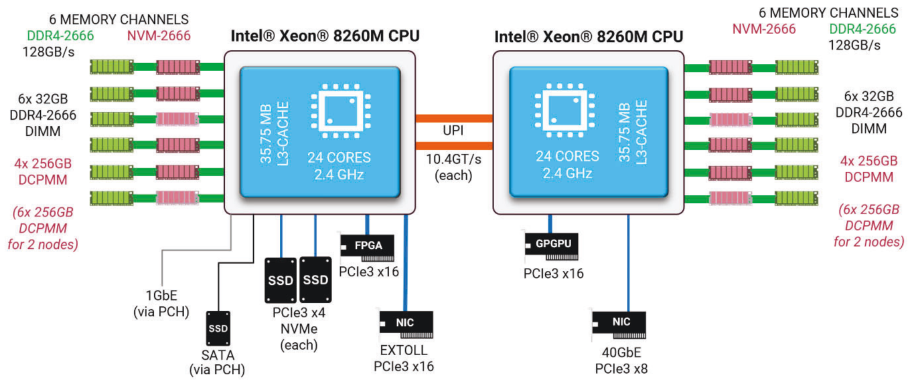

Figure 1.5: Architecture of the DAM

1.3.4 The Scalable Storage and Service Module (SSSM)

The SSSM provides storage capacity based on conventional spinning-disks and uses a parallel file system (BeeGFS). It provides storage capacity for the workloads while they are running on the DEEP-EST prototype. It is not included in any backup scheme. The SSSM is accessible under `/work`. For more information on the storage and file system see Chapter 8, Section 8.6.

1.3.5 The All-Flash Storage Module (AFSM)

The AFSM complements the SSSM and is based on modern PCIe3 NVMe SSD storage devices to provide scalable, high-performance global I/O and storage capabilities and better match the computational power of the DEEP-EST Prototype modules for data- and storage-intensive applications and workloads. On the ASFM, the BeeGFS global parallel file system is used to make 1.8 PB of data storage capacity available, supported by two Metadata servers and six volume data server systems which are interconnected by a 100 Gbps EDR-InfiniBand fabric. The AFSM is integrated into the DEEP-EST EDR fabric topology of the CM, ESB and DAM.

1.4 Software

The DEEP-EST software architecture (see Figure 1.6) was designed in the early stages of the project taking careful account of the requirements determined from the co-design applications. The languages, programming and parallelization paradigms, libraries, and needed tools were integrated in the stack, and platform adaptations were identified and implemented wherever needed.

The lower layers of the software stack have been adapted to provide the best support for the underlying hardware, while hiding these modifications from the end user by keeping the higher-level layers of the stack in APIs familiar to users. For example, low-level interconnect management features were developed and integrated with MPI, but without changing the MPI calls that are directly used by the application developers.

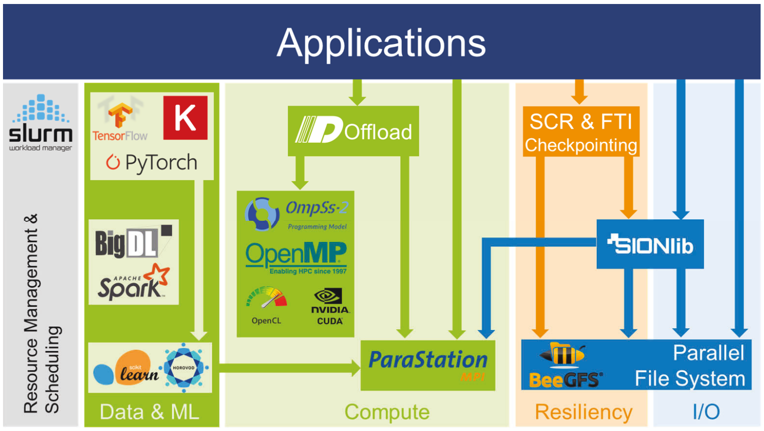

Figure 1.6: Software stack in the DEEP-EST project

The MSA software stack enables application developers to map the intrinsic scalability patterns of their applications and workflows onto the hardware: highly parallel code parts run on the large-scale, energy-efficient Booster, while less scalable code parts can profit from the high single-thread performance of the Cluster, or from the high memory capacity of the Data Analytics Module. Users can freely decide how many nodes to use in each module, so that the highest application efficiency and system usage can be achieved[6].

[6] A. Kreuzer, J. Amaya, N. Eicker, E. Suarez, *Application performance on a Cluster-Booster system*, 2018 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), HCW (20th International Heterogeneity in Computing Workshop), Vancouver (2018), p: 69 - 78. [doi: 10.1109/IPDPSW.2018.00019] https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8425386

1.4.1 Resource management and scheduling

Resource management and job scheduling play a pivotal role for the success of the overall MSA. Existing job schedulers guarantee efficient use of monolithic supercomputers. However, the MSA requires capabilities to manage heterogeneous resources, to enable co-scheduling of resource sets across modules, and to handle dynamically varying resource-profiles. For this reason, the resource management and scheduling software packages psslurm and Slurm[7] have been widely extended to enable an optimal utilization of the heterogeneous resources in a modular supercomputer. These extensions include the ability to dynamically allocate nodes in all compute modules, as well as global resources. Also, better support for the Multiple-Program Multiple-Data (MPMD) programming paradigm has been implemented, which is needed for heterogeneous jobs running different executables on different parts of the job allocation. Furthermore, a new switch (`--delay`) and a clause (`--module list`) have been implemented in Slurm: the former enables workflows consisting of jobs with data dependencies to overlap the executions of their different steps, so that data can be transferred directly between them without writing and then again reading them from the file system; the latter provides in order of preference the list of modules on which the job-steps can run, giving the scheduler more flexibility in the allocation of resources, depending on the demand in the given point on time.

A user may need to pre-process data before running a long simulation, then perform data-reduction, and ultimately visualize the final result. Running these codes on different modules consists simply on indicating to the scheduler on which nodes to execute each step. Data is typically transferred between the jobs of a workflow via the file-system, which means that data is written onto the external storage in one step, and then re-read in the next workflow-step. Taking into account the time and power consumed in such write-read operations, this approach is not necessarily the fastest and certainly not the most energy efficient. Because of that, the DEEP-EST project has investigated the potential implementation and benefits of directly transferring data between workflow steps via MPI.

Workflows running on a modular system architecture can benefit from dynamic scheduling support. Workload trace files including the different project features and specific metrics for workflow analysis were generated and analysed, thus comparing results from modular and homogeneous systems[8]. Special attention was put on

[7] Slurm. https://slurm.schedmd.com/documentation.html

[8] M. D'Amico and J. C. Gonzalez, *Energy hardware and workload aware job scheduling towards interconnected HPC environments*, IEEE Transactions on Parallel and Distributed Systems, doi: 10.1109/TPDS.2021.3090334. https://arxiv.org/abs/2106.12007

evaluating the scheduling features developed within DEEP-EST: module flexibility and workflow dependencies. The system configuration chosen for the scheduling modelling was based on the characteristics of the DEEP-EST prototype and the workload-mix reproduced elements of project's applications. The analysis showed that the new scheduling features provide benefits for application workflows without penalizing traditional jobs.

1.4.2 Programming Environment

The DEEP-EST programming environment has been designed to provide all the functionality required by the co-design applications in the most user-friendly possible manner. The hardware complexity of the MSA is abstracted behind the interfaces and parallel programming paradigms that have become the de-facto standard in HPC: MPI and OpenMP. Specific implementations of these standards have been extended to achieve the maximum application performance on the hardware components constituting the DEEP-EST prototype. MPI and OpenMP are complemented by parallel programming tools supporting acceleration devices (CUDA, OpenACC, OpenCL) and by frameworks for machine learning and deep learning applications (TensorFlow, PyTorch, Keras, Horovod, etc.).

The MSA programming paradigm is based on MPI, in particular using the **ParaStation MPI** implementation[9,10]. In DEEP-EST, ParaStation has been made network topology aware at different levels of the software stack. Besides providing means and extensions for MPI applications to adapt their program flow by creating communicators reflecting the modular architecture, collective MPI operations have been optimised for modular systems and, in particular, those using the latest generation GPUs. For example, ParaStation MPI has been extended with CUDA awareness features to improve both productivity and performance of hybrid MPI codes.

The **OmpSs-2** programming model[11], spearhead of the OpenMP standard and developed by BSC, has been enhanced in the areas of tasking, programmability, support for hardware accelerators, and support for distributed shared memory systems using MPI. A new scheduler-design and dependency system has improved the performance and scalability of the OmpSs-2 runtime on many-core processors. In

[9] ParaStation Modulo. https://par-tec.com/software/

[10] S. Pickartz, Virtualization as an enabler for dynamic resource allocation in HPC, Dissertation, RWTH AachenUniversity, Aachen, 2019. https://doi.org/10.18154/RWTH-2019-02208.

[11] F. Sainz, J. Bellón, V. Beltran, and J. Labarta, "Collective Offload for Heterogeneous Clusters", 2015 IEEE 22nd International Conference on High Performance Computing (HiPC), p. 376-385 (2015) [doi = {10.1109/HiPC.2015.20}]. https://www.bsc.es/printpdf/research-and-development/publications/collective-offload-heterogeneous-clusters

addition, a new lightweight instrumentation plugin has been created to analyse OmpSs-2 applications, and the `taskloop` directive has been extended to support data dependencies. To ease the programming of accelerators, experimental support for OpenACC and array reductions for CUDA have been integrated into the main OmpSs-2 distribution. Also, the TAMPI library has been extended to support MPI RMA by supporting one-sided operations. Many of the new OmpSs features developed in the context of the DEEP-EST project have been already presented to the OpenMP committee for a future inclusion into the OpenMP standard.

Data analytics/machine learning components and frameworks required by the DEEP-EST applications and early access users have been installed and regularly updated on the hardware prototype. Also, the Intel oneAPI implementation has been installed and experiments were done to test its use for programming GPGPU and FPGA accelerators. It is worth mentioning that, in the same manner as it is done on the JSC production machines, the full DEEP-EST software stack has been integrated on EasyBuild, which is used for the maintenance and regular software updates on the prototype.

1.4.3 I/O and resiliency

DEEP-EST has also addressed the topics of I/O and resiliency. The efficient management of data between different modules poses a great challenge for I/O systems, such as **BeeGFS** and **SIONlib**, which leverage new non-volatile memory technologies to cope with this new scenario. Moreover, traditional check-pointing libraries (e.g. FTI or SCR) have been enhanced with new features and a simpler interface to deal with new application requirements.

The European BeeGFS parallel file system, developed by FHG-ITWM, has been enhanced with new features to support storage pools and various storage hardware. BeeOND has also been re-implemented and integrated into the SLURM job manager, allowing system users to create a temporary BeeGFS file system on their allocated nodes with the options that suits their jobs best. Furthermore, the time series-based monitoring solution for BeeGFS (beegfs-mon) has been implemented, released, and integrated into the DCBD monitoring system. Users can now investigate the current and past status of the file system using Grafana panels. JUELICH's SIONlib library has been extended with a new MSA-aware algorithm for the selection of collector processes for collective I/O. The algorithm is portable and relies on platform specific plug-ins to identify processes, which run on parts of the system that are well suited for the role of I/O collector. In reaction to the GPU-based ESB concept, SIONlib's read and write functions have been made CUDA aware. They now allow the user to pass input and output buffers that reside on a CUDA device and transparently handle the transfer of data in that case.

To improve resiliency, incremental checkpoint (iCP) and differential checkpointing (dCP) features have been implemented in the **FTI** library. A theoretical model has been created that accurately predicts the overhead reduction for dCP depending on performance characteristics of the architecture and shows a linear dependency between the reduction of overhead and the data reduction factor. This approach introduces the advantage that the checkpoint data of former checkpoints is preserved inside the checkpoint file. Furthermore, an HDF5 interface has been developed inside FTI, which enables writing with all processes into one shared file.

1.4.4 Benchmarking, performance analysis, modelling and monitoring tools

A benchmark suite composed of a wide range of synthetic benchmarks and selected applications has been integrated in the **JUBE** benchmarking environment and has periodically run on the DEEP-EST prototype to measure its performance and identify potential variations. A visualization environment for the results was put in place to ease the interpretation of data, automatic emails were sent when something was not working, and automatic backups of the benchmarking results were implemented. For instance, thanks to these benchmarking activities, drawbacks with the initial BeeGFS file system configuration were identified and the sweet-spot for its configuration was found. The benchmarks were also used to assess the performance of the DEEP-EST prototype in detail.

Traces of application workloads were used to conduct efficiency analysis and project the performance of the applications at large scale. With this analysis, low parallel efficiency was identified and communicated to the application developers, which could identify and address its sources. Projection data allowed to predict the performance of the applications improved within DEEP-EST. Furthermore, the **Extrae** instrumentation software tools from BSC were extended in order to properly instrument CUDA codes.

Last, but not least, new system-monitoring capabilities have been created. The **DCDB** and **Wintermute** frameworks from BADW-LRZ have been further developed to reach a production-ready state within the project, and all components and plug-ins required for the DEEP-EST prototype have been designed and implemented. In addition to all data supplied natively by DCDB, the sensors exposed by BeeGFS covering file system activity have been integrated. The collection of all this wide variety of system-monitoring information, is accessible to users and operators through the user-friendly visualization tool Grafana. Furthermore, the monitoring tools have been integrated with the resource manager to enable energy-saving scheduling mechanisms. With the

DEEP-EST monitoring infrastructure, full control over the system utilization has been provided, opening opportunities for optimised operation policies. The power consumption of an application running on an HPC system depends on the amount of resources that it uses, and the model in which these resources are run, cooled, and operated. For this reason, an energy-model has been developed to evaluate the energy used by applications on each of the modules of the DEEP-EST prototype, assuming different operational frequencies and configurations.

1.5 Co-design Applications

For the DEEP-EST project several important and ambitious scientific codes from the HPC and HPDA area have been selected as the DEEP-EST co-design applications. Most of the codes combine HPC computation with advanced data processing and analytics. Thus, they do consist of multiple parts with different resource requirements and are eminently suitable to assess the potential of the MSA and the DEEP-EST prototype. The applications belong to six scientific fields:

- Neuroscience: In DEEP-EST, three applications were used to simulate functional models of brain structure. The NEST simulator investigates the dynamics of brain-scale neuronal network models. It does so at the level of resolution of neurons and synapses, where neurons are brain cells connected to each other by synapses. NEST is combined with two types of in situ analysis: computation of electrical local field potentials using the Arbor and HybridLFPy packages, and statistical analysis of spike activity using the Elephant package.
- Molecular Dynamics (MD): A MD simulation generally tracks the trajectories of many particles evolving over time. It solves differential equations of motion in time steps. GROMACS is one of the world's best MD software packages. It is a toolbox allowing users to prepare the structure that they want to simulate, run the simulation and analyse the results at the end.
- Radio Astronomy: In DEEP-EST, two parts of the imaging pipeline of the LOFAR radio telescopes were studied: the correlator and the imager. The correlator combines the data from all receivers and operates on streaming data, normally in real time. First it performs filtering, then corrections, and finally correlates the data from multiple receivers. The imager creates sky images partitioning incoming data in small blocks that are convolved and gridded onto small subgrids. These subgrids are fast Fourier transformed and then added to a large grid, which is finally inversely FFTed to a sky image.
- Space Weather: The space weather workflow consists of three applications: DLMOS, xPic and GMM. DLMOS is a Deep Learning Model of the Solar Wind

to forecast the plasma conditions at the orbit of the Earth from images of the Sun. The particle-in-cell code xPic consists of a field solver that calculates in a Cartesian grid the Maxwell equations of electromagnetism, and a particle solver calculating the motion of billions of charged particles using Newton's equations of motion. The large data volume generated by xPic's particle solver, is analysed with the machine learning model GMM.

- Data Analytics in Earth Science: In DEEP-EST, three applications are used: NextDBSCAN, NextSVM and Deep Learning (DL) frameworks. NextDBSCAN is a new parallel DBSCAN algorithm used for density-based clustering of large three-dimensional point-clouds. NextSVM is a new parallel Support Vector Machine (SVM) used for supervised learning classification tasks with labelled datasets (such as remote sensing images). The TensorFlow framework with the Keras extension is used for computer vision.
- High Energy Physics: CMSSW is the software framework for the Compact Muon Solenoid (CMS) Experiment at CERN. In DEEP-EST, two workflows were used: CMS Reconstruction and CMS Classification. The former takes the raw data coming out of the CMS detector and builds high level physics objects, which are then used for the physics analysis. The CMS Classification takes the reconstructed quantities as input and tries to identify the type of collision event. This is an analytics type of workflow that involves the use of Deep Learning for the purpose of classification.

The above mentioned applications have been analysed in detail to find out e.g. how to best map them to the modules of the DEEP-EST prototype, potentially splitting the application into separate parts (e.g. HPC computation and data analytics). Based on the codes requirements, co-design input was provided through a detailed questionnaire covering, amongst other, aspects such as: primary metric for success (e.g., throughput, accuracy, etc.); programming languages and parallelism paradigms used; rate, type, and volume of inter-process communication; computation-to-communication balance; and I/O requirements (see deliverable D1.1). Later in the project, while the DEEP-EST prototype and its SW were in development, in depth co-design discussions took place around specific design questions, e.g., preferred configuration of the DCPMM non-volatile memory; preferred network topology; kind of collective operations between MSA modules, etc.

In parallel to this co-design feedback, the application codes were adapted to the target hardware, ported to the actual DEEP-EST prototype and its software environment, and then optimised. The results of all these experiences are reported in Chapters 2 to 7 of this book.

1.6 Summary and outlook

The DEEP-EST project has developed the Modular Supercomputing Architecture (MSA) deploying a hardware prototype and implementing its software stack following the codesign input of six application development teams.

From the data centre operator's point of view, the MSA has several advantages. First of all, the better and more efficient system utilisation made possible by the MSA will immediately benefit the data centre operator, leading to a higher ROI (return on investment). To maximise this effect, it is secondly possible to optimise the system configuration, i.e. the number and characteristics of modules to the best match of the specific requirements of the centre and its application portfolio and mix. Additionally, maintenance of individual modules is possible without disturbing the rest of the system, reducing the overall down-times of the machine. Furthermore, the long-term sustainability is improved: new modules can be added to an existing system and old ones substituted at different points in time, keeping the rest of the system and its central resources (e.g. storage) for a much longer lifetime. At some point this might require bridging between older and newer network generations. Finally, procurement processes, which typically depend on different funding sources (e.g. regional, national, project-bound, etc.) can be split and handled individually for the independent modules in an easier way.

From the user's perspective, DEEP-EST provides a very flexible architecture that can match the requirements of very diverse classes of applications, making use of the modules according to their respective needs. The six HPC and HPDA applications in DEEP-EST have tested a variety of different scenarios. Monolithic applications may well use only one of the modules, but more complex, multi-physics or multi-scale applications distribute their code-constituents among several modules of the system and achieve better scalability and efficiency. This also offers the opportunity for more complex workflows or to conduct simulation and data analysis/visualisation concurrently, with the high-speed connection between different modules facilitating necessary data transfers.

Following the success of the DEEP project series, the Modular Supercomputing Architecture is already being applied in production systems. The JUWELS Booster[12] – as of May 2021 the fastest computer in Europe – builds up together with the JUWELS Cluster a Petascale-level modular system. Also, the recently installed EuroHPC

[12] https://apps.fz-juelich.de/jsc/hps/juwels/booster-overview.html

Petascale system MeluXina is modular[13], and more supercomputers in Europe and worldwide have been announced to follow the same philosophy[14],[15].

The implementation of MSA in large-scale production systems is not the end of its development roadmap, which continues in various upcoming EuroHPC JU projects. The software environment for MSA-platforms will be enhanced within the SEA-projects, which started in April 2021: DEEP-SEA is the direct continuation of the software efforts in DEEP-EST and aims at easing application porting to MSA systems and making their programming environment more dynamic by leveraging more malleability and composability[16], IO-SEA will improve the IO-capabilities of MSA systems with a novel data management and storage platform based on object store support, hierarchical storage management (HSM) and intelligent data placement[17]; RED-SEA finally will develop next generation European network technologies with better capabilities for intra- and inter-module communication[18]. Furthermore, at least two of the three pilot projects selected for funding within the EuroHPC-2020-01 call[19] will apply a MSA approach: the EUPEX project will build a modular pilot system integrating European technologies (including the EPI general purpose processor, ParaStation Modulo, etc.), while the HPCQS project will integrate a Quantum Module into an existing MSA system[20].

With this elaborated development roadmap, the Modular Supercomputing Architecture and the overall results of the DEEP-EST project are in the best possible position to be part of the first European Exascale platforms.

[13] https://eurohpc-ju.europa.eu/news/meluxina-live-eurohpc-ju-supercomputer-luxembourg-operational

[14] Slide 40 in https://www.r-ccs.riken.jp/R-CCS-Symposium/2019/slides/Wang.pdf

[15] https://www.hpcwire.com/2021/02/25/japan-to-debut-integrated-fujitsu-hpc-ai-supercomputer-this-spring/

[16] https://cordis.europa.eu/project/id/955606

[17] https://cordis.europa.eu/project/id/955606

[18] https://cordis.europa.eu/project/id/955776

[19] https://eurohpc-ju.europa.eu/calls/advanced-pilots-towards-european-exascale-supercomputers-pilot-quantum-simulator

[20] The EUPEX and HPCQS projects are in the GA-preparation phase at the time of writing.

1.7 Acknowledgements

The authors thank all the institutions and individuals involved in the DEEP series of projects and, in particular, all the members of the DEEP-EST team who have contributed to the development of the MSA architecture, its prototype hardware implementations and its software environment.

This work has been partially funded by the European Union's Seventh Framework (FP7/20017-2013) and Horizon 2020 Framework Programmes for Research and Innovation under grant agreements 287530 (DEEP), 610476 (DEEP-ER), 754304 (DEEP-EST), and 955606 (DEEP-SEA). The present publication reflects only the authors' views. The European Commission is not liable for any use that might be made of the information contained therein.