PROCEEDINGS OF SCIENCE

Modular Supercomputing and its Role in Europe's Exascale Computing Strategy

Sarah Neuwirth[a,b,*]

[a] *Institute of Computer Science, Goethe University Frankfurt, Germany*
[b] *Jülich Supercomputing Centre, Forschungszentrum Jülich, Germany*

E-mail: s.neuwirth@em.uni-frankfurt.de

Reaching Exascale compute performance at an affordable budget requires increasingly heterogeneous HPC systems, which combine general purpose processing units (CPUs) with acceleration devices such as graphics processing units (GPUs) or many-core processors. The Modular Supercomputing Architecture (MSA) developed within the EU-funded DEEP project series breaks with traditional HPC system architectures by orchestrating these heterogeneous computing resources at system-level, organizing them in compute modules with different hardware and performance characteristics. Modules with disruptive technologies, such as quantum devices, can also be included in a modular supercomputer to satisfy the needs of specific user communities. The goal is to provide cost-effective computing at extreme performance scales fitting the needs of a wide range of Computational Sciences. This approach brings substantial benefits for heterogeneous applications and workflows since each part can be run on exactly matching computing resources, therefore improving the time to solution and energy use. It is therefore ideal for supercomputer centers running a heterogeneous mix of applications.
This work introduces the Modular Supercomputing Architecture – which constitutes the central element in Europe's roadmap to Exascale computing –, including its history, its role in Europe's Exascale computing strategy, its hardware and software elements, and experiences from mapping applications and workflows to MSA systems.

The 39th International Symposium on Lattice Field Theory,
8th-13th August, 2022,
Rheinische Friedrich-Wilhelms-Universität Bonn, Bonn, Germany

*Speaker

PoS(LATTICE2022)245

1. Introduction

Numerous technologies and concepts have been developed and researched to meet the ever-increasing demands on computational power. The most common scheme is to maximize exploitable parallelism and efficiency by using high performance computing (HPC) resources, where many processors operate simultaneously to achieve exceptional computational performance and significantly reduce the overall execution time. In general, scalability is used as the corresponding metric to indicate the ability of hardware and software to deliver greater computational performance when the amount of resources or problem size is scaled.

In recent years, highly specialized processors suitable for various tasks and use cases have been developed, including graphics processing units (GPUs), tensor processing units (TPUs), and quantum processing units (QPUs). The idea for modular supercomputing systems emerges from the desire to allow a larger number of codes to take advantage of these heterogeneous resources while improving the power efficiency, system utilization, and scalability of cluster computers. Therefore, the Modular Supercomputer Architecture (MSA) [1–3] introduces hardware disaggregation at the system level by dividing the hardware into compute modules with different hardware and performance characteristics to create a single heterogeneous system. This approach brings significant benefits to heterogeneous applications and workflows by allowing each code part to run on the most suitable module, which improves execution time and energy consumption.

In the following, the Modular Supercomputing Architecture – which constitutes the central element in Europe's roadmap to Exascale computing – is introduced, including its historical development, its role in Europe's Exascale computing strategy, its hardware and software components, and experiences from mapping applications and workflows to MSA systems. The paper concludes with a discussion about the Exascale data challenges and how to overcome them, and provides some insights on how to integrate disruptive technologies, such as quantum devices, into modular supercomputers to meet the needs of specific user communities.

2. Evolution of the Modular Supercomputing Architecture

In this section, the motivation for the development of the modular supercomputing architecture is explained first. Then, a historical overview of the development of the MSA is given. At the end, the DEEP-EST prototype, the first true MSA system, is introduced.

2.1 Motivation

For a fixed problem size, *Amdahl's Law* [4] states that the speedup of a code is always limited by the sequential part. This is called *strong scaling*. It provides the upper limit of the speedup and can be defined as $S = 1/(s + p/N)$, where s and p are the fractions of execution time of the scalar and parallelizable code parts respectively, and N the number of processors. In reality, the sizes of problems scale with the amount of available resources. This is known as *weak scaling* and described in *Gustafson's Law* [5], which is based on the assumptions that the parallel part scales linearly with the amount of resources and that the sequential part does not increase with respect to the size of the problem resulting in the scaled speedup $S = s + p \times N$.

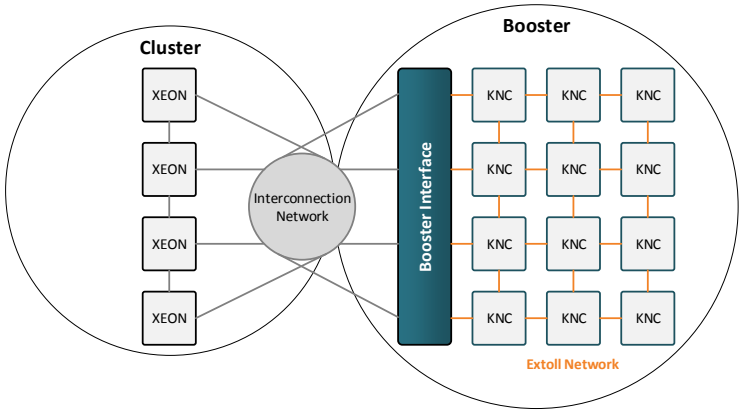


Figure 1: Cluster-Booster concept introduced in DEEP [6].

While Amdahl's law suggests minimizing the sequential code fraction, Gustafson's law states that this fraction is negligible if we have a sufficiently large number of processors. However, Amdahl's law is derived from a very simple model with idealized settings, and Gustafson can be very misleading, since good speedup does not always equate to good efficiency. By mapping different parts of a code on distinct hardware modules, Amdahl's Law can be extended and generalized to provide the maximal asymptotic speedup given by the dominant concurrency of a code.

Let a code have $r-1$ parts with different concurrencies $n_i < n_h = n_r$. The r^{th} code part has concurrency $n_h = n_r$, which can be scaled to infinity according to Amdahl. For simplicity, all n_i are defined for the r code parts in consideration as the numbers where 80% of the maximum parallel speedup has been achieved or where the critical path does not allow for more processors. All code fractions add up to one: $\sum_{i=1}^{r} p_i = 1$. Then, the *Generalized Amdahl's Law* (GAL) can be expressed through

$$S^{GAL} = \frac{T_{single}}{T_{parallel}} = \frac{\sum_{i=1}^{r} p_i}{\sum_{i=1}^{r} \frac{p_i}{n_i}} = \frac{1}{\sum_{i=1}^{r} \frac{p_i}{n_i}}. \tag{1}$$

Let us now assume that n_d is the dominating lower concurrency. For the asymptotes, we assume that all n_i with $i > d$ can be scaled. For n_d to be the dominating lower concurrency, the following condition must be met: $n_d \cdot \sum_{i=1}^{d-1} \frac{p_i}{n_i} \ll p_d$. Pushing n_h to infinity gives us the asymptotic formula for GAL:

$$S_{\infty}^{GAL} = \lim_{n_h \to \infty} \frac{1}{\sum_{i=1}^{d-1} \frac{p_i}{n_i} + \frac{p_d}{n_d} + \frac{p_h}{n_h}} \to \frac{1}{\sum_{i=1}^{d-1} \frac{p_i}{n_i} + \frac{p_d}{n_d}} \to \frac{n_d}{p_d} \tag{2}$$

The maximal asymptotic speedup is, in this case, given by the dominant concurrency of the code. In real life this is obviously only a convenient approximation.

2.2 Modular Supercomputing Architecture

The DEEP project series [7, 8] addresses the research of Exascale computing challenges following a stringent co-design approach. In phase one of the project series, DEEP introduced a new heterogeneous system architecture called the *Cluster-Booster concept* [6, 9, 10]. Key objectives

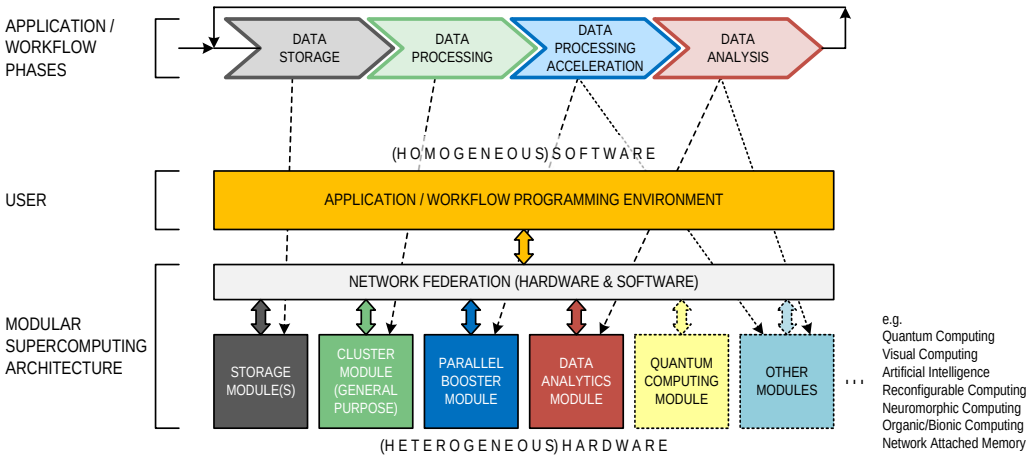
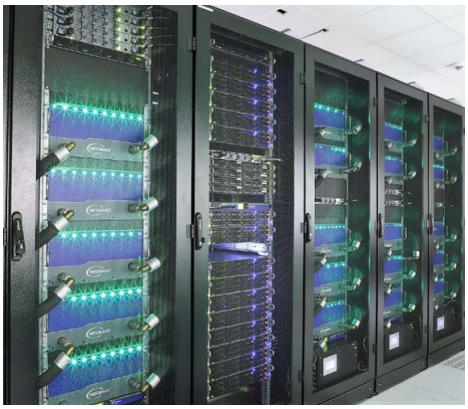


Figure 2: Modular Supercomputing Architecture.

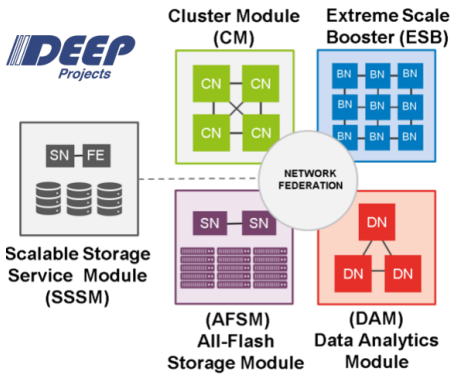
included enabling more codes to take advantage of highly scalable systems and improving the power efficiency and scalability of cluster computers. The Cluster-Booster architecture, as depicted in Figure 1, proposed to integrate heterogeneous computing resources at the system level by extracting the accelerators from the host nodes, moving them into a stand-alone cluster of accelerators, called *Booster*, and connecting this cluster to a standard HPC-Cluster via a high-speed network. Key benefit of this approach is that no constraints are put on the combination of nodes that an application may select, and resources can be reserved and allocated dynamically at runtime, which is in line with the extension of Amdahl's Law. The standard cluster nodes are fit for scalar code, while the booster nodes offer better energy efficiency and are suited for highly scalable code.

DEEP – Extended Reach (DEEP-ER) [11, 12], the second phase of the DEEP project series, extended the Cluster-Booster architecture by implementing a multi-level memory hierarchy introducing *non-volatile memory devices* for efficient I/O buffering and checkpointing, and *network-attached memory* (NAM) [13]. DEEP-ER also introduced a highly scalable and efficient I/O system based on Fraunhofer's BeeGFS file system [14, 15] supporting I/O intensive applications, using the optimized I/O middleware SIONlib [16–18]. In addition, a multi-level checkpoint scheme was developed that allowed the powerful I/O subsystem and fast network-attached storage to be leveraged to reduce the overhead of storing state for long-running tasks.

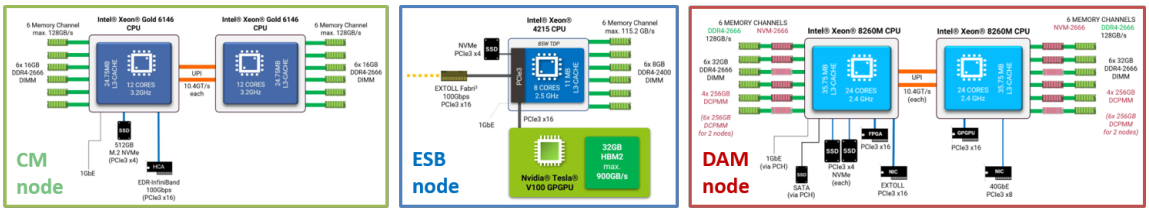
The third installment, *DEEP – Extreme Scale Technologies (DEEP-EST)*, generalized the previous work by introducing the *Modular Supercomputing Architecture* (MSA) [1–3]. In addition, the first fully working MSA system prototype was built, consisting of three modules (see Section 2.3), each one tuned to best match the needs of a certain class of algorithms. Figure 2 shows a generic sketch of the Modular Supercomputing Architecture. MSA breaks with traditional HPC system architectures by orchestrating heterogeneous computing resources at system-level, organizing them in compute modules with different hardware and performance characteristics. Modules with disruptive technologies, such as quantum devices, can also be included in a modular supercomputer to satisfy the needs of specific user communities. The goal is to provide cost-effective computing at extreme performance scales fitting the needs of a wide range of computational sciences. This approach brings substantial benefits for heterogeneous applications and workflows. In a modular



(a) Modular supercomputing prototype.



(b) Architecture of the DEEP-EST system.



(c) Overview of the DEEP-EST compute modules.

Figure 3: First MSA prototype: the DEEP-EST system.

supercomputer, each application can dynamically decide which kinds and how many nodes to use, mapping its intrinsic requirements and concurrency patterns onto the hardware. Codes that perform multi-physics or multi-scale simulations can run across compute modules due to a global system-software and programming environment. Application workflows that execute different actions after (or in parallel) to each other can also be distributed in order to run each workflow-component on the best suited hardware, and exchange data either directly (via message-passing communication) or via the filesystem. A modular supercomputing system can supply any combination or ratio of resources across modules and is not bound to fixed associations between, for instance, CPUs and accelerators as will be found in clusters of heterogeneous nodes. It is therefore ideal for supercomputer centers running a heterogeneous mix of applications (higher throughput and energy efficiency).

2.3 DEEP-EST Prototype

The DEEP-EST prototype [19, 20] (see Figures 3a and 3b) has been developed in a close co-design loop between applications, system software, and system component architects. It consists of three computing modules and two storage modules. Figure 3c shows an overview of the node architectures in the different compute modules. Each hardware module provides different characteristics, therefore exploiting the benefits introduced by the Generalized Amdahl's Law:

- ***Cluster Module (CM):*** 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.

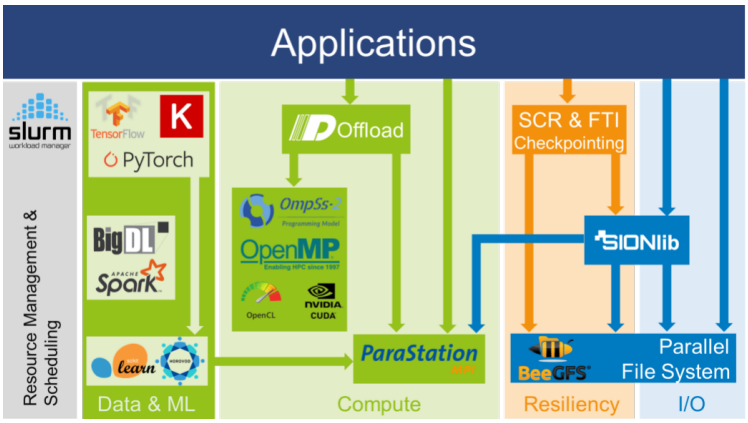


Figure 4: Software stack in the DEEP-EST project [19].

Usage targets include applications and code parts requiring high single-thread performance and a modest amount of memory, which typically show moderate scalability.

- ***Extreme Scale Booster (ESB):*** Booster cluster with 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 nodes are interconnected via InfiniBand EDR providing a 100 Gbit/s bandwidth. Usage targets include compute-intensive applications and code parts with regular control and data structures, which typically show high parallel scalability.

- ***Data Analytics Module (DAM):*** Specifically designed cluster for high-performance data analytics (HPDA) and artificial intelligence (AI) workloads with 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. Flexibility, non-volatile memory and different acceleration capabilities are key features of the DAM to support HPDA and ML/AI workloads requiring large memory capacity, data streaming, and bit- or small datatype processing.

- ***Scalable Storage Service Module (SSSM):*** Provides storage capacity based on conventional spinning-disks and uses the BeeGFS parallel file system.

- ***All-Flash Storage Module (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. BeeGFS is used to make 1.8 PB of data storage capacity available. AFSM deploys two Metadata servers and six volume data server systems, which are interconnected by a 100 Gbps EDR-InfiniBand fabric.

As shown in Figure 4, the hardware environment is complemented by a software architecture [21, 22] designed to carefully address the requirements of the co-design applications supported in DEEP-EST. It focuses on the most relevant parts of the software stack required to run a supercomputer while also exploiting the modular supercomputer architecture. For example, the ParaStation MPI communication library [23] is designed to leverage distributed memory systems

and the OmpSs-2 programming model [24] is used to exploit many-core processors, deep memory hierarchies, and accelerators. Furthermore, the software stack provides support for resiliency (i.e.; checkpointing), parallel I/O, file system and storage. To support the complex I/O and storage system, DEEP-EST has focused on MSA-aware extensions [22] for BeeGFS and SIONlib.

3. Porting Example Workloads to MSA Systems

As part of the DEEP project series, several co-design applications were selected. Six of these applications have been ported to the DEEP-EST prototype system [19, 25, 26]. 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. The chosen workloads are from six science domains: neuroscience, modular dynamics (MD), radio astronomy, space weather, data analytics in earth science, and high energy physics.

In the following, two use cases are introduced. First, the mapping of a Space Weather system simulation to the DEEP-EST prototype is presented. Afterwards, Lattice QCD workflows on MSA-based production systems will be briefly discussed.

3.1 Space Weather Simulations on the DEEP-EST Prototype

The space weather simulation connecting the Sun to the magnetosphere of Earth, which has been ported to the DEEP-EST prototype, consists of three applications [19]:

- **DLMOS** (Deep Learning Modelling of the Solar wind): used to forecast the solar wind conditions in front of the Earth from images of the Sun. DLMOS is coded in Python, uses PyTorch and is deployed in two modes: training and scoring (inference). While scoring can be performed quickly on regular CPU architectures, training the model requires sufficient resources in disk access, memory size and computing power.

- **xPic** (extended Particle-in-cell): a first-principles plasma physics code used to study the plasma environment in the solar wind and the magnetosphere. The code has been partitioned in two solvers: filed solver and particle solver. The field solver is a numerical algorithm that solves Maxwell's equations for electromagnetism in a 3D Cartesian grid, while the particle solver uses Newton's equations of motion to compute the movement of billions of charged particles, which integrate the system.

- **GMM** (Gaussian Mixture Model): a machine learning algorithm that analyses velocity distributions functions extracted from particle information generated in the xPic code. The GMM analysis runs "on the fly". The execution frequency depends on the particular simulation tested, but typically every few hundred iterations of the xPic particle solver, detailed particle data is transferred to the GMM for analysis.

Figure 5 shows how the space weather workflow is mapped onto the DEEP-EST prototype.The training part of DLMOS is characterized by extensive and continuous movement of data from disk to processor memory. In addition, large amounts of data are used to train a deep neural network that relies on the continuous use of tensor operations (matrix and vector multiplications) and therefore

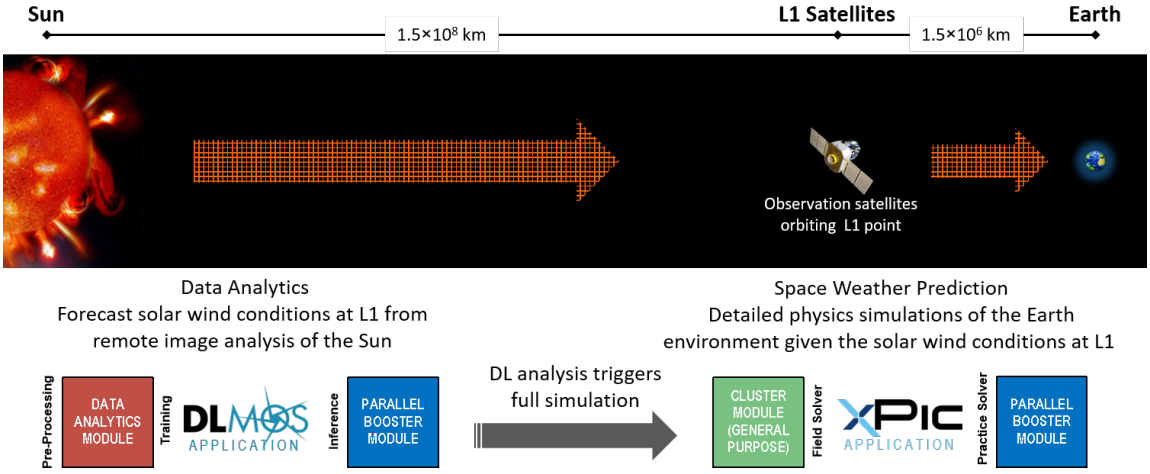


Figure 5: Mapping the space weather simulation onto the DEEP-EST system.

requires high memory bandwidth. Therefore, DLMOS benefits from the large number of cores and the high memory bandwidth offered by the DAM.

The DLMOS inference code is mapped to the ESB, since this is also the current module used for I/O in the xPic code. The inference process is very fast and requires minimal resources, so it is not necessary to use the DAM for this procedure. Once the training and inference phase have been completed, the space weather prediction via xPic is triggered.

The xPic code is run in the cluster-booster configuration [11], i.e., on the CM and ESB, leveraging the results of the DEEP-ER project. The field solver is mapped to the CM to take advantage of the high single-thread performance, while the particle solver is mapped to the ESB performing very fast calculations on a very large number of independent particles. GMM, a second machine learning model used to further analyze the particle data, is run on the DAM, while the xPic simulation code is running.

An initial evaluation on the DEEP-EST prototype demonstrated satisfactory performance results. For example, the particle solver had near-optimal strong and weak scaling results on the ESB. GMM showed good weak scaling results on the DAM. A detailed discussion of the experimental evaluation can be found here [19].

3.2 Lattice QCD Workflows on Modular Supercomputers

Lattice chromodynamics (QCD) is a very homogeneous problem, which leaves lots of room for tasks, threads, and single instruction multiple data (SIMD) lanes to do the same operation on different parts of the data. A typical Lattice QCD workflow can be outlined as follows:

1. Generate an ensemble of *lattice gauge configurations* with statistical weight:

$$e^{-\tilde{S}[U]} = \det M[U] e^{-S_g[U]}$$

2. Measure some operator O_i on each configuration i.

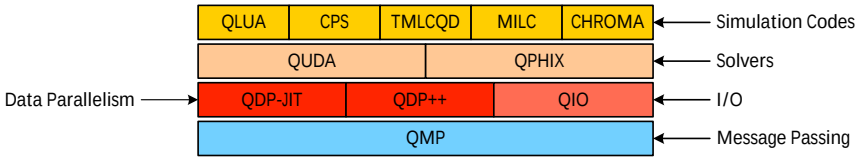


Figure 6: Lattice QCD community software stack based on USQCD [27].

3. Average of measurements is Euclidean path integral calculation of expected value:

$$\langle O \rangle = \frac{1}{N} \sum_{i}^{N} O_i \exp[-\tilde{S}_{\text{Euc}}\{U_i\}]$$

In the limits $N \to \infty$, $L \to \infty$ and $a \to \infty$, this is equivalent to the path integral:

$$\langle O \rangle = \frac{1}{N} \int \mathcal{D}[\overline{\psi}, \psi, A] O(\overline{\psi}, psi, A) \mathrm{e}^{-iS_{QCD}[\overline{\psi}, \psi, A]}$$

The homogeneity of the Lattice QCD data often allows near-perfect weak scaling up to more than a million threads distributed across thousands of MPI tasks. Furthermore, it can be oberserved that Lattice QCD workflows often rely on pure data parallelism or are of embarrassingly parallel nature (e.g., measurement on a gauge ensemble or multiple Markov chains).

The USQCD [27] suite provides a community stack of codes and libraries designed to provide common functionality, optimized for a wide range of HPC architecture. The software maps physics problems to hardware in an optimized yet flexible way. At the top are the supported simulation codes such as Chroma [28], which is also one of the co-design applications in the DEEP project series. The next layer comprises the solver, including QUDA, a Level 3 solver for a variety of fermion actions for use on NVIDIA GPU based systems. The data parallelism components QDP-JIT and QDP++ parallelize lattice field data structures, while QIO provide I/O functionality. QMP provides a standard communications layer for Lattice QCD similar to MPI.

MPI	QDP	Solver
MVAPICH2	qdp-jit	QUDA
MVAPICH2	qdpxx	QUDA
PS-MPI	qdp-jit	QUDA
PS-MPI	qdpxx	QUDA
PS-MPI	qdpxx	QPHIX (AVX512)

Table 1: Chroma software installation on the JUWELS modular supercomputer.

Table 1 outlines the Chroma software installation on the JUWELS supercomputer derived from USQCD. JUWELS is a multi-petaflop modular supercomputer operated by Jülich Supercomputing Centre currently consisting of two modules: Cluster and Booster. The Cluster is a BullSequana X1000 system with Intel Xeon Skylake-SP processors and Mellanox EDR InfiniBand. The Booster is a BullSequana XH2000 system with 2nd generation AMD EPYC processors, NVIDIA Ampere GPUs and NVIDIA/Mellanox HDR Infiniband. The QUDA solver library makes the JUWELS GPU nodes the ideal platform for Lattice QCD codes. It has been demonstrated that the solver performance on the JUWELS Booster nodes is $\sim 60\times$ faster than on the JURECA Booster nodes [29].

In order to further advance Lattice QCD codes on future modular supercomputing systems, the Jülich Supercomputing Centre has been focusing on the QMOD project [30]. The goal of this project is to further investigate different unexploited task-based concurrencies in the context of modular supercomputing. The QMOD project will enable Lattice QCD task-parallelism across different architectures, exchange of lattice data structures (i.e., gauge fields, propagators, eigenvectors) between separate communicators, and lattice field I/O, i.e., intercommunicator data exchange. QMOD is designed as an add-on for the community stack USQCD and consists of two libraries: QMPadd and QMOD. QMPadd establishes the communication link between partitions and the QMOD library provides data transfer functionality. An example application would be a group of nodes of one architecture solves a quark matrix system, passes the solutions (propagators) to a separate hardware group, which begins assembling them into multi-nucleon correlation functions.

4. Europe's Exascale Computing Strategy

The Frontier system at Oak Ridge National Laboratory, USA has recently surpassed the 1 exaFLOP/s barrier [31], making it the first Exascale system in the world. It is expected that China and Japan soon will introduce their first Exascale systems as well. As a result, the PRACE Scientific Steering Committee [32] has called attention to the urgent need for an expanded European advanced computing infrastructure to cover a broad range of applications, education and workforce training. European researchers are leaders in many areas of algorithm and software development, which has helped to create a thriving computational science community. To further drive the expansion of Europe's HPC infrastructure and remain internationally competitive, the modular supercomputing architecture plays a central role in Europe's roadmap to exascale computing.

In this section, first an overview of current EU research projects promoting MSA is given. Then, Europe's exascale computing strategy is introduced. Finally, two research questions related to MSA that are currently being investigated at Goethe University Frankfurt are highlighted.

4.1 Current European Research Projects Advancing the MSA

Several EU-funded projects are currently underway with the target to further advance the hardware and software ecosystem of MSA systems. *DEEP – Software for Exascale Architectures (DEEP-SEA)* [33] is a continuation of the DEEP project series targeting an open-source software and programming environment for future European Exascale systems based on MSA. The goal is to support a broad variety of different workloads with varying compute node and memory configuration requirements. The *Input/Output Software for Exascale Architectures (IO-SEA)* project [20, 34] seeks to improve the I/O and data management in large-scale MSA systems and builds upon the results of the SAGE 1-2 [35] projects and MAESTRO. Its main focus is to provide a novel data management and storage platform for exascale computing based on hierarchical storage management (HSM) and on-demand provisioning of storage services. The platform will efficiently make use of storage tiers spanning NVMe and NVRAM at the top all the way down to tape-based technologies. The *Network Solution for Exascale Architectures (RED-SEA)* project [36] is developing a European network solution for MSA-based Exascale systems by leveraging existing European technology, including BXI (Bull eXascale Interconnect) and Exanest. Further projects are the *European Processor*

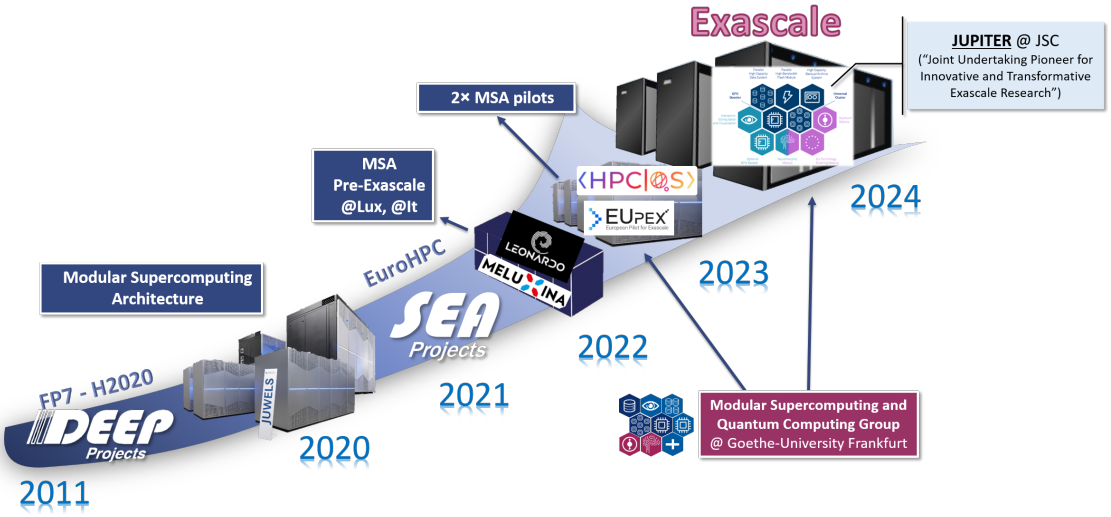


Figure 7: Perspective for the European road to exascale.

Initiative (EPI) [37], the *European Pilot for Exascale (EUPEX)* [38], and the *Adaptive multi-tier intelligent data manager for Exascale (ADMIRE)* [39].

4.2 Roadmap to Exascale Computing

As highlighted in the previous section, several research projects are currently underway to advance the MSA. As shown in Figure 7, modular supercomputing is at the core of the European Exascale strategy. Since 2011, the DEEP project series has been the main driver, leading to the first MSA prototype in DEEP-EST and several production systems with different compute modules.

The Jülich Supercomputing Centre (JSC) currently operates two modular supercomputers: *JURECA* and *JUWELS*. JURECA is a pre-Exascale modular supercomputer, which combines a flexible Data Centric (DC) module, based on the Atos BullSequana XH2000 with a selection of best-of-its-kind components, and a scalability-focused Booster module, delivered by Intel and Dell Technologies based on the Xeon Phi many-core processor. JUWELS is a multi-petaflop modular supercomputer consisting of two modules: Cluster and Booster (see Section 3.2 for details).

In 2022, two European pre-Exascale modular supercomputers have been introduced. In June, the new *LUMI system* [40], an HPE Cray EX system installed at the EuroHPC center at CSC in Kajaani, Finland, deputed as the new No. 3 in the TOP500 with a peak performance of 0.309 EFlop/s. The LUMI supercomputer achieves its high performance with a large number of nodes with accelerators (GPUs). In addition, the system is complemented by a CPU-only partition, IaaS cloud services, and a large object storage solution. In November, the *Leonardo system* [41] at EuroHPC/CINECA in Bologna, Italy deputed as the new No. 4 in the TOP500. The machine achieved an HPL score of 0.174 EFlop/s. The system has two main computing modules, named Booster and Data Centric. A customized version of NVIDIA Ampere GPUs equips the Booster module, while the Data Centric module features Intel's new generation Sapphire Rapids processors.

In addition to the productive pre-exascale supercomputers, two projects are currently underway with the goal of building two pilot systems. The EUPEX project [38] will deploy a pilot hardware

and software platform integrating the full spectrum of European technologies, and will demonstrate the readiness and scalability of these technologies, and particularly of the modular supercomputing architecture, towards Exascale. In addition, the HPCQS project [42] has the goal to deploy a quantum simulator pilot and HPC system integration with the intent to host an open European federated hybrid HPC-QS infrastructure that will provide non-commercial cloud access.

JUPITER [43] ("Joint Undertaking Pioneer for Innovative and Transformative Exascale Research") will be the first European modular supercomputer to make the leap to the Exascale class. The system is scheduled to be installed at JSC starting in 2023 and is expected to be in full operation by the end of 2024. In its basic configuration, JUPITER will have an extremely powerful booster module with highly efficient GPU-based accelerators and a universal CPU cluster. The system will be supported by a high-capacity data module, a high-bandwidth flash module, and a high-capacity backup/archive system module that incorporate the results of previous and ongoing EU projects.

4.3 Exascale Data Challenges and Emerging Workloads

As exascale-class modular supercomputing systems come within reach, scientific advances that match the growth in theoretical computing power will require massive innovations in software stacks for storage and data processing, including the following core capabilities [20]:

- **System Scalability:** Exascale supercomputers will span hundreds of thousands of nodes. However, conventional parallel file systems cannot operate efficiently at this scale. Meanwhile, the now data-centric architecture of today's HPC centers has a direct impact on the design of applications and workflows, which will become even more data-oriented as a result.
  => *Access to data becomes a critical issue to avoid blocked systems.*

- **Data Scalability:** Bigger machines mean more data and therefore more records or files.
  => *I/O systems need to be able to store hundreds of exabytes or even zettabytes of data.*

- **Data Heterogeneity:** The data sets within the I/O system are typically not equivalent. Some are very small (a few kilobytes), while others are quite large (up to petabytes). The way they are used and accessed can also vary widely: Some files are accessed in the form of streams, while others are accessed more randomly (e.g., database files and machine learning training sets). Some files are accessed by a single client, others by many concurrent clients.
  => *Quite complex "data taxonomy" needs to be supported.*

- **Data Placement:** Supercomputers, especially modular ones, are becoming more and more fragmented and have a complex network federation. Therefore, data should be used and generated as close as possible to where the simulation code is executed. Otherwise, the data path takes many network hops, resulting in lower performance and higher energy consumption.

The exascale data challenges are further intensified through a new class of codes, known as emerging workloads. HPC applications are evolving to include not only traditional scale-up modeling and simulation bulk-synchronous workloads but also scale-out workloads like artificial intelligence (AI), data analytics methods, deep learning, big data and complex multi-step workflows. Exascale workflows are projected to include multiple different components from both scale-up and

scale-out communities operating together to drive scientific discovery and innovation. Another performance aspect is the intensifying complexity of parallel file and storage systems in large-scale cluster environments. The changing landscape of emerging hybrid HPC workloads along with the ever increasing gap between the compute and storage performance capabilities reinforces the need for an in-depth understanding of extreme-scale I/O and for rethinking existing data storage and management techniques. This is especially true for modular supercomputing systems [20].

Given the complexity of modern supercomputers and HPC systems, achieving theoretical peak performance depends on myriad parameters. To optimize system performance and make efficient use of underlying resources, various methods can be applied [44], including simulation, benchmarking, and monitoring. However, these methods and the tools are not compatible with each other, i.e., both the individual tools and the approaches consider only a certain part of a certain domain, e.g., network, I/O, or resource allocation. Furthermore, each of these approaches generates specific insights that can be applied to similar problems or specific system configurations. To avoid generating such knowledge for one-time purposes only, and also to support other users, this knowledge must be easily accessible and available to the community. Goethe University Frankfurt is pursuing the open-source project MAWA-HPC [45] (Modular and Automated Workload Analysis for HPC Systems) to develop a generic workflow and tool suite that can be applied to various use cases from different science domains. Its modular design allows the workflow to support different community tools at each stage, increasing the compatibility of each tool and covering new use cases.

4.4 A Quantum Future for HPC?

The integration of modules with disruptive technologies, such as quantum processors, naturally fits into the MSA concept and its extension of Amdahl's Law. However, the main limiting factor for current quantum processor units (QPUs) and pure quantum algorithms is often the high error rates of the qubits, which limit the circuit depth, i.e., the number of quantum operations that can be performed before a measurement needs to be made. These error prone QPUs are called *Noisy Intermediate-Scale Quantum Computers (NISQ)*. Their limitations in terms of feasible circuit depths led to the development of *Variational Quantum Algorithms (VQAs)* [46], which are hybrid algorithms consisting of a classical and a quantum part. VQAs have been tested on simulators and on real quantum processors for small problem sizes. Current QPUs are not yet good enough to perform VQAs for reasonable problem sizes, but they are a potential candidate to demonstrate quantum advantage for a real-world problem in the near future [46].

The integration of quantum processors fits naturally into the modular supercomputing architecture as an additional *Quantum Computing module*, as indicated in Figure 2. The fact that quantum processors are excellent at accelerating certain parts of a code (e.g., evaluating the cost function) but poor for other parts makes them an ideal component for a modular system. In an ideal modular supercomputer, the quantum processor is tightly coupled with the other modules, enabling low-latency, high-bandwidth communication. System-level computing resources are managed by a scheduler that is aware of applications currently running and their requirements, resulting in high utilization of the quantum processor and reduced waiting times. Goethe University Frankfurt is currently working on a prototype integration by extending the OmpSs programming model [24] to include support for QPUs as an additional type of accelerator.

5. Summary

The Modular Supercomputing Architecture plays a central role in the European exascale strategy. This work has highlighted the major milestones in the development of this new architectural paradigm and the three key principles associated with it: (1) scalability, (2) parallel efficiency, and (3) optimal system utilization. By disaggregating hardware resources and bundling them in dedicated modules with specific hardware and performance characteristics, applications and workflows can be mapped to the available system resources at runtime according to the principle of the extended and generalized Amdahl's Law. This enables quasi-optimal utilization of the system and breaks with the traditional approach of static assignment of CPUs and GPUs.

It was shown which state-of-the-art production systems use the MSA approach. These systems include the DEEP-EST prototype, JUWELS, LUMI and Leonardo. The principle of co-design was presented and applied to two science problems. Using the Space Weather Simulation and a typical Lattice QCD workflow, it was demonstrated how code can be mapped to the corresponding compute modules. In addition, current EU projects that are investigating the further development of the MSA were discussed. JUPITER, the first European exascale-class system, will combine the results and findings of past and current research projects. It is scheduled for completion by 2024. The next major challenges for extending the MSA include better understanding of emerging workloads, new methods for addressing exascale data challenges, and integrating new compute modules, such as the quantum computing module. Goethe University Frankfurt is currently working closely with its European colleagues to meet these challenges.

References

[1] E. Suarez, N. Eicker and T. Lippert, *Supercomputing Evolution at JSC*, vol. 49 of *Publication Series of the John von Neumann Institute for Computing (NIC) NIC Series*, pp. 1–12, NIC Symposium 2018, Jülich (Germany), 22 Feb 2018 - 23 Feb 2018, John von Neumann Institute for Computing, Feb., 2018, http://hdl.handle.net/2128/17546.

[2] E. Suarez, N. Eicker and T. Lippert, *Modular Supercomputing Architecture: from Idea to Production; 3rd*, in *Contemporary High Performance Computing: From Petascale toward Exascale, Volume 3*, vol. 3, (FL, USA), pp. 223–251, CRC Press (2019).

[3] E. Suarez, W. Frings, N. Attig, S. Achilles, J. De Amicis, T. Eickermann et al., *Developing Exascale Computing at JSC*, in *NIC Symposium 2020*, vol. 50 of *Publication Series of the John von Neumann Institute for Computing (NIC) NIC Series*, (Jülich), pp. 1 – 19, NIC Symposium 2020, Jülich (Deutschland), 27 Feb 2020 - 28 Feb 2020, Forschungszentrum Jülich GmbH Zentralbibliothek, Verlag, Feb, 2020.

[4] G.M. Amdahl, *Validity of the Single Processor Approach to Achieving Large Scale Computing Capabilities*, in *Proceedings of the April 18-20, 1967, Spring Joint Computer Conference*, AFIPS '67 (Spring), (New York, NY, USA), p. 483–485, Association for Computing Machinery, 1967, DOI.

[5] J.L. Gustafson, *Reevaluating Amdahl's Law*, *Commun. ACM* **31** (1988) 532–533.

[6] S. Neuwirth, *Accelerating Network Communication and I/O in Scientific High Performance Computing Environments*, Ph.D. thesis, Heidelberg University, Germany, 2018.

[7] N. Eicker, T. Lippert, T. Moschny, E. Suarez and D. project, *The DEEP Project - An alternative approach to heterogeneous cluster-computing in the many-core era*, *Concurrency and computation: Practice and Experience* **28** (2016) 2394.

[8] "DEEP Project Series: Objectives." https://www.deep-projects.eu/project/objectives.html.

[9] N. Eicker and T. Lippert, *An accelerated Cluster-Architecture for the Exascale*, *PARS '11, PARS-Mitteilungen, Mitteilungen - Gesellschaft für Informatik e.V., Parallel-Algorithmen und Rechnerstrukturen* **28** (2011) 110.

[10] N. Eicker, T. Lippert, T. Moschny and E. Suarez, *The DEEP project: Pursuing cluster-computing in the many-core era*, in *Proceedings of the 42nd International Conference on Parallel Processing Workshops (ICPPW) 2013, Workshop on Heterogeneous and Unconventional Cluster Architectures and Applications*, pp. 885–892, 2013, DOI.

[11] A. Kreuzer, J. Amaya, N. Eicker, R. Leger and E. Suarez, *The DEEP-ER Project: I/O and Resiliency Extensions for the Cluster-Booster Architecture*, in *2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, pp. 109–116, 2018, DOI.

[12] A. Kreuzer, N. Eicker, J. Amaya and E. Suarez, *Application Performance on a Cluster-Booster System*, in *2018 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, pp. 69–78, May, 2018, DOI.

[13] J. Schmidt, *Accelerating Checkpoint/Restart Application Performance in Large-Scale Systems with Network Attached Memory*, Ph.D. thesis, Heidelberg University, Germany, 2017.

[14] C. Manzano, *BeeGFS in the DEEP/-ER Project*, BeeGFS User Meeting 2016, Kaiserslautern (Germany), May, 2016, http://juser.fz-juelich.de/record/811106.

[15] "BeeGFS - The leading parallel file system." https://www.beegfs.io/.

[16] W. Frings, F. Wolf and V. Petkov, *Scalable Massively Parallel I/O to Task-Local Files*, in *Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis*, SC '09, (New York, NY, USA), Association for Computing Machinery, 2009, DOI.

[17] J. Freche, W. Frings and G. Sutmann, *High-Throughput Parallel-I/O using SIONlib for Mesoscopic Particle Dynamics Simulations on Massively Parallel Computers*, in *Parallel Computing: From Multicores and GPU's to Petascale*, pp. 371–378, IOS Press (2010).

[18] K. Chasapis, J.-T. Acquaviva and S. Lührs, *Integration of Parallel I/O Library and Flash Native Accelerators: An Evaluation of SIONlib with IME*, in *Proceedings of the Workshop on*

Challenges and Opportunities of Efficient and Performant Storage Systems, CHEOPS '21, (New York, NY, USA), Association for Computing Machinery, 2021, DOI.

[19] A. Kreuzer, E. Suarez, N. Eicker and T. Lippert, eds., *Porting applications to a Modular Supercomputer - Experiences from the DEEP-EST project*, vol. 48 of *Schriften des Forschungszentrums Jülich IAS Series*, Forschungszentrum Jülich GmbH Zentralbibliothek, Verlag, Jülich (2021).

[20] S. Neuwirth, *Assessment of the I/O and Storage Subsystem in Modular Supercomputing Architectures*, in *2022 IEEE International Conference on Cluster Computing (CLUSTER)*, pp. 589–596, 2022, DOI.

[21] K. Thust, *D4.4 I/O software packages*, Tech. Rep. DEEP Extended Reach, Grant Agreement Number: 610476 (2017).

[22] V. Beltran and P. Martinez, *D6.3 Complete Programming Environment Implementation*, Tech. Rep. DEEP Extreme Scale Technologies, Grant Agreement Number: 754304 (2021).

[23] "ParaStation MPI Repository." https://github.com/ParaStation/psmpi.

[24] A. Duran, E. Ayguadé, R.M. Badia, J. Labarta, L. Martinell, X. Martorell et al., *OmpSs: a Proposal for Programming Heterogeneous Multi-Core Architectures*, *Parallel Process. Lett.* **21** (2011) 173.

[25] E. Suarez, S. Kunkel, A. Küsters, H.E. Plesser and T. Lippert, *Modular Supercomputing for Neuroscience*, in *International Workshop on Brain-Inspired Computing*, pp. 63–80, Springer, Cham, 2019.

[26] P. Kumbhar, M. Hines, A. Ovcharenko, D.A. Mallon, J. King, F. Sainz et al., *Leveraging a Cluster-Booster Architecture for Brain-Scale Simulations*, in *International conference on high performance computing*, pp. 363–380, Springer, 2016.

[27] "US Lattice Quantum Chromodynamics (USQCD)." http://usqcd-software.github.io/.

[28] R.G. Edwards and B. Joo, *The Chroma Software System for Lattice QCD*, in *22nd International Symposium on Lattice Field Theory (LATTICE 2004)*, 2004.

[29] E. Gregory and M. Wagner, *Scaling Lattice QCD on many GPU Nodes of JUWELS*, Tech. Rep. MSA Seminar at Jülich Superomcputing Centre (2020).

[30] E.B. Gregory, *QCD on the Modular Supercomputer*, in *37th International Symposium on Lattice Field Theory (LATTICE 2019)*, 2019, DOI.

[31] "TOP500." https://top500.org/.

[32] PRACE Scientific Steering Committee, *The Scientific Case for Computing in Europe 2018–2026*, 2018.

[33] "DEEP-SEA: Programming Environment for European Exascale Systems." https://www.deep-projects.eu/.

[34] "IO-SEA: Storage I/O and Data Management for Exascale Architectures." https://iosea-project.eu/.

[35] "SAGE Project Series." https://sagestorage.eu/.

[36] "RED-SEA: Network Solution for Exascale Architectures." https://redsea-project.eu/.

[37] "Euroapean Processor Initiative." https://www.european-processor-initiative.eu/.

[38] "Euroapean Pilot for Exascale." https://eupex.eu/.

[39] "ADMIRE: Adaptive multi-tier intelligent data manager for Exascale." https://www.admire-eurohpc.eu/.

[40] "LUMI Supercomputer at CSC's data center in Kajaani, Finland." https://lumi-supercomputer.eu/lumi_supercomputer/.

[41] "Leonardo Supercomputer." https://leonardo-supercomputer.cineca.eu/about/.

[42] "High Performance Computer – Quantum Simulator hybrid (HPCQS)." https://www.hpcqs.eu/.

[43] "First European Exascale Supercomputer Coming to Jülich." https://www.fz-juelich.de/en/news/archive/press-release/2022/first-european-exascale-supercomputer-coming-to-julich.

[44] S. Neuwirth and A.K. Paul, *Parallel I/O Evaluation Techniques and Emerging HPC Workloads: A Perspective*, in *2021 IEEE International Conference on Cluster Computing (CLUSTER)*, pp. 671–679, IEEE, 2021.

[45] Z. Zhu, S. Neuwirth and T. Lippert, *A Comprehensive I/O Knowledge Cycle for Modular and Automated HPC Workload Analysis*, in *2022 IEEE International Conference on Cluster Computing (CLUSTER)*, pp. 581–588, 2022, DOI.

[46] M. Cerezo, A. Arrasmith, R. Babbush, S.C. Benjamin, S. Endo, K. Fujii et al., *Variational quantum algorithms*, *Nature Reviews Physics* **3** (2021) 625.