Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 255 (2025) 140–149

Procedia Computer Science

www.elsevier.com/locate/procedia

Proceedings of the Second EuroHPC User Day

Towards a European HPC/AI ecosystem: a community-driven report

Petr Taborsky[a,*], Iacopo Colonnelli[b], Krzysztof Kurowski[c], Rakesh Sarma[d], Niels Henrik Pontoppidan[e], Branislav Jansík[f], Nicki Skafte Detlefsen[a], Jens Egholm Pedersen[g], Rasmus Larsen[h], Lars Kai Hansen[a]

[a]*Technical University of Denmark, Anker Engelunds Vej 1, Bygning 101A, 2800 Kongens Lyngby, Denmark*
[b]*University of Torino, Computer Science Dept., Corso Svizzera 185, 10149, Torino, Italy*
[c]*Poznań Supercomputing and Networking Center, Jana Pawła II 10, 61-139 Poznań, Poland*
[d]*Forschungszentrum Jülich GmbH, Jülich Supercomputing Centre, Wilhelm-Johnen-Straße, 52425 Jülich, Germany*
[e]*Eriksholm Research Centre, Rørtangvej 20, 3070 Snekkersten, Denmark*
[f]*IT4Innovations, VSB – Technical University of Ostrava, Ostrava, Czech Republic*
[g]*KTH Royal Institute of Technology, Brinellvägen 8, 114 28 Stockholm, Sweden*
[h]*Alexandra Instituttet, Rued Langgaards Vej 7, 2300 Copenaghen, Denmark*

Abstract

The rapid advancements in AI and Machine Learning necessitate a robust computational infrastructure to support cutting-edge research and industrial applications. From the academic and industrial AI community perspective, voiced in the recent ELISE project, the European AI platform is recommended to center around the EuroHPC growing ecosystem. It should be user-driven, easily accessible, powerful, and compliant with European regulations. AI-optimized and dedicated supercomputers for the European AI community are also coming, in addition to upgrading partitions of existing EuroHPC systems to 'AI enabled' stage. Related calls have been initiated in September 2024. Further, conventional EuroHPC systems are suggested to be extended with quantum computing, edge AI, and neuromorphic computing to cater to AI models deployed on network edge devices and sustainability in the long run. The challenges are presented in three case studies, ranging from training Transformers on HPC to LLMs trained federally across three different Euro HPC systems to recent results on hybrid classical-quantum application. This paper concludes with case studies results-informed next steps believed to benefit AI practitioners and the broader AI community.

© 2025 The Authors. Published by Elsevier B.V.
This is an open access article under the CC BY 4.0 license (https://creativecommons.org/licenses/by/4.0)
Peer-review under responsibility of the scientific committee of the Proceedings of the Second EuroHPC user day

Keywords: Artificial Intelligence; High-Performance Computing; HPC; ELISE; ELLIS; EuroHPC Joint Undertaking; Quantum Computing; Federated Learning

* Corresponding author.
E-mail address: ptab@dtu.dk

1877-0509 © 2025 The Authors. Published by Elsevier B.V.
This is an open access article under the CC BY 4.0 license (https://creativecommons.org/licenses/by/4.0)
Peer-review under responsibility of the scientific committee of the Proceedings of the Second EuroHPC user day
10.1016/j.procs.2025.02.269

1. Introduction

The European Union is well positioned to accelerate its Artificial Intelligence (AI) and Machine Learning (ML) research by leveraging advancements in High-Performance Computing (HPC) and Quantum Computing (QC), with prospects of a more sustainable Neuromorphic platform in the long run. Among other activities, the European R&D community, represented by ELLIS Community and the European Commission, initiated the ELISE project to investigate and recommend to AI practitioners and researchers the options for a joint pan-European AI R&D platform for the near future. This paper presents excerpts from the recent ELISE project report, comprising inputs from 100+ R&D teams, hardware providers (e.g., NVIDIA), and industry partners (e.g., OTICON) that are the most relevant to the EuroHPC community. In particular, it focuses on EuroHPC systems.

HPC systems have been pivotal in scientific research, providing the computational power for complex simulations and data analysis. With the rise of AI and ML, a broader R&D community (e.g., ELLIS) is interested in utilizing HPC systems for these fields. This report examines the feasibility and implications of using HPC systems like MeluXina, LUMI, or the upcoming JUPITER exascale supercomputer for AI/ML research in Europe.

The article analyzes two large-scale case studies taken from the AI community, i.e., a large neural network training on a single EuroHPC facility (Sec. 2), a federated training of a Large Language Model (LLM) across three EuroHPC systems (Sec. 3), and one hybrid classical-quantum case study (Sec. 4), demonstrating some challenges that industrial and scientific AI practitioners have to face and proposing a way forward.

2. Training Large Neural Networks on HPC system

Traditional HPC applications are large-scale scientific simulations from diverse domains, like life sciences, weather forecasting, quantum chemistry, and physics [32]. Typically, these applications rely on floating-point operations (double precision, via CPUs and, increasingly, GPUs) to run predetermined models that generate data, often organized in a few large data files. On the other hand, AI applications rely on given data to fit a model, i.e., data produce a model using some ML probabilistic algorithm. The forward-backward pass in the backpropagation algorithm [38] typically leverages hardware accelerators (e.g., GPUs or TPUs) to perform a set of fast and possibly noisy matrix multiplications. In the meantime, a small set of CPUs performs preprocessing and transfer tasks on lots of little data chunks called mini-batches. Given that, AI and simulations are quite opposite approaches that naturally prefer different settings of the underlying systems. From the AI perspective, the following is beneficial:

- *Heterogeneous accelerators*. While traditional HPC systems rely on CPUs and possibly GPUs, primarily integrating accelerators (e.g., GPUs or TPUs) is essential for optimizing AI/ML workloads [18, 7].
- *Mixed precision*. While simulation and numerical differential equation solvers used in HPC workflows benefit high precision, training neural networks may prefer more noisy iterations to fewer precise ones [14, 33, 42].
- *Parallel processing*. AI/ML tasks benefit model and data parallelisms, which typically require hardware-specific and topology-specific code optimizations to be effectively implemented on HPC systems [3, 7].
- *Large accelerator memory*. Training AI/ML models involves handling large datasets, necessitating substantial memory resources. Moreover, it is beneficial when memory is 'close' to a processing unit [26].
- *Distributed file system and bandwidth*. The distributed file system and interconnect in existing HPC facilities may not be optimized for AI/ML tasks. For example, transferring thousands of small data chunks across processing units may not suit the file system used, and I/O often becomes the training time bottleneck [7].

2.1. Why to Use HPC Systems for AI/ML?

Despite HPC systems not being necessarily AI-centric, it is still pragmatic and often the best choice of industrial or academic researchers to run AI tasks on them, all things considered:

- *Enhanced computational power*. HPC systems offer unparalleled computational capabilities essential for training large AI/ML models. For instance, LUMI, the fastest supercomputer in Europe, can perform at 379.7 petaflops. So, even suboptimized code runs faster than on smaller GPU clusters, see Fig. 1.

- *Scalability*. HPC systems are made to handle large-scale computations, making them suitable for AI/ML tasks that require significant parallel processing.
- *Existing infrastructure*. Leveraging existing HPC infrastructure can reduce the need for additional investments in new hardware, facilitating a more cost-effective approach to AI/ML research.
- *Energy Efficiency*. Modern HPC systems like JUPITER are designed with energy efficiency in mind, which is crucial for the sustainability of large-scale AI/ML operations.
- *Data privacy compliance*. EuroHPC systems often provide GDPR compliance 'by default', and the upcoming AI Factories will advise on even higher standards, including the AI Act.

2.2. Training Large Neural Network on MeluXina EuroHPC

The previous paragraph gave a general reason for using HPCs for AI. Next, we dig deeper into the AI challenges on EuroHPC systems, where things are further complicated by the diversity of these systems and the diverse nature of AI architectures used for AI tasks (e.g., transformers vs. xLSTMs). In 2021, a dedicated and detailed study [18] executed a set of Scientific Machine Learning (SciML) tasks across several HPC architectures in the US, analyzing the challenges of applying AI tasks on HPC. According to the study: "[. . .] Input and activation sizes limit batching and will ultimately mandate the exploitation of model parallelism; AI-optimized GPUs running SciML demand more PCIe, NVMe, and Lustre bandwidth than currently provided; Local NVMe used to feed SciML training workloads does not provide clear performance benefits at scale and should be evaluated against centralized fabric-attached storage or strong scaling with static partitioning of training data; Data scientists should structure models to exploit unused resources to reduce time per epoch. [. . .]".

2.3. Experiments, Results & Recommendations

To corroborate the findings above on EuroHPC systems and to obtain hands-on user experience, the ELISE project team executed the following minimalistic experiments in 2022. In addition, they also provided insight into how large an efficiency gap of a naive straightforward approach is: take a 'laptop' code and run it on better (HPC) hardware. The obtained results are summarized in Fig. 1.

Experimental setup. **HW**: European 'Tier 0' MeluXina EuroHPC[1], 'Tier 1' Danish life sciences supercomputer Computerome HPC, and 'Tier 2' University GPU cluster node. **Model**: Wave2Vec2.0 [1]. It represents a large enough transformer-based model ($\simeq 10^8$ parameters) to impose a challenge for training on a 'standard' Tier-2 GPU cluster. **Task**: To fine-tune a pre-trained (self-supervised on LibriSpeech dataset) speech-to-text model on the PolyAI/minds14 dataset. **Data**: PolyAI/minds14, the transcribed noisy audio files with intents extracted from a commercial system in the e-banking domain, associated with spoken examples in 14 diverse language varieties, see [13], also available at: https://paperswithcode.com/dataset/minds14. **Code**: Wave2Vec2 on HF, a Hugging Face implementation of the Wave2Vec2.0 model offering reproducibility.

Results & Recommendations. The two major challenges identified in the experiment can be summarized as follows. They demonstrate that while an infamous 'number of GPUs' may be a bottleneck for the largest models with optimized code, tackling challenges of many R&D AI tasks is more tangible and often within the reach of AI practitioners.

GPU memory (HBM) bottlenecks
HPC systems offer extensive resources, yet they show that it is not straightforward to use them efficiently. For instance, larger GPU DRAM allows practitioners to load and process larger mini-batches per GPU. Our experiments and the

[1] MeluXina EuroHPC provides NVIDIA Ampere A100 GPUs with 40GB HBM and AMD EPYC CPU. Tier 1 scratch Storage uses a Lustre file system (400 G/s). Its Accelerator Module offers 200 GPU nodes, each featuring 2 AMD Rome CPUs (32 cores @ 2.35 GHz - 128HT cores total) and 4 NVIDIA A100-40 GPUs. These nodes have 512 GB of RAM, a local SSD of 1.92 TB, and 2 HDRcards connecting them to the InfiniBand network. Additional supporting experiments on Computerome HPC (DK) using NVIDIA V100 (Volta) GPUs and 'Tier 2' University GPU cluster node with NVIDIA Titan X GPU and 12GB DRAM were conducted.

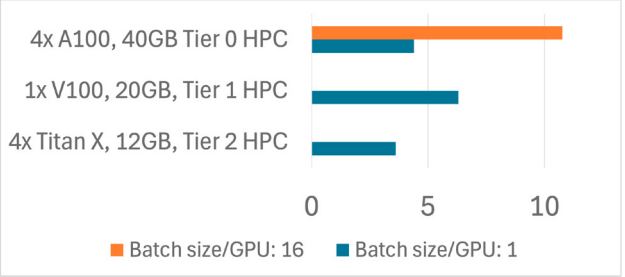

Fig. 1. **Migrating AI code across HPCs without optimizing it for underlying HW is inefficient and sub-optimal**. Throughput (number of processed training samples per second per GPU) of vanilla 'off-the-shelf' Hugging Face code during fine-tuning of the pre-trained Wave2vec2.0 transformer model are shown. Fine-tuning was done on three different HPC systems, ranging from local GPU cluster (Tier-2) to EU level HPC (Tier-0). The code was not optimized for any of the three HPC systems used. Results show that migrating code from the local university GPU cluster (Tier 2 HPC) to (Tier 0 HPC) without a change, i.e., keeping the same number of GPUs, only provides negligible gains. Significant gains are indicated (orange vs. blue) when batch size per GPU is increased from 1 to 16. However, despite using approximately half of GPU DRAM in the case of batch 16, the GPUs were heavily underutilized during training, operating at 6% of their peak performance on average. While the total gain from increasing batch size 16x is less than 3x in this case, it also leaves room for improvement in a range of 1 order of magnitude. **Throughput metric (x-axis)** = the number of training samples processed per second by a single GPU (an average over GPUs, 1 hour of training, and 5 runs).

study mentioned above [18] demonstrated that significant gains can be obtained by increasing batch size per GPU (Fig. 1, orange vs. blue). However, in the case of Wave2Vec2.0 fine-tuning, with all data samples ($\simeq$ 100 MBs) loaded on every GPU, we experienced GPUs running idle most of the time, waiting for gradients synchronization. This overhead was due to the Wave2Vec2.0 transformer with more than 10^8 parameters being large enough to make GPU-GPU gradient reduction take more than ten times longer than doing forward-backward passes[2]. This situation is common for fine-tuning pre-trained models for specific applications and training models from scratch. The good news is that code optimization may lead to a more than an order of magnitude reduction in training time.

Due to many models utilizing a 'funnel' architecture, the largest bottleneck will be the few widest layers. Thus, layer parallelization may not reduce the bottleneck, and intra-layer parallelization or other techniques may be required [18, 45]. Instead, when performance is bound by file system or interconnection bandwidth, it is possible to increase model complexity (e.g., making it deeper), balance hyperparameters such as learning rate and batch size, and choose a more advanced optimizer (e.g., see PyTorch [35]) to reduce the total number of iterations. Generally, a creative approach is advised. EuroHPC JU not only offers training and courses, e.g., on LUMI, but also provides so-called 'Development Access' and, recently, 'AI and Data Intensive Access' to EuroHPC facilities for these purposes.

I/O (Storage & Interconnect)

Handling and processing large datasets efficiently requires robust data management systems. Current HPC systems may not be fully optimized for the specific needs of AI/ML, such as the rapid access and processing of large volumes of unstructured data that often come in many small files scattered over the file system [21]. Besides high GPU memory and GPU-to-GPU bandwidth within one compute node, a node-to-node interconnect plays an immense role depending on AI application. For instance, training ResNet of 50 layers and 25×10^6 parameters using 32-bit precision on mini-batch of 32 on *A*100 GPU or higher would fully utilize throughput of $\approx$ 900 GB/s or higher, while the inter-node fabric currently available at HPC centers are, e.g., $\approx$ 50GB/s per each AMD MI250x GPU module (LUMI-G) at the time of writing[3]. While data parallelization may be hindered by node-to-node communication bottlenecks, model parallelization may also have limited effects due to many models utilizing a 'funnel' architecture, as discussed above.

While EuroHPC infrastructure updates are in progress, models will likely keep increasing their size. Thus, AI practitioners may consider other techniques, such as a gradient accumulation on GPUs over several batches (often used to overcome batch size limits imposed by available GPU memory) combined with an adjusted learning rate [40] before the all-reduce operation that concludes the round. This idea can be further extended to asynchronous training.

[2] To eliminate other communication costs (e.g., inter-node data transfers), we only used 4 GPUs on the same compute node for fine-tuning.
[3] Latest generation of NVLink(Switch) provides up to 1.8TB/s theoretical throughput

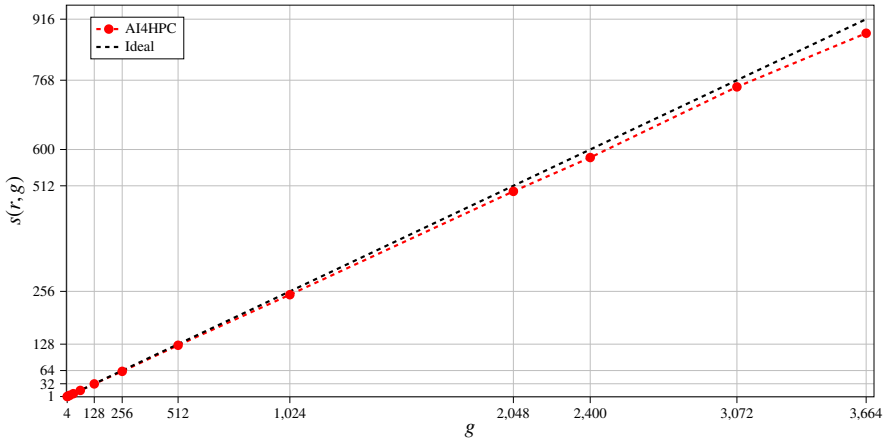

Fig. 2. AI4HPC scaling on the JUWELS system was demonstrated on 3,664 NVIDIA A100 GPUs with 96% efficiency. Here, a U-Net model consisting of 52 million parameters is trained with a synthetic dataset using the *Horovod* backend. The black line shows the ideal speed-up. g denotes the number of GPUs, while $s(r, g)$ is the speed-up.

Such approaches have common traits with federated learning, which also allows to tackle communication bottlenecks by reducing the size and frequency of inter-node communications (see Sec. 3).

2.4. Towards an HPC-optimized library for AI

The increasing focus on including accelerators in the EuroHPC hosting sites requires the next generation of AI codes to be *HPC-ready*. The challenge for the upcoming exascale systems is potentially even higher, as they should demonstrate good scalability with increasing communication overhead under more node-to-node interconnects.

In order to address these challenges and to better utilize HPC systems, the AI4HPC library was developed in the EU-funded CoE RAISE project. AI4HPC is targeted to allow users to seamlessly and efficiently train their AI models in a distributed setting on the EU HPC infrastructure while including various code optimizations to improve training performance. The library has already been built and tested across various European HPC centers. At the time of writing this manuscript, the tested sites are JUWELS at Jülich Supercomputing Center (JSC), JURECA at JSC, DEEP-EST at JSC, LUMI at IT Center for Science (CSC), CTE-AMD at Barcelona Supercomputing Center (BSC), Leonardo at CINECA, and JEDI at JSC, the first module of the upcoming exascale supercomputer JUPITER.

AI4HPC includes data manipulation routines, the collection of various ML architectures, optimization routines for efficient training, the Hyperparameter Optimization (HPO) module, and monitoring and performance benchmarking tools. The library includes multiple distributed training backends, which users can exploit for their training workloads. The integrated backends are PyTorch DDP, Horovod, HeAT and DeepSpeed. In terms of large-scale performance measurement, the scalability of the library has been demonstrated (shown in Fig. 2) on the JUWELS system with up to 96% efficiency on 3,664 NVIDIA A100 GPUs.

As mentioned in the challenges above, apart from performance requirements, submitting jobs on HPC systems requires building the correct environment to enable execution. This task involves understanding the modules and their continuously updated versions. For the AI4HPC library, a proper execution environment is automatically created for each configured HPC center. Integration of new centers in the library is straightforward. The CoE RAISE project also developed another tool, LAMEC, which specifically manages the environment and generates job submission scripts for multiple HPC centers in a simplified GUI. LUMI, Vega, Karolina, and Leonardo are already integrated into LAMEC.

3. Cross-facility Deep Learning

If squeezing every last drop of computing performance from larger and larger HPC centers is pivotal for sustaining the scales of modern AI, exploring cross-facility federated science [10] is fundamental for many practical reasons:

- *Reaching even larger scales*. If a complex application can be decomposed into (almost) embarrassingly parallel modules, these modules can be efficiently offloaded to different HPC facilities, increasing concurrency and reducing time-to-solution. Typical AutoML tasks, e.g., hyperparameter optimization and neural architecture search, fall into this category [17].
- *Enhancing resource utilization*. During peak load or maintenance periods, jobs can linger in a pending state for a long time. The introduction of a federated meta-scheduler [23] or a decentralized scheduling plane [28] that distributes jobs across multiple HPC facilities would benefit both users, reducing time-to-solution, and systems, reducing idle times in underloaded machines.
- *Exploiting data locality*. Since the advent of physics-informed neural networks [37], Deep Learning has often been coupled with large-scale scientific simulations to improve accuracy and reduce time-to-solution [27, 20]. In situ data processing [4], i.e., analysing data where they are generated, is a promising approach to avoid network communications and reduce inference latency [9]. A federated HPC ecosystem would allow scientists to set up largely distributed training processes, allocating model replicas near data sources.
- *Ensuring data privacy*. In some cases, data cannot be moved from their original, trusted location for privacy and security reasons. Still, cross-silo federated learning approaches [30] that train models on multiple datasets without disclosing them can benefit all involved parties. With the advent of Deep Learning in sensitive sectors like healthcare [11] and finance [34], supporting this kind of scenario is becoming crucial.
- *Guaranteeing fairness*. Foundation models [8] with trillions of parameters necessitate an entire exascale data center for training from scratch [29]. However, prolonged exclusive resource allocations to a single European initiative can undermine the principle of fair resource usage, disadvantaging smaller, national projects. Distributing computation among different centers can ensure a more equitable European HPC ecosystem.

The challenges of cross-facility science have been studied for various application domains, from astrophysics to genomics [43], to molecular dynamics [36], to Deep Learning [5], and orchestrating cross-site experiments on pairs of European HPC facilities has already proven feasible. In [36], the authors run a large-scale plasma simulation analysis using a sparse grid combination technique to mitigate the curse of dimensionality. The experiment ran on top of two facilities, i.e., HAWK at HLRS (Stuttgart, DE) and SuperMUC at LRZ (Garching, DE). In [5], a LLaMAv2-7B model is trained from scratch using a federated learning approach to reduce the inter-site communication overhead. Again, the experiment involved two facilities: Leonardo at CINECA (Bologna, IT) and Karolina at IT4I (Ostrava, CZ). Both cases involve small federations of homogeneous (x86-64 CPUs, NVIDIA GPUs) and relatively close HPC facilities.

3.1. Experiments, Results & Recommendations

In order to assess the EuroHPC readiness for large-scale cross-facility experiments, we ran an extended version of the federated learning experiment described in [5]. This time, we tried a federated training of a LLaMAv3 8B model on top of three different HPC sites: Leonardo in Bologna, Italy (Intel CPUs, NVIDIA GPUs), LUMI in Kajaani, Finland (AMD CPUs, AMD GPUs), and MeluXina in Luxembourg (AMD CPUs, NVIDIA GPUs). Compared to the previous attempts, this is a far more challenging setting, with three geographically far machines mounting heterogeneous accelerators. Fig. 3 compares the land surface covered by the three European cross-facility experiments, showing how the one discussed here is by far the largest European cross-HPC experiment described in the literature. This work focuses on the challenges encountered while setting up the federation and orchestrating the application's life cycle. A detailed description of the technical aspects behind this experiment and the obtained results will be provided elsewhere.

Results & Recommendations. Standard distributed training processes can be modelled as Bulk Synchronous Programming (BSP) workloads [44], where each superstep locally computes the forward pass and backpropagation on each node, communicates the gradients to all other nodes, and globally synchronizes the process to compute the result-

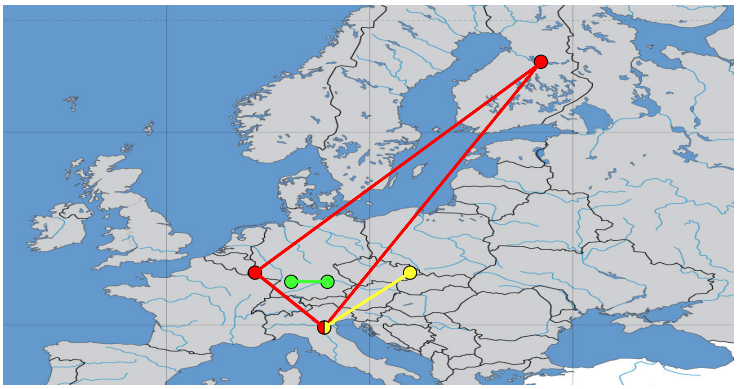

Fig. 3. Map of the European cross-facility experiments: in green the plasma simulation analysis on SuperMUC and HAWK [36], in yellow the federated learning experiment on Leonardo and Karolina [5], and in red the federation among Leonardo, MeluXina, and LUMI HPC facilities. The last configuration requires data to be exchanged across a total of about 5.200 km. The total land surface covered is about 680.000 km^2, more or less the 16% of the EU surface area.

ing gradient [2]. Federated learning workloads minimize cross-cluster data transfers by only requiring synchronization and communication of model weights at the end of each FL round [5]. However, since BSP workloads contain global barriers that are very sensitive to stragglers, performance fluctuations in different HPC facilities significantly undermine the overall performance of the training workload. **Challenge**: Reduce cross-site performance fluctuations, which depend on several factors:

- *Heterogeneous hardware and software stacks*. Deep learning workloads heavily rely on hardware accelerators, especially GPUs. Even if different accelerators can scale very well for a wide range of dataset sizes, the absolute duration of a training step varies significantly between different hardware, with AMD GPUs on LUMI being up to 6 times slower than NVIDIA GPUs on Leonardo and MeluXina. This difference is due to the different hardware and heterogeneous software stacks available in different computing facilities. **Solution**: Provide unified AI software collections on different facilities optimised for the underlying hardware accelerators. Better support for HPC package managers (e.g., Spack [12]) and software containers (e.g., Singularity [25]) could also help in achieving (performance) portability.
- *Unbalanced queuing times*. Different HPC facilities reach peak load during different periods, making queuing times highly variable among sites. Plus, large HPC facilities are complex systems that still need frequent maintenance periods, making it difficult to find all involved sites up and running simultaneously. **Solution**: Implement a cross-facility orchestration plane. Several meta-scheduling [23] or distributed scheduling [28] algorithms can be derived from grid computing frameworks and combined with a vendor-agnostic compatibility layer [15] to avoid lock-in. Plus, cross-facility workflow management systems like StreamFlow [6], PyCOMPSs [41], or JAWS [22] can help orchestrate cross-HPC workloads and offer non-functional requirements out of the box, e.g., portability, reproducibility, fault tolerance, provenance tracking, and secure efficient data transfers.
- *Slow and unstable inter-site network*. Relying on the public Internet for heavy data transfers leads to suboptimal and highly variable transfer times that slow down the communication phase of BSP supersteps. With huge, trillion-parameter models, this overhead can become significant. **Solution**: Procure a dedicated high-speed interconnection plane among EuroHPC facilities.

Another challenge we faced during the federation setup was the extreme heterogeneity in HPC resource access. The PRACE program already allows cross-facility resource requests. However, the way different sites manage resource grants varies significantly in several aspects: monthly vs bulk allocation of compute hours, core hours vs node hours grants, and significantly different amounts of computing hours granted by different facilities. **Challenge**: Revise the resource allocation program with HPC federations as first-class citizens, allowing users to submit cross-facility experiment proposals.

4. Quantum/Classical AI in EuroQCS-Poland

Until recently, there has been 'classical' digital computing and quantum computing world. Despite its impressive theoretical yet elusive benefits, quantum computing (QC) has been approached by academics and industry primarily using quantum simulators. Only recently, hybrid-quantum computing, which combines classical and quantum computation within one task, has been shown by experiments and benchmarks to be practically helpful. In practice, 'hybrid' computation is a back-and-forth collaboration approach where different aspects of a problem are passed between the quantum and classical tools best suited for each stage, thus accelerating the overall process and delivering a performance boost.

EuroHPC is also working to integrate quantum computing into its broader supercomputing ecosystem, funding research projects and building infrastructure to support this hybrid approach. The idea is to develop quantum-ready HPC systems where different quantum computers can be efficiently integrated with classical supercomputers. This initiative offers a novel interpretation of quantum computers as accelerator platforms in genuine HPC environments in Europe. The foreseen integration will require essential R&D developments towards a hybrid software stack managing both HPC and QC workloads. This effort positions Europe at the forefront of developing quantum-accelerated HPC systems to address next-generation computational challenges.

Owned by the EuroHPC JU, the first quantum system EuroQCS-Poland will be hosted in 2025 at the Poznan Supercomputing and Networking Center (PSNC) and integrated into the local HPC infrastructure, allowing for remote access via the co-located supercomputer connected to the PIONIER NREN and Pan-European GEANT networks. The quantum system will be a digital, gate-based quantum computer based on trapped ions offering 20-plus physical qubits delivered by AQT [16].

To select the best quantum system for EuroQCS-Poland, a set of application benchmarks, including AI/ML algorithms, have been developed to evaluate the overall performance of available European quantum computing technologies [24]. The well-known MNIST dataset, containing images of handwritten digits collected for image classification, has been used as input. The Quantum Support Vector Machine (QSVM) was proposed as AI/ML benchmark as it is an algorithm that applies a quantum kernel to a Support Vector Machine (SVM) for classification and regression [39]. In a nutshell, SVM finds an optimal hyperplane between classes, but when data is not linearly separable, a kernel function maps it to a higher-dimensional space. The proposed QSVM use case was successfully tested on a trapped-ion quantum computer to create a feature map, potentially identifying complex patterns that classical methods could not identify.

5. Conclusion and future work

The rapid advancements in AI and Machine Learning necessitate a robust computational infrastructure to support cutting-edge research and industrial applications on a European scale. The EuroHPC systems are well suited for AI advancements, yet they still require substantial effort from AI practitioners to exploit their resources efficiently. The ELISE Horizon Europe project's recent report recommends that the EuroHPC systems move further toward a community-driven R&D ecosystem that is easily accessible and powerful and has links to quantum, edge AI, and neuromorphic computing. While edge computing is an increasingly important deployment platform for many AI applications, such as hearing devices [19], neuromorphic computing constitutes an emerging and energy-efficient alternative to von Neumann architectures [31].

Extending the existing EuroHPC systems with the upcoming *hybrid classical-quantum systems* and an AI-optimized supercomputer, announced recently with AI Factories, will address several challenges with porting AI tasks to HPC. In addition, by connecting to over 100 international research teams and platform providers over the last three years, the ELISE project formulated a set of recommendations for the R&D community to advance AI-driven innovation. These include supporting the *active participation* of the AI R&D community in designing the platform, e.g., within the newly set up EuroHPC User Forum, where the AI-focused R&D community is currently underrepresented.

Moreover, the AI community and the EuroHPC JU should join forces to support a *platform-agnostic* format of AI models. This effort will enable a smooth transition and *federation* of training, research and models across EuroHPC systems, whose current diversity may become a crucial bottleneck for R&D in Europe in the future and should be addressed, for instance, adopting some of the solutions suggested in this work.

Acknowledgements

This project was funded by ELISE EU Grant agreement ID: 951847. We would like to acknowledge the participation of over a hundred anonymous research teams, from academia, government, and private sectors, incl. banking, automotive, telecommunications, etc., who took part in our workshops, surveys, and work groups including following: MLOPs Summer school, DTU; On-line Crash Course for LLMs on EuroHPCs; MLOPs Course for Industry, DTU; European AI Platform workshop, European Networks of AI Excellence and the Center of Excellence in Exascale Computing, CoE RAISE; AI on LUMI HPC; VISION, RIAG, INFRAG meeting; First HPC User Day; AI Platform of the Future, Workshop No.1 & 2; BEST summer school. Your willingness to share your experiences and perspectives provided essential data and nuanced understanding that have been crucial. The same goes to Nvidia, Dr. rer. nat. Maria Athelogou and Frédéric Parienté. Special thanks to the EuroHPC JU sites for providing access to HPC systems, which made this project possible. And to European Commission and EuroHPC JU representatives, Mladen Skelin, Jan Hückmann, Dr. Daniel Opalka, Dr. Lilit Axner, Miguel Rubio and many others for inviting the ELISE team to coordination and discussion meetings, making the project influential and actionable. We are also grateful to the administrative and technical staff for their continuous assistance and to our peer reviewers for their constructive feedback.

References

[1] Baevski, A., Zhou, H., Mohamed, A., Auli, M., 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. arXiv:2006.11477.

[2] Ben-Nun, T., Hoefler, T., 2019. Demystifying parallel and distributed deep learning: An in-depth concurrency analysis. ACM Comput. Surv. 52, 65:1–65:43. doi:10.1145/3320060.

[3] Brewer, W., Behm, G., Scheinine, A., Parsons, B., Emeneker, W., Trevino, R.P., 2020. Inference benchmarking on hpc systems, in: 2020 IEEE High Performance Extreme Computing Conference (HPEC), IEEE. pp. 1–9.

[4] Choi, J.Y., Chang, C., Dominski, J., Klasky, S., Merlo, G., Suchyta, E., Ainsworth, M., Allen, B., Cappello, F., Churchill, M., Davis, P.E., Di, S., Eisenhauer, G., Ethier, S., Foster, I.T., Geveci, B., Guo, H., Huck, K.A., Jenko, F., Kim, M., Kress, J., Ku, S., Liu, Q., Logan, J., Malony, A.D., Mehta, K., Moreland, K., Munson, T.S., Parashar, M., Peterka, T., Podhorszki, N., Pugmire, D., Tugluk, O., Wang, R., Whitney, B., Wolf, M., Wood, C., 2018. Coupling exascale multiphysics applications: Methods and lessons learned, in: 14th IEEE International Conference on e-Science, e-Science 2018, Amsterdam, The Netherlands, IEEE Computer Society. pp. 442–452. doi:10.1109/ESCIENCE.2018.00133.

[5] Colonnelli, I., Birke, R., Malenza, G., Mittone, G., Mulone, A., Galjaard, J., Chen, L.Y., Bassini, S., Scipione, G., Martinovič, J., Vondrák, V., Aldinucci, M., 2024. Cross-facility federated learning. Procedia Computer Science 240, 3—-12. doi:10.1016/j.procs.2024.07.003.

[6] Colonnelli, I., Cantalupo, B., Merelli, I., Aldinucci, M., 2021. Streamflow: Cross-breeding cloud with HPC. IEEE Trans. Emerg. Top. Comput. 9, 1723–1737. doi:10.1109/TETC.2020.3019202.

[7] De Sensi, D., Pichetti, L., Vella, F., De Matteis, T., Ren, Z., Fusco, L., Turisini, M., Cesarini, D., Lust, K., Trivedi, A., et al., 2024. Exploring gpu-to-gpu communication: Insights into supercomputer interconnects. arXiv preprint arXiv:2408.14090 .

[8] Devlin, J., Chang, M., Lee, K., Toutanova, K., 2019. BERT: Pre-training of deep bidirectional transformers for language understanding, in: NAACL-HLT, Association for Computational Linguistics. pp. 4171–4186. doi:10.18653/V1/N19-1423.

[9] Do, T.M.A., Pottier, L., Caíno-Lores, S., da Silva, R.F., Cuendet, M.A., Weinstein, H., Estrada, T., Taufer, M., Deelman, E., 2021. A lightweight method for evaluating *in situ* workflow efficiency. J. Comput. Sci. 48, 101259. doi:10.1016/J.JOCS.2020.101259.

[10] Enders, B., Bard, D., Snavely, C., Gerhardt, L., Lee, J., Totzke, B., Antypas, K., Byna, S., Cheema, R., Cholia, S., Day, M.R., Gaur, A., Greiner, A., Groves, T.L., Kiran, M., Koziol, Q., Rowland, K., Samuel, C., Selvarajan, A., Sim, A., Skinner, D., Thomas, R.C., Torok, G., 2020. Cross-facility science with the superfacility project at LBNL, in: 2nd IEEE/ACM Annual Workshop on Extreme-scale Experiment-in-the-Loop Computing, XLOOP@SC 2020, Atlanta, GA, USA, IEEE. pp. 1–7. doi:10.1109/XLOOP51963.2020.00006.

[11] Esteva, A., Robicquet, A., Ramsundar, B., Kuleshov, V., DePristo, M., Chou, K., Cui, C., Corrado, G., Thrun, S., Dean, J., 2019. A guide to deep learning in healthcare. Nature Medicine 25, 24–29. doi:10.1038/s41591-018-0316-z.

[12] Gamblin, T., LeGendre, M.P., Collette, M.R., Lee, G.L., Moody, A., de Supinski, B.R., Futral, S., 2015. The spack package manager: bringing order to HPC software chaos, in: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, SC 2015, Austin, TX, USA, ACM. pp. 40:1–40:12. doi:10.1145/2807591.2807623.

[13] Gerz, D., Su, P.H., Kusztos, R., Mondal, A., Lis, M., Singhal, E., Mrkšić, N., Wen, T.H., Vulić, I., 2021. Multilingual and cross-lingual intent detection from spoken data. arXiv:2104.08524.

[14] Goodfellow, I., Bengio, Y., Courville, A., Bengio, Y., 2016. Deep learning. MIT press Cambridge.

[15] Hategan-Marandiuc, M., Merzky, A., Collier, N.T., Maheshwari, K., Ozik, J., Turilli, M., Wilke, A., Wozniak, J.M., Chard, K., Foster, I.T., da Silva, R.F., Jha, S., Laney, D.E., 2023. PSI/J: A portable interface for submitting, monitoring, and managing jobs, in: 19th IEEE International Conference on e-Science, e-Science 2023, Limassol, Cyprus, IEEE. pp. 1–10. doi:10.1109/E-SCIENCE58273.2023.10254912.

[16] Humble, T.S., McCaskey, A., Lyakh, D.I., Gowrishankar, M., Frisch, A., Monz, T., 2021. Quantum computers for high-performance computing. IEEE Micro 41, 15–23. doi:10.1109/MM.2021.3099140.

[17] Hutter, F., Kotthoff, L., Vanschoren, J. (Eds.), 2019. Automated Machine Learning - Methods, Systems, Challenges. The Springer Series on Challenges in Machine Learning, Springer. doi:10.1007/978-3-030-05318-5.

[18] Ibrahim, K.Z., Nguyen, T., Nam, H.A., Bhimji, W., Farrell, S., Oliker, L., Rowan, M., Wright, N.J., Williams, S., 2021. Architectural requirements for deep learning workloads in hpc environments, in: 2021 International Workshop on Performance Modeling, Benchmarking and Simulation of High Performance Computer Systems (PMBS), IEEE. pp. 7–17.
[19] Iftikhar, S., Gill, S.S., Song, C., Xu, M., Aslanpour, M.S., Toosi, A.N., Du, J., Wu, H., Ghosh, S., Chowdhury, D., et al., 2023. Ai-based fog and edge computing: A systematic review, taxonomy and future directions. Internet of Things 21, 100674.
[20] Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., Bridgland, A., Meyer, C., Kohl, S.A.A., Ballard, A.J., Cowie, A., Romera-Paredes, B., Nikolov, S., Jain, R., Adler, J., Back, T., Petersen, S., Reiman, D., Clancy, E., Zielinski, M., Steinegger, M., Pacholska, M., Berghammer, T., Bodenstein, S., Silver, D., Vinyals, O., Senior, A.W., Kavukcuoglu, K., Kohli, P., Hassabis, D., 2021. Highly accurate protein structure prediction with alphafold. Nature 596, 583–589. doi:10.1038/s41586-021-03819-2.
[21] Khan, A., Paul, A.K., Zimmer, C., Oral, S., Dash, S., Atchley, S., Wang, F., 2022. Hvac: Removing i/o bottleneck for large-scale deep learning applications, in: 2022 IEEE International Conference on Cluster Computing, pp. 324–335. doi:10.1109/CLUSTER51413.2022.00044.
[22] Kirton, E., Foster, B., Froula, J.L., Sul, S.J., Trong, S., Kollmer, A., Melara, M., Rowland, K., Rath, G., USDOE, 2020. Joint genome institute analysis workflow service (jaws) v2.0. doi:10.11578/dc.20210617.3.
[23] Kurowski, K., Nabrzyski, J., Oleksiak, A., Weglarz, J., 2008. A multicriteria approach to two-level hierarchy scheduling in grids. J. Sched. 11, 371–379. doi:10.1007/S10951-008-0058-8.
[24] Kurowski, K., Rydlichowski, P., Wojciechowski, K., Pecyna, T., Slysz, M., 2023. Application performance benchmarks for quantum computers. CoRR abs/2310.13637. doi:10.48550/ARXIV.2310.13637, arXiv:2310.13637.
[25] Kurtzer, G.M., Sochat, V., Bauer, M.W., 2017. Singularity: Scientific containers for mobility of compute. PLOS ONE 12, 1–20. doi:10.1371/journal.pone.0177459.
[26] Kwon, Y., Rhu, M., 2018. A case for memory-centric hpc system architecture for training deep neural networks. IEEE Computer Architecture Letters 17, 134–138. doi:10.1109/LCA.2018.2823302.
[27] Lam, R., Sanchez-Gonzalez, A., Willson, M., Wirnsberger, P., Fortunato, M., Alet, F., Ravuri, S., Ewalds, T., Eaton-Rosen, Z., Hu, W., Merose, A., Hoyer, S., Holland, G., Vinyals, O., Stott, J., Pritzel, A., Mohamed, S., Battaglia, P., 2023. Learning skillful medium-range global weather forecasting. Science 382, 1416–1421. doi:10.1126/science.adi2336.
[28] Lu, K., Subrata, R., Zomaya, A.Y., 2007. On the performance-driven load distribution for heterogeneous computational grids. J. Comput. Syst. Sci. 73, 1191–1206. doi:10.1016/J.JCSS.2007.02.007.
[29] Ma, Z., He, J., Qiu, J., Cao, H., Wang, Y., et al., 2022. BaGuaLu: targeting brain scale pretrained models with over 37 million cores, in: ACM PPoPP, pp. 192–204. doi:10.1145/3503221.3508417.
[30] McMahan, B., Moore, E., Ramage, D., Hampson, S., y Arcas, B.A., 2017. Communication-efficient learning of deep networks from decentralized data, in: Singh, A., Zhu, X.J. (Eds.), Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, Fort Lauderdale, FL, USA, PMLR. pp. 1273–1282.
[31] Mead, C., 2023. Neuromorphic engineering: In memory of misha mahowald. Neural Computation 35, 343–383.
[32] NICULESCU, V., 2019. High performance computing in big data analytics. Applied Medical Informatics .
[33] Noh, H., You, T., Mun, J., Han, B., 2017. Regularizing deep neural networks by noise: Its interpretation and optimization. Advances in Neural Information Processing Systems 30.
[34] Özbayoglu, A.M., Gudelek, M.U., Sezer, O.B., 2020. Deep learning for financial applications : A survey. Appl. Soft Comput. 93, 106384. doi:10.1016/J.ASOC.2020.106384.
[35] Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A., 2017. Automatic differentiation in pytorch .
[36] Pollinger, T., Craen, A.V., Niethammer, C., Breyer, M., Pflüger, D., 2023. Leveraging the compute power of two HPC systems for higher-dimensional grid-based simulations with the widely-distributed sparse grid combination technique, in: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, SC 2023, Denver, CO, USA, ACM. pp. 84:1–84:14. doi:10.1145/3581784.3607036.
[37] Raissi, M., Perdikaris, P., Karniadakis, G.E., 2019. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J. Comput. Phys. 378, 686–707. doi:10.1016/J.JCP.2018.10.045.
[38] Rumelhart, D.E., Hinton, G.E., Williams, R.J., 1986. Learning representations by back-propagating errors. Nature 323, 533–536. doi:10.1038/323533a0.
[39] Slysz, M., Kurowski, K., Waligóra, G., Węglarz, J., 2023. Exploring the capabilities of quantum support vector machines for image classification on the mnist benchmark, in: Computational Science – ICCS 2023, Springer Nature Switzerland, Cham. pp. 193–200.
[40] Smith, S.L., Kindermans, P., Ying, C., Le, Q.V., 2018. Don't decay the learning rate, increase the batch size, in: 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings, OpenReview.net.
[41] Tejedor, E., Becerra, Y., Alomar, G., Queralt, A., Badia, R.M., Torres, J., Cortes, T., Labarta, J., 2017. PyCOMPSs: Parallel computational workflows in python. Int. J. High Perform. Comput. Appl. 31, 66–82. doi:10.1177/1094342015594678.
[42] Thomas, V., Pedregosa, F., Merriënboer, B., Manzagol, P.A., Bengio, Y., Le Roux, N., 2020. On the interplay between noise and curvature and its effect on optimization and generalization, in: International Conference on Artificial Intelligence and Statistics, PMLR. pp. 3503–3513.
[43] Tyler, N., Jr., R.A.K., Bard, D., Nugent, P., 2022. Cross-facility workflows: Case studies with active experiments, in: IEEE/ACM Workshop on Workflows in Support of Large-Scale Science, WORKS 2022, Dallas, TX, USA, IEEE. pp. 68–75. doi:10.1109/WORKS56498.2022.00014.
[44] Valiant, L.G., 1990. A bridging model for parallel computation. Commun. ACM 33, 103–111. doi:10.1145/79173.79181.
[45] Zeng, Z., Liu, C., Tang, Z., Chang, W., Li, K., 2021. Training acceleration for deep neural networks: A hybrid parallelization strategy, in: 2021 58th ACM/IEEE Design Automation Conference (DAC), pp. 1165–1170. doi:10.1109/DAC18074.2021.9586300.