Enabling Hyperparameter-Tuning of AI Models for Healthcare using the CoE RAISE Unique AI Framework for HPC

M. Riedel*†, C. Barakat*†, S. Fritsch†, M. Aach†, J. Busch†, A. Lintermann†, A. Schuppert‡, S. Brynjólfsson*, H. Neukirchen*, M. Book*

* School of Engineering and Natural Sciences, University of Iceland, Iceland
† Jülich Supercomputing Centre, Forschungszentrum Jülich, Germany
‡ Joint Research Centre for Computational Biomedicine, University Hospital RWTH Aachen, Germany
c.barakat@fz-juelich.de, morris@hi.is, s.fritsch@fz-juelich.de, m.aach@fz-juelich.de, j.busch@fz-juelich.de, a.lintermann@fz-juelich.de, schuppert@aices.rwth-aachen.de, sb@hi.is, helmut@hi.is, book@hi.is

***Abstract*—The European Center of Excellence in Exascale Computing "Research on AI- and Simulation-Based Engineering at Exascale" (CoE RAISE) is a project funded by the European Commission. One of its central goals is to develop a Unique AI Framework (UAIF) that simplifies the development of AI models on cutting-edge supercomputers. However, those supercomputers' High-Performance Computing (HPC) environments require the knowledge of many low-level modules that all need to work together in different software versions (e.g., TensorFlow, Python, NCCL, PyTorch) and various concrete supercomputer hardware deployments (e.g., JUWELS, JURECA, DEEP, JUPITER and other EuroHPC Joint Undertaking HPC resources). This paper will describe our analyzed complex challenges for AI researchers using those environments and explain how to overcome them using the UAIF. In addition, it will show the benefits of using the UAIF hypertuning capability to make AI models better (i.e., better parameters) and faster by using HPC. Also, to demonstrate that the UAIF approach is indeed simple, we describe the adoption of selected UAIF building blocks by healthcare applications. The examples include AI models for the Acute Respiratory Distress Syndrome (ARDS). Finally, we highlight other AI models of use cases that co-designed the UAIF.**

***Keywords*—High-Performance Computing; Software Framework; Machine Learning; Deep Learning; Quantum Computing**

I. Introduction

Artificial Intelligence (AI) is a technology that is still relatively new in exploiting High-Performance Computing (HPC) systems for a wide variety of application domains compared to traditional simulation sciences using numerical methods based on known physical laws. Therefore, the European Centre of Excellence - Research on AI- and Simulation-Based Engineering at Exascale (CoE RAISE)[1] develops a Unique AI Framework (UAIF) to provide seamless solutions for a wide variety of complex AI applications. The timespan of CoE RAISE is from 2020 until the end of 2023 whereby several EuroHPC JU National Competence Centers (NCCs) have already agreed to maintain the UAIF beyond its project lifetime. Several NCCs (e.g., Germany, Iceland) use parts of UAIF already with different use cases while we highlight one particular use case in this paper. The UAIF aims at the convergence of HPC and innovative AI techniques making it easier for AI researchers to make use of cutting-edge hardware infrastructures. In addition to HPC hardware infrastructures, a seamlessly usable and versatile Software Infrastructure is critical for accelerating convergence through new AI toolsets that are ready to scale for enormous quantities of datasets. CoE RAISE considers AI requirements of Computing-driven Use Cases using numerical methods based on known physical laws. Use case examples include AI used in Computational Fluid Dynamics (CFD) problems (i.e., wind wheels, turbulence, hydrogen research, coating, etc.). CoE RAISE also addresses AI requirements of Data-driven Use Cases with large datasets of measurement devices. Use case examples include AI used in additive manufacturing (i.e., 3D printing quality assurance), dataset analysis from the Large Hadron Collider (LHC) at CERN, and remote sensing data analysis. The UAIF methodologies are co-designed by the above use cases to ensure its usage by a wide variety of scientific and engineering applications. The co-design activities leverage the Interaction Room Methodology (1, 2) using Mural Boards[2].

HPC environments getting more complex with a massive increase in technology (e.g., hierarchical storage and malleability) and heterogenity (e.g., different GPU vendors). Domain users and AI researchers are often overwhelmed when using HPC resources and cant focus on their specific domain problem. For example, the high amount of AI packages on various HPC machines in different versions and the vast availability of different module environments[3] for those is hard to understand. Healthcare experts such as those addressed in this paper are overwhelmed with these low-level environments and demand a lower barrier to use HPC/AI methods such as hyperparameter optimization (HPO) of AI models.

This work was performed in the Center of Excellence (CoE) Research on AI- and Simulation-Based Engineering at Exascale (RAISE) receiving funding from EU's Horizon 2020 Research and Innovation Framework Programme H2020-INFRAEDI-2019-1 under grant agreement no. 951733. Icelandic HPC National Competence Center is funded by the EuroCC-1 projects that has received funding from the EU HPC Joint Undertaking (JU) under grant agreement No 951732. Parts of the work have been also supported by the European Digital Innovation Hub (EDIH) of Iceland (EDIH-IS) funded in parts by the Digital Europe Programme. Finally, parts of this work were performed in the SMITH Project receiving funding from the German Federal Ministry of Education and Research.

[1] https://www.coe-raise.eu/

[2] Mural Boards https://mural.co

[3] https://modules.sourceforge.net/

This paper provides a comprehensive overview of the UAIF software layout design that addresses the challenge mentioned above, including major updates since its last publication by M. Riedel et al. (3). Its main contribution is to describe each component of UAIF and the comprehensive application co-design efforts of use cases over two years. One key component of the UAIF that is used in many use cases is the use of HPO tools. Therefore, we also show a short adoption example of the UAIF in healthcare leveraging this popular HPO component. While more healthcare-related AI models leverage HPC as shown in (4), this paper focuses particularly on the adoption of the HPO through the CoE RAISE UAIF.

The remainder of this paper is structured as follows. Section 1 starts with the introduction to CoE RAISE and the approach of co-designing the UAIF with distinct engineering and scientific use cases. A review of related work is given in Section 2, while Section 3 describes in detail the main contribution of this paper that is the result of the CoE RAISE UAIF co-design process. Section 4 then described one adoption of the UAIF components by use cases that were not involved in the original co-design process with a special emphasis on hyperparameter tuning. The paper ends with some concluding remarks.

II. Related Work

Despite the fact that the HPC environments are getting more complex for domain-specific researchers, there is quite a history of tools and platforms that make it easier for researchers to access HPC resources. One example is a HPC-driven Grid middleware Uniform Interface to Computing Resources (UNICORE)[4] (5). It offers a ready-to-run system including client and server software that makes HPC resources available in a seamless and secure way. In contrast to the UAIF, UNICORE is not focused on a streamlined set of components for AI on HPC resources. UNICORE is much more general and operates basically one level above the module environments and schedulers providing REST APIs or GUIs.

Other similar ideas of a framework or collection of tools are existing in the commercial space from key vendors. One example is the Intel oneAPI HPC Tools for Developers[5]. These tools enable developers to build, analyze, optimize, and scale HPC applications across multiple types of architectures more easily using the Intel oneAPI Base Toolkit and Intel oneAPI HPC Toolkit. The tools include approaches such as state-of-the-art techniques in vectorization, multithreading, multinode parallelization, and memory optimization so that one can more easily build software that is HPC-ready. In contrast to UAIF, these tools are not fully open source and the overall design of the oneAPI is not focused on HPC at Exascale but more general use by HPC applications. For the sake of completeness, several solutions also exist in Clouds like in Amazon Web Services (AWS) with SageMaker[6]. Those services do not work on EuroHPC JU nor EU HPC systems.

[4]https://www.unicore.eu/

[5]https://www.intel.com/content/www/us/en/high-performance-computing/hpc-software-and-programming.html

[6]https://aws.amazon.com/sagemaker/

III. CoE RAISE UAIF Co-Design Layout Results

This section provides a comprehensive overview of the framework software layout after the final co-design phase of CoE RAISE, including major updates since the last publication by Riedel et al. in (3). It briefly describes each component of the framework. Fig. 1 includes several components that are not directly under the control of CoE RAISE, but are identified as important dependencies for the UAIF running on HPC systems. For releases of a wide variety of UAIF components, we refer to the CoE RAISE Web page.

A. Applications

The application layer contains different use cases that contributed to the co-design of the Unique AI Framework (UAIF), but also initiatives that might adopt the UAIF. In Fig. 1, the large red arrow represents the co-design activities that influence the reference architecture components. The large green arrows represent the benefits and adoption potential for external project activities in the larger HPC ecosystem.

Component (A) in Fig. 1 represents the co-design efforts of the UAIF based on compute- and data-intensive use cases[7]. Hence, several different use case types contributed to benchmarking and proof of scalability of components of the UAIF on various HPC systems. During the project, and especially in the last few months, a clear picture is provided of what components are relevant for the UAIF.

Also, a wide variety of CoEs[8] have been funded in different domain-specific areas providing use cases that leverage simulation sciences or AI/HPC methods to utilize the emerging Exascale computing. As shown in Fig. 1 (B), UAIF adoption is recommended to CoEs as a way to prevent AI developers in domain-specific sciences from wasting a lot of effort in configuring the correct AI setup on HPC.

In addition, National Competence Centers (NCCs) has been created under the EuroCC-1 and EuroCC-2 project umbrella to enable industry and Small and Medium Enterprises (SMEs) to leverage HPC resources of EuroHPC[9]. Component (C) of Fig. 1 represent adoptions of the UAIF by NCCs and the significant potential to governmental, academic, industry, and SME partners to speed-up and scale-up their applications.

Finally, Digital Twins (DTs) and their workflows using HPC, e.g., in the Destination Earth[10] or Inter-Twin[11] projects, are becoming important for HPC users in Europe. Component (D) has been added to Fig. 1 to represent the processing-intensive applications of DTs that are also highly relevant for CoE RAISE, either the DTs adopting parts of UAIF components or including new use cases in CoE RAISE.

B. Reference Architecture Elements

This section describes the reference architecture components of the UAIF for Exascale HPC/AI methods, which are

[7]CoE RAISE Use Cases https://www.coe-raise.eu/use-cases

[8]EU HPC Centres of Excellence https://www.hpccoe.eu/eu-hpc-centres-of-excellence2/

[9]EuroHPC JU HPC Systems https://eurohpc-ju.europa.eu/about/our-supercomputers_en

[10]Destination Earth https://digital-strategy.ec.europa.eu/en/policies/destination-earth

[11]InterTwin https://www.intertwin.eu

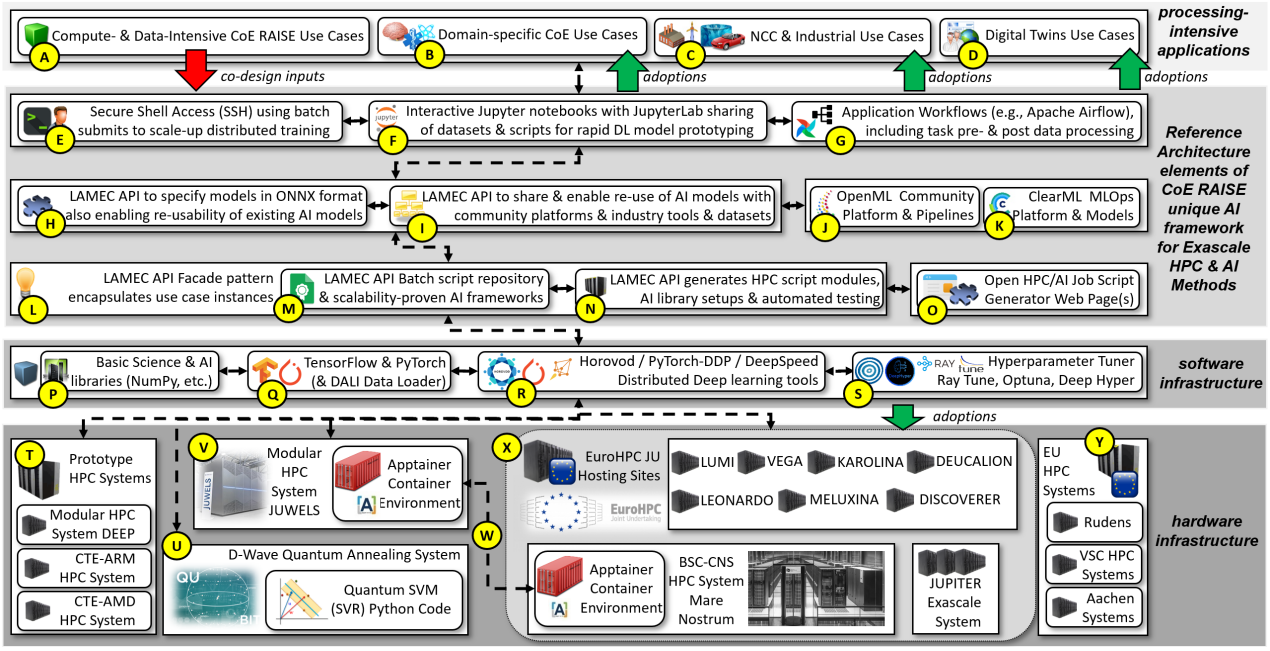

Figure 1. Application co-design result creating a unique AI software framework (UAIF) for HPC towards Exascale.

listed in Fig. 1 in the second layer. As shown in Fig. 1 (E), the first element includes the use of the Secure Shell (SSH) protocol. Principally, as a means to remotely log into HPC systems and submit batch scheduler scripts, e.g., via the Simple Linux Utility for Resource Management (SLURM) (6) tool, it remains one of the integral access methods for HPC applications. Also, AI researchers frequently require interactive access to HPC systems to facilitate quick and rapid prototyping of Machine Learning (ML) and DL models. Component (F) in Fig. 1 addresses this need in the UAIF by offering Jupyter notebooks and JupyterLabs[12]. The component (G) in Fig. 1 is a new addition to the UAIF and supports application workflows and workflow automation, including task pre- and post- data processing capabilities. The UAIF recommends the Apache Airflow[13] tool that is a platform to programmatically create, schedule, and monitor workflows.

As shown in Fig. 1 (H), a key component of the overall UAIF Load AI Modules, Environments, and Containers (LAMEC) API is the fast portability between different DL frameworks and reproducibility achieved by using the standard Open Neural Network Exchange (ONNX). The implementation of the overall UAIF LAMEC API using ONNX is still a work in progress. Another element of the overall UAIF LAMEC API shown in Fig. 1 is component (I), which represents a seamless integration with other tools. The goal is to use the LAMEC API to share and re-use existing AI models with community platforms (see below), industry tools, datasets, and to enable Transfer Learning (TL). While initial discussions with community platforms have taken place, the implementation of the overall UAIF LAMEC API integration and provisioning of AI models is still work in progress.

OpenML[14] is an open community platform for sharing datasets, algorithms, models, and experiments using a wide variety of traditional ML approaches. One ansatz to enlarge the user community of UAIF is to integrate its components into the OpenML platform such that experiments can be also run on cutting-edge HPC systems. Hence, component (J) in Fig. 1 represents how this community might leverage the LAMEC API integration components. On the other hand, ClearML[15] is an ML Operations (MLOps) platform that can be used to develop, orchestrate, and automate ML workflows at scale. CoE RAISE provides one installation of ClearML for its internal and external users. Hence, another approach to enlarge the user community of UAIF is to integrate components into MLOps platforms such as ClearML that are often used in the industry such that its tasks can be also run on cutting-edge HPC systems. Hence, the component (K) in Fig. 1 represents how this community might leverage the LAMEC API components through integration with MLOps platforms.

To map the abstract specifications of software and hardware needs by AI researchers to specific software and hardware HPC infrastructure elements, a facade pattern is used by the UAIF LAMEC API general design. Hence, as represented by component (L) in Fig. 1, the UAIF design employs an abstract wrapper functionality that maps the abstract specifications from users to specific software and hardware configurations. The LAMEC API core is split into the two following elements. The first core element is a batch script repository and the second is an API using this repository to generate new batch script elements, and both are described below. Thus the first core element of the UAIF LAMEC API is a batch script repository. It consists of batch scripts for specific HPC systems

[12]Project Jupyter https://jupyter.org/
[13]Apache Airflow https://airflow.apache.org/
[14]OpenML https://www.openml.org/
[15]ClearML https://clear.ml/

with a correct setup of modules needed for using specific UAIF AI tools As described above and represented by component (M) in Fig. 1, one idea is to use this repository with the UAIF LAMEC API as follows with the second core below. But it quickly becomes clear that the repository in itself is also a great resource for AI/HPC researchers that already know how to deal with changing HPC modules in batch scripts. The second core element of the UAIF LAMEC API, which is represented by component (N) in Fig. 1, is using the above-mentioned repository to generate new batch script segments. This lowers the barrier for entry to leveraging HPC systems for AI researchers that may not have much experience working with modules in HPC environments, as well as saving valuable time through automation for experienced users. Additional components beyond verified site AI modules and libraries, such as AI model scripts or datasets for training and inference, are planned for later addition (although these are usually elements of a job script that inexperienced AI researchers do not find challenging). Maintenance of the job script repository is needed to keep it up-to-date, but that is effort is lower than letting all users find the right modules themselves.

Finally, the open HPC/AI job script generator web page(s) shown as component (O) in Fig. 1 also uses the implementation of the UAIF LAMEC API. This concept is derived from existing job script generators available at the Swiss National Supercomputing Centre (CSCS)[16] or the National Energy Research Scientific Computing Center (NERSC)[17], where the difference to these existing tools lies in the use of UAIF AI toolsets.

C. Software Infrastructure

The software infrastructure layer components (P) – (S), which are depicted in Fig. 1. are presented in this section. Despite the increase of DL tools, and their uptake in the AI communities, there remains a core of basic science libraries heavily used by CoE RAISE communities. Examples are NumPy[18] and scikit-learn[19]. In addition, the building block (P) in Fig. 1 of the UAIF also includes simulation science codes, e.g., those using numerical methods based on known physical laws and that have the potential to benefit from coupling to AI models. Since CoE RAISE focuses on AI models, the various relevant simulation science codes have been kept out of the UAIF layout. As shown in Fig. 1 (Q), the UAIF recommends the use of PyTorch[20] and TensorFlow[21]. CoE RAISE has tested their scalability in depth using various applications. NVIDIA Data Loading Library (DALI)[22] further increases the performance of PyTorch and TensorFlow. This inclusion is represented by component (Q) in Fig. 1 in parenthesis due to the proprietary nature with NVIDIA GPUs. CoE RAISE investigates still other GPU vendors such as Advanced Micro Devices (AMD). Component (R) in Fig. 1 outlines three supported tools used for accelerating distributed AI model training by leveraging the large number of GPUs available at HPC sites today. PyTorch-Distributed Data Parallel (DDP)[23] and Horovod[24] are included in the UAIF software layout. More recently, the component also added DeepSpeed. One of the most successful aspects of the current adoptions of the UAIF are Hyperparameter Optimization (HPO) tools represented by component (S) in Fig. 1. The most adopted component is Ray Tune tool[25], and other UAIF components in that context are Optuna[26] and DeepHyper[27]. This paper describes more about the adoption of RayTune in the later adoption section.

D. Hardware Infrastructure

The hardware infrastructure layer components (T) – (Y) depicted in Fig. 1 are presented in this section. The benchmarking and porting activities of Coe RAISE have been performed on a number of interesting prototype HPC systems that feature new and emerging technologies. The Dynamical Exascale Entry Platform (DEEP)[28] system has been used to experiment with the Modular Supercomputing Architecture (MSA) (7, 8) type of HPC architecture. This component (T) in Fig. 1 also includes two new prototype systems, the Advanced Reduced Instruction Set Computer Machine (ARM)-based CTE-ARM and CTE-AMD, hosted at the Barcelona Supercomputing Centre (BSC) in Spain. The CTE-ARM[29] is a supercomputer based on 192 A64FX ARM processors, with a Linux Operating System (OS) and an Tofu interconnect network (6.8GB/s). CTE-AMD[30] is a cluster based on AMD EPYC processors, with a Linux OS and an Infiniband interconnection network. It includes two AMD MI50 GPUs per node. Quantum Computing (QC) is gaining momentum as the EuroHPC JU recently funded, together with national contributions, several QC systems [31]. Multiple use case applications (9, 10) have successfully engaged in QC by utilizing the D-Wave QA system available via the Juelich UNified Infrastructure for Quantum computing (JUNIQ)[32] at JSC in Germany. As represented by component (U) in Fig. 1, the quantum AI models implemented were Support Vector Machines (SVMs). They were used for regression tasks via Support Vector Regression (SVR).

The MSA-based HPC system JUWELS is massively used for co-designing the UAIF and performing necessary speed-up and scaling benchmarks of its components, see component

[16] CSCS job script generator https://user.cscs.ch/access/running/jobscript_generator/
[17] NERSC job script generator https://my.nersc.gov/script_generator.php
[18] NumPy https://numpy.org/
[19] scikit-learn https://scikit-learn.org/stable/
[20] PyTorch https://pytorch.org/
[21] TensorFlow https://www.tensorflow.org/
[22] DALI https://developer.nvidia.com/dali
[23] PyTorch Distributed Data Parallel https://pytorch.org/tutorials/beginner/dist_overview.html
[24] Horovod https://github.com/horovod/horovod
[25] Ray Tune https://www.ray.io/ray-tunel
[26] Optuna https://optuna.org/l
[27] DeepHyper https://deephyper.readthedocs.io/en/latest/l
[28] DEEP Prototype HPC System hosted by JSC https://www.fz-juelich.de/en/ias/jsc/systems/prototype-systems/deep_system
[29] CTE-ARM HPC System https://www.bsc.es/innovation-and-services/technical-information-cte-arm
[30] CTE-AMD HPC System https://www.bsc.es/innovation-and-services/technical-information-cte-amd
[31] EuroHPC JU Quantum Computers https://eurohpc-ju.europa.eu/selection-six-sites-host-first-european-quantum-computers-2022-10-04_en
[32] JUNIQ https://www.fz-juelich.de/en/ias/jsc/systems/quantum-computing/juniq-facility

(V) in Fig. 1. It is an ideal HPC system for AI workloads as described by Kesselheim et al. in (11). Container technologies are an important tool within larger AI communities to facilitate porting of applications and datasets between systems. One example is shown as component (W) in Fig. 1, where the porting operation of a containerized application from JUWELS at JSC to the MARE NOSTRUM 4 system at BSC is shown. This transparent deployment of containerized code is made possible by the support of Apptainer[33] (previously named Singularity) available at both sites (e.g., see container runtime on JUWELS[34]). Initial test have been performed with containers on HPC platforms at scale, but more application use cases are still work in progress. This component of the UAIF is crucial to support industry applications that have not used HPC before.

Component (X) in Fig. 1 covers the EuroHPC JU hosting sites[35] that may adopt the UAIF. Several European HPC systems contributed to co-design with applications to the UAIF design. It is the goal of the UAIF developer community to support many EuroHPC JU systems. Initial discussions with some of these sites have been started by CoE RAISE partners to encourage the adoption of the UAIF. They reveal however that many of the components of the UAIF are already partly adopted by EuroHPC JU hosting sites. The broader adoption strategy is in its initial stages, while components such as the LAMEC API are considered to be further developed adding more EuroHPC JU systems support over time. One highlight of the adoption will be the integration to the first European Exascale system JUPITER[36], which will be installed in 2024.

It is observed that new users of the UAIF are often starting using regional or university-level systems before scaling up to larger systems. Component (Y) in Fig. 1 contains examples such as the university-level systems Rudens[37] of the Riga Technical University (RTU), or the HPC systems of Rheinisch-Wesfälische Technische Hochschule Aachen - RWTH Aachen University (RWTH)[38]. These sites are in the process of adopting parts of the UAIF framework through users in CoE RAISE. Another example are Belgium regional HPC systems such as the Vlaams Supercomputer Centre (VSC)[39] that are in use by CoE RAISE. There is a wide variety of other HPC systems, such as industrial systems in Iceland (e.g., Responsible Compute[40]) that are not shown in Fig. 1 (Y), but are in progress to adopt elements of the UAIF.

IV. Framework Adoption Example in Healthcare

Critically ill patients who require treatment on an intensive care unit (ICU) in hospitals are at high risk of developing respiratory diseases. Therefore, the Simulation and Data Lab (SDL) Health and Medicine[41] of the Icelandic National Competence Center (NCC) for HPC and AI performs research in that area with a special emphasis on leveraging cutting-edge HPC systems. One of the main research areas of the SDL, in cooperation with the Smart Medical Information Technology for Healthcare (SMITH)[42] project, is Acute Respiratory Distress Syndrome (ARDS), a condition that was first described by Ashbaugh et al. (12) and which has a high mortality rate among affected ICU patients due to its heterogeneity (13).

This section describes short insights on the evolution of using HPC for that research based on earlier work by C. Barakat in (4) with an emphasis on adopting the UAIF for HPO (i.e., component S in Fig. 1). More recently, the SDL Health and Medicine developed and improved a deep learning-based surrogate model of one tool for modeling ARDS onset in a virtual patient called the Nottingham Physiology Simulator (NPS) (14). The model development process takes advantage of current ML and data analysis techniques, as well as efficient HPO methods using the UAIF deployment of the DEEP MSA system at JSC (i.e., component T in Fig. 1). Also, this healthcare use case adopts the basic science libraries like NumPy (i.e., component P in Fig. 1) and DL tools with distributed training (i.e., component Q and R in Fig. 1).

As shown in Fig. 1 (S), the concrete HPO tool used is Ray Tune (15), which in turn employs different scheduling algorithms in order to simplify the task of finding the optimal hyperparameters for training the final ARDS model. DL model hyperparameters are the variables that affect the way in which a model is built and its training process, and can be altered either through a process of trial and error, or automatically using optimization algorithms. The schedulers used by Ray Tune in the optimization process are HyperBand, Asynchronous HyperBand, Population-Based Training (PBT), and the default First-In, First-Out (FIFO). The whole process of finding hyperparameters and using the aforementioned schedulers and algorithms are computationally expensive and thus HPC is required to perform it. Medical experts in the SDL are not experts in HPC and thus leverage the seamless access of HPO tools on HPC via UAIF components such as Jupyter (i.e., Fig. 1 F) and the LAMEC API (i.e., Fig. 1 N).

In order to achieve results, these algorithms distribute the tuning task over the available HPC resources and may interfere with the process by introducing perturbations, or by shutting down under-performing tasks. The comparison of the different algorithms is enabled to highlight the most efficient in terms of resource use and accuracy of ARDS model results. Figure 2 showcases the performance of the different schedulers. It is clear where the most efficient schedulers, namely HyperBand (Figure 2b) and Asynchronous HyperBand (Figure 2c), begin stopping trials that seem to underperform, thereby reducing unnecessary resource use. On the other hand, PBT (Figure 2d)consumes the most resources, but does so while intro-

[33] Apptainer https://apptainer.org/
[34] JUWELS Container Runtime https://apps.fz-juelich.de/jsc/hps/juwels/container-runtime.html
[35] EuroHPC JU Hosting Sites https://eurohpc-ju.europa.eu/about/our-supercomputers_en
[36] Path to JUPITER https://www.fz-juelich.de/en/ias/jsc/news/news-items/news-flashes/2023/path-to-jupiter
[37] Rudens HPC System https://www.rtu.lv/en/research/science-and-innovation-centre/scientific-equipment-unit/hpc-center
[38] RWTH Aachen University HPC Systems https://help.itc.rwth-aachen.de/en/service/rhr4fjjutttf/
[39] VSC HPC Systems https://www.vscentrum.be/
[40] 52Responsible Compute https://responsiblecompute.com/
[41] https://ihpc.is/simulation-and-data-lab-health-and-medicine/
[42] https://www.smith.care/en/

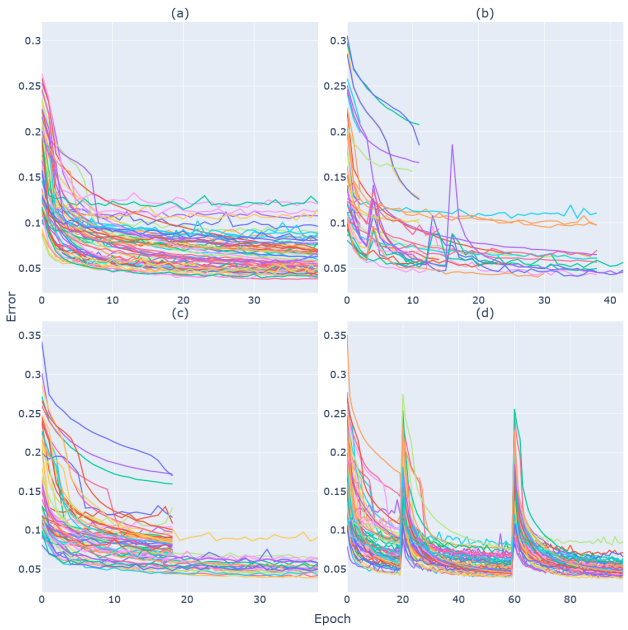

Figure 2. Mean Absolute Error of the ARDS model hyperparameter tuning process using different schedulers.

ducing perturbations to the tuning process, through which more insight into the best parameter combination could be achieved. The tuner is therefore adaptable to different system architectures, where researchers can choose the approach that best suits the available hardware. The UAIF and HPO tools are used to tune parameters like the learning rate, the batch size, the dropout rate, the loss function, and the presence of an intermediate fully-connected layer before the output layer in the network architecture. While concrete medical results are out of scope of this paper, we conclude that the results have been not only faster achieved with UAIF and HPC speed-ups, but also enabled better AI model accuracies through HPO.

V. Conclusion

The implementation process of the UAIF goes forward as planned with respect to scalability tests and benchmarking of UAIF components, but also with the implementation and design of its LAMEC API to make it easier for non-technical users to adopt it. We conclude that the co-design use cases of CoE RAISE for the UAIF helped to be able to adopt the UAIF in other scientific or engineering domains such as healthcare. For the mentioned healthcare use case application of UAIF in ARDS research, the UAIF components enabled the researchers to create better AI models faster than before. While there are many benefits, one drawback of the framework is its maintenance and keeping the repository of job scripts up-to-date which requires some development efforts. Initial discussions with partners in CoE RAISE and NCCs reveal that many want to contribute to an open-source solution with development efforts that keeps the UAIF alive even beyond project lifetime. There is work to be done on the broader deployment of the UAIF across EuroHPC JU hosting sites and other EU HPC resources. Also the adoption of the UAIF in EuroCC NCC industrial use cases, other CoEs, and selected Digitial Twins is work in progress.

References

[1] Book, M. et al, "Facilitating collaboration in high-performance computing projects with an interaction room," in Proceedings of the 4th ACM SIGPLAN International Workshop on Software Engineering for Parallel Systems, 2017, pp. 46–47.

[2] Book, M. et al, "Facilitating collaboration in machine learning and high-performance computing projects with an interaction room," in 2022 IEEE 18th International Conference on e-Science (e-Science). IEEE, 2022, pp. 529–538.

[3] Riedel, M. et al, "Practice and experience using high performance computing and quantum computing to speed-up data science methods in scientific applications," in 2022 45th Jubilee International Convention on Information, Communication and Electronic Technology (MIPRO). IEEE, 2022, pp. 281–286.

[4] Barakat, C. et al, "Lessons learned on using high-performance computing and data science methods towards understanding the acute respiratory distress syndrome (ards)," in 2022 45th Jubilee International Convention on Information, Communication and Electronic Technology (MIPRO). IEEE, 2022, pp. 368–373.

[5] Streit, A. et al, "Unicore 6—recent and future advancements," Annals of Telecommunications-annales des Télécommunications, vol. 65, pp. 757–762, 2010.

[6] Yoo, A.B., Jette, M.A. and Grondona, M., "Slurm: Simple linux utility for resource management," in Job Scheduling Strategies for Parallel Processing: 9th International Workshop, JSSPP 2003, Seattle, WA, USA, June 24, 2003. Revised Paper 9. Springer, 2003, pp. 44–60.

[7] Eicker, N. et al, "The deep project an alternative approach to heterogeneous cluster-computing in the many-core era," Concurrency and computation: Practice and Experience, vol. 28, no. 8, pp. 2394–2411, 2016.

[8] Suarez, E., Eicker, N. and Lippert, T., "Modular supercomputing architecture: from idea to production," in Contemporary high performance computing, pp. 223–255. CRC Press, 2019.

[9] Riedel, M., Cavallaro, G. and Benediktsson, J.A., "Practice and experience in using parallel and scalable machine learning in remote sensing from hpc over cloud to quantum computing," in 2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS. IEEE, 2021, pp. 1571–1574.

[10] Pasetto, E. et al, "Quantum svr for chlorophyll concentration estimation in water with remote sensing," IEEE Geoscience and Remote Sensing Letters, vol. 19, pp. 1–5, 2022.

[11] Kesselheim, S., Herten, A. et al, "Juwels booster–a supercomputer for large-scale ai research," in High Performance Computing: ISC High Performance Digital 2021 International Workshops, Frankfurt am Main, Germany, June 24–July 2, 2021, Revised Selected Papers 36. Springer, 2021, pp. 453–468.

[12] Ashbaugh, D. et al, "Acute respiratory distress in adults," The Lancet, vol. 290, no. 7511, pp. 319–323, 1967.

[13] Barakat, C. et al, "An HPC-Driven Data Science Platform to Speed-up Time Series Data Analysis of Patients with the Acute Respiratory Distress Syndrome," in 2021 44th International Convention on Information, Communication and Electronic Technology (MIPRO), Opatija, Croatia, pp. 311–316, IEEE.

[14] Hardman, J., Bedforth, N. et al, "A physiology simulator: validation of its respiratory components and its ability to predict the patient's response to changes in mechanical ventilation.," British journal of anaesthesia, vol. 81, no. 3, pp. 327–332, 1998.

[15] Moritz, P. et al, "Ray: A distributed framework for emerging {AI} applications," in 13th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 18), 2018, pp. 561–577.