Unique AI Framework (UAIF)

-

CoE RAISE & Call for Collaboration

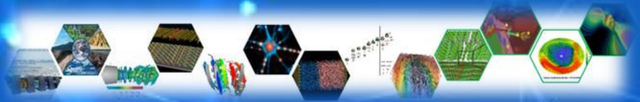
Prof. Dr. – Ing. Morris Riedel et al.

School of Engineering & Natural Sciences, University of Iceland, Iceland

National Competence Center (NCC) for HPC & AI in Iceland – IHPC

2023-07-26, CASTIEL2 Webinar Series "Code of The Month" Vol 2 Event, Online

IHPC National Competence Center (NCC) for HPC & AI in Iceland



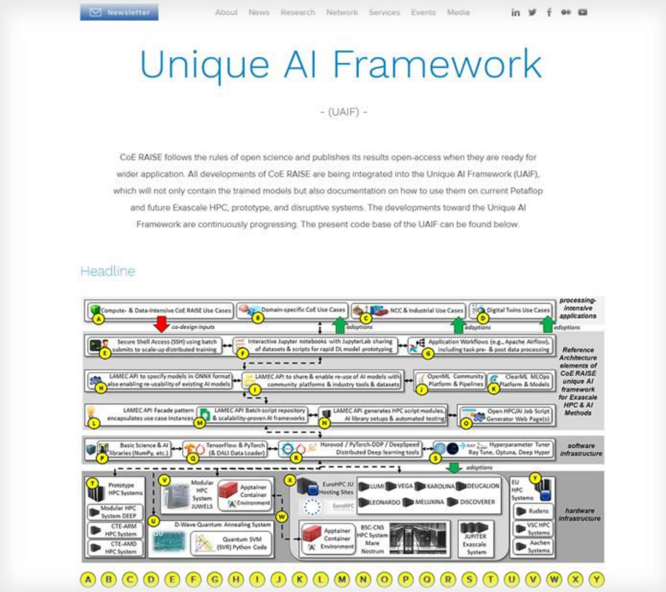
@ProfDrMorrisRiedel **@Morris Riedel** **@MorrisRiedel** **@MorrisRiedel**

https://www.youtube.com/channel/UCWC4VKHmL4NZgFfKoHtANKg **morris@hi.is**

Welcome & Agenda

- CoE RAISE & AI Focus
  - (Talk given by Prof. Dr. Morris Riedel, UoIceland)
  - Motivation: Selected Challenges for AI/HPC Users
  - Unique AI Framework (UAIF) Solutions Overview
  - Collaboration Initiative with NCCs & EuroHPC Hosting Sites
- Load AI Modules, Environments & Containers (LAMEC)
  - (Talk given by Dr. Xin Liu, Juelich Supercomputing Centre)
  - Example of Solutions for Simplifying AI/HPC Access
- AI4HPC Library
  - (Talk given by Dr. Eray Inanc, Juelich Supercomputing Centre)
  - Example of Solutions for Training AI Models in CFD
- Discussions – Q&A

→ https://www.coe-raise.eu



→ https://www.coe-raise.eu/uaif

Motivation: Selected Challenges for AI/HPC Users

- Questions on "AI at scale" using HPC
  - Many distributed training tools for deep learning are available – what scales best?
  - Scaling up means larger batch sizes – what are the limits?
  - Not only faster training of models – but also better models – how?
  - Simple access like Google Colab – how exactly?
  - Examples of Batch Job Scripts – Where?
- Increasing complexity of using HPC systems
  - Module names for AI tools vary heavily on different systems & have many dependencies
  - Different types of hardware need different libraries for AI tools (e.g., Nvidia vs AMD GPUs)
  - Broader availability of EuroHPC JU Hosting Sites & need for porting applications w.r.t. computing time

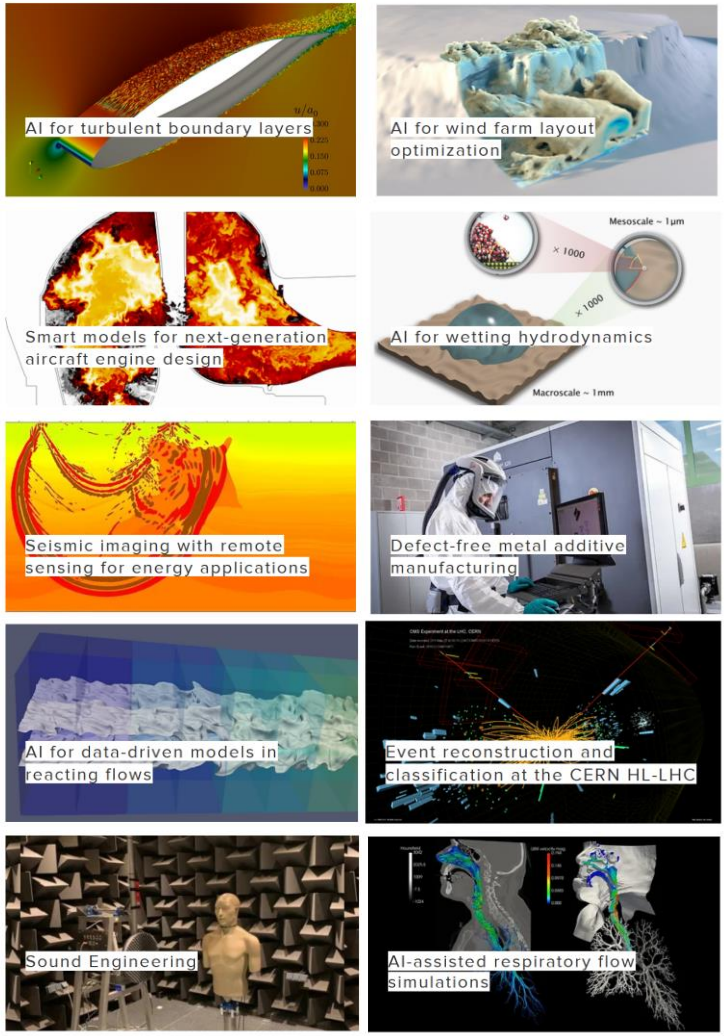
HEAT Helmholtz Analytics Toolkit, PyTorch, TensorFlow, DeepSpeed, DeepHyper, HOROVOD, RAY, tune

One Common "Frustration":
AI/HPC researchers spend approximately 2-3 days per month setting up the right environment and sending working & outdated job scripts around in emails

```
#!/usr/bin/env bash

# Slurm job configuration
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=4
#SBATCH --cpus-per-gpu=20
#SBATCH --account=hai_so2sat
#SBATCH --output=output.out
#SBATCH --error=error.er
#SBATCH --time=6:00:00
#SBATCH --job-name=BENTF2
#SBATCH --gres=gpu:1 --partition=booster

#load modules
ml Stages/2020  GCC/9.3.0  OpenMPI/4.1.0rc1
ml Horovod/0.20.3-Python-3.8.5
ml TensorFlow/2.3.1-Python-3.8.5
#activate my virtualenv
#source /p/project/joaiml/remote_sensing/rocco_sedona/ben_TF2/scripts/env_tf2_juwels_booster/bin/activate

#export relevant env variables
#export CUDA_VISIBLE_DEVICES="0,1,2,3"

#run Python program
srun --cpu-bind=none python -u train_hvd_keras_aug.py
```

Compute & Data-Driven Use Cases of HPC/AI Methods

RAISE Center of Excellence

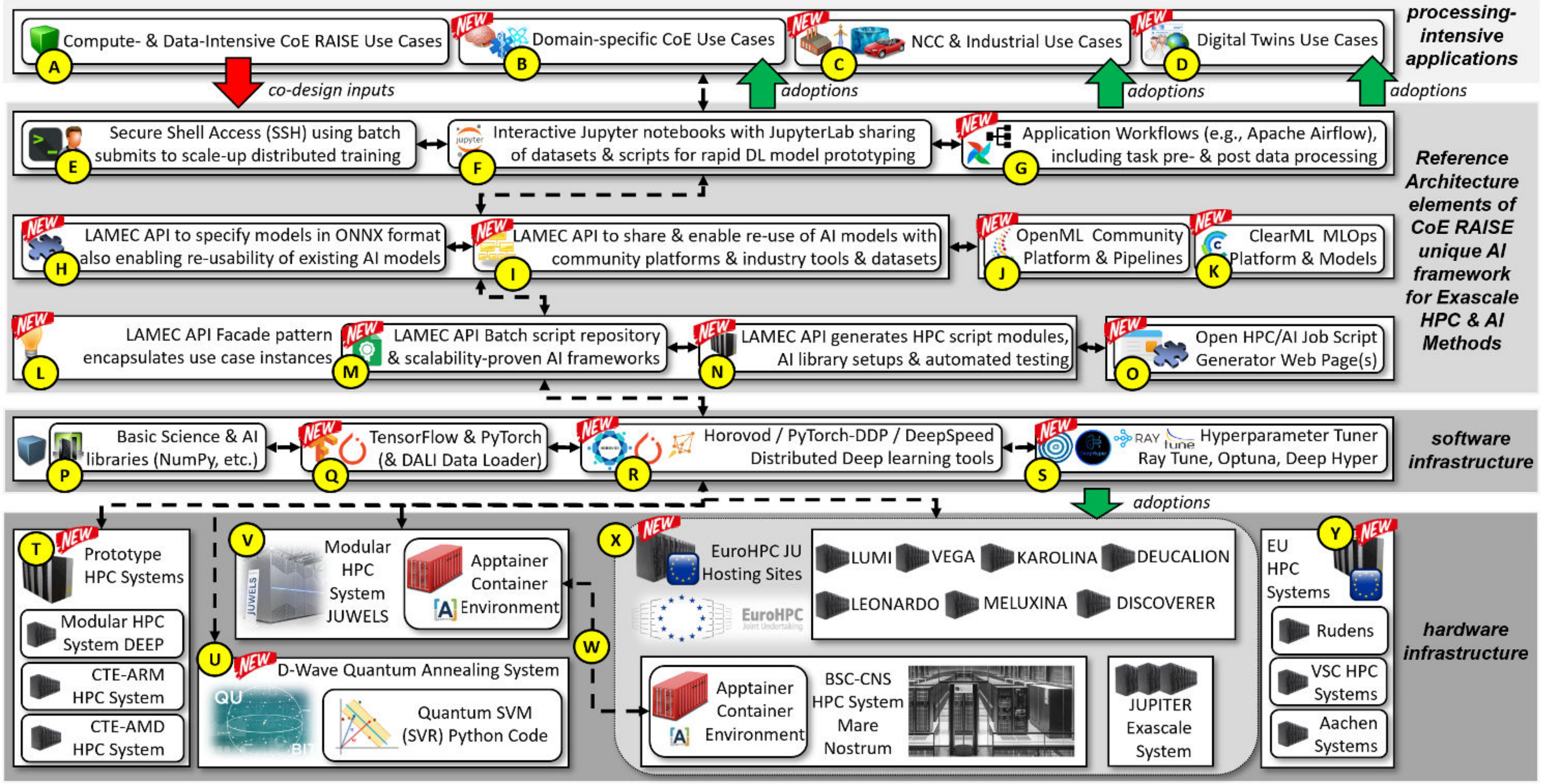


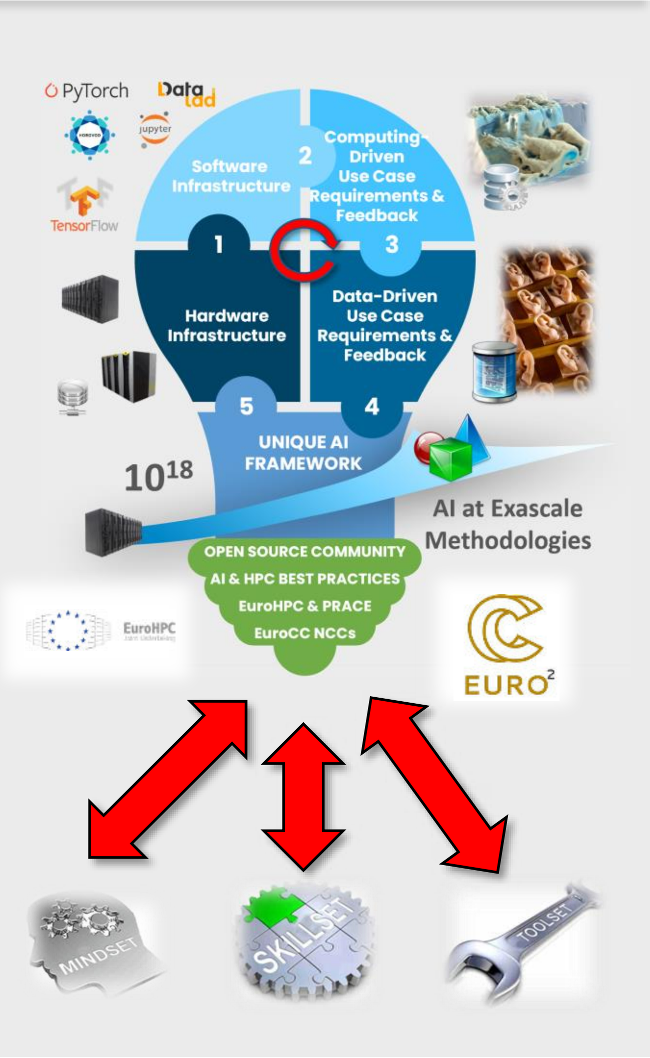
CoE RAISE works on a wide variety of novel HPC/AI Methods including cutting-edge deep learning algorithms.

NEW

Use Case	Task	AE		PINN	ANNs		CNN			NO	GNN		RNN		GAN		TF				QC SVM	RF	CP
Details	*#*	*CAE*	*VQ-VAE*	*PINN*	*ANN*	*RBF-ANN*	*U-Net*	*RES NET*	*CNN*	*FNO*	*MLPF*	*GAT*	*LSTM*	*GRU*	*WGAN*	*CGAN*	*MViT*	*ViViT*	*Swin*	*TF*			
AI for turbulent boundary layers	3.1	X		X	X										X								
AI for wind farm layout optimization	3.2					X			X											X			
AI for data-driven models in reacting flows	3.3						X					X											X
Smart models for next generation aircraft engine design	3.4						X					X											X
AI for wetting hydrodynamics	3.5	X		X						X			X										
Event reconstruction and classification at the CERN HL-LHC use case	4.1										X										X		
Seismic imaging with remote sensing for energy applications	4.2	X		X				X					X	X		X				X	X	X	
Defect-free metal additive manufacturing	4.3	X	X		X												X	X	X				
Sound Engineering	4.4	X			X																		
NHR4CES Project	ext.				X																		X

Unique AI Framework (UAIF) Solutions – Overview



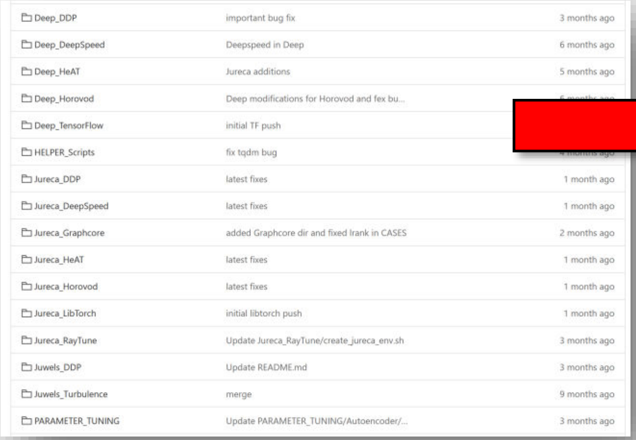


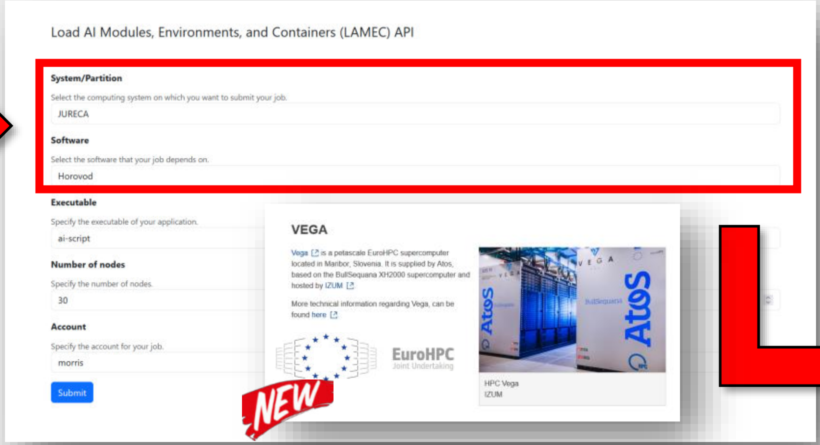
M. Riedel and C. Barakat et al., "Enabling Hyperparameter-Tuning of AI Models for Healthcare using the CoE RAISE Unique AI Framework for HPC," 2023 46th MIPRO ICT and Electronics Convention (MIPRO), Opatija, Croatia, 2023, pp. 435-440, doi: 10.23919/MIPRO57284.2023.10159755.

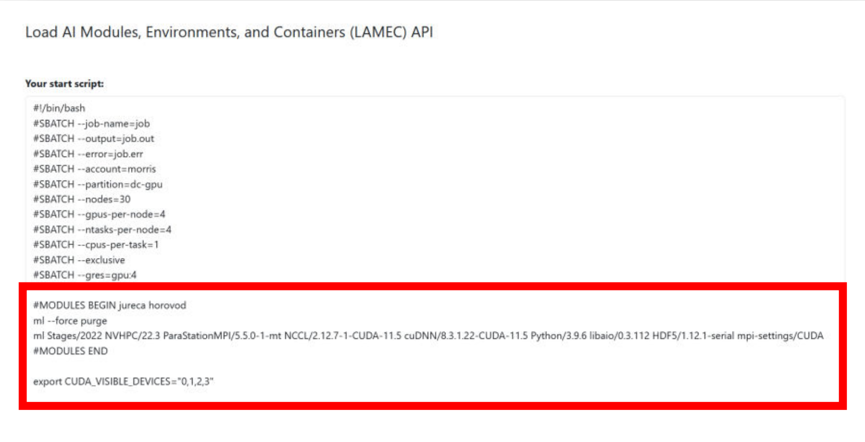
Unique AI Framework (UAIF) – HPC is Usable for AI!

- Addressing the Mindset of AI/HPC Users: Using HPC by Simplifying AI/HPC Access!
  - Load AI Modules, Environments & Containers (LAMEC) API
  - **(More information by talk given by Dr. Xin Liu, Juelich Supercomputing Centre)**
  - E.g., job script generator for the right module setup & Jupyter Notebooks
  - Examples of Batch Job Scripts – Where? Fine-Tuning with own AI/HPC scripts!

CoE RAISE addresses "Frustration":
AI/HPC researchers spend approximately 2-3 days per month setting up the right environment and sending working & outdated job scripts around in emails





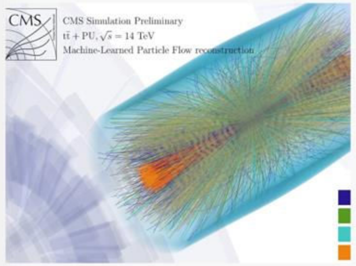


→ https://apps.fz-juelich.de/jsc/lamec-api/

Unique AI Framework (UAIF) – Ready-to-use Tools

RAISE Center of Excellence

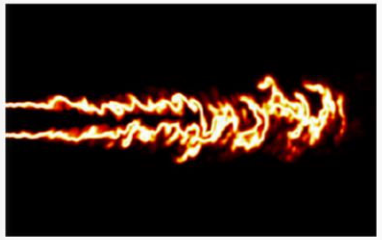
- Enable "AI at scale" using HPC via UAIF Components
  - Addressed: Many distributed training tools for deep learning are available – what scales best?
  - All UAIF Components have been benchmarked & tested for scalability on different hardware
- Selected Ready-to-Use Tools using UAIF Components with innovative AI methods
  - Provides open source-code & data for specific models (e.g., graph neural networks, coupling, etc.)
  - **E.g., AI4HPC for CFD researchers (Talk given by Dr. Eray Inanc, Juelich Supercomputing Centre)**



Machine-Learned Particle-Flow (MLPF)

Machine-Learned Particle-Flow (MLPF) is an algorithm based on Graph Neural Networks (GNN) and is aimed at performing efficient, GPU-accelerated particle flow reconstruction at large particle detector experiments. It takes particle tracks and calorimeter clusters as input and gives higher-level physics objects, for instance electrons, hadrons and photons, as output. This repository contains the code necessary to train MLPF using single or multiple GPUs, to perform large-scale hyperparameter optimization (HPO) using multiple compute nodes on HPC systems, to evaluate the model performance as well as to export the model for later use in inference. The main model and training is implemented in TensorFlow while Ray Tune is used for HPO. A publicly available dataset is available at [1]. MLPF was first introduced in [2] and later versions appeared in [3,4].

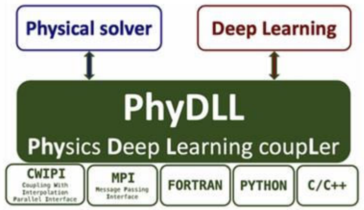
GIT Repository



AI4Sim Model Collection

This repository proposes convolutional (CNN) and graph-based neural network (GNN) architectures as physical surrogates in various industrial use cases. While CNNs can reach state-of-the-art performances on structured grids, GNNs offer natural surrogates for simulations relying on complex unstructured meshes. Several use cases are presented, with comparisons of training pipelines based on CNNs and on GNNs.

GIT Repository



PhyDLL

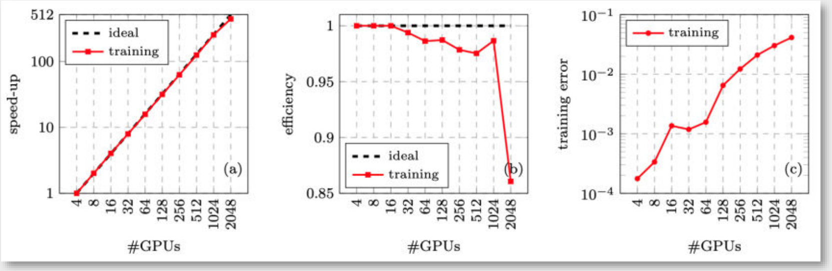
PhyDLL (Physics Deep Learning coupLer) is an open-source coupling library (https://phydll.readthedocs.io). It allows a performant data transfer and processing between massively parallel physical solvers and distributed deep learning inferences. PhyDLL proposes different coupling schemes that suit the context and the data-structure topology. Currently, Fortran and Python interfaces are available for physical solvers and deep learning engines respectively. The ongoing collaborations within CoE RAISE, will enable the creation of a C/C++ interface in order to making PhyDLL even more accessible for a wider range of users. Toward exascale, the development of inter-GPU communications in multi-nodes settings within PhyDLL is under way.

GIT Repository

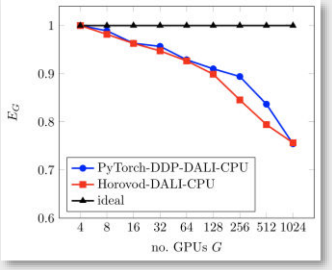
→ https://www.coe-raise.eu/coderepositories-uaif

Unique AI Framework (UAIF) – Better AI/HPC Models

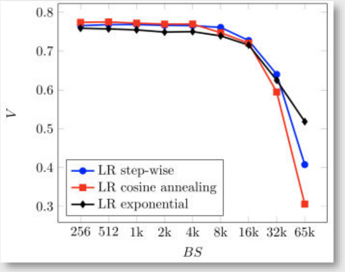
- Addressing another Dimension of Complexity beyond just using AI/HPC Tools
  - Addressing: Scaling up means larger batch sizes – what are the limits?
  - Addressing: Not only faster training of models – but also better models – how?



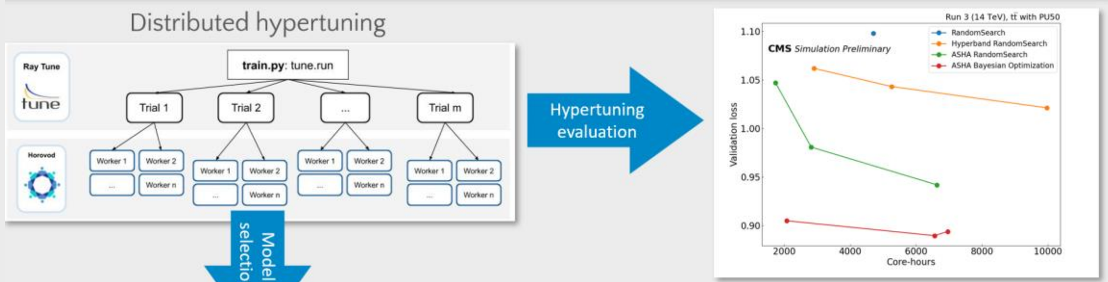
Autoencoder AI model in CFD & Parallel performance using PyTorch-DDP on JUWELS Booster (4 x NVIDIA A100 / node) → scalable - but is it efficient & useful ?



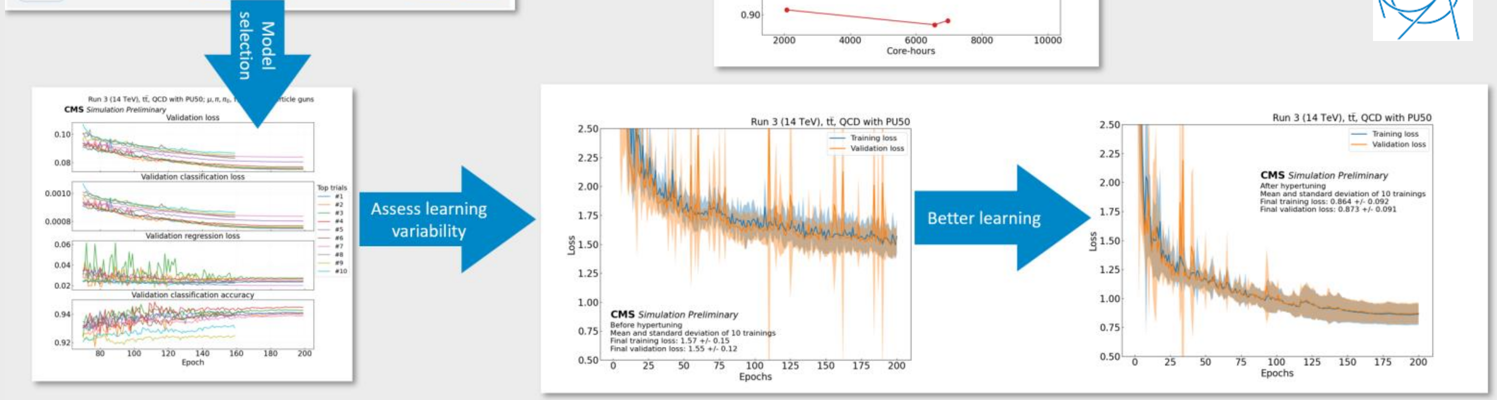
Performance of Horovod & PyTorch-DDP (with DALI dataloader) on up to 1,024 GPUs using ImageNet as scaling benchmark



Validation accuracy over batch size showing impact of learning rate schedulers on ImageNet as benchmark

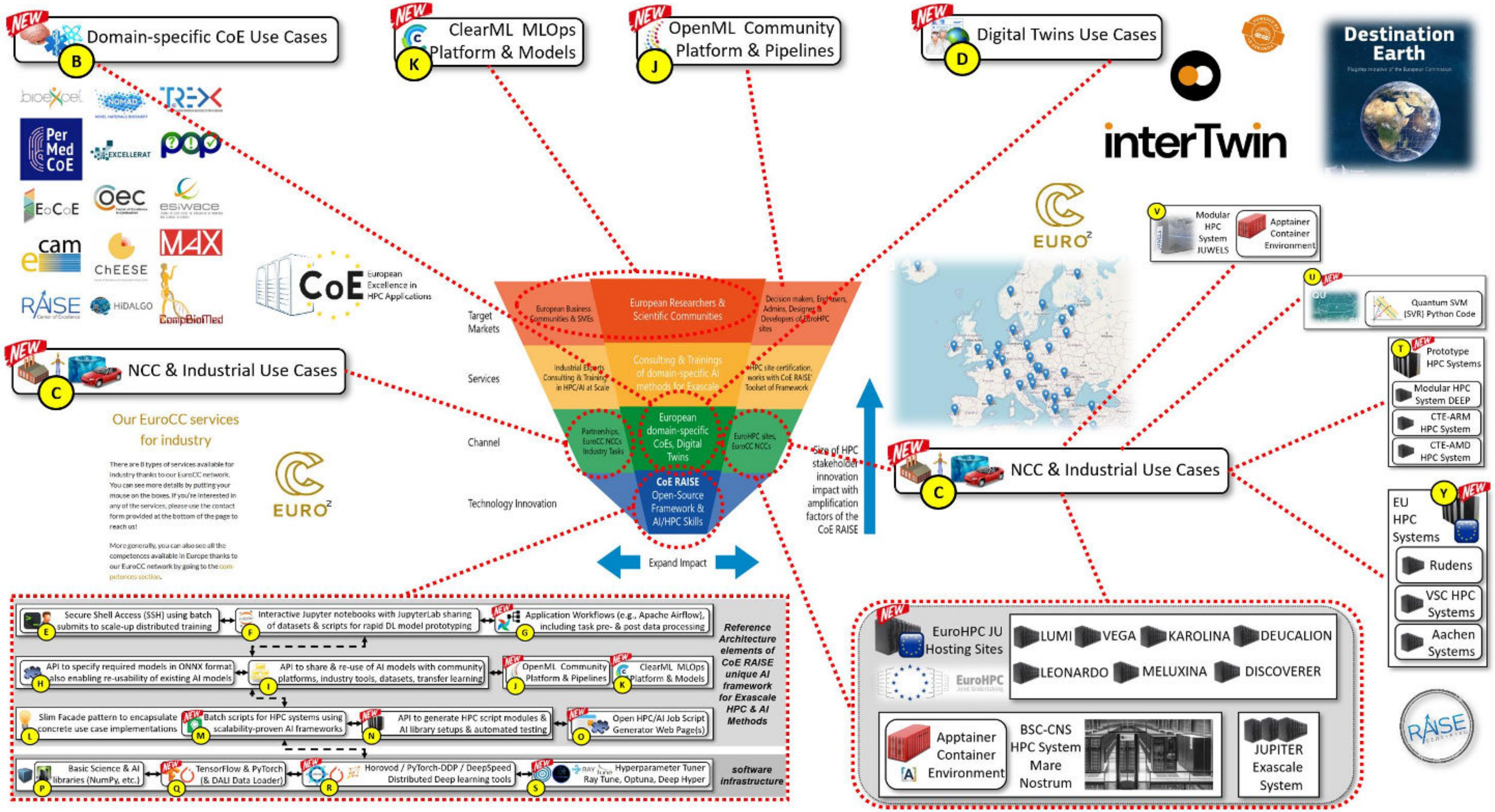


- Mean validation loss decreased by **~44%** giving a significant performance improvement



Two benefits for our NCC users: Using HPC for distributed training of deep learning models in combination with hyperparameter–tuning skills on HPC enables better AI models much faster!

Call for Collaboration – Towards a Community!



- Lessons learned talking already with many NCCs & CoEs
  - Problems of mindset, skillset, and toolset w.r.t. AI/HPC use shared across our EuroHPC JU community
  - E.g., UAIF-LAMEC job script generator useful for all NCCs with industry or "non-tech-savvy" users
  - Join our efforts!

Code of the Month – Unique AI Framework

LAMEC (Load AI Modules, Environments and Containers)

Jóhannes Nordal BSc, Þór Arnar Curtis MSc, Xin Lin PhD

Motivation

- Software modules vary heavily between different HPC systems.
- AI developers spend 2-3 days per months setting up the right environment on HPC systems.
- The goal is to simplify setup of components.

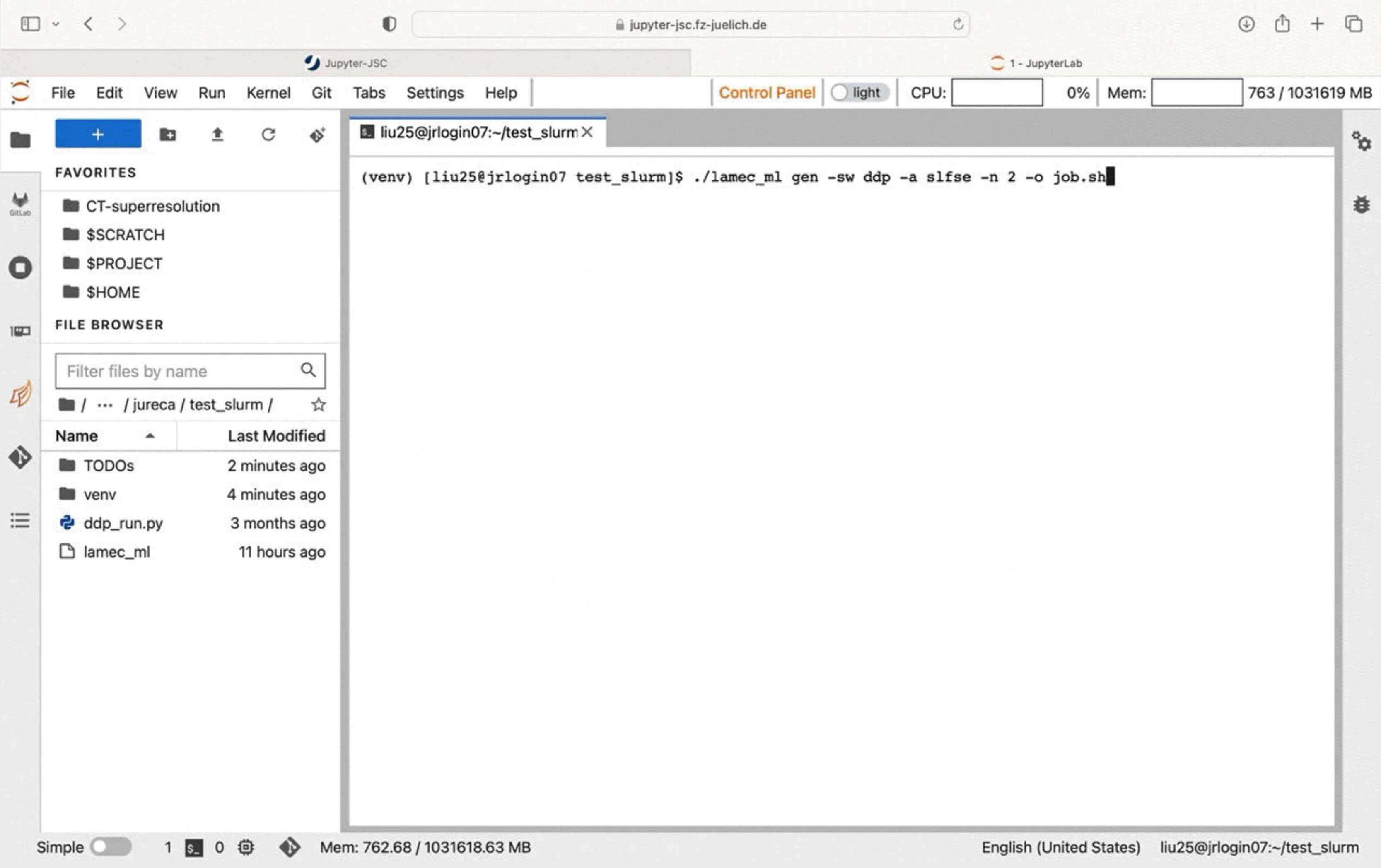
Load AI Modules, Environments and Containers

- LAMEC job script generator automatically selects the right module setup
- Loads correct modules for a given ML framework
- Sets sensible default values for SLURM, based and software and HPC system
- Parses up-to-date job scripts maintained in a git repository
- A command-line and webpage tool

https://gitlab.jsc.fz-juelich.de/CoE-RAISE/FZJ/lamec-oa

https://apps.fz-juelich.de/jsc/lamec-api/

Demonstration



https://gitlab.jsc.fz-juelich.de/CoE-RAISE/FZJ/lamec-oa

https://apps.fz-juelich.de/jsc/lamec-api/

Upcoming

- Adoption plan: LUMI, CTEAMD, Vega, PizDaint, VSC, Rudens
- Extend LAMEC support for containers, for ONNX format and enable re-usability of exsiting AI models
- Automated testing

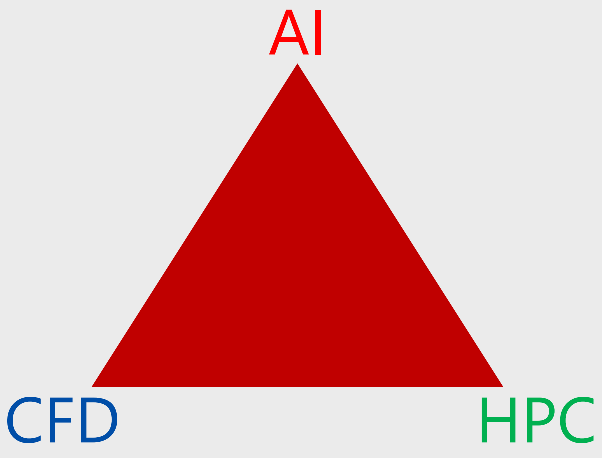
RAISE Center of Excellence

Code of the Month – Unique AI Framework

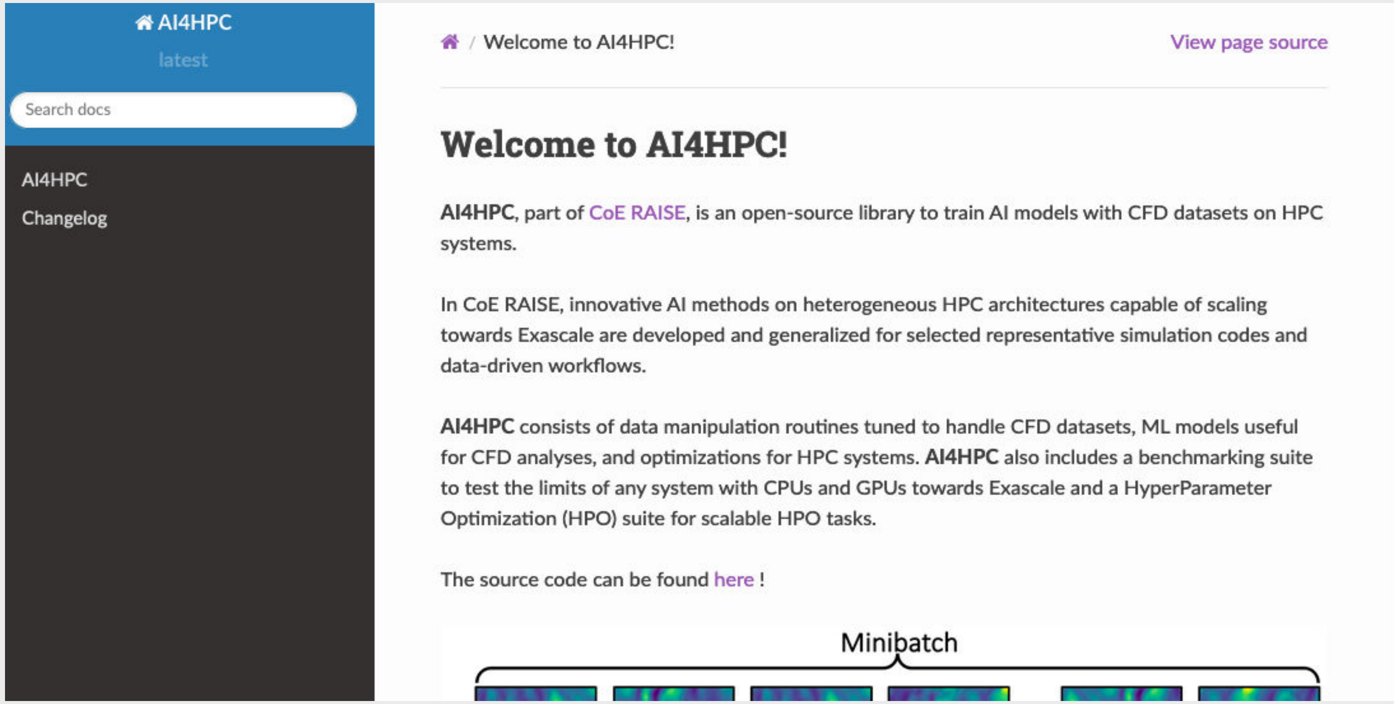
AI4HPC

by Eray Inanc

AI4HPC



Is an open-source library to train AI models with CFD datasets on HPC systems


AI4HPC
latest
Search docs
AI4HPC
Changelog

Welcome to AI4HPC!   View page source

Welcome to AI4HPC!

AI4HPC, part of CoE RAISE, is an open-source library to train AI models with CFD datasets on HPC systems.

In CoE RAISE, innovative AI methods on heterogeneous HPC architectures capable of scaling towards Exascale are developed and generalized for selected representative simulation codes and data-driven workflows.

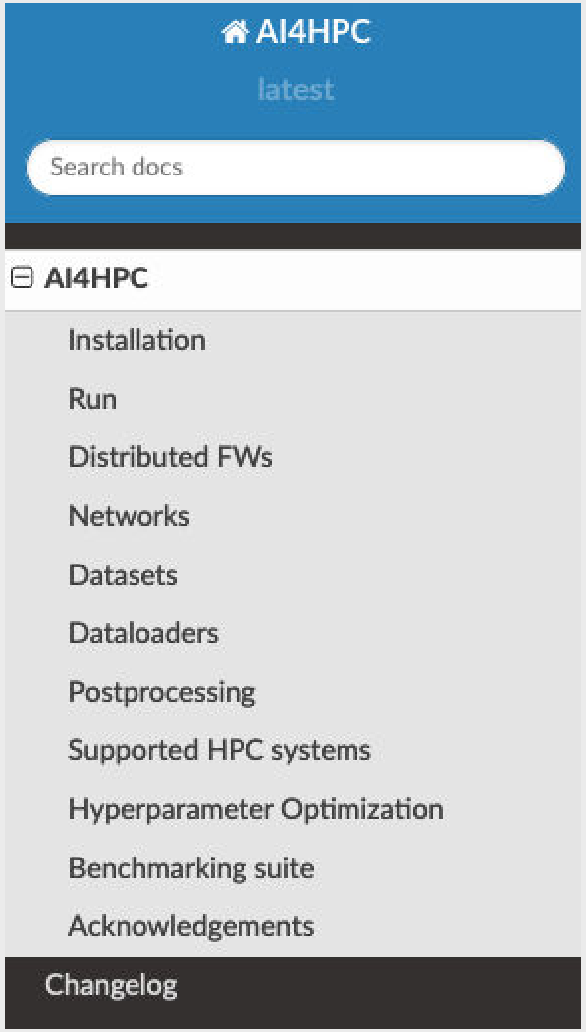
AI4HPC consists of data manipulation routines tuned to handle CFD datasets, ML models useful for CFD analyses, and optimizations for HPC systems. **AI4HPC** also includes a benchmarking suite to test the limits of any system with CPUs and GPUs towards Exascale and a HyperParameter Optimization (HPO) suite for scalable HPO tasks.

The source code can be found here !

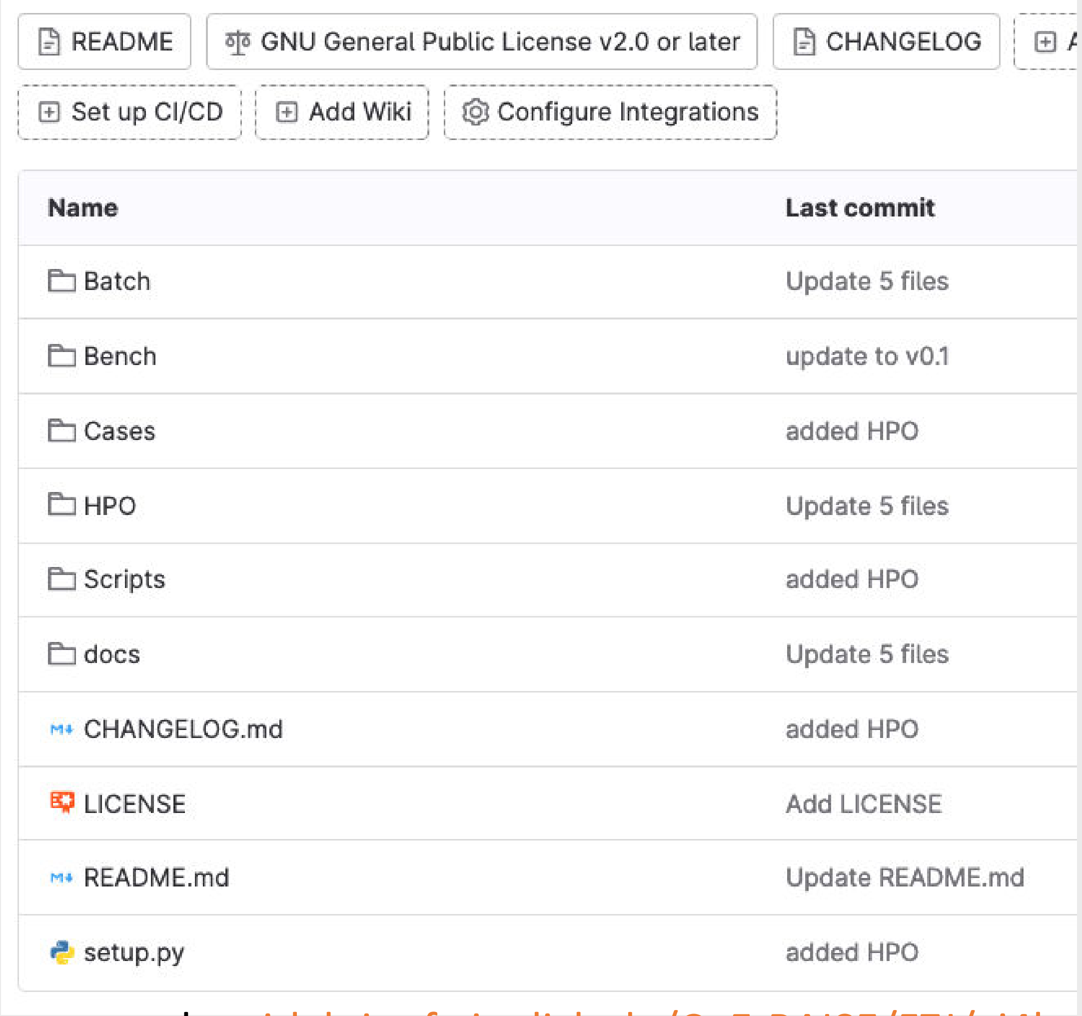
Minibatch

ai4hpc.readthedocs.io

What AI4HPC offers



- Pre-processing routines
- ML models for CFD
- HPC optimizations
- Post-processing routines
- Benchmarking suite
- HPO suite



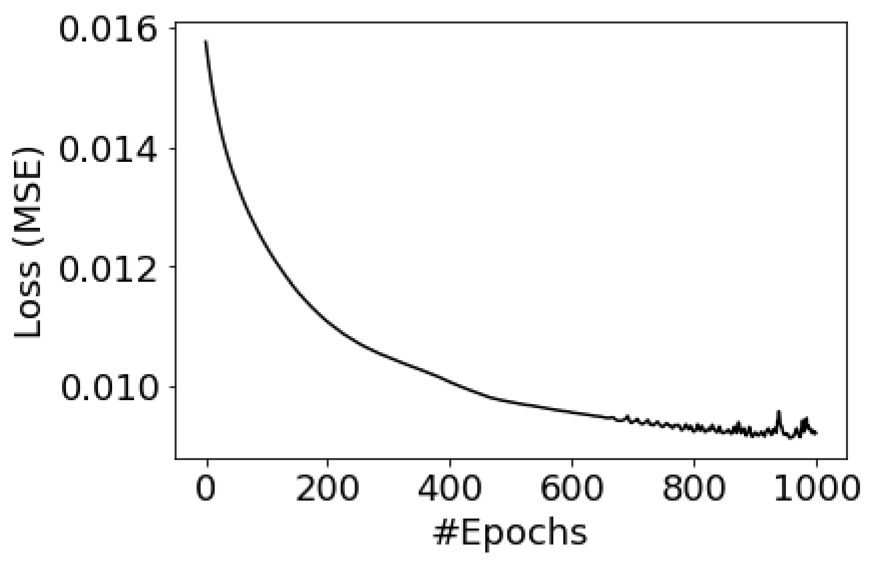
Source code: gitlab.jsc.fz-juelich.de/CoE-RAISE/FZJ/ai4hpc

Why need of HPCs?

RAISE Center of Excellence

Training using 1 GPU*:

- 2 hours per epoch
- Each training would take 1 year…
  - + many runs for development ☹
  - + even more runs for tuning



CAE model developed by Dr. Sarma in FZJ

Solution? Reduce runtimes! Use HPCs!

but how?

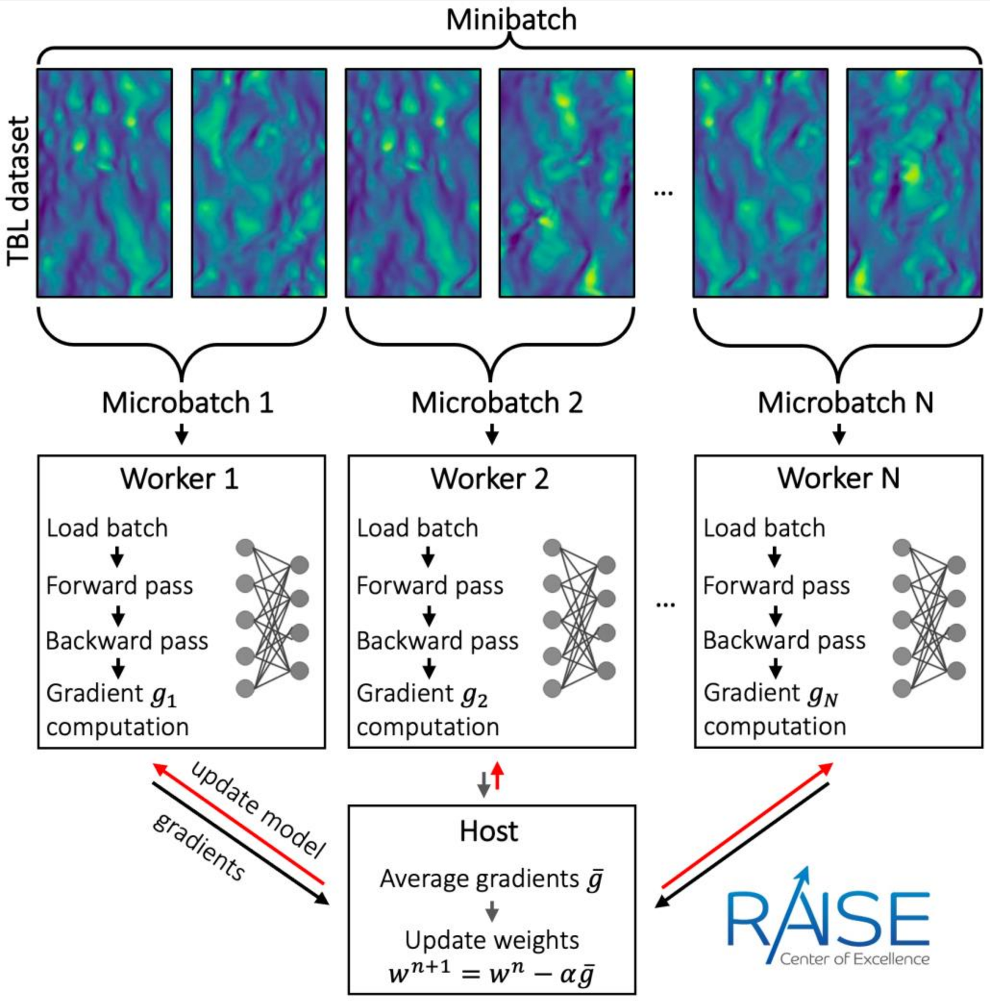
where to start?

*NVIDIA A100 on JURECA DC

Parallelisation

Distributed Trainings - DDP

- mini-batch is split to smaller batches
- Identical NN each GPU
- Server gathers, updates and sends NN params
  - NCCL/RCCL
  - MPI with CUDA/HIP support
  - Gloo (experimental)
- Minibatch = Microbatch * #GPU
  - Model accuracy?
  - Tuning required!



DDT workflow with N GPUs

Backend frameworks

1. Distributed Data Parallel (DDP) - PyTorch
2. Horovod – Uber
3. DeepSpeed – Microsoft
4. HeAT – Helmholtz Analytics Framework

Hardwares tested

RAISE Center of Excellence

- Tier-0/1 HPCs
- Prototypes

Location	System name	CPU	GPU
CINECA	Marconi100	1,960 IBM POWER9	3,920 NVIDIA V100
FZJ	Juwels Cluster + Booster	2,560 Intel Xeon	3,774 NVIDIA A100
FZJ	Jureca DC	1,536 AMD EPYC	768 NVIDIA A100
FZJ	DEEP-EST	147 Intel Xeon	75 NVIDIA V100
FZJ	JUAWEI	11 ARM HiSilicon	
HLRS	Hawk	11,264 AMD EPYC	192 NVIDIA A100 + 64 V100*
CSCS	Piz Daint	9,330 Intel Xeon	68,448 NVIDIA P100
BSC	Marenostrum4	6,912 Intel Xeon	
BSC	CTE-AMD	33 AMD EPYC	66 AMD MI150
BSC	CTE-ARM	192 ARM A64FX	
BSC	HUAWEI	16 ARM Kunpeng 920	
CEA	Joliot-Curie	2,484 Intel Xeon + 2,292 AMD EPYC	
LRZ	SuperMUC-NG	6,480 Intel Xeon	64 NVIDIA V100*
CSC	LUMI	5,632 AMD EPYC	10,240 AMD MI250x

*cloud nodes

50,000 CPUs and 90,000 GPUs!

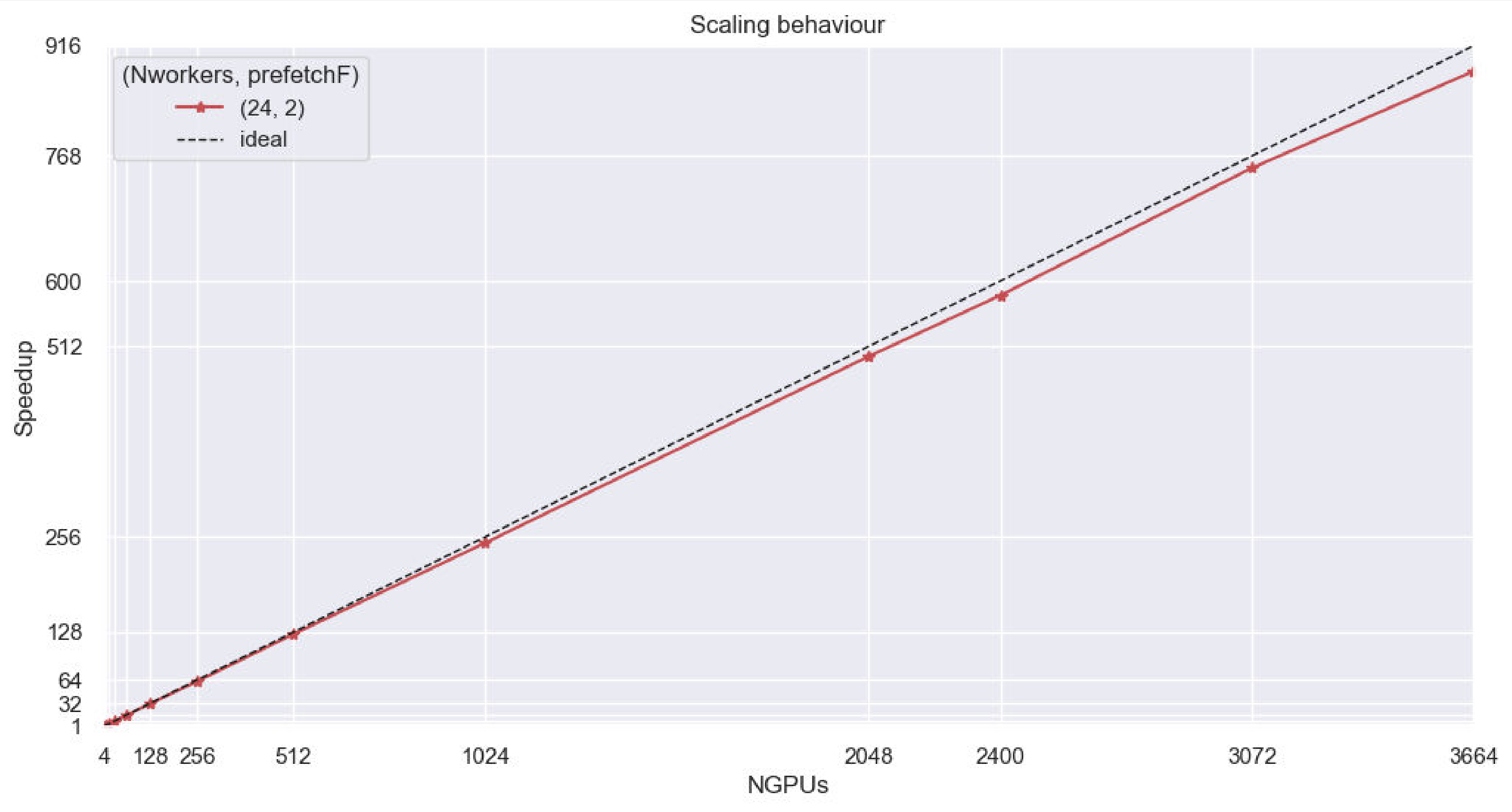
Performance

Super scaling

- Test on JUWELS-BOOSTER
- Up to 3,664 GPUs
- E>0.93

Details:

- DDP
  - PyTorch[2] w/ Horovod[3]
- I/O disabled – synthetic data

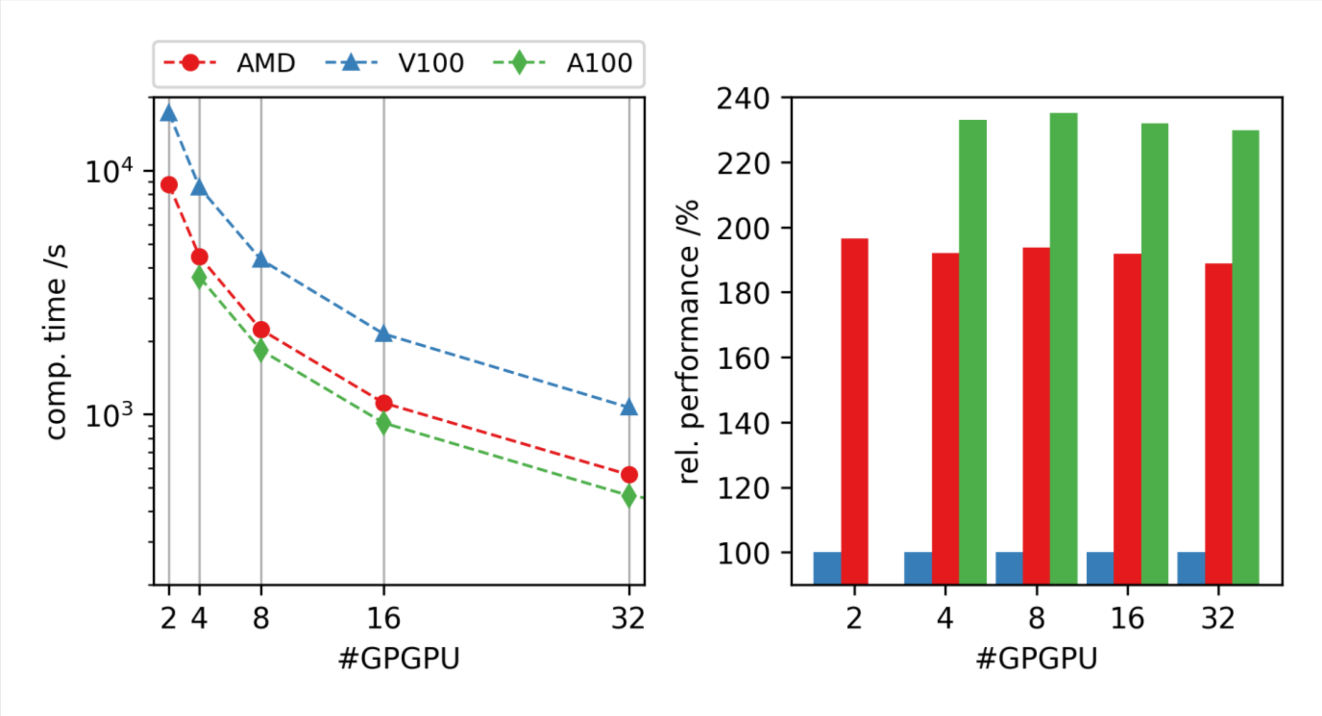


AI4HPC benchmarking suite tested on JUWELS-BOOSTER*

*https://www.fz-juelich.de/en/ias/jsc/systems/supercomputers/juwels

GPU matters

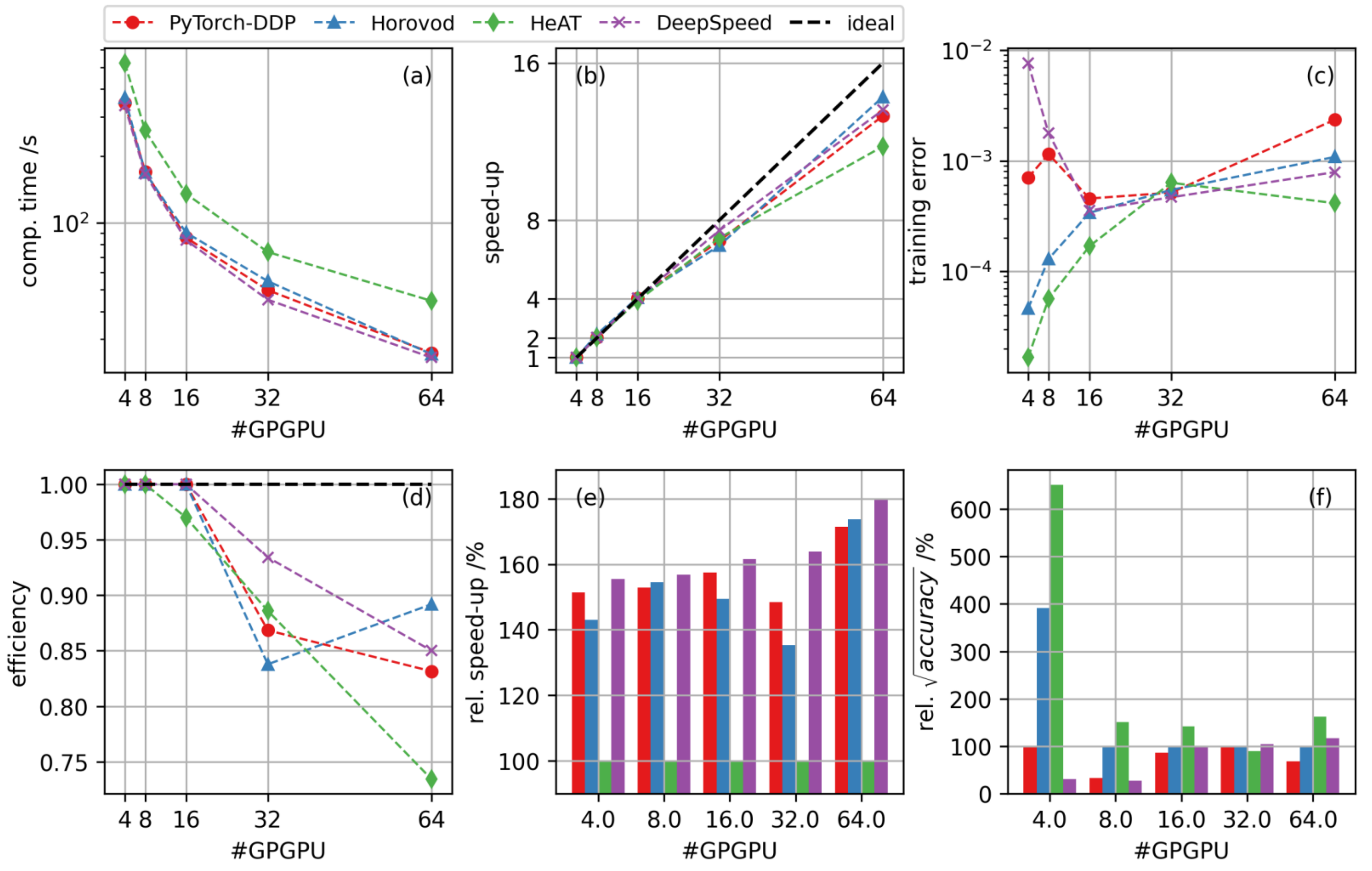
- AMD: prototype CTE-AMD
- NVIDIA V100: prototype DEEP-EST
- NVIDIA A100: JUWELS
- Experimental!
  H100 ~40% faster than A100



CAE (10K params) with TBL-small

Framework comparison

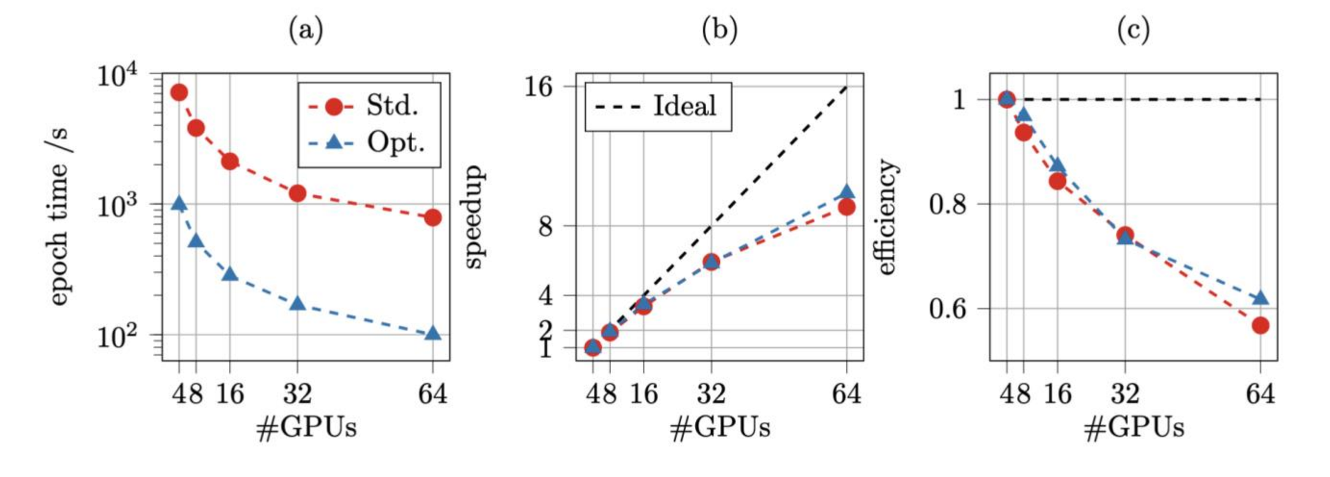
- HPC: JURECA-DC
- Network: CAE (10K params.)
- Dataset: TBL-small (22GB)
- Hyperparameters:
  - Epoch=10
  - Learning Rate=0.01
  - Batch size=96



Implementations / optimizations

- Multi-process dataloader for irregular data shapes
- Reduced precision definitions
- Adaptive summation algorithm
- Deterministic test-runs
- Training error for CFD problems*
- Gradient accumulation
- Nvsight & Torch.profiler

10x speed-up

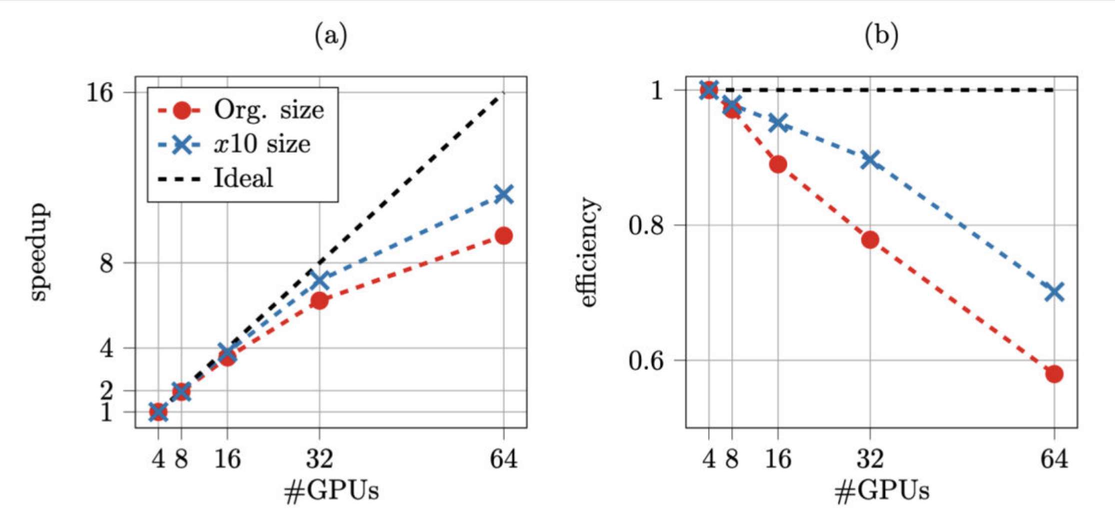


CAE (65K params) with TBL-large (8.3TB)

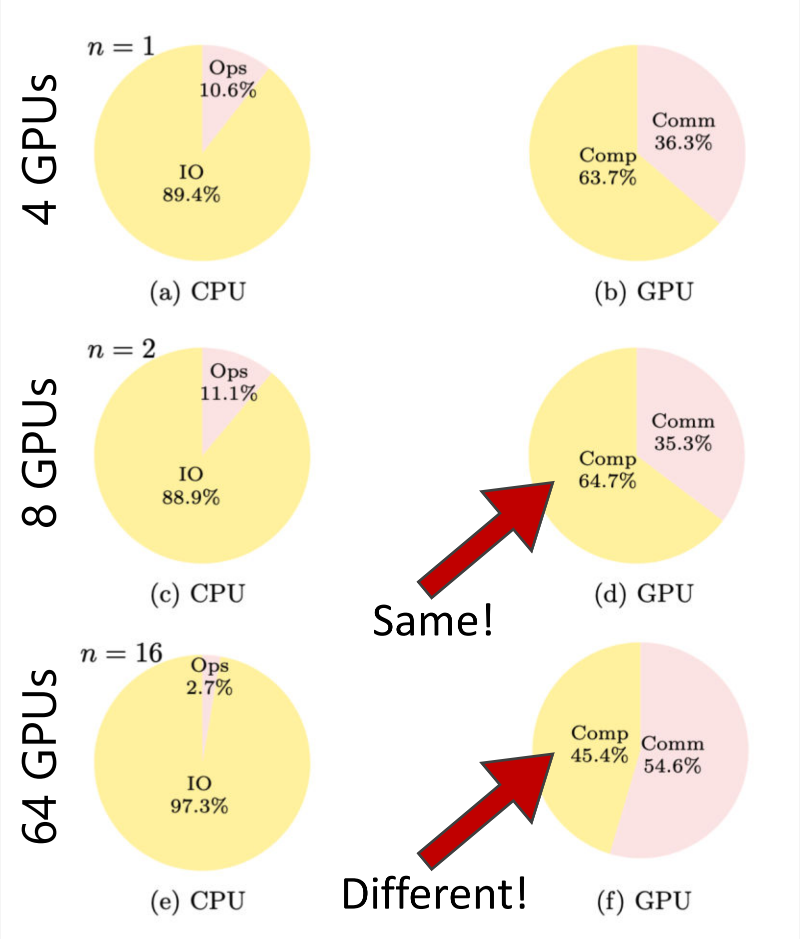
*from the work by Jin et al. https://doi.org/10.1063/1.5024595

Influence of I/O: dataset size

- Data per GPU is limited!
- Too few data -> comm overhead
- Test: try 10x8.3=<u>83TB</u> dataset



CAE (65K params)

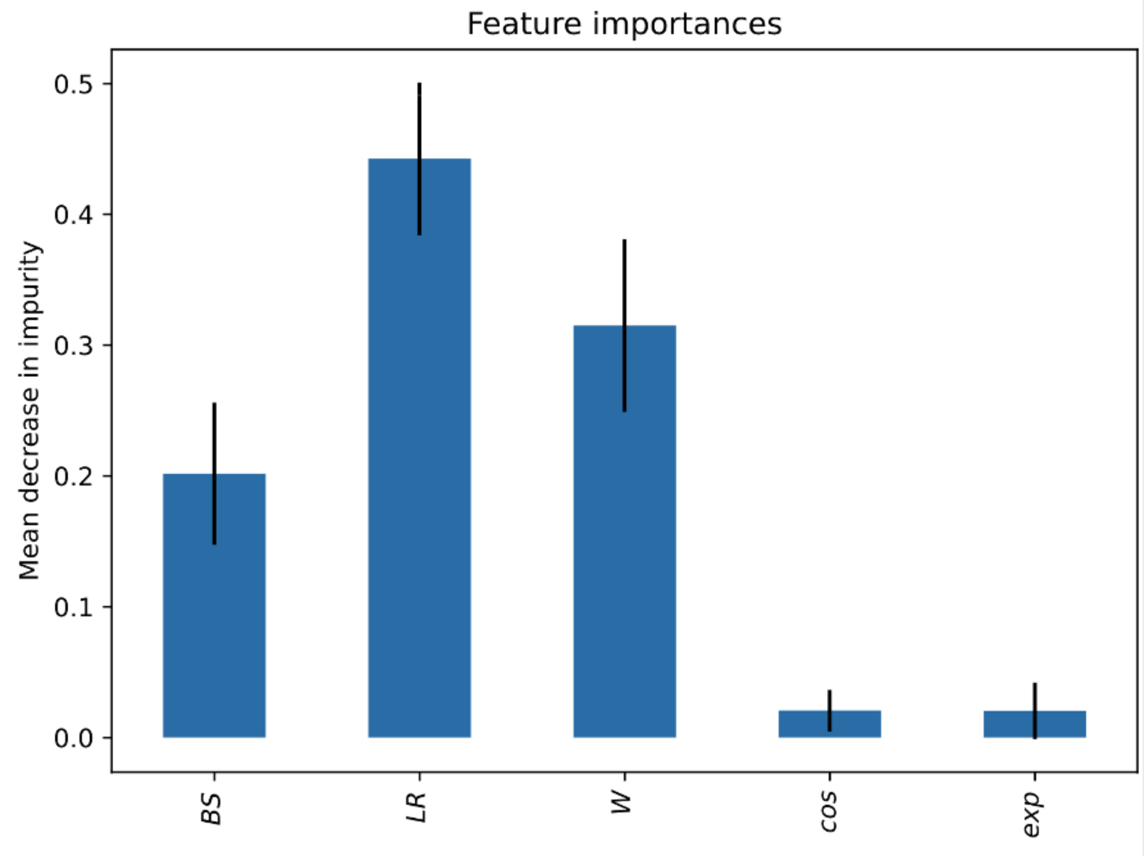


Profiling shares with [1,2,16] nodes or [4,8,64] GPUs

Influence of minibatch size & learning rate

- Minibatch = Microbatch * #GPU
  1024 GPUs -> Minibatch$_{min}$ =! 1024
- LR affects training error!
- 2 solution to fix large Minibatch:

1. Scale learning rate LR*
   - Simple to implement
   - Try-and-error
2. Use adaptive summation (*Adasum*) algorithm**
   - Hard to implement
   - LR independent



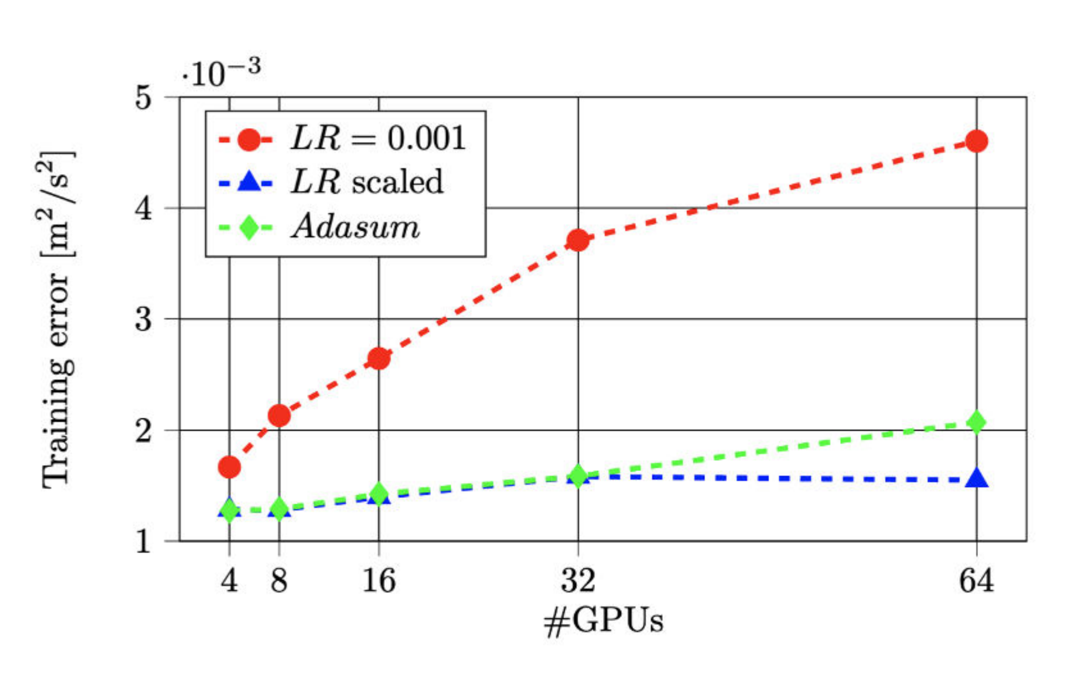
Automated Hyperparameter tuning (thanks to M. Aach)

*Goyal et al. https://arxiv.org/pdf/1706.02677.pdf
**Maleki et al. http://arxiv.org/abs/2006.02924

Influence of minibatch size & learning rate

- Minibatch = Microbatch * #GPU
  1024 GPUs -> Minibatch$_{min}$ =! 1024
- LR affects training error!
- 2 solution to fix large Minibatch:

1. Scale learning rate LR*
   - Simple to implement
   - Try-and-error
2. Use adaptive summation (*Adasum*) algorithm**
   - Hard to implement
   - LR independent

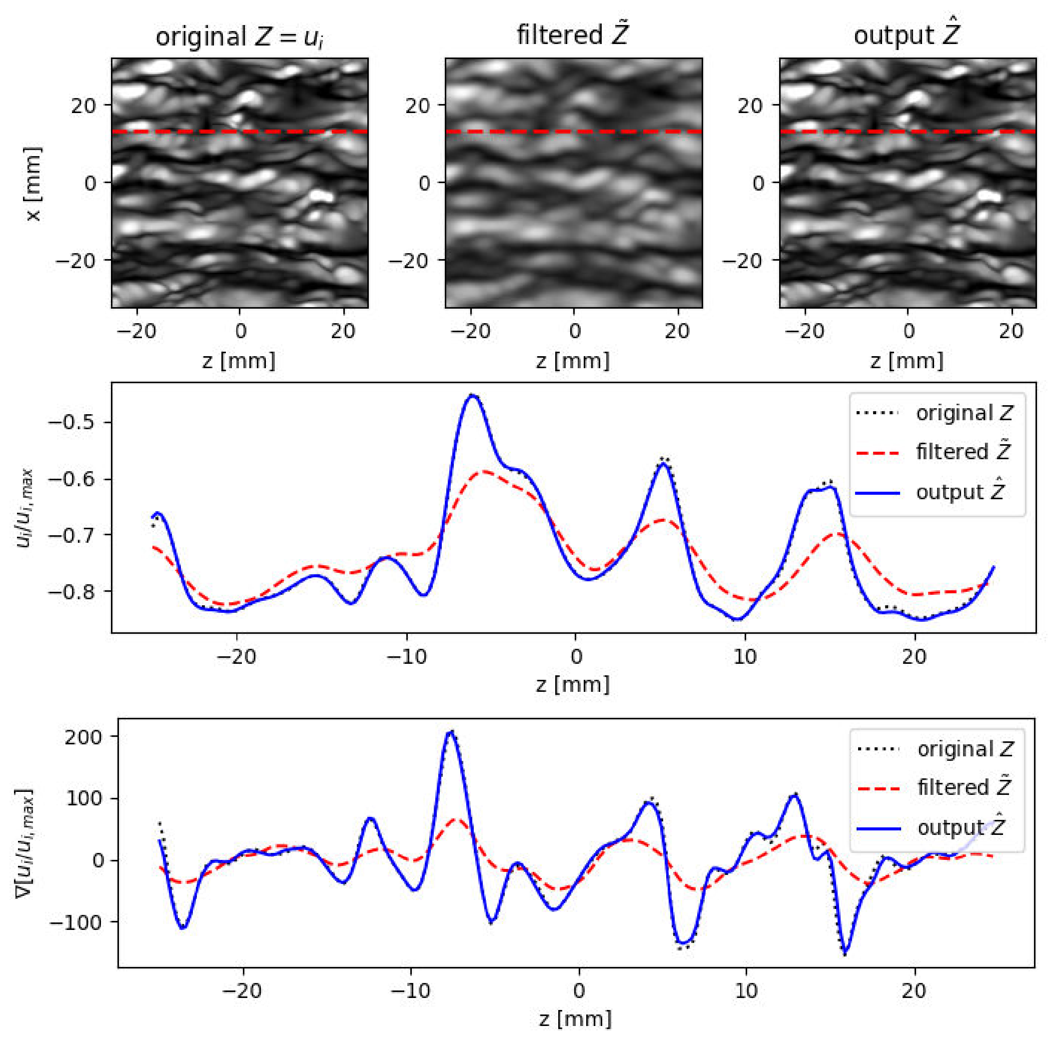


CAE (65K params) with TBL-large (8.3TB)

*Goyal et al. https://arxiv.org/pdf/1706.02677.pdf
**Maleki et al. http://arxiv.org/abs/2006.02924

A result

- Case: TBL
- Motivation: Super-resolution
- Aim: recover 5 times coarse grid
- Model: Convolutional Defiltering (CDM)
- HPC: 32 GPUs on JURECA-DC
- Shown: Streamvice velocity results
  - Black line -> fine grid
  - Red line -> 5x coarse grid
  - Blue line -> super-resolution



Remarks

1. AI4HPC is a first step to combine
   - AI
   - CFD
   - HPC
2. Complete package
   - Pull the repo and start running!
3. Great performance
   - Data size and I/O bottleneck
4. Constant development and user support
   - Not limited to CFD!

Thank you for your attention

drive. enable. innovate.

RAISE Center of Excellence

The CoE RAISE project receives funding from the European Union's Horizon 2020 – Research and Innovation Framework Programme H2020-INFRAEDI-2019-1 under grant agreement no. 951733

Follow us: