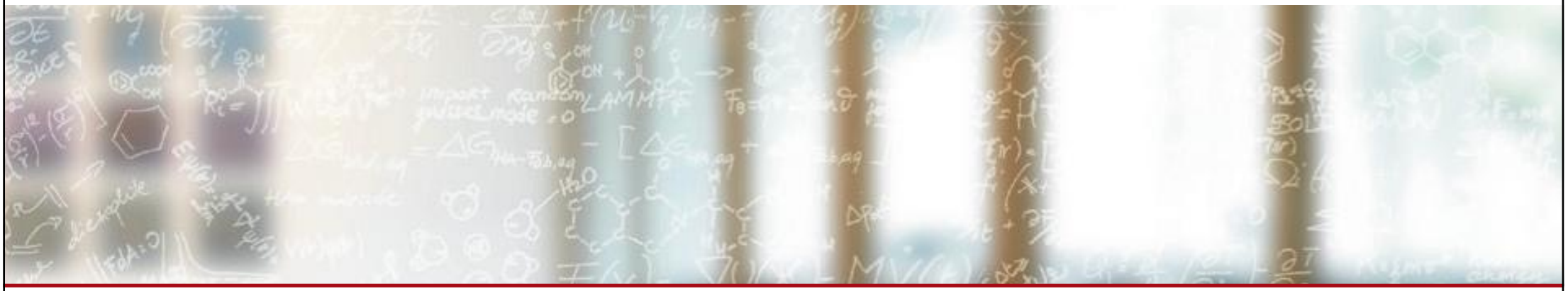
CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

ETH zürich



Evolving HPC services to enable ML workloads on HPE Cray EX

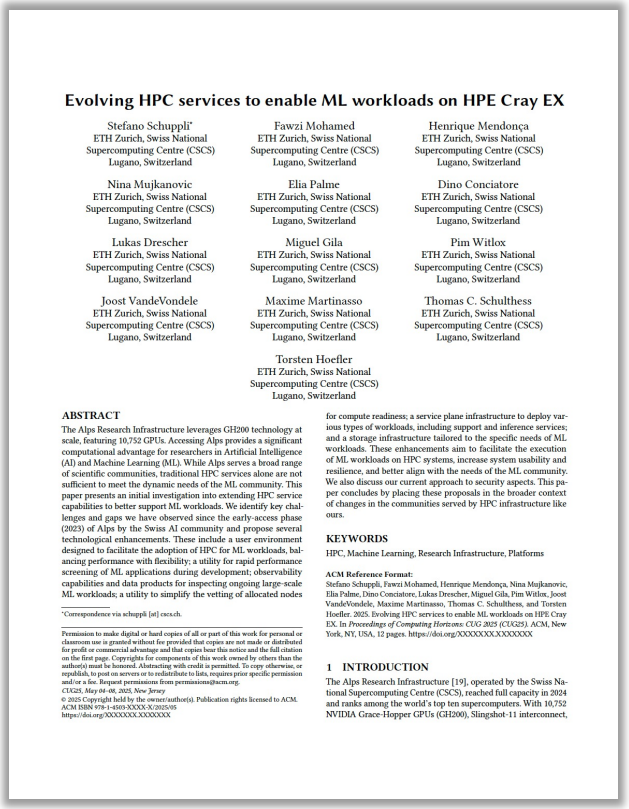
CUG'25, May 7th, 2025

Stefano Schuppli, Fawzi Mohamed, Henrique Mendonça, Nina Mujkanovic, Elia Palme, Dino Conciatore, Lukas Drescher, Miguel Gila, Pim Witlox, Joost VandeVondele, Maxime Martinasso, Thomas C. Schulthess, Torsten Hoefler

CSCS

ETH zürich

Introduction



Evolving HPC services to enable ML workloads on HPE Cray EX

Stefano Schuppli*
ETH Zurich, Swiss National Supercomputing Centre (CSCS)
Lugano, Switzerland

Fawzi Mohamed
ETH Zurich, Swiss National Supercomputing Centre (CSCS)
Lugano, Switzerland

Henrique Mendonça
ETH Zurich, Swiss National Supercomputing Centre (CSCS)
Lugano, Switzerland

Nina Mujkanovic
ETH Zurich, Swiss National Supercomputing Centre (CSCS)
Lugano, Switzerland

Elia Palme
ETH Zurich, Swiss National Supercomputing Centre (CSCS)
Lugano, Switzerland

Dino Conciatore
ETH Zurich, Swiss National Supercomputing Centre (CSCS)
Lugano, Switzerland

Lukas Drescher
ETH Zurich, Swiss National Supercomputing Centre (CSCS)
Lugano, Switzerland

Miguel Gila
ETH Zurich, Swiss National Supercomputing Centre (CSCS)
Lugano, Switzerland

Pim Witlox
ETH Zurich, Swiss National Supercomputing Centre (CSCS)
Lugano, Switzerland

Joost VandeVondele
ETH Zurich, Swiss National Supercomputing Centre (CSCS)
Lugano, Switzerland

Maxime Martinasso
ETH Zurich, Swiss National Supercomputing Centre (CSCS)
Lugano, Switzerland

Thomas C. Schulthess
ETH Zurich, Swiss National Supercomputing Centre (CSCS)
Lugano, Switzerland

Torsten Hoefler
ETH Zurich, Swiss National Supercomputing Centre (CSCS)
Lugano, Switzerland

ABSTRACT

The Alps Research Infrastructure leverages GH200 technology at scale, featuring 10,752 GPUs. Accessing Alps provides a significant computational advantage for researchers in Artificial Intelligence (AI) and Machine Learning (ML). While Alps serves a broad range of scientific communities, traditional HPC services alone are not sufficient to meet the dynamic needs of the ML community. This paper presents an initial investigation into extending HPC service capabilities to better support ML workloads. We identify key challenges and gaps we have observed since the early-access phase (2023) of Alps by the Swiss AI community and propose several technological enhancements. These include a user environment designed to facilitate the adoption of HPC for ML workloads, balancing performance with flexibility; a utility for rapid performance screening of ML applications during development; observability capabilities and data products for inspecting ongoing large-scale ML workloads; a utility to simplify the vetting of allocated nodes for compute readiness; a service plane infrastructure to deploy various types of workloads, including support and inference services; and a storage infrastructure tailored to the specific needs of ML workloads. These enhancements aim to facilitate the execution of ML workloads on HPC systems, increase system usability and resilience, and better align with the needs of the ML community. We also discuss our current approach to security aspects. This paper concludes by placing these proposals in the broader context of changes in the communities served by HPC infrastructure like ours.

*Correspondence via schuppli [at] cscs.ch.

CUG25, May 04–08, 2025, New Jersey

ACM ISBN 978-1-4503-XXXX-X/2025/05
https://doi.org/XXXXXXX.XXXXXXX

KEYWORDS

HPC, Machine Learning, Research Infrastructure, Platforms

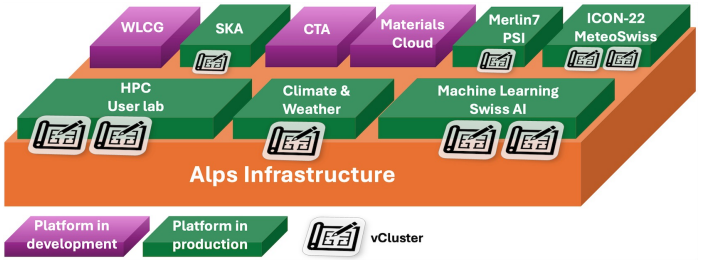
ACM Reference Format:

Stefano Schuppli, Fawzi Mohamed, Henrique Mendonça, Nina Mujkanovic, Elia Palme, Dino Conciatore, Lukas Drescher, Miguel Gila, Pim Witlox, Joost VandeVondele, Maxime Martinasso, Thomas C. Schulthess, and Torsten Hoefler. 2025. Evolving HPC services to enable ML workloads on HPE Cray EX. In *Proceedings of Computing Horizons: CUG 2025 (CUG25).* ACM, New York, NY, USA, 12 pages. https://doi.org/XXXXXXX.XXXXXXX

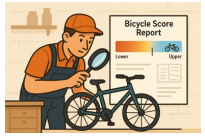
1 INTRODUCTION

The Alps Research Infrastructure [19], operated by the Swiss National Supercomputing Centre (CSCS), reached full capacity in 2024 and ranks among the world's top ten supercomputers. With 10,752 NVIDIA Grace-Hopper GPUs (GH200), Slingshot-11 interconnect,

- Recap of our journey
  - How did we come to this?
- Our work in a nutshell
  - Pragmatic approaches for extending HPC services capabilities to better support ML workloads.



Support for Container-based User Environments

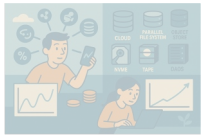


A GPU Saturation Scorer for ML Applications

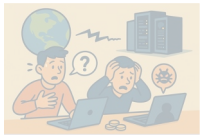
Infrastructure Observability for ML Workloads

A Node Vetting and Early Abort System

A Service Plane for Supporting and Inference Services



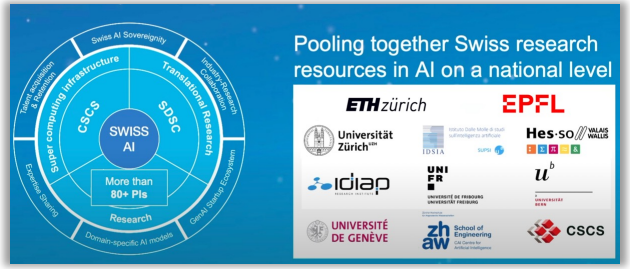
Storage Services for ML Workloads*

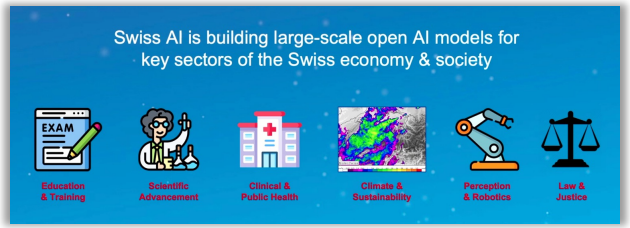


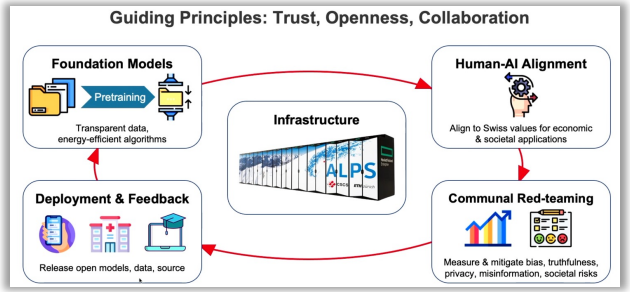
Security Implications of ML Workloads*

** Not covered in today's presentation*

The Swiss AI Initiative

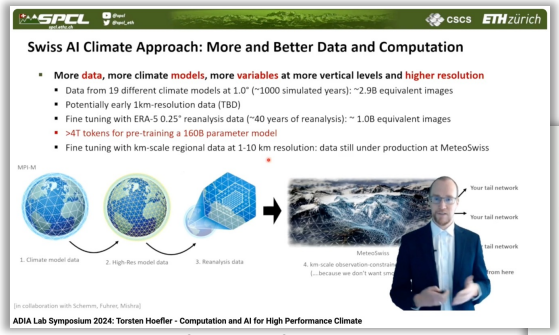




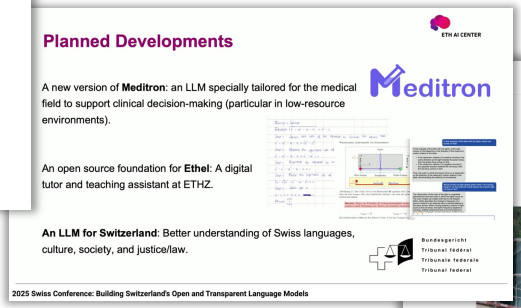


From "*Community access for ML on ALPS: The Swiss AI Initiative*" *by Joost VandeVondele* - https://www.youtube.com/watch?v=xLuuE0HuXAU

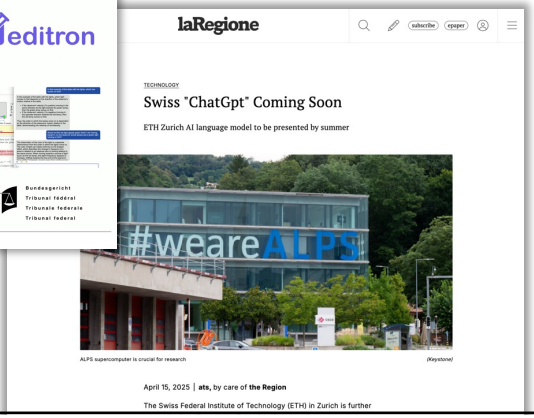
- Our efforts are guided by our tight collaboration with world-class ML experts working across scientific domains.
  - 80+ research groups, representing a wide range of domains.
  - Wide spectrum of model architectures, usage modes, datasets, scales and scaling libraries, and project phases.



"Computation and AI for High Performance Climate" by Torsten Hoefler - https://www.youtube.com/watch?v=QLmqJmj39qs



"Building Switzerland's Open and Transparent Language Models" by Imanol Schlag https://www.youtube.com/watch?v=gjvakQQOCdw



Translated from https://www.laregione.ch/rubriche/tecnologia/1830519/zurigo-politecnico-svizzero-modello-ia

CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

ETH zürich

Technological components

Motivations

a) **HPC knowledge gap**
ML users might lack time or HPC expertise, potentially leading to inefficient usage of resources and disruptive system use.

b) **Diverse and evolving needs**
ML users need to react rapidly to innovations. They require agility and services beyond “usual” training phases.

c) **Repetitive work affecting productivity**
Certain efforts are duplicated across time and across teams.

d) **Reproducibility and portability**
There is a need for moving ML applications across infrastructures.

e) **Storage offering alignment**
ML access patterns might strain traditional storage options, which in some cases might even be over-provisioned compared to needs.

f) **Operational and support resource constraints**
Limited staff and funding hinder robust, extensive and detailed coverage of all community needs and wishes.

g) **Security considerations**
ML introduces risks that amplify existing common security concerns.

Support for Container-based User Environments

- Background
  - ML users need agility and so value familiar interfaces that allow for rapid start of activities and provide greater flexibility
  - The ML ecosystem is vibrant
    - Many components are available *off-the-shelf*
- Our approach
  - Bring your own stack
  - Lower the barrier of entry for users new to HPC environments
  - Support teams' productivity
    - Facilitate container-based ML workloads on HPC systems
  - Prepare for future project needs

Support for Container-based User Environments

```
image = "ubuntu:latest"

mounts = [
    "/scratch/project01/dataset:/scratch/project01/dataset:ro",
    "/scratch/${USER}"
]

workdir = "/scratch/${USER}/project01_code"
writable = true

[annotations.com.hooks.aws_ofi_nccl]
variant = "cuda12"

[annotations.com.hooks.ssh]
enabled = "true"
authorize_ssh_key = "<public key file path>"
```

Listing 1: Example of an Environment Definition File (EDF).

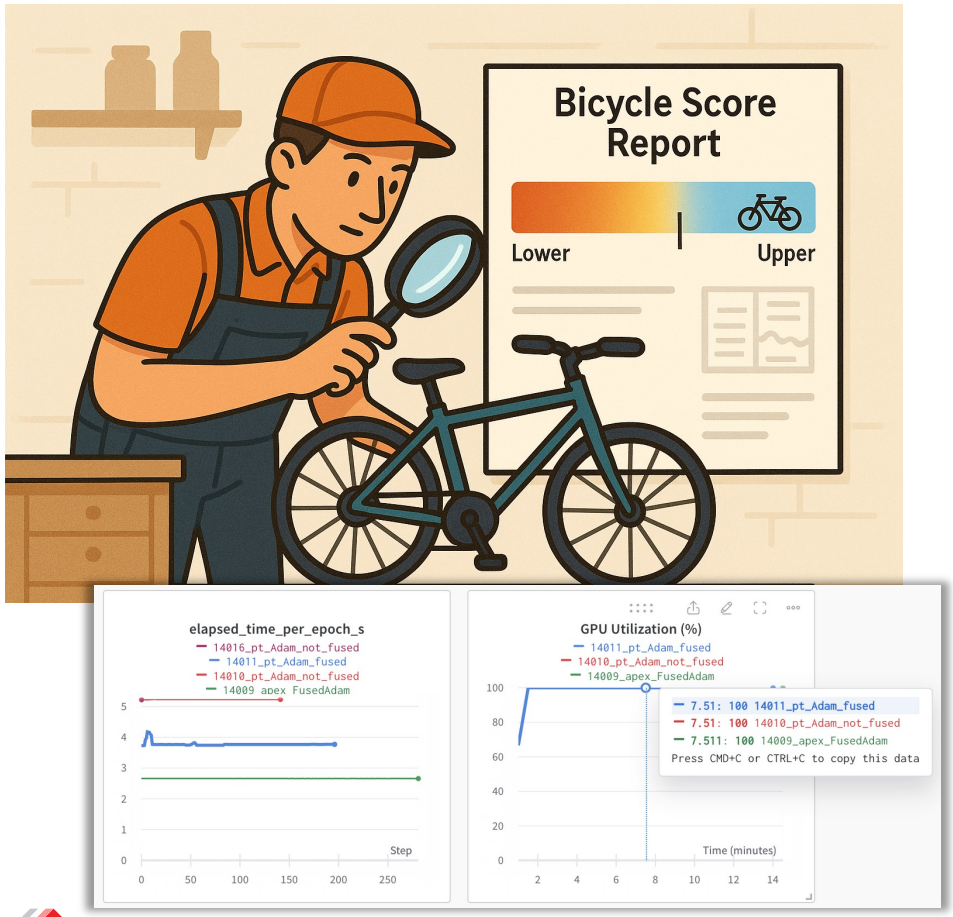
```
#!/bin/bash

#SBATCH --nodes 64
#SBATCH --ntasks-per-node 4

srun --environment=./my_environment.toml python train.py
```

Listing 2: Example of EDF files usage with Slurm.

User documentation and usage examples:
https://confluence.cscs.ch/x/YANYNQ

- Design elements
  - Toolset for transparently using containers to run SLURM jobs
  - Environment Definition File (EDF), TOML-based
  - Catalog of containers customizations
    - Performance related customizations
    - Functional customizations
- Ongoing work
  - Shifting to Podman
  - Integration with vulnerabilities scanning services
  - Image building support
  - Support for start-up steps definition
  - Greater leverage of CDIs

A GPU Saturation Scorer for ML Applications



- Background
  - Utilization ≠ Saturation
  - Users might lack time or the necessary background to understand computational aspects of their applications
  - Profiling is a difficult task
- Our approach
  - We are developing a lightweight utility to aggregate profiling data and lower the bar for users who might not have time or the necessary background.
  - Boosts productivity by simplifying profiling analysis

A GPU Saturation Scorer for ML Applications

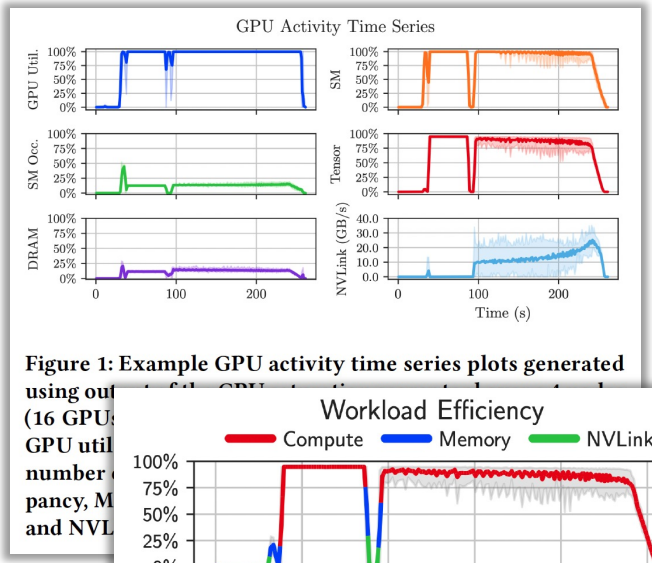


Figure 1: Example GPU activity time series plots generated using out (16 GPU GPU util number pancy, M and NVL

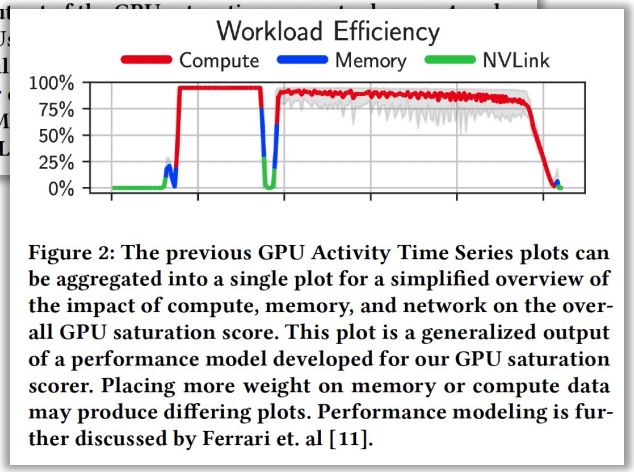


Figure 2: The previous GPU Activity Time Series plots can be aggregated into a single plot for a simplified overview of the impact of compute, memory, and network on the overall GPU saturation score. This plot is a generalized output of a performance model developed for our GPU saturation scorer. Placing more weight on memory or compute data may produce differing plots. Performance modeling is further discussed by Ferrari et. al [11].

- Design elements
  - CLI usage and SLURM integration
  - Avoid the need for the user to instrument the application.
    - Currently using data from services such as NVIDIA DCGM
- Ongoing work
  - Immediate needs are around Grace-Hopper and ML applications, but our interest are more general
  - Improve aggregation logic
  - Making it suitable for the adoption within the project's proposal process used to apply for resources

Infrastructure Observability for ML Workloads

- Background
  - Transient issues and inefficiencies might go unnoticed or left unexplored
  - Debugging is hard, scattered across tools
  - Current solutions might be incomplete for large-scale ML undertakings (e.g., W&B)
- Our approach
  - Provide practical and effective ways to analyze a running ML workloads (job-scoped data) and to identify root causes of common issues.

Infrastructure Observability for ML Workloads

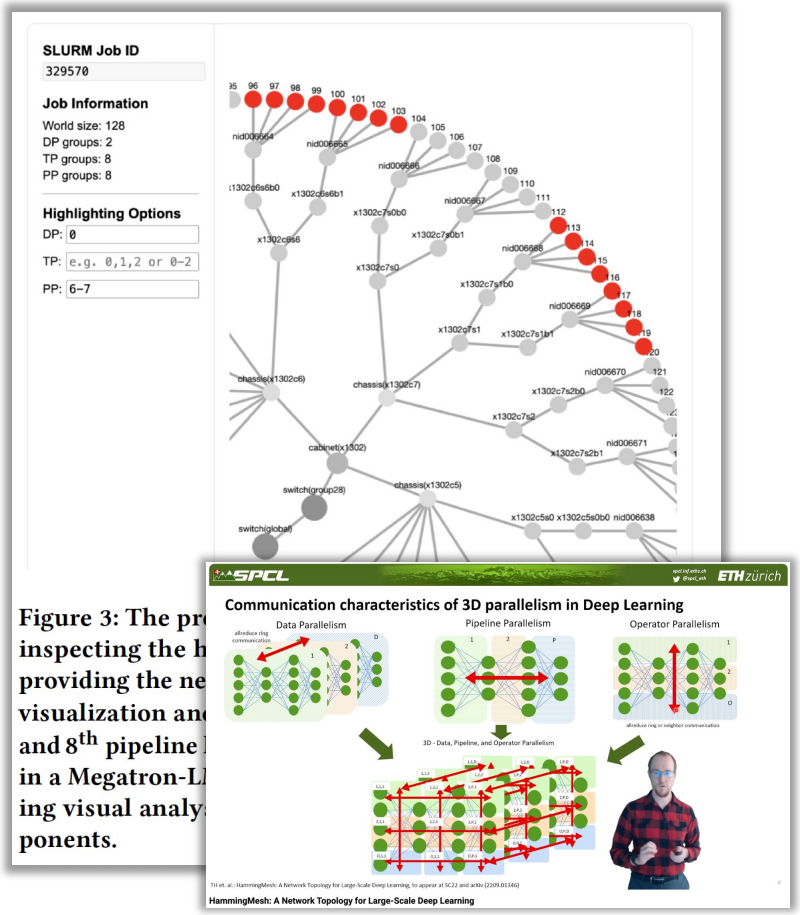


Figure 3: The pr
inspecting the h
providing the ne
visualization an
and 8th pipeline
in a Megatron-L
ing visual analy
ponents.

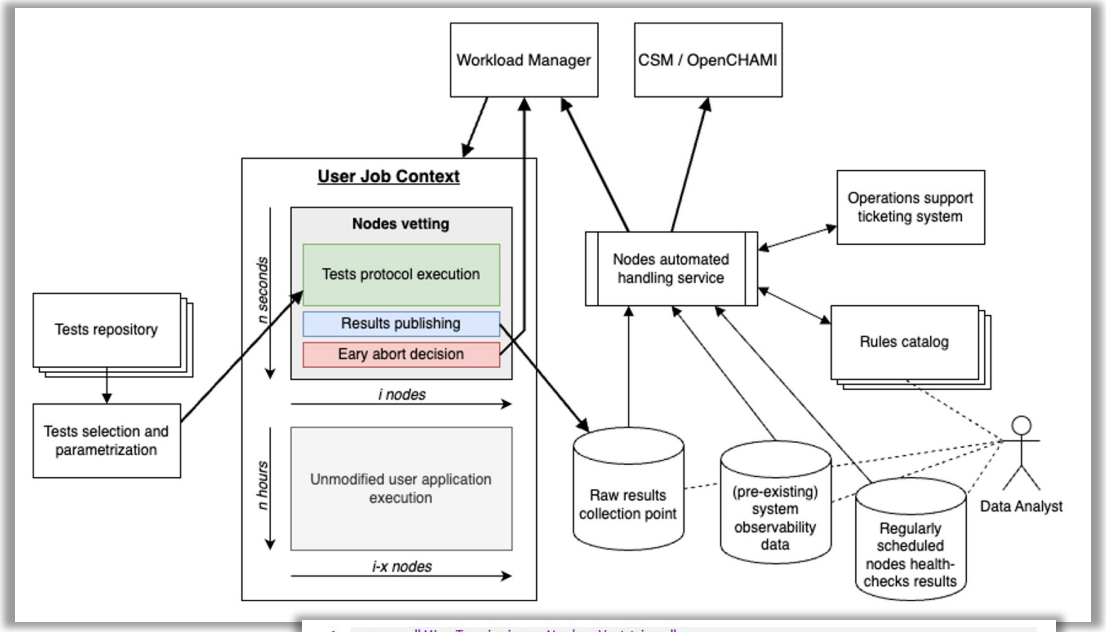
"HammingMesh: A Network Topology for Large-Scale Deep Learning"
by Torsten Hoefler - https://www.youtube.com/watch?v=4_Ma01QtttQ

- Design elements
  - Comprehensive data coverage of multiple system aspects to ready the system for unexpected correlation needs.
  - Value out-of-the-box and progressive opt-in for additional features and gradually deeper insights
  - Dashboards might not be enough!
    - Data analysis web tools to support more advanced logic
- Ongoing work
  - Increase data coverage and provide users with data quality information
  - Define how to preserve knowledge from past major efforts
    - Longer-term idea: automated analysis of data based on existing knowledge of the system

A Node Vetting and Early Abort System

- Background
  - Reliability of nodes is critical for large-scale ML
  - Project and overall cluster-productivity impacted
  - Explicit nodes exclusion by users is not desirable
- Our (pragmatic) approach
  - Support users' productivity by providing ways to verify the compute-readiness of allocated nodes, before the application is executed
    - Opt-in and focus on large-scale jobs
  - A tool that is common to all users also helps us collecting data needed for operation purposes

A Node Vetting and Early Abort System




```yaml
name: "ML Training Node Vetting"
evals:
- name: "Check GPU"
  type: vetnode.evaluations.gpu_eval.GPUEval
  max_temp:  30 #(celsius)
  max_used_memory: 0.2 #(%)
- name: "NCCLBandwidth"
  type: vetnode.evaluations.nccl_eval.NCCLEval
  min_bandwidth: 90.0 #(GBps)
  requirements:
    - torch
- name: "CudaKernel"
  type: vetnode.evaluations.cuda_eval.CUDAEval
  requirements:
    - cuda-python
    - numpy
```

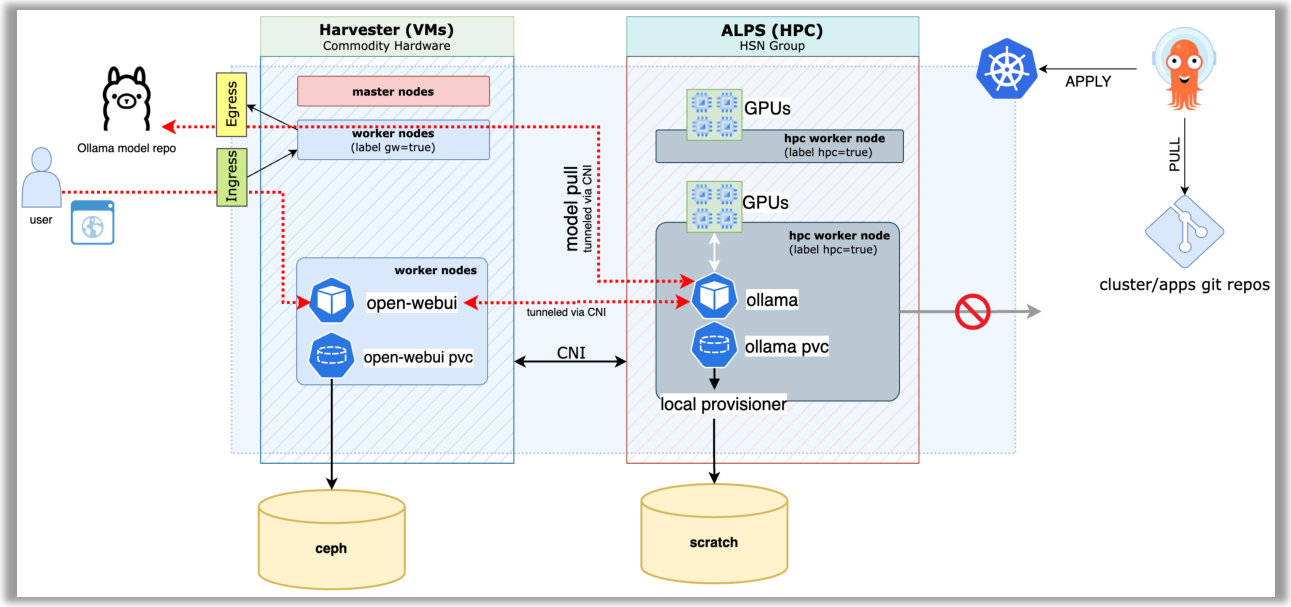
Listing 3: Example of Node Vetting Protocol

- Design elements
  - A lightweight utility to rapidly verify the compute-readiness of allocated nodes
    - A catalog of tests – which users can select depending on their specific needs
  - A centralized system to handle repetitive offending nodes
    - A catalog of rules to trigger automated node exclusion and healing actions
- Ongoing work
  - Development of the central handling system
  - Evaluating ways for capturing the application outcome

A Service Plane for Supporting and Inference Services

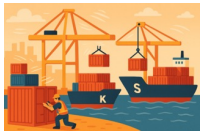
- Background
  - There is an ecosystem of established tools to facilitate activities throughout the life-cycle of ML projects
    - These are usually web-based, long-running applications and do not perform significant computational work
    - The absence of viable alternatives might force users to utilize high-end nodes to operate such services
  - Deploying and operating such services on behalf of users is not desirable.
  - There is increased interest in deploying inference services – which have similar characteristics.
    - The desire is for a MLOps-inspired paths between training and inference.
- Our approach
  - Introducing a dedicated infrastructure tailored for deploying long-running services.

A Service Plane for Supporting and Inference Services



- Design elements
  - A RKE2-based Kubernetes cluster is configured to unify an heterogenous set of nodes:
    - VMs (commodity hardware)
    - GPU nodes from the Alps infrastructure
  - While only services on commodity hardware are internet-accessible, CNI enables their communication with services deployed on the Alps infrastructure.
  - ArgoCD is used to implement GitOps processes through which users deploy their services.
- Ongoing work
  - Integration with existing IAM services.
  - Integration with accounting workflows.
  - Identifying the appropriate interfaces to facilitate user adoption

Summary and Key Points



Support for Container-based User Environments

A GPU Saturation Scorer for ML Applications

Infrastructure Observability for ML Workloads

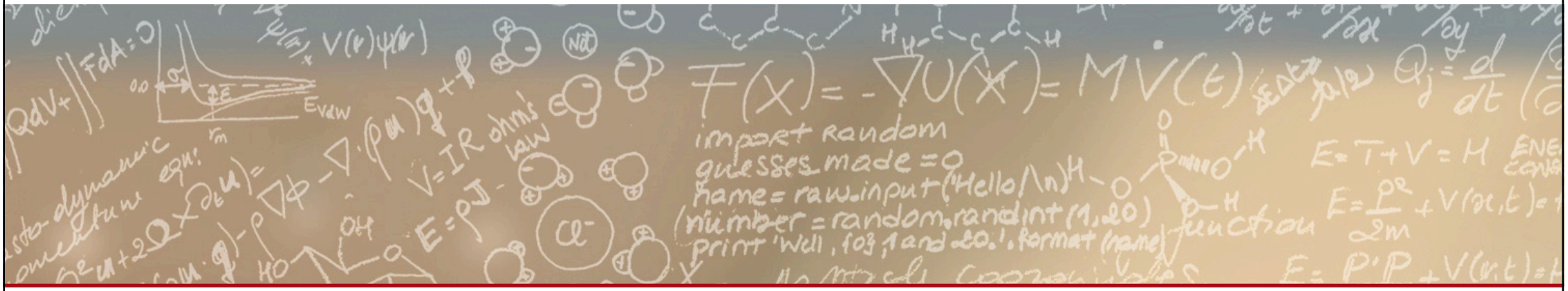
A Node Vetting and Early Abort System

A Service Plane for Supporting and Inference Services

- These approaches are meant to be general and applicable beyond our current infrastructure.
- These proposed solutions could be considered as structures on which our accumulated expertise can be organized, maintained and exposed for the benefit of our users and our operational teams.
- Traditional HPC services alone may not be sufficient for large ML projects. Through these solutions we aim to advantage our users by helping them focus on their core work and better compete on the world stage.

CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

ETH zürich



Thank you for your attention.