Efficiency, Reproducibility, and Portability in HEP Machine Learning Training - ML Training Facility at Vanderbilt University

Project Close-Out Report

Principal Investigator: Andrew Melo
Postdoc: Jethro Gaglione
Institution: Vanderbilt University

The goal of this project was to design, deploy, and validate a proof-of-concept Machine Learning Training Facility (MLTF) to support efficient, scalable, and reproducible machine learning (ML) training, with a particular focus on high-energy physics (HEP) applications within the CMS collaboration. As ML models used in HEP have grown substantially in size and complexity, training increasingly depends on specialized GPU hardware, high-speed interconnects, and optimized storage. At the same time, the community faces challenges related to reproducibility and limited access to GPU-based computing resources.

This project aimed to address these challenges by creating a centralized ML training facility that combines high-performance hardware with a standardized software stack based on MLflow. A core design principle was to enable both local and external users to submit and reproduce ML training jobs without requiring expertise in site-specific infrastructure or account access to Vanderbilt's HPC's (ACCRE) computing resources.

The MLTF proof-of-concept hardware consists of 16 NVIDIA A6000 GPUs, interconnected using NVLink for fast intra-node communication and RoCE (RDMA over Converged Ethernet) for GPU-to-GPU communication across nodes. High-performance NVMe storage was incorporated to enable fast data access, including use of GPUDirect Storage (GDS) to allow data transfers directly between storage and GPUs. GPUs and storage devices are housed in PCIe expansion enclosures, enabling flexible configurations and future scalability.

A significant portion of the project effort was spent validating and stabilizing this hardware stack. Early testing identified firmware and BIOS limitations, particularly related to PCIe ports and NVMe support. These issues required extended coordination with vendors to develop and deploy updated firmware. After several iterations, fully functional firmware was delivered, allowing the originally purchased hardware to be used as intended and eliminating reliance the loaner systems. Later in the project, ACCRE underwent a major system transition to Rocky Linux 9, along with updates to CUDA, NCCL, and NVIDIA drivers. RoCE functionality was revalidated under the new operating system, and "hello-world" demonstrations of GPUDirect Storage using local NVMe disks were successfully completed.

The MLTF software ecosystem is built around MLflow, which serves as the primary tool for experiment tracking, system metric logging, model and checkpoint archiving, and reproducibility. MLflow provides a unified interface across different ML frameworks and enables systematic preservation of training configurations and artifacts for future reuse and inference staging. A major deliverable of the project was the development of a custom MLflow submission gateway. This gateway enables remote users to submit ML training jobs to the MLTF via SLURM while interacting exclusively through MLflow Projects. Authentication is handled through the CERN single sign-on mechanism, allowing non-Vanderbilt users to access the facility without local computing accounts. The MLflow tracking infrastructure itself was deployed on a dedicated Kubernetes server and updated as needed to accommodate new MLflow releases and feature requirements. Extensive documentation was developed and expanded throughout the project, including tutorials covering TensorFlow and PyTorch workflows, hyperparameter optimization with Optuna, and novice-friendly examples of data- and model-parallel training.

A primary scientific use case for MLTF was the integration of CMS-specific ML training workflows, particularly those developed by the CMS BTV Physics Object Group (POG) using the B-Hive framework. Early project milestones included adding MLflow logging functionality to B-Hive, enabling automatic tracking of system metrics, training parameters, and model artifacts. Using this infrastructure, we successfully conducted training of the CMS BTV POG's DeepJet model within MLTF, validating both hardware performance and software reproducibility. This exercise provided a realistic benchmark for large-scale CMS models and demonstrated the benefits of using centralized logging/tracking in ML training.

Subsequent work focused on enabling data-parallel training within B-Hive using PyTorch Distributed Data Parallel (DDP). This effort required close collaboration with B-Hive developers and iterative refinement to remain compatible with updated PyTorch features, including compiled models. A functional DDP implementation was achieved and tested locally at MLTF. Early parallel training tests on the CMS production-level ParT model indicate a ×1.65 training speed efficiency improvement under conditions similar to those the model was nominally trained on, with further improvements expected as the DDP implementation is refined and training parameters optimized. This test was conducted on the ZToQQ subset of the JetClass dataset which the nominal model was trained on.

Engagement with the HEP ML community was an important component of the project. Early outreach included a presentation at the CERN Fast ML Co-Processing Group meeting in July 2024, where the MLTF concept and initial progress were introduced [1]. This talk generated early interest and led to several external users joining a beta testing phase. The MLTF was later presented at the Computing in High Energy Physics (CHEP) 2024 conference, using the DeepJet training workflow as a concrete demonstration of the facility's capabilities and benefits [2]. The presentation was well received and sparked follow-up discussions with CMS and non-CMS groups interested in using MLTF for their own training workflows. A proceedings paper describing this work was subsequently published in EPJ Web Conferences [3].

Additional internal presentations included a presentation at a B-Hive developers meeting detailing MLflow integration and data-parallel training developments [4], as well as participation in the B-Hive developers hackathon at CERN, where MLflow integration was completed and a merge request submitted [5]. These activities helped ensure that project outcomes were aligned with overall CMS software development and would benefit the broader collaboration.

Despite challenges associated with hardware, system software/OS transitions, and temporary administrative disruptions, the project successfully delivered a functional and validated machine learning training facility. The project demonstrated that a centralized ML training facility built on modern GPU hardware and standardized tooling such as MLflow can significantly reduce barriers to large-scale ML training while improving reproducibility and collaboration. Work on MLTF will continue beyond the scope of this project, with plans to finalize gateway deployment, push remaining B-Hive enhancements, and expand support to additional CMS and HEP ML use cases.

References

[1] MLTF Presentation, *CERN Fast ML Co-Processing Group Meeting*, July 19, 2024
https://indico.cern.ch/event/1438068/

[2] MLTF and DeepJet Training Demonstration, *Computing in High Energy Physics (CHEP) 2024*
https://indico.cern.ch/event/1338689/contributions/6010710/

[3] J. Gaglione, A. Melo, S. Dhakal, and P. Koirala. *Efficiency, Reproducibility, and Portability in HEP Machine Learning Training - ML Training Facility at Vanderbilt University*. **EPJ Web Conf., 337** (2025) 01172 DOI: https://doi.org/10.1051/epjconf/202533701172

[4] MLflow Tracking and Data-Parallel Training in B-Hive, *B-Hive Developers Meeting*
https://indico.cern.ch/event/1577553/

[5] CMS B-Hive Developers Hackathon, CERN, October 6–9, 2024
https://gitlab.cern.ch/cms-btv/b-hive/-/merge_requests/38