Distributed Machine Learning Workflow with PanDA and iDDS in LHC ATLAS

Wen Guan[1,*], *Tadashi* Maeno[1], *Rui* Zhang[2], *Christian* Weber[1], *Torre* Wenaus[1], *Aleksandr* Alekseev[3], *Fernando Harald* Barreiro Megino[3], *Kaushik* De[3], *Edward* Karavakis[1], *Alexei* Klimentov[1], *Tatiana* Korchuganova[4], *FaHui* Lin[3], *Paul* Nilsson[1], *Zhaoyu* Yang[1], and *Xin* Zhao[1]

[1]Brookhaven National Laboratory, Upton, NY, USA
[2]University of Wisconsin-Madison, Madison, USA
[3]University of Texas at Arlington, Arlington, TX, USA
[4]University of Pittsburgh, Pittsburgh, PA, USA

Abstract. Machine Learning (ML) has become one of the important tools for High Energy Physics analysis. As the size of the dataset increases at the Large Hadron Collider (LHC), and at the same time the search spaces become bigger and bigger in order to exploit the physics potentials, more and more computing resources are required for processing these ML tasks. In addition, complex advanced ML workflows are developed in which one task may depend on the results of previous tasks. How to make use of vast distributed CPUs/GPUs in WLCG for these big complex ML tasks has become a popular research area. In this paper, we present our efforts enabling the execution of distributed ML workflows on the Production and Distributed Analysis (PanDA) system and intelligent Data Delivery Service (iDDS). First, we describe how PanDA and iDDS deal with large-scale ML workflows, including the implementation to process workloads on diverse and geographically distributed computing resources. Next, we report real-world use cases, such as HyperParameter Optimization, Monte Carlo Toy confidence limits calculation, and Active Learning. Finally, we conclude with future plans.

1 Introduction

Machine Learning (ML) has emerged as a crucial tool within the ATLAS experiment [1] at the LHC [2]. The continuously growing data volume and the complexity of the physics analysis have intensified the interest in large-scale ML applications. In the context of this paper, a distributed ML workflow involves a sequential chain of workloads, where one or more workloads incorporate machine learning tasks and these workloads can be distributed across geographically computing resources. It encompasses various types of workloads, and achieving full automation in their execution is crucial.

The Production and Distributed Analysis (PanDA) system [3, 4] is a robust workload management system that excels in handling distributed computing resources and is well-suited for large-scale ML workloads. It can efficiently schedule workloads to large-scale

*e-mail: wguan2@bnl.gov

distributed heterogeneous computing resources among different institutes and organizations. Another key advantage of PanDA is that it provides users a uniform view for different computing resources. It abstracts the underlying computing resources, presenting a unified interface for submitting and managing workloads. Users can submit workloads to PanDA without needing to be aware of the specific details of the computational resources. PanDA's capabilities to transparently manage diverse computing resources are vital for the execution of distributed ML workloads.

The intelligent Data Delivery Service (iDDS) is a workflow orchestration system designed to automate complex and dynamic workflows. It orchestrates the execution of workloads within each workflow, considering the topological dependency among these workloads, with advanced functions such as workflow description with Directed Acyclic Graph (DAG), conditional branching, iterative sequences, polymorphic workloads, and so on. iDDS is flexible in handling a wide variety of complex workflows, making it suitable for a wide range of automation scenarios. iDDS provides an easy way to build large-scale distributed ML workflows.

We can create a powerful framework for distributed large-scale ML workflows based on PanDA and iDDS, as shown in Figure 1. PanDA serves as the execution engine for large-scale ML workloads on distributed computing resources. iDDS orchestrates the workflow, automating the execution chain of consistent workloads. It triggers the execution of downstream workloads based on the results of upstream workloads, allowing advanced automation and processing.

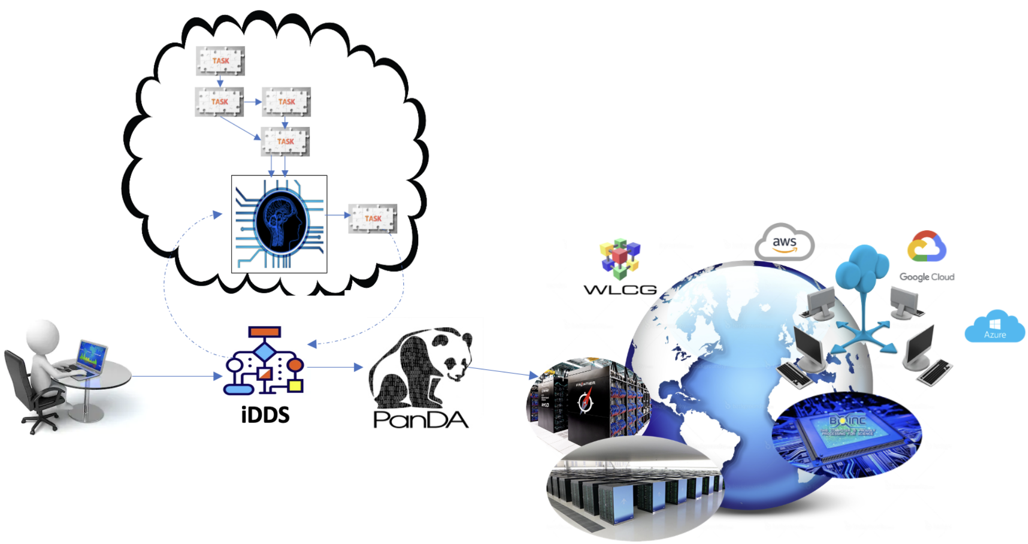

Figure 1: An integrated workflow with PanDA and iDDS, where iDDS automates complex and dynamic workflows, and PanDA schedules workloads to large-scale distributed heterogeneous computing resources.

2 Workflow Orchestration for distributed ML

iDDS orchestrates the execution of different workloads within each workflow. It supports conditional branching in workflows with condition functions. A condition function is one of system-defined functions, a user-defined function, or the results of a workload. In some ML use cases, certain workloads collect the results of upstream workloads to make decisions regarding the selection and execution of downstream workloads. Conditional branching enables a wide range of distributed ML use cases.

Figure 2 shows an iterative workflow, where iDDS aggregates the results from previous workloads to generate new workloads and submit them to PanDA. PanDA analyzes the characteristics of those tasks and schedules them on distributed heterogeneous computing resources, which improves overall efficiency in large-scale ML workloads. PanDA and iDDS work together to streamline distributed ML workflows.

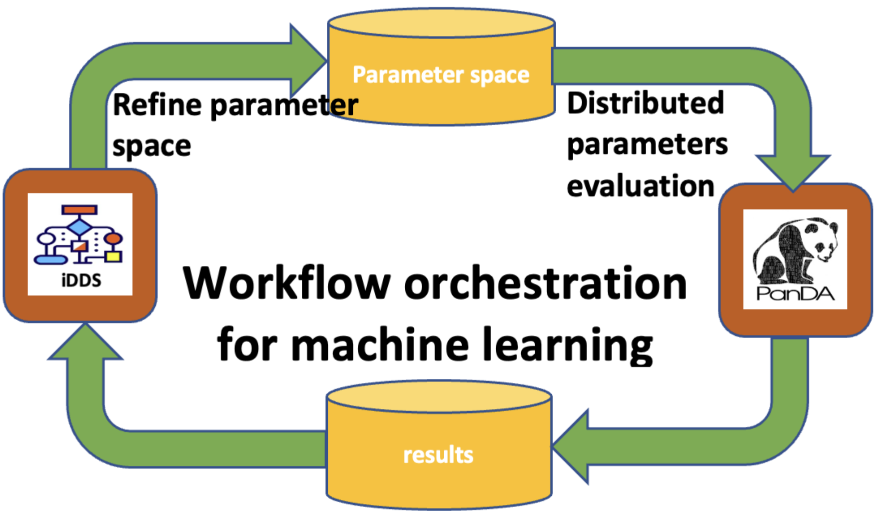

Figure 2: An iterative workflow, where iDDS aggregates the results from previous workloads to generate new workloads and submit them to PanDA.

3 Use Cases

3.1 Distributed HyperParameter Optimization

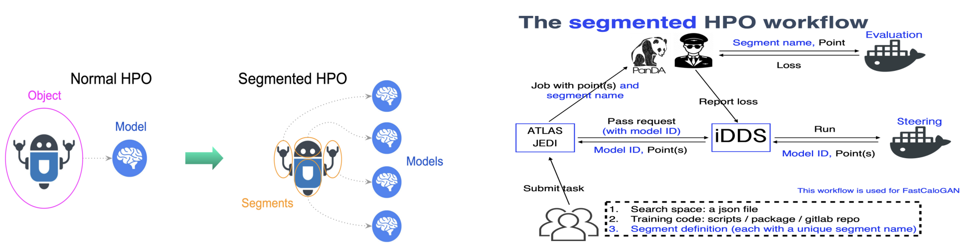

Figure 3: Segmented HyperParameter Optimization

In ML, HyperParameter Optimization (HPO) is a critical task to tune the parameters that control the training process iteratively. In geographically distributed environments, implementing the iterative process of HPO can be challenging due to the lack of direct access to the results of previous workloads. To overcome this challenge, iDDS has provided a fully automated platform that specifically addresses the HPO requirements in distributed ML, ensuring the efficient execution HPO workflows with a capability to collect the results of upstream workloads across distributed computing resources.

The workflow of iDDS for HPO involves several steps. First, iDDS centrally scans the hyperparameter search space using advanced ML optimization algorithms, such as the Bayesian optimization method [5], to generate hyperparameter points. These points are then asynchronously evaluated on remote CPU/GPU resources through PanDA. The training results are reported back to iDDS, which further optimizes the search space and generates a new round of hyperparameter points. This iterative process continues until the optimal hyperparameter point and the associated trained models are obtained, marking the completion of the iterations. In addition to HPO, iDDS also implemented an enhanced segmented HPO method to optimize multiple ML models at the same time. This new method improves the training efficiency and also improves the performance by reducing the bias, as shown in Figure 3.

The HPO service provided by PanDA and iDDS offers a fully automated platform for ML users. By utilizing geographically distributed CPU/GPU resources, it enables large-scale application of computational power to tackle extensive HPO tasks. It is currently in production for ATLAS ML users, for example for the FastCaloGAN analysis [6]. However, it is important to note that the service is designed to be experiment-agnostic, its design allowing for easy adaptation and utilization outside of ATLAS.

3.2 Monte Carlo Toy-Based Confidence Limits

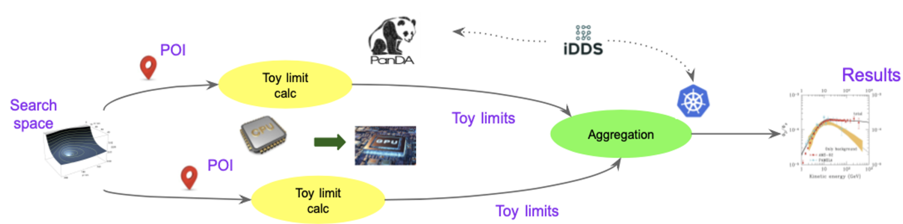

Figure 4: Multiple-steps Monte Carlo Toy-based confidence limits calculations and aggregations.

Confidence limits need to perform statistical tests or hypothesis tests to show that the obtained results are significantly different from what could have been obtained by chance, which typically involves computationally intensive grid scans. To minimize random grid scanning, efficient Monte Carlo (MC) Toy-based confidence limits require multiple steps of grid scans in different granularity, where each step results in certain phase space ranges that need to be excluded in the analyses.

To address the requirements of multi-step workflow for confidence limits, an integrated workflow with PanDA and iDDS has been developed, where iDDS automates the multi-step workflow by coordinating the toy limits calculations and aggregations. As shown in Figure 4, Points of Interest (POI) are generated based on the search space. The toy calculations are then scheduled on distributed computing resources through PanDA. At the end of each loop, the results are aggregated to generate new search spaces. iDDS triggers the aggregation of results and schedules new steps in the workflow, ensuring a seamless progression.

3.3 Active Learning

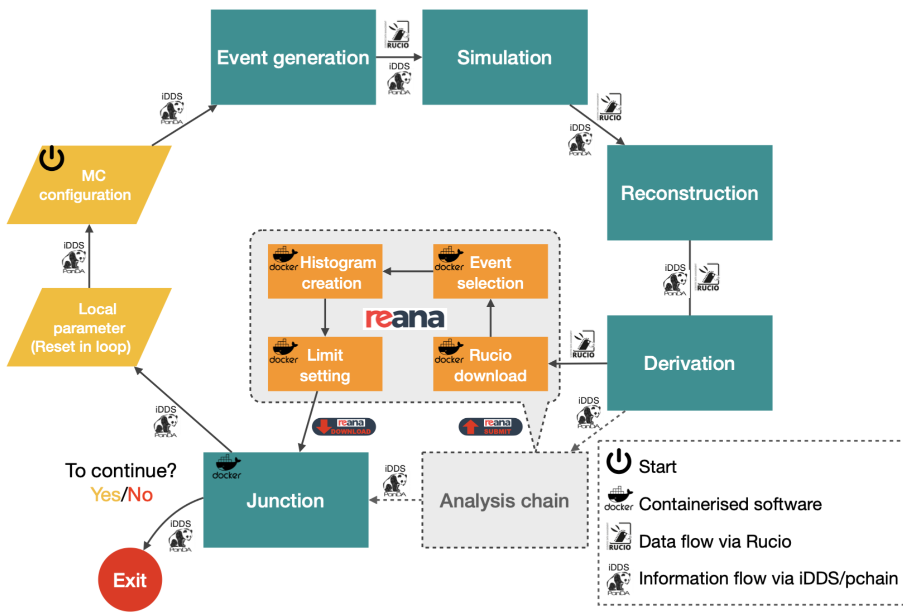

Figure 5: A schematic view of the AL workflow with a $H \to ZZ_d \to 4l$ optimized analysis.

Active Learning (AL) [7] is an iterative parameter search technique by refining the parameter space for the next iteration based on the previous results. ML methods are used to efficiently guide the parameter search process, improving the overall efficiency compared to a single-step processing approach. It can enhance the efficiency in new physics searches.

By integrating PanDA and iDDS, an automated iterative ML-assisted workflow has been developed. In this workflow, iDDS orchestrates the processing between different steps to enable iterative parameter search.

One AL workflow is demonstrated in a $H \to ZZ_d \to 4l$ optimized analysis [8], as shown in Figure 5. This workflow includes the production chain and analysis chain. The production chain starts with a template MC configuration, until the step to generate a corresponding Derived AOD (DAOD) sample for each physics parameter point. Once a DAOD sample is produced, the analysis chain starts. It includes PanDA jobs which run a REANA workflow task with a series of analysis steps in a REANA cluster [9] and PanDA jobs that run AL logic, with Bayesian optimization to look for the maximal difference between the observed and expected limits to identify excesses, to generate new parameter points. In this workflow, every step is coordinated by iDDS for automation, without human intervention. This workflow has demonstrated AL driven re-analysis for dark sector analysis. The result has been published in the ATLAS PUB Note [8]. In addition, another onging AL workflow is in a generic Heavy Higgs→ WW search analysis.

4 Conclusions

The intelligent Data Delivery Service has been developed to support emerging use cases [10, 11] in ATLAS and other experiments. It is in production in ATLAS, Rubin Observatory (LSST) experiment [12, 13] and sPHENIX [14, 15]. An integrated PanDA and iDDS service to enable distributed large-scale ML workflow is an enabler for efficient utilization of resources and streamlined execution of ML tasks.

In the future we will continue to generalize these services as a contribution to the HEP ML ecosystem, making them more widely accessible.

Acknowledgements

This manuscript has been authored by employees of Brookhaven Science Associates, LLC under Contract No. DE-SC0012704 with the U.S. Department of Energy. The publisher by accepting the manuscript for publication acknowledges that the United States Government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes.

References

[1] ATLAS Collaboration, *The ATLAS Experiment at the CERN Large Hadron Collider*, J. Inst. **3**, S08003 (2008)

[2] L. Evans and P. Bryant (editors), *LHC Machine*, J. Inst. **3**, S08001 (2008)

[3] T. Maeno et al., *PanDA: Production and Distributed Analysis System*, Computing and Software for Big Science (to be submitted)

[4] T. Maeno et al., *Overview of ATLAS PanDA Workload Management*, J.Phys.: Conf. Ser. **331**, 072024 (2011)

[5] P. Frazier, *A tutorial on bayesian optimization* (2018), `arXiv:1807.02811`

[6] J. M., *The Fast Simulation Chain in the ATLAS experiment*, EPJ Web Conf. **251**, 03012 (2021)

[7] K. Cranmer, *Active learning for excursion set estimation* (2019), `ACAT 2019`, `https://indico.cern.ch/event/708041/contributions/3269754/`

[8] ATLAS Collaboration, *Demonstrating an active learning driven pipeline for optimised analysis reinterpretation: an extended search for Higgs bosons decaying into four-lepton final states via an intermediate dark Z boson* (2023), `https://atlas.web.cern.ch/Atlas/GROUPS/PHYSICS/PUBNOTES/ATL-PHYS-PUB-2023-010/`

[9] *REANA*, `https://reanahub.io/`

[10] W. Guan et al., *An intelligent Data Delivery Service for and beyond the ATLAS experiment*, EPJ Web Conf. **251**, 02007 (2021)

[11] W. Guan et al., *Towards an Intelligent Data Delivery Service*, EPJ Web Conf. **245**, 04015 (2020)

[12] Z. Ivezic et al., *LSST: From Science Drivers To Reference Design And Anticipated Data Products*, Astrophys. J., **873**, 111 (2019)

[13] LSST Science Collaboration, *Lsst science book, version 2.0* (2009), `arXiv:0912.0201`

[14] A. Adare et al., *An upgrade proposal from the PHENIX Collaboration* (2015), `arXiv:1501.06197`

[15] sPHENIX Collaboration, *Design, status and schedule of the sphenix experiment at rhic*, Nuclear Physics A **967**, 548 (2017)