An intelligent Data Delivery Service for and beyond the ATLAS experiment

Wen Guan[1,*], *Tadashi Maeno*[1], Brian Paul Bockelman[2], Torre Wenaus[1], Rui Zhang[3], Christian Weber[1], Fernando Harald Barreiro Megino[4], Fahui Lin[4] and Aleksandr Alekseev[5]

[1]Brookhaven National Laboratory, Upton, USA
[2]Morgridge Institute for Research, Madison, USA
[3]University of Wisconsin-Madison, Madison, USA
[4]University of Texas at Arlington, USA
[5]Moscow State U.; Andres Bello Natl. U.; Moscow, INR

Abstract. The intelligent Data Delivery Service (iDDS) has been developed to cope with the huge increase of computing and storage resource usage in the coming LHC data taking. It has been designed to intelligently orchestrate workflows and data management systems, decoupling data pre-processing, delivery, and primary processing in large scale workflows. It is an experiment-agnostic service that has been deployed to serve data carousel (orchestrating efficient processing of tape-resident data), machine learning hyperparameter optimization, active learning, and other complex multi-stage workflows defined via DAG (Directed Acyclic Graph), CWL (Common Workflow Language) and other descriptions, including a growing number of analysis workflows. We will at first introduce some deployed use cases in a summary. Then we will focus on new improvements and use cases under developments in ATLAS, Rubin Observatory and sPHENIX, together with future efforts.

1 Introduction

The iDDS is a project developed for intelligent granular data delivery and orchestration which supports complex workflows to efficiently use resources such as storages, networks, processing CPUs and so on. It's an experiment-agnostic service which has been employed by LHC ATLAS [1][2], Vera Rubin Observatory (LSST) [3][4] and sPHENIX at RHIC [5][6]. It has been successfully deployed for different use cases:

- Fine-grained Data Carousel for LHC ATLAS [7][8]: The iDDS enables grouping processing in proper granularities to efficiently use disk storages. It has added the capability to the WorkFlow Management system (WFM system) [9][10] to work with fine-grained file-level data. Input data is incrementally processed based on more detailed knowledge on the status of input data, to reduce the overhead and get rid of redundant data transfers and caching. Processed data is released from the cache promptly with similarly fine granularity, such that the full workflow minimizes the input data footprint on disk. iDDS has been integrated with the ATLAS computing system since mid 2020 and has been used for bulk data reprocessing campaigns.

[*] Email: wen.guan@cern.ch

- Scalable Hyperparameter Optimization (HPO) service [7][8]: It's a scalable Machine Learning (ML) service to efficiently distribute hyperparameter optimization tasks and other ML workflows to distributed CPU/GPU resources. The iDDS has provided a fully automated platform for HPO on top of geographically distributed CPU/GPU resources among the Grid, HPC, and clouds, such that large scale resources can be applied for large HPO tasks. Meanwhile, the same architecture has been adapted to more and more use cases, such as the Monte Carlo Toy based confidence limits workflow.
- DAG (Directed Acyclic Graph) workflow [7][8]: The iDDS has implemented a DAG workflow support for the Rubin Observatory exercise, in which a single workflow can consist of a hundred thousand jobs forming the vertexes of a DAG. iDDS allows jobs to be incrementally released based on a messaging service, to avoid long waiting time in each Work.

The iDDS continues its efforts on the current use cases to improve the user experience and efficiency. At the same time, new developments have been enhancing and extending its usage in different experiments. In this paper, we will present the improvements in iDDS, with also improvements and developments in current use cases and new use cases.

2 Enriched Workflow Management

The iDDS has implemented a high-level workflow engine to automate complex production and analysis workflows. It interacts with workload management systems such as PanDA, to drive workload scheduling.

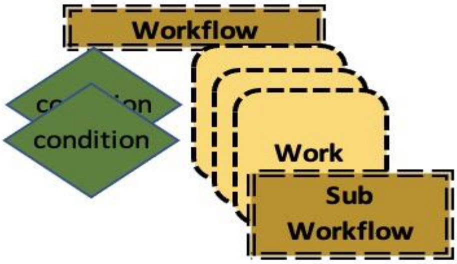


Fig. 1. Task level workflows/sub workflows.

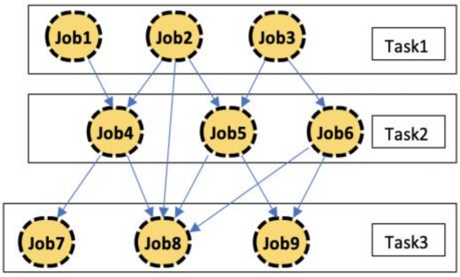


Fig. 2. Job level DAG

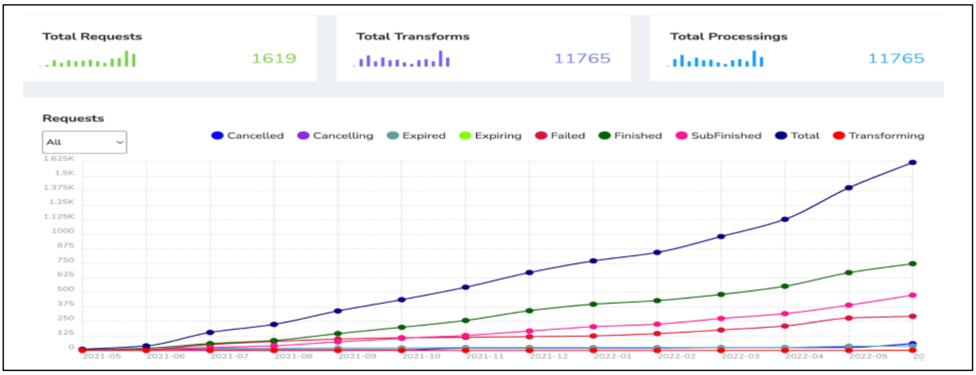
The high-level workflow engine in iDDS works to manage task chains and job chains. In the task level, the iDDS has implemented the DG (Directed Graph) workflow management which not only supports DAG (Directed Acyclic Graph), but also supports graphs with cycles. It has implemented templates for DAG workflows and Loop workflows, in which pre-defined conditions and custom conditions can be used to control how to select different branches to be executed. In the job level, DAG has been implemented to manage the dependencies between different jobs. The iDDS automatically evaluates the dependencies to release jobs in an incremental mode.

3 Use Cases

3.1 DAG management for Rubin Observatory

The Rubin Observatory exercise employs PanDA as both a workflow and workload management system. The iDDS is integrated as a workflow manager to manage the DAG dependencies in the task level and job level. In the job level, iDDS optimizes for managing

DAG dependencies of jobs in and between tasks. When a job terminates, a trigger system in iDDS will be triggered to evaluate corresponding child jobs and release jobs. In the task level, iDDS optimizes for triggering the finalizing tasks, such as the merging tasks. It has been in production since the summer of 2021. In the first half year of 2022, the DP0.2 (Phase 2 of Data Preview 0) campaign has been successfully processed. After that, an even bigger processing campaign, the HSC (Hyper Suprime-Cam) processing, has been decided to use PanDA and iDDS to process the camera data, which is still ongoing currently. Until the presentation, iDDS-PanDA has processed more than 11000 tasks, where many tasks have more than10K jobs, as shown in Figure 3. The Figure 4 shows the task dependency map of an example workflow.



Total Requests 1619

Total Transforms 11765

Total Processings 11765

Requests

All

Fig. 3. Since late 2021 in Rubin Observatory, iDDS-PanDA within the LSST framework has processed more than 11000 tasks, where many tasks have more than 10K jobs.

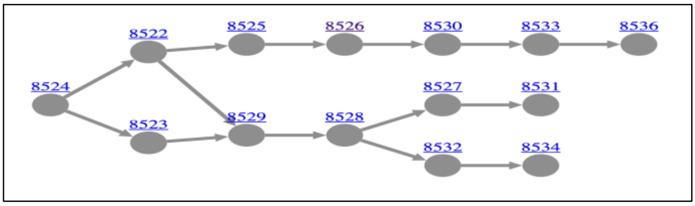


Fig. 4. The DAG monitor shows the relationship between different tasks in a workflow.

3.2 Task Chains for sPHENIX

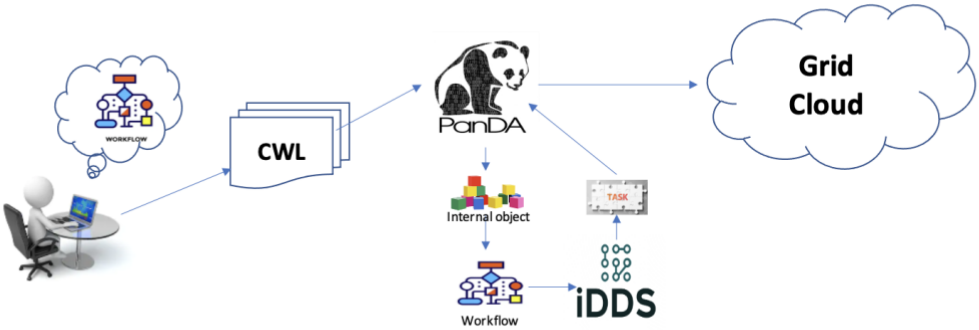
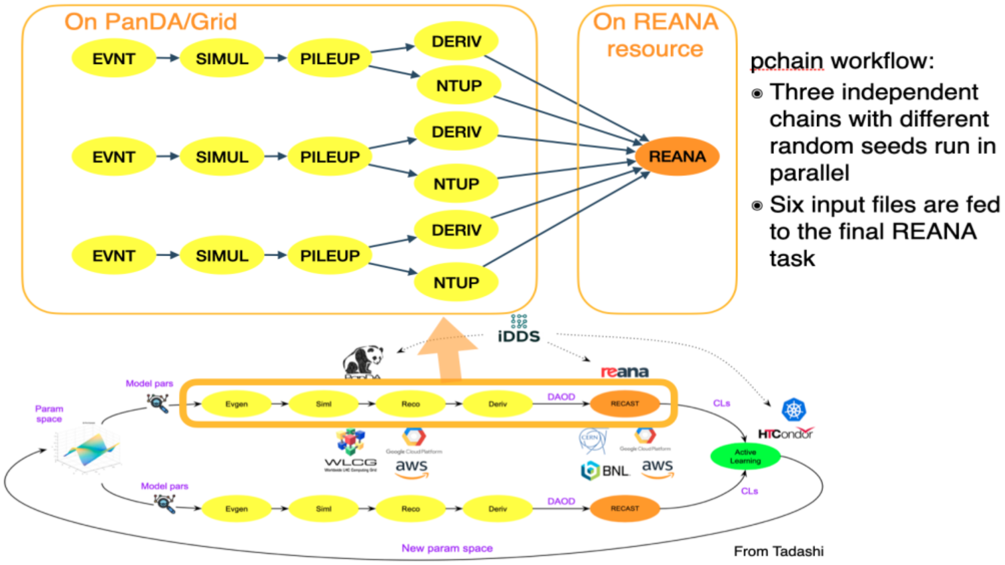


Fig. 5. The iDDS for sPHENIX

The sPHENIX experiment at RHIC adopted PanDA/iDDS at about the same timeline as Rubin Observatory. The iDDS is adopted for task chain management and PanDA is adopted for task management. In sPHENIX, the task chain is defined with Common Workflow Language (CWL) [11], which is transformed into internal workflow objects. The iDDS works to manage the internal workflow objects and triggers to execute tasks in the chain, as shown in Figure 5.

3.3 LHC ATLAS Analysis

3.3.1 Multiple-Steps Task Chain



pchain workflow:

- Three independent chains with different random seeds run in parallel
- Six input files are fed to the final REANA task

Fig. 6. Multiple-steps processing: The domain space is redefined for the next iteration based on previous iterations. In this way, the previous results can improve the efficiency of next iterations.

In LHC ATLAS, with more and more data, it requires more and more computing resources to process it. At the same time, to explore the physics potentials, physicists work hard to dig the data in more details, which also increases the requirements of computing resources. To reduce the reliance on computing resources, advanced complex workflows are designed. In the new workflow, the domain space is redefined for the next iteration based on previous iterations, to make sure the next iteration can only focus on a potential area, instead of scanning all areas blindly.

The iDDS workflow engine works to automate the multiple-step processing. It parses the results of previous tasks, triggers an optimization step to calculate the new domain space for the next step, and then schedules new processing steps based on the new domain space, as shown in Figure 6. The support of loop workflows in iDDS makes it possible. Currently it's already integrated with PanDA and REANA [12]. It has successfully been tested with a mono-Hbb analysis. We are working on optimizing and simplifying the workflow for production usages.

3.3.2 MC Toy Based Confidence Limits

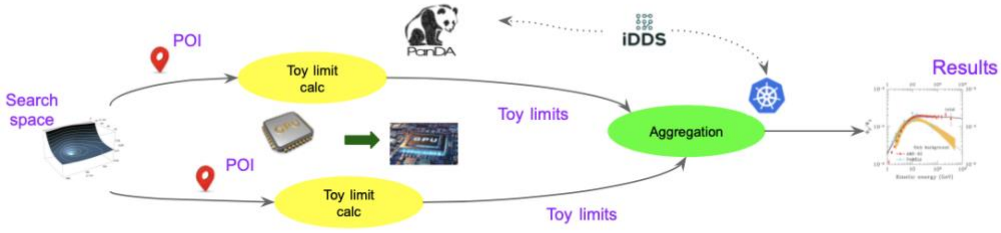


Fig. 7. Multiple-steps Monte Carlo Toy based confidence limits calculations and aggregations.

An efficient Monte Carlo Toy based confidence limits workflow requires multiple steps of grid scans, where current steps depend on previous steps. The iDDS automates this workflow of the toy limits calculations and aggregations. In this workflow, Points of Interest (POI) are generated based on the search space. Then the toy calculations are scheduled to distributed computing resources through PanDA. At the end of one loop, the results are aggregated to generate new search spaces. In the workflow, iDDS triggers to aggregate results and then to schedule new steps, as shown in Figure 7.

4 Summary and Outlook

iDDS has been developed to support various emerging use cases in ATLAS and other experiments. It has already been in production in ATLAS and Rubin Observatory (LSST) experiment and is under integration for sPHENIX. We will continue to support and improve the user experience and efficiency for the current use cases. At the same time, we also plan to put more efforts in distributed machine learning for advanced complex workflows in the future.

References

1. ATLAS Collaboration, *2008 JINST* 3 S08003.
2. L. Evans and P. Bryant, *Journal of Instrumentation*, 3 (2008).
3. LSST Science Collaboration, arXiv: 0912.0201 (2009).
4. Z. Ivezic et al., *ApJ*, 873 111 (2019).
5. sPHENIX Collaboration, *Nucl.Phys.A* 967, 548-551 (2017).
6. R. Reed, *J. Phys.: Conf. Ser.* 779, 012019 (2017).
7. W. Guan et al., *EPJ Web Conf.* 245, 04015 (2020).
8. W. Guan et al., *EPJ Web of Conf.* 251, 02007 (2021).
9. J. Elmsheuser and A. D. Girolamo, *EPJ Web of Conferences* 214, 03010 (2019).
10. F. H. Barreiro et al., *J. Phys.: Conf. Ser.* 898, 052016 (2017).
11. M.R. Crusoe et al., *Communications of the ACM*, Vol. 65, No. 6, 54-63 (2022).
12. T. Simko et al., *Front. Big Data*, 2021.661501 (2021).