Brookhaven National Laboratory

U.S. DEPARTMENT OF ENERGY

intelligent Data Delivery Service (iDDS) for and beyond the ATLAS experiment

Wen Guan on behalf of the ATLAS Collaboration

International Conference for High Energy Physics (ICHEP 2022)
July 8, 2022

@BrookhavenLab

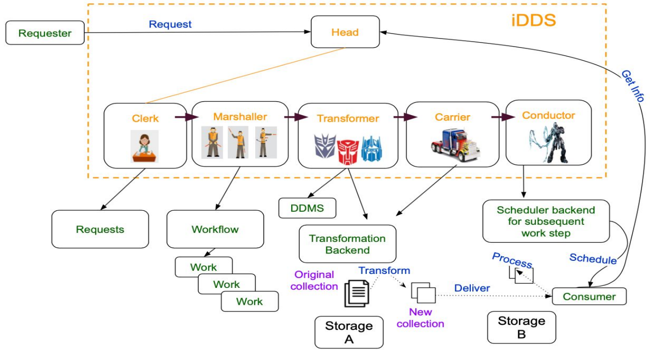
intelligent Data Delivery Service (iDDS)

Intelligent granular data delivery and orchestration supporting complex workflows that are efficient in the use of storage, network and processing.

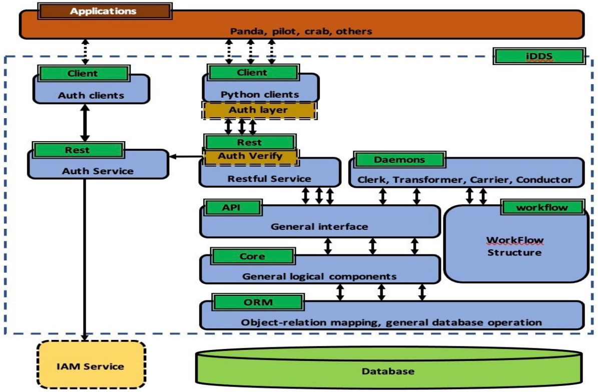
- Joint project between **US-ATLAS and IRIS-HEP**.
- HEP Software Foundation hosts the project

Significance and Impact

- **Experiment-agnostic service**, employed by **LHC ATLAS** and **Vera Rubin Observatory**, **sPHENIX at RHIC** is exploring it.
- Fine-grained **Data Carousel** for LHC ATLAS enables processing in proper granularities and grouping to efficiently use disk storage. It has been running smoothly for last 2 years.
- Scalable Machine Learning service to efficiently distribute ML **hyperparameter optimization** tasks and other ML workflows to distribute CPU/GPU resources. Being adapted to more and more use cases.
- Complex dynamic **workflow management**, such as DAG (Directed Acyclic Graphs), Loop workflow, Template Workflow and Condition workflow to automate complex production and analysis workflows.
- **Ongoing R&D** - a growing number of analysis use cases that are demanding in their complexity and resource needs: active learning, integrated REANA (Reusable Analyses) workflows, fine-grained data transform and delivery.



iDDS Schematic View

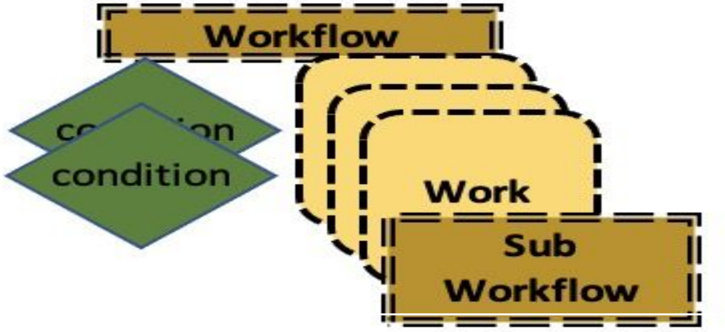


iDDS Architecture

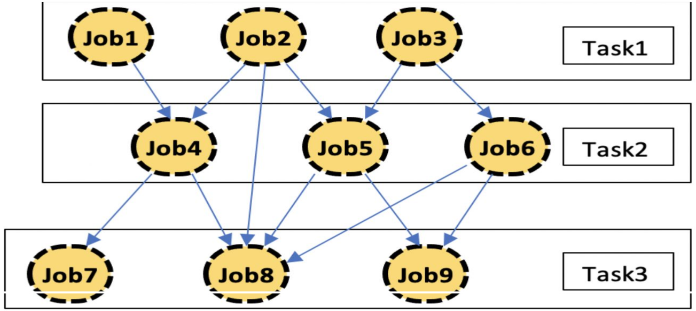
iDDS Workflow Management

Implemented a high-level workflow engine, to manage task/job chains. Interacting with software such as PanDA, to drive workload scheduling.

- **Task/Job dependency management**
  - Task Level
    - DAG (Directed Acyclic Graphs)
    - SubWorkflow, TemplateWorkflow, LoopWorkflow
  - Job Level
    - DAG
- **Conditions** to manage the dependencies between different tasks during execution, to trigger executing different tasks based on previous tasks' results
  - AndCondition, OrCondition and their combinations
  - Attributes of different work instances can be used as a condition.
  - A task's output can also be used as a condition.
- **Use cases**
  - Rubin Observatory (deployed)
  - sPHENIX (testing)
  - LHC Analysis (R&D)



Task level workflow/subworkflow supports, with conditions and templates to trigger executing different tasks based on previous tasks' results.

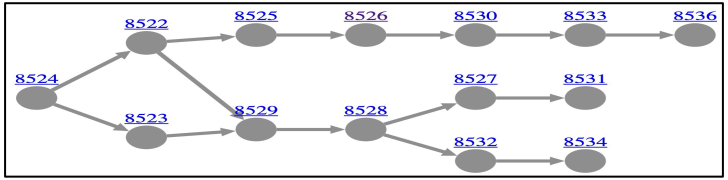


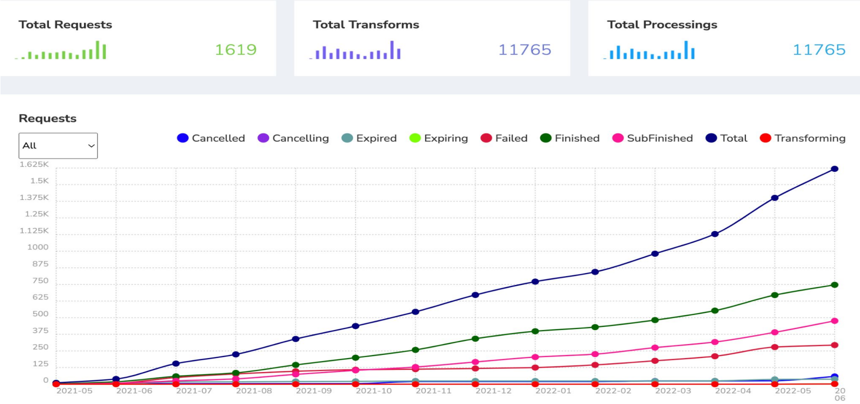
Job Level DAG (Directed Acyclic Graphs) management to manage the executing order of dependent jobs.

iDDS for Rubin Observatory

Employ job-level DAG to manage the dependencies of different jobs, task-level DAG to manage the final merging task.

- Integration done in first half of 2021
- Culminated during spring/summer in successful scaling tests (~200k concurrent jobs) that drove the Rubin decision to adopt
- Since then, evolved from a demonstrator to a production system
- Job level DAG optimized for managing DAG dependencies of big tasks (80K jobs per task)
- Task level DAG optimized for triggering the final merging task
- Largely stable since Oct 2021
- **DP0.2 (Phase 2 of Data Preview 0) campaign successfully** from the beginning of 2022 to June 2022.
  - From Jan-Jun 2022. Workflows: 351, tasks: 2732, jobs: 16483753
- **HSC (Hyper-Suprime Cam) processing ongoing.**
  - Started from Jun 2022



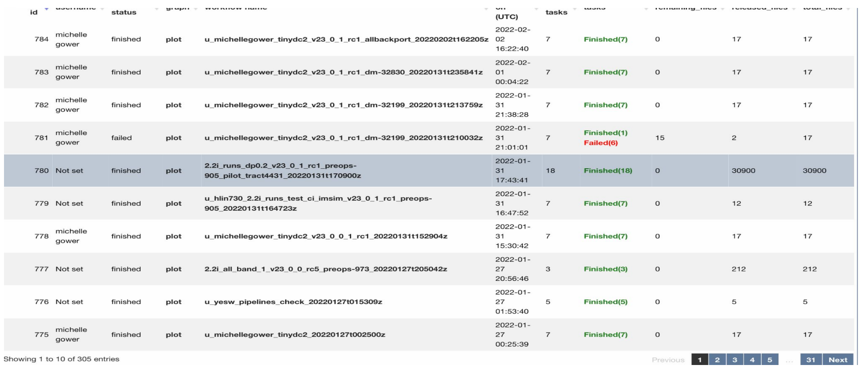

Total Requests 1619

Total Transforms 11765

Total Processings 11765

Requests

Since late 2020 in Rubin Observatory, iDDS-PanDA within the LSST framework has processed more than 11000 tasks.



id	username	status	graph	workflow name	on (UTC)	tasks	tasks	remaining_files	released_files	total_files
784	michelle gower	finished	plot	u_michellegower_tinydc2_v23_0_1_rc1_allbackport_20220202t162205z	2022-02-02 16:22:40	7	Finished(7)	0	17	17
783	michelle gower	finished	plot	u_michellegower_tinydc2_v23_0_1_rc1_dm-32830_20220131t235841z	2022-02-01 00:04:22	7	Finished(7)	0	17	17
782	michelle gower	finished	plot	u_michellegower_tinydc2_v23_0_1_rc1_dm-32199_20220131t213759z	2022-01-31 21:38:28	7	Finished(7)	0	17	17

781	michelle gower	failed	plot	u_michellegower_tinydc2_v23_0_1_rc1_dm-32199_20220131t210032z	2022-01-31 21:01:01	7	Finished(1)
							Failed(6)
								15	2	17

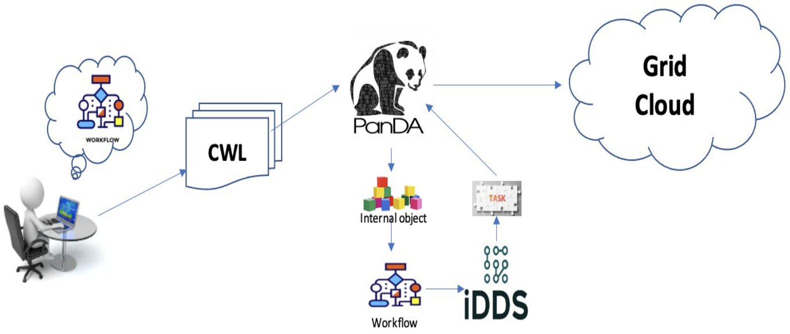
780	Not set	finished	plot	2.2i_runs_dp0.2_v23_0_1_rc1_preops-905_pilot_tract4431_20220131t170900z	2022-01-31 17:43:41	18	Finished(18)	0	30900	30900
779	Not set	finished	plot	u_hlin730_2.2i_runs_test_ci_imsim_v23_0_1_rc1_preops-905_20220131t164723z	2022-01-31 16:47:52	7	Finished(7)	0	12	12
778	michelle gower	finished	plot	u_michellegower_tinydc2_v23_0_0_1_rc1_20220131t152904z	2022-01-31 15:30:42	7	Finished(7)	0	17	17
777	Not set	finished	plot	2.2i_all_band_1_v23_0_0_rc5_preops-973_20220127t205042z	2022-01-27 20:56:46	3	Finished(3)	0	212	212
776	Not set	finished	plot	u_yesw_pipelines_check_20220127t015309z	2022-01-27 01:53:40	5	Finished(5)	0	5	5
775	michelle gower	finished	plot	u_michellegower_tinydc2_20220127t002500z	2022-01-27 00:25:39	7	Finished(7)	0	17	17

Showing 1 to 10 of 305 entries

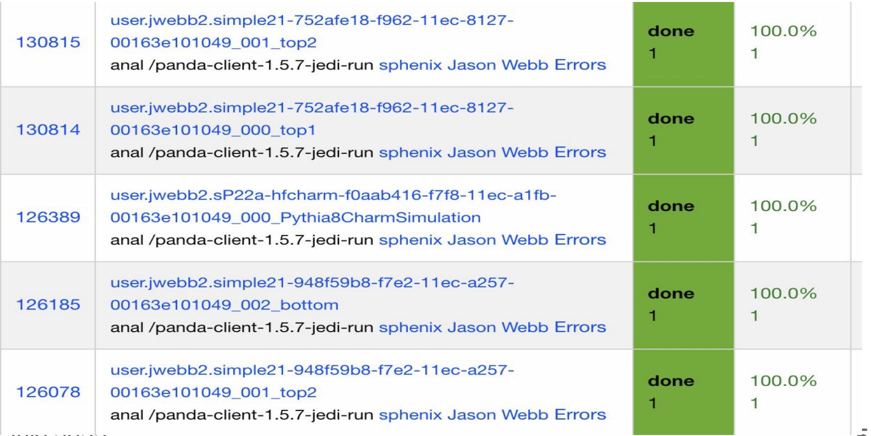
iDDS for sPHENIX

Testing PanDA and iDDS for tasks and task-chains management.

- sPHENIX adopted PanDA/iDDS at about the same time and with the same timeline as Rubin
  - iDDS for task chain management and PanDA for workload management.
  - Kuberneters (or variant, like OKD) based deployment of services
  - PostgreSQL backends (a new capability: ATLAS/CERN instances use Oracle)
- Integration tests are submitting to PanDA/iDDS.



iDDS works to manage workflows

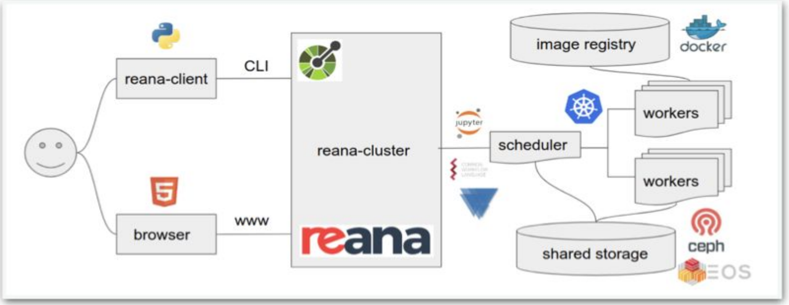


130815	user.jwebb2.simple21-752afe18-f962-11ec-8127-00163e101049_001_top2 anal /panda-client-1.5.7-jedi-run sphenix Jason Webb Errors	done 1	100.0% 1
130814	user.jwebb2.simple21-752afe18-f962-11ec-8127-00163e101049_000_top1 anal /panda-client-1.5.7-jedi-run sphenix Jason Webb Errors	done 1	100.0% 1
126389	user.jwebb2.sP22a-hfcharm-f0aab416-f7f8-11ec-a1fb-00163e101049_000_Pythia8CharmSimulation anal /panda-client-1.5.7-jedi-run sphenix Jason Webb Errors	done 1	100.0% 1
126185	user.jwebb2.simple21-948f59b8-f7e2-11ec-a257-00163e101049_002_bottom anal /panda-client-1.5.7-jedi-run sphenix Jason Webb Errors	done 1	100.0% 1
126078	user.jwebb2.simple21-948f59b8-f7e2-11ec-a257-00163e101049_001_top2 anal /panda-client-1.5.7-jedi-run sphenix Jason Webb Errors	done 1	100.0% 1

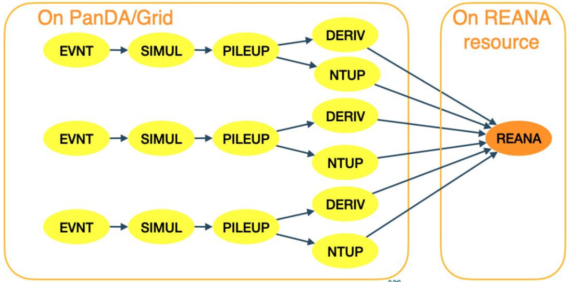
iDDS for LHC Analysis

Improving analysis efficiencies by using re-definition of the domain space for the next iteration based on results in the previous iteration. Automate multiple-step processing for LHC ATLAS processing and analysis tasks.

- Automate multiple-step task chains.
- Integration of PanDA/iDDS and REANA
- Manage and trigger tasks based on previous tasks' results.
- Adapting mono-Hbb analysis
  - Successfully tested the system, e.g., task chains, interaction with REANA (Resuable Analyses), passing secrets to the payload, etc, working on some physics issues together with physicists.



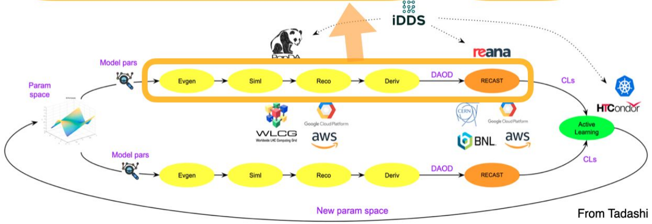
reference



R. Zhang

pchain workflow:

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

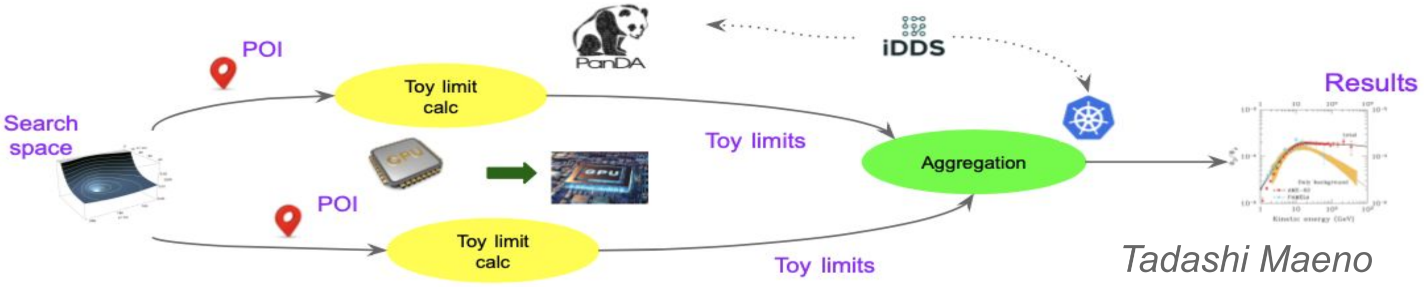


From Tadashi

iDDS for LHC Analysis (Cont)

An efficient MC toy based confidence limits workflow requires multiple steps of grid scans, where the current steps depend on the previous steps. Automate the workflow via PanDA and iDDS.

- Automate the workflow of toy limits calculation and aggregation
  - Point Of Interest (POI) generation based on search space.
  - Toy limits calculation.
  - Results aggregation.
- Schedule processing to distributed computing resources.
- A wrapper tool on top of 'phpo' (a PanDA CLI) to construct complicated execution strings on behalf of users will be developed.



Tadashi Maeno

iDDS for LHC Analysis (Cont)

More efforts to support ML for analysis will be put to simplify the ML based analysis workflow.

- Distributed ML based analysis
  - Full chain workflow integration and automation for analysis
    - Integration of application such as ServiceX for data transformation: root-based data to array-based data
    - Distribute ML tasks to CPUs/GPUs, Cluster/HPC.
  - Container + user code management
- User interface
  - Assessment and improvements of current CLI tools.
  - GUI tools to simplify the job submission and bookkeeping.
  - Jupyter tools.
- Enhancement of monitoring.
  - Realtime logging to keep user informed.
  - Tools/Monitors to filter logs.

Summary: iDDS current status

Main architecture

- iDDS database, core, REST API
- Plugins
- Agents
- Watchdogs

K8s deployment

- Containerization
- Helm deployments

Instances in production

- ATLAS, DOMA (Rubin/sPHENIX)

Supported use cases

- Data carousel
- HPO(ToyMC)
- Job level DAG(Rubin exercise)
- Task level DAG(ActiveLearning,Rubin,sPHENIX)

Experiments

- ATLAS
- Rubin Observatory
- sPHENIX

Near-term tasks

- Main Structure improvements
  - Messaging based trigger system between agents, instead of database polling based system
  - Carrier splitting (to split it into poller and manager)
- Memcache based improvement
  - Improve the database efficiency
- Client improvements
  - CLI improvements
  - Jupyter
- Improvements based on Rubin/sPHENIX requirements

Future developments

- Workflow structure improvements
  - To support data delivery as a step in the workflow.
  - IRIS-HEP ServiceX integration.
- New use cases
  - Dynamic transformation and placement on demand, for example Derivation on Demand, MxAOD on demand.
  - Fine-grained data transformation and delivery, such as Event Streaming Service

Brookhaven National Laboratory

Resources

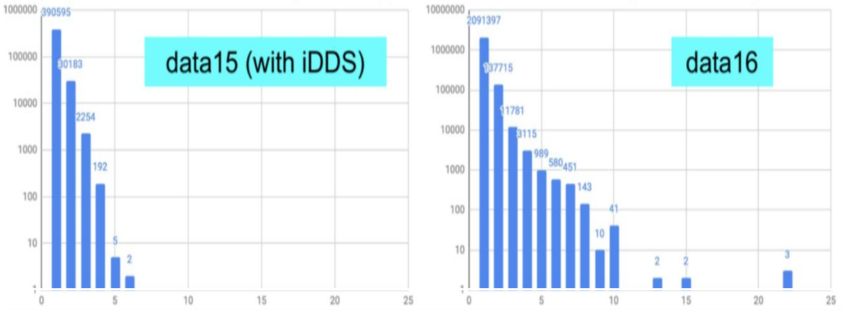
- **iDDS:**
  - HOME Page: https://iddsserver.cern.ch/website/
  - Github: https://github.com/HSF/iDDS
  - Documents: https://iris-hep.org/projects/idds.html
  - ATLAS instance monitor: https://aipanda181.cern.ch/monitor/
  - DOMA instance monitor: https://aipanda017.cern.ch/monitor/
- **PanDA**: https://panda-wms.readthedocs.io/en/latest/

Backups

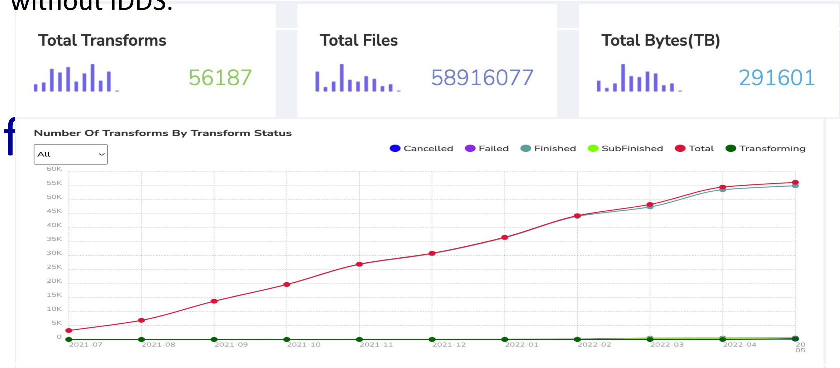
iDDS Data Carousel

iDDS Data Carousel orchestrates Rucio to collect and digest file information, and triggers JEDI/PanDA to process only prestaged files with proper granularities and grouping.

- It has been **in production** since May 2020.
- Has processed in total about 290 Pb data.
- Has reduced a lot of redundant job attempts.
  - With iDDS, the tail of jobs with a lot of attempts becomes shorter.



With iDDS for data15 reprocessing, the tail of jobs with a lot of attempts because shorter, comparing with data16 reprocessing without iDDS.



Total Transforms 56187

Total Files 58916077

Total Bytes(TB) 291601

Number Of Transforms By Transform Status

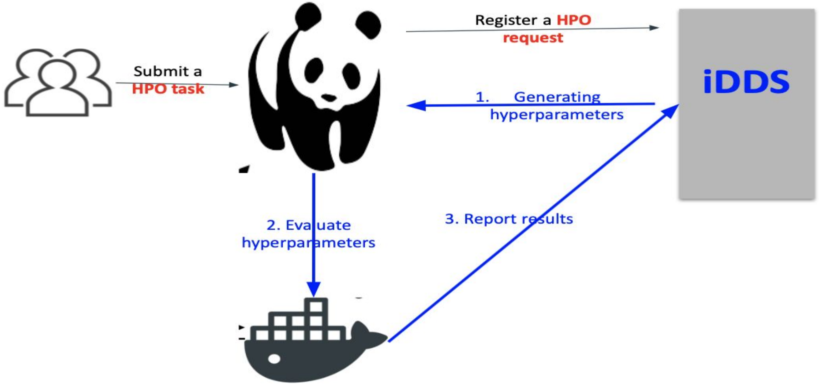
All

ATLAS data carousel with iDDS has been in production since May 2020. From July 2021(older data are archived), it has processed 56K transforms, 290 Pb data.

iDDS HyperParameter Optimization (HPO)

iDDS HPO provides a fully-automated platform for hyperparameter optimization on top of geographically distributed CPU/GPU resources on the Grid, HPC and Clouds.

- ❖ A group of optimized hyperparameters can greatly improve the physics analysis performance. A lot of LHC analyses are using HPO tc enhance the performance.
- ❖ iDDS HPO automates hyperparameter generation and evaluation with many iterations: new hyperparameters are generated automatically from previous evaluation results.
- ❖ iDDS HPO distributes ML tasks to CPU/GPUs on potentially geographically distributed resources.
- ❖ iDDS HPO has been **used by ATLAS ML users**, not specific to ATLAS.
- ❖ Different use cases are using the HPO framework to automate distributed tasks.
  - FastCaloGAN
  - Monte Carlo toy based confidence limits estimation (requiring multiple steps of grid scans, where current steps depend on previous steps)
  - AtlFast3 fast simulations

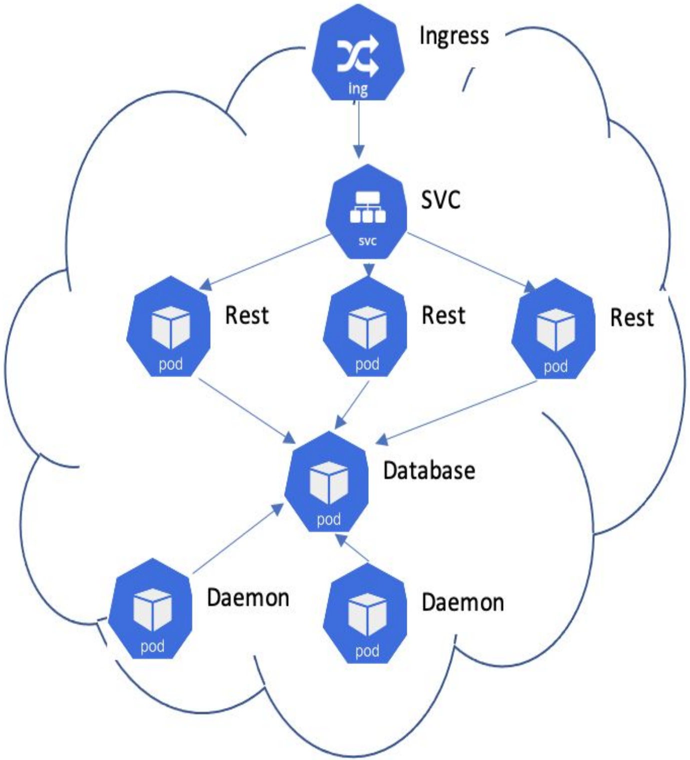


- ❖ **A HPO task should include two parts**
  - ➢ Hyperparameter generating:
    - ■ Option 1: define search space with predefined methods
    - ■ Option 2: develop user container
  - ➢ Evaluation
    - ■ User ML training/learning process

iDDS K8S deployment

Deploy iDDS on Kubernetes, to simplify the deployment and management

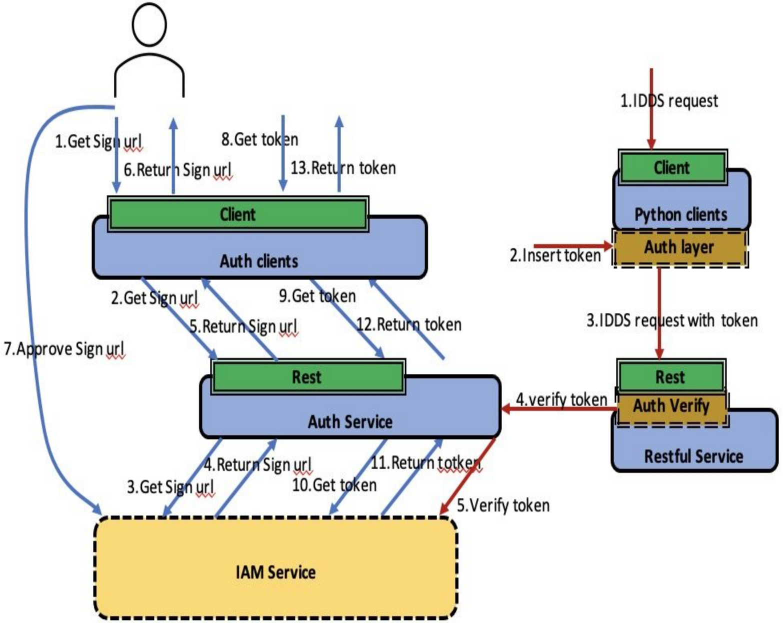
- **Containerization** of iDDS components:
  - Rest: Restful interface
  - Database: Postgresql database.
  - Daemon: Agents to process requests.
- **Deployment with helm on k8s.**
- **Integration with PanDA on k8s to provide workload management services for different experiments.**
- **Deploying for Rubin and sPHENIX**



iDDS JWT Authorization

JWT (JSON Web Token) based authorization

- **Token generation**
  - To generate token with Indigo IAM service.
- **Token based authorization on Rest service.**
  - Token validation and authorization based or token.
- **Deployed for Rubin**



iDDS for LHC Analysis (Cont)

More efforts to support ML for analysis will be put to simplify the workflow.

- User code management
  - For ML based analysis, various software can be used and the user codes can be very different. Currently all codes need to be in a container which is difficult for users.
  - Solutions:
    - Predefined containers to support different ML tools.
    - User specific codes based on the containers.
- User interface
  - Assessment and improvements of current CLI tools.
  - GUI tools to simplify the job submission and bookkeeping.
  - Jupyter tools.
- Enhancement of monitoring.
  - Realtime logging to keep user informed.
  - Tools/Monitors to filter logs.