ATLAS EXPERIMENT

iris hep

THE UNIVERSITY of WISCONSIN MADISON

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

Wen Guan, Tadashi Maeno
Brian Bockelman, Torre Wenaus
Fernando Barreiro, Fahui Lin, Rui Zhang, Misha Borodin, Paul Nilsson

2021 Meeting of the Division of Particles and Fields of

the American Physical Society (DPF21)

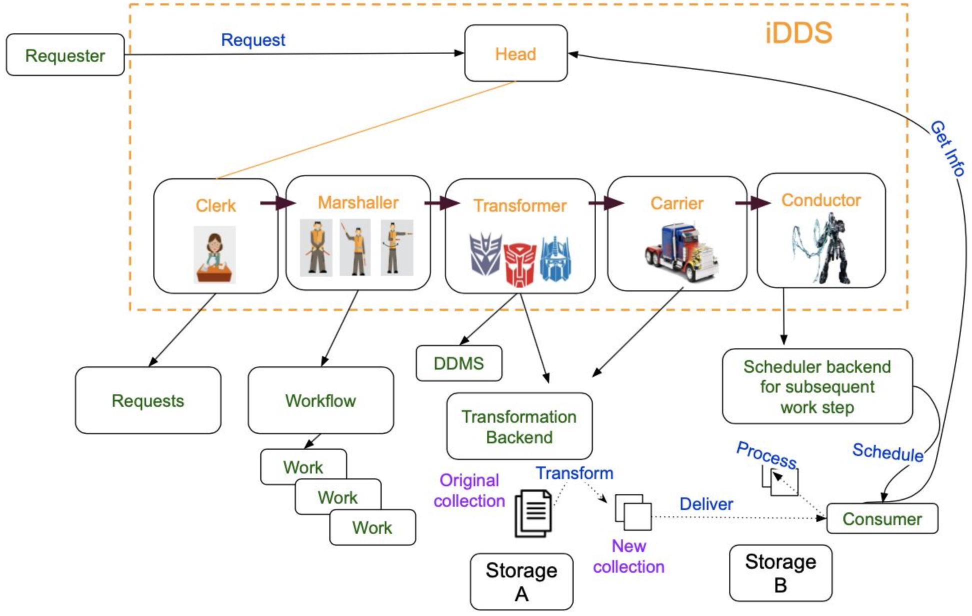
July 12, 2021

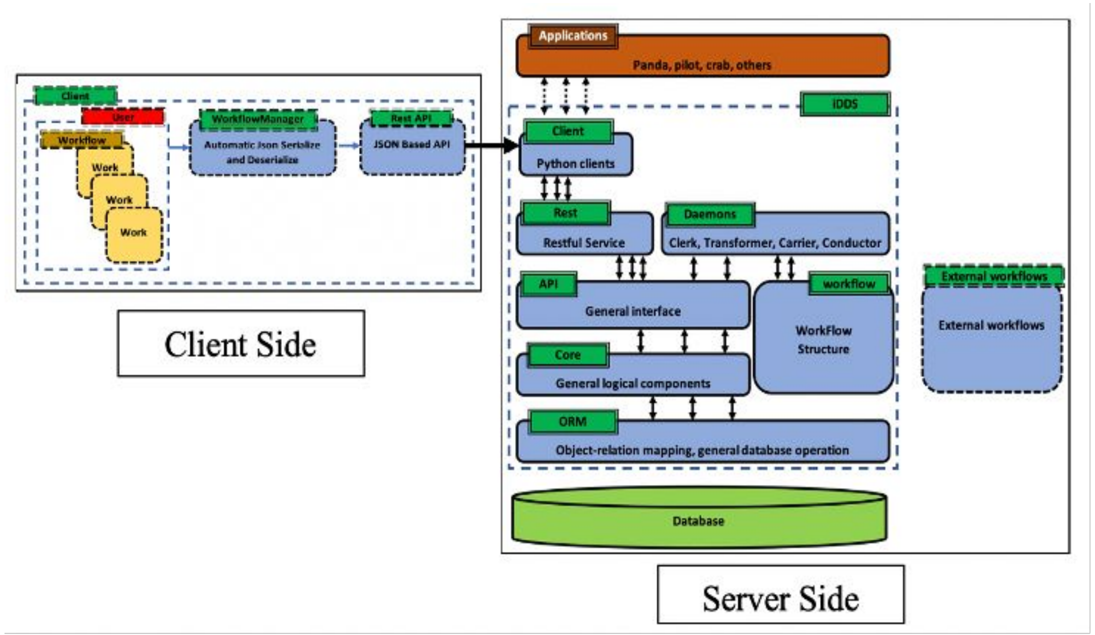
iDDS (a joint project with IRIS-HEP and ATLAS)

- **A general service to transform and deliver needed data to consumers, mainly to deliver offline data to users.**
  - **Orchestration of workload manager and data manager with generalized workflows, to improve the workflow between the workload manager and data manager.**
    - **Experiment agnostic based on the generalization**
    - **Extraction and abstraction of functions for orchestration**
    - **Maintainability and extensibility with plugin architecture**
    - **Improvement on the granularities**
      - **Fine grained tape carousel**
      - **Fine grained data transformation**
  - **Flexibility to support new workflows**
    - **Automate multiple-step tasks to reduce manual operations**
      - **DAG (Directed Acyclic Graph): Active Learning, ...**
      - **Hyperparameter tuning**
    - **Reduce storage usage and reduce data size to be transferred**
      - **Remote data transformation/reduction**
      - **On-demand production of analysis format data**
    - **Others**

iDDS Architecture

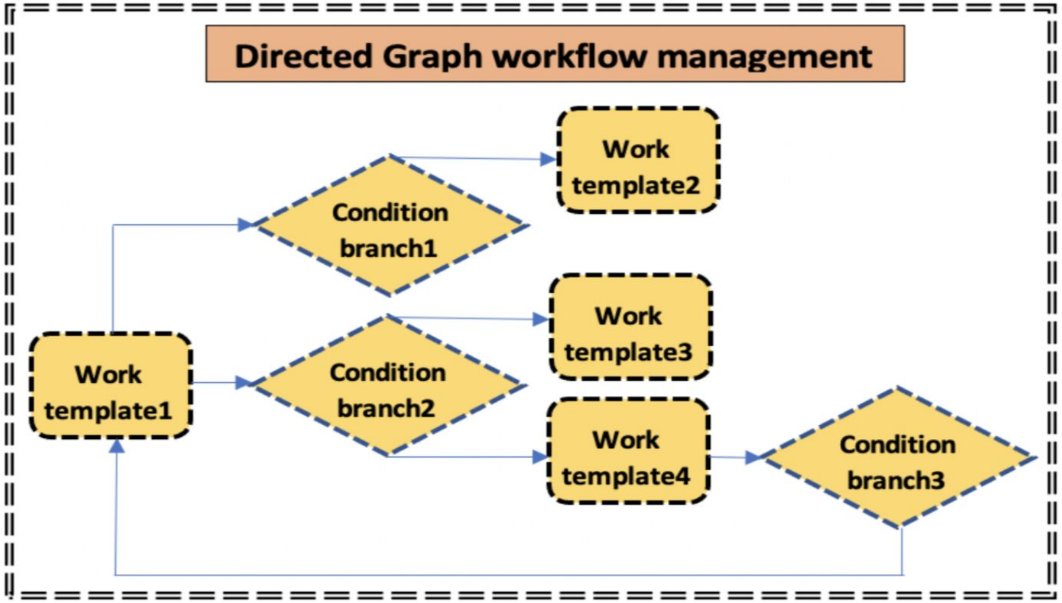
- **Client/Server**
  - **Head service to receive requests from users.**
  - **Agents to process requests**
    - **Employ different backends for transformation: PanDA workload manager, Rucio data manager, Condor cluster system and so on.**
- **Layered architecture of the server**
  - **Every layer abstracts a group of functions, hiding the complexity of different logics on different layers.**



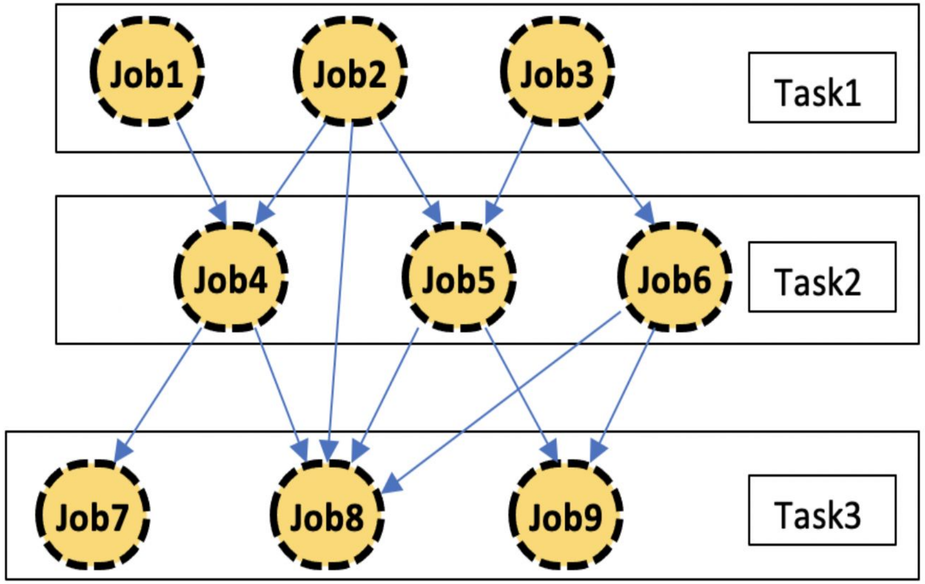


iDDS DAG

- **DAG (Directed Acyclic Graph) workflow management**
  - **Automate workflows with multiple-step processing.**
  - **Reduce manual operations.**
- **Two types of DAG in iDDS**
  - **Task Level DAG**
    - **Relation description is among tasks.**
    - **When there are new outputs in a task, new jobs are generated for the dependent tasks.**
    - **When a task is terminated, dependent tasks are triggered.**
  - **Job Level DAG**
    - **Relation description is among pre-defined jobs.**
    - **Job grouping based on workflow management system.**
    - **When a job is terminated, dependent jobs are triggered.**



Task Level DAG



Job Level DAG

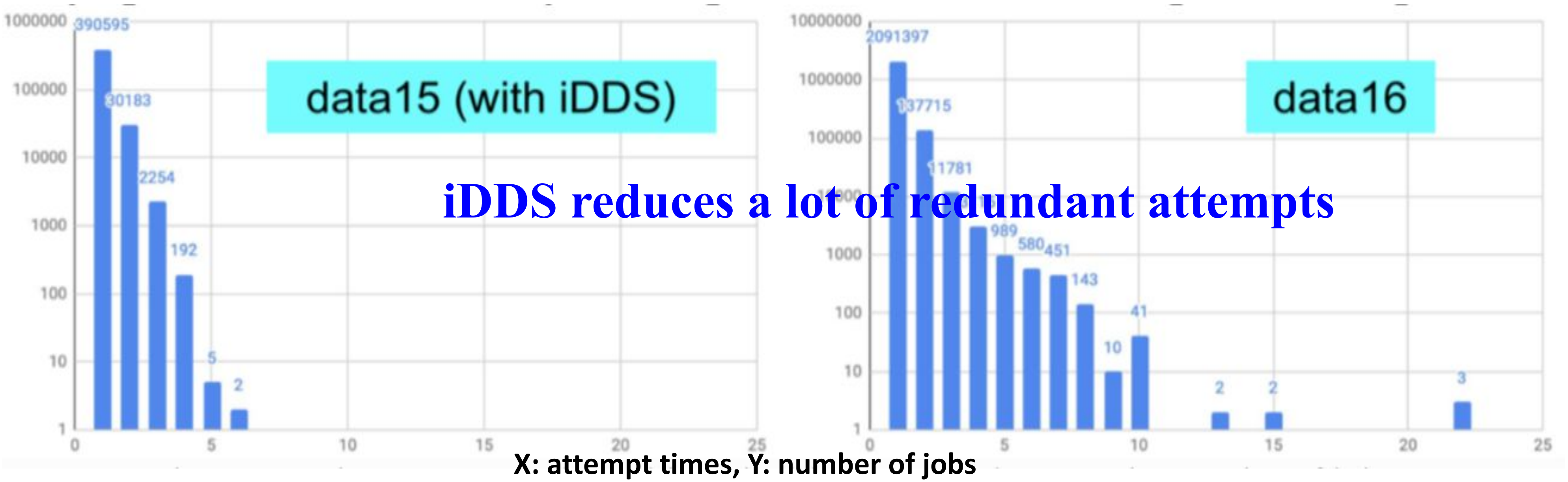
iDDS available use cases

- Data Carousel
- HPO(Hyper Parameter Optimization)
- Job Level DAG (LSST exercise)
- Task Level DAG (Active Learning)

iDDS Data Carousel

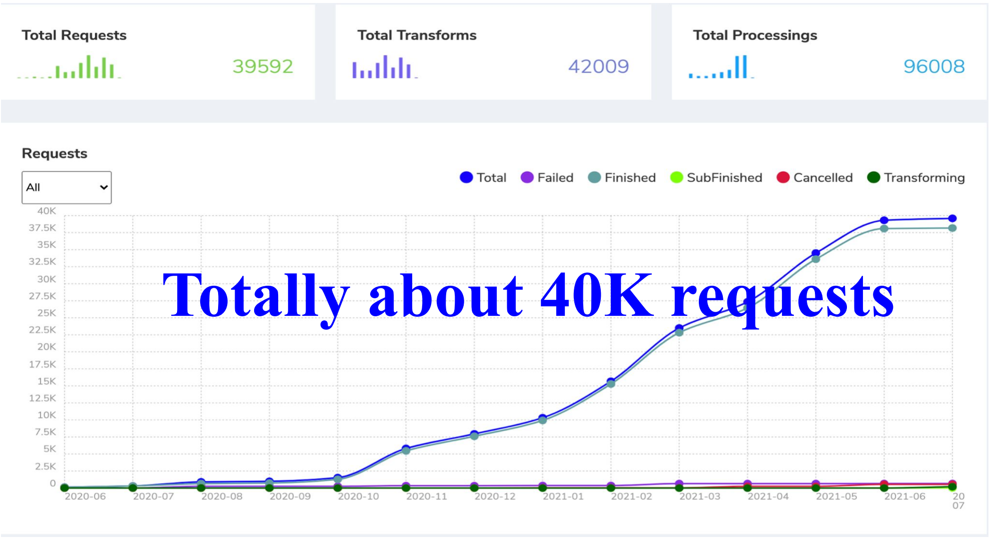
- Conventional data carousel issues:
  - Data management is based on dataset level and the task can be released when 90% of the data is available (It can take very long time to stage in the tail of last few files. If waiting until all files are staged, the waiting time will be too long).
  - Many attempts because jobs may be released before the inputs staged
  - Delayed task release to avoid too many attempts
    - Large buffer to keep big dataset for long time
- iDDS orchestrates Rucio data manager and PanDA workload manager
  - collect and digest file information in file level.
  - trigger JEDI/PanDA workload manager to only release jobs with prestaged files
  - proper granularities and grouping.
- In ATLAS production since June 2020
- Has processed in total about 60 PB data.
- Reduced a lot of redundant job attempts.
  - With iDDS, the tail of jobs with a lot of attempts becomes shorter.

iDDS Data Carousel



iDDS reduces a lot of redundant attempts

X: attempt times, Y: number of jobs



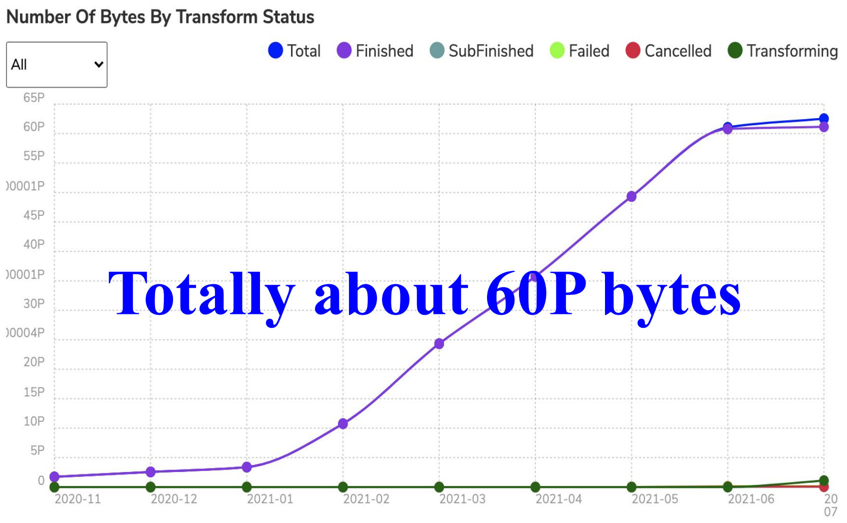
Total Requests 39592

Total Transforms 42009

Total Processings 96008

Requests

Totally about 40K requests



Number Of Bytes By Transform Status

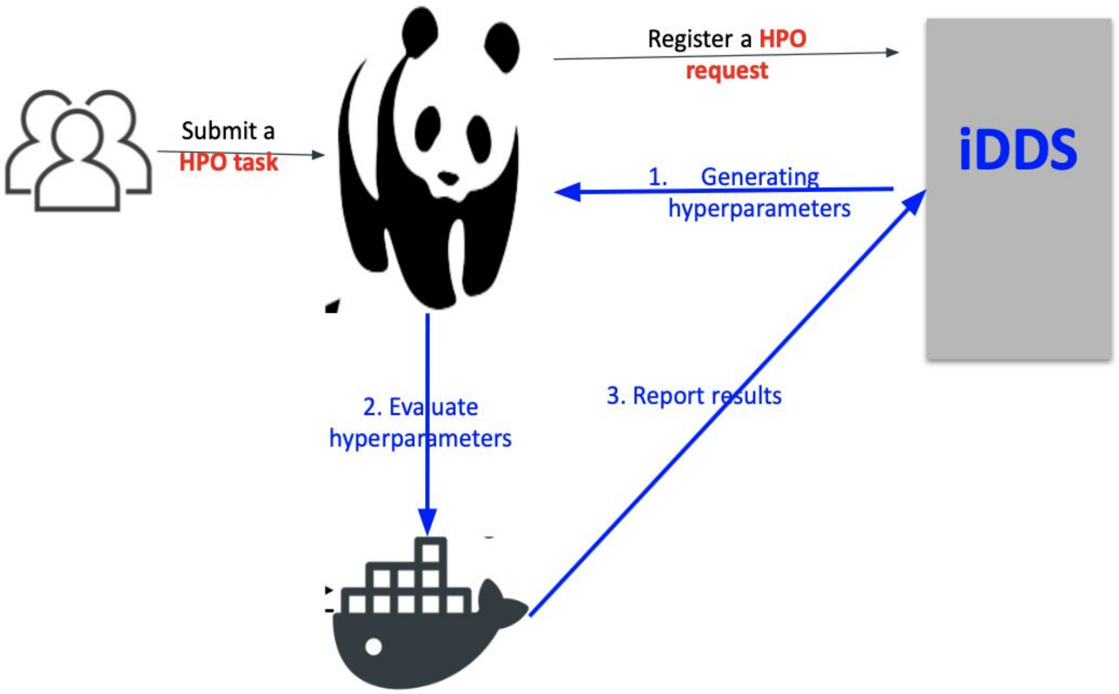
Totally about 60P bytes

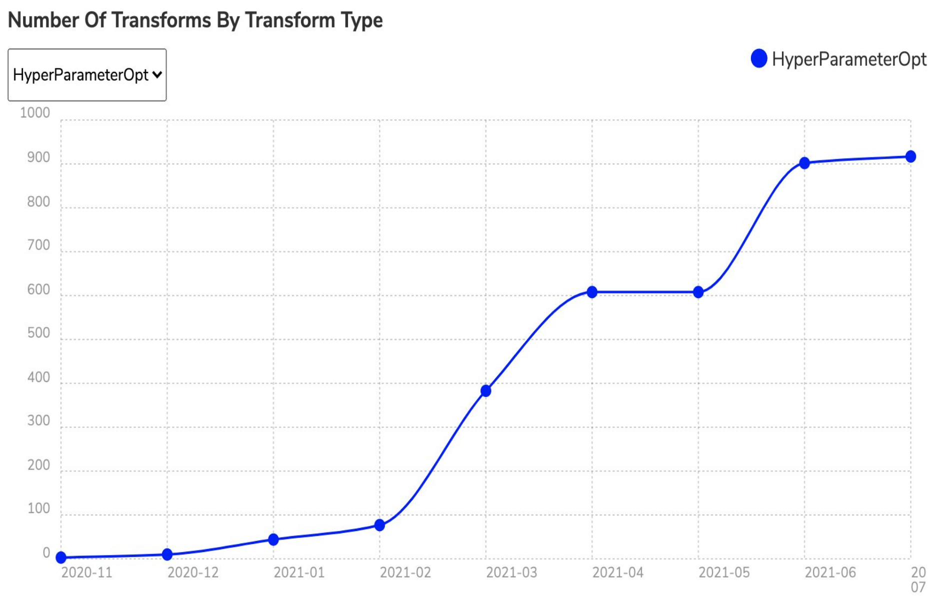
iDDS HPO(Hyper Parameter Optimization)

- **Hyper Parameter**
  - A parameter to control the learning/training process in machine learning.
- **Hyper Parameter Optimization**
  - To find a set of optimal parameters for a learning algorithm.
- **Ways to find hyper parameters:**
  - Randomly/Gridly generate a lot of hyper parameters in one round to evaluate and then select the one producing the best results -- not efficient in fine granularity
  - Searching hyperparameters based on previous hyper parameter results using some algorithms -- multiple rounds, require manual operations for every round on distributed workload system.
- **iDDS HPO**
  - To provide a fully-automated platform for hyper parameter optimization on top of geographically distributed GPU resources on the grid, HPC, and clouds.
  - Automatically generate hyper parameters based on previous results.
  - Distribute hyper parameters to Grid resources to be evaluated and collect results for the next round of hyper parameters generation.
  - Automate the whole workflow with one submission, reduced a lot of manual operations.
  - Support multiple users and multiple tasks with one server.

iDDS HPO(Hyper Parameter Optimization)

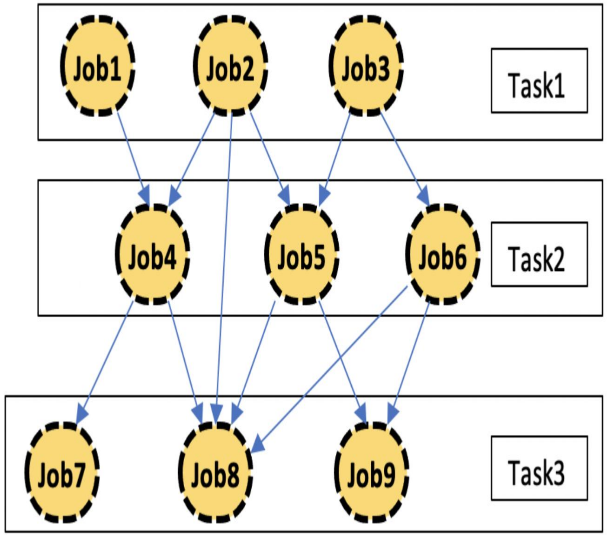
- **iDDS HPO status**
  - Used by ATLAS machine learning users, not specific to ATLAS.
- **The usage is increasing**
  - Different use cases are using the HPO framework to automate distributed tasks.
    - FastCaloGAN fast simulation.
    - ToyMC simulation.
- **More Materials**:
  - IMLHPO20201022.pdf, idds_ml_20200707.pdf

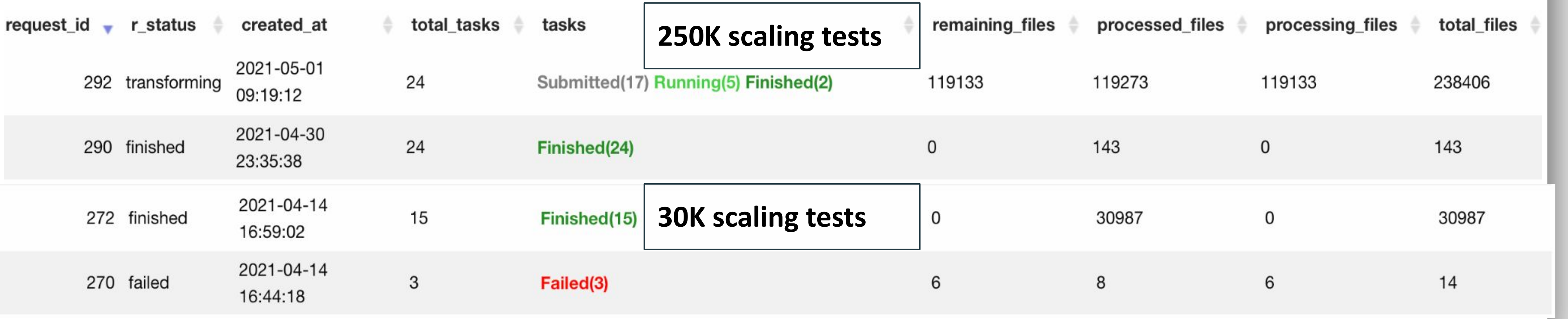




iDDS Rubin Observatory (LSST)

- **LSST exercise**
  - Employ iDDS DAG to automate multiple-step processings with one request.
  - Use the experiment-agnostic DOMA PanDA instance to schedule jobs to Google Cloud.
  - Job Level DAG: cascade of chains for multiple-step processing.
  - iDDS manages the dependencies and triggers to release jobs incrementally when all dependencies are ready, instead of blocking tasks until all previous tasks finish, to avoid long waiting.
  - Scaling tested with 250K DAG jobs
  - Tutorials to Users.





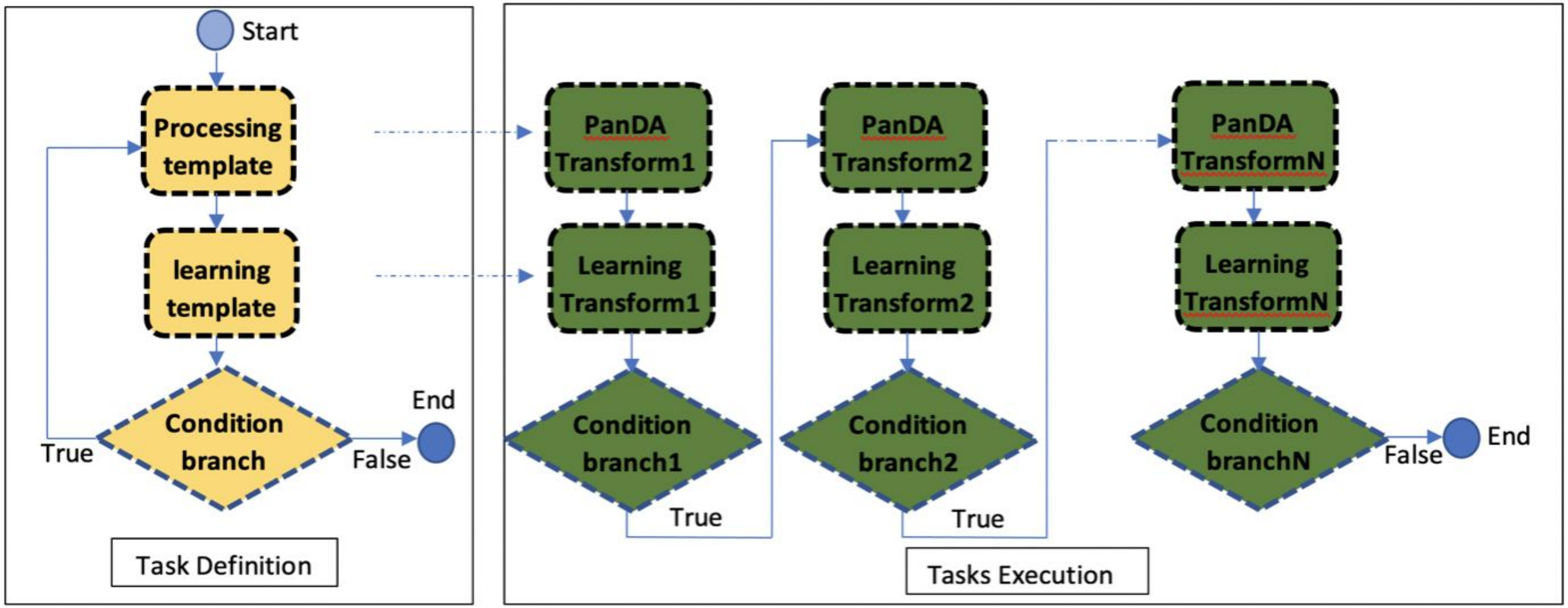
250K scaling tests

request_id	r_status	created_at	total_tasks	tasks	remaining_files	processed_files	processing_files	total_files
292	transforming	2021-05-01 09:19:12	24	Submitted(17) Running(5) Finished(2)	119133	119273	119133	238406
290	finished	2021-04-30 23:35:38	24	Finished(24)	0	143	0	143
272	finished	2021-04-14 16:59:02	15	Finished(15)	0	30987	0	30987
270	failed	2021-04-14 16:44:18	3	Failed(3)	6	8	6	14

30K scaling tests

iDDS ActiveLearning

- ## ActiveLearning
  - Automate multiple-step processings based on some decision-making of previous tasks' results.
  - A simple Task Level DAG use case to chain processing and learning tasks.
  - To define the subsequent processing task based on the decision making in the learning task which analyzes the results of the previous processing task.
  - Task templates to generate concrete tasks, and condition branches to control the workflow.
  - Under integration with PanDA workload manager.



Summary: iDDS Current Status

Main architecture

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

Documents & monitors

- Home page: https://idds.cern.ch
- Codes: https://github.com/HSF/iDDS
- Documents: https://idds.readthedocs.io
- ATLAS monitor: https://bigpanda.cern.ch/idds/
- Different monitors are being enriched.

Instances in production

- **ATLAS, DOMA**

Available use cases

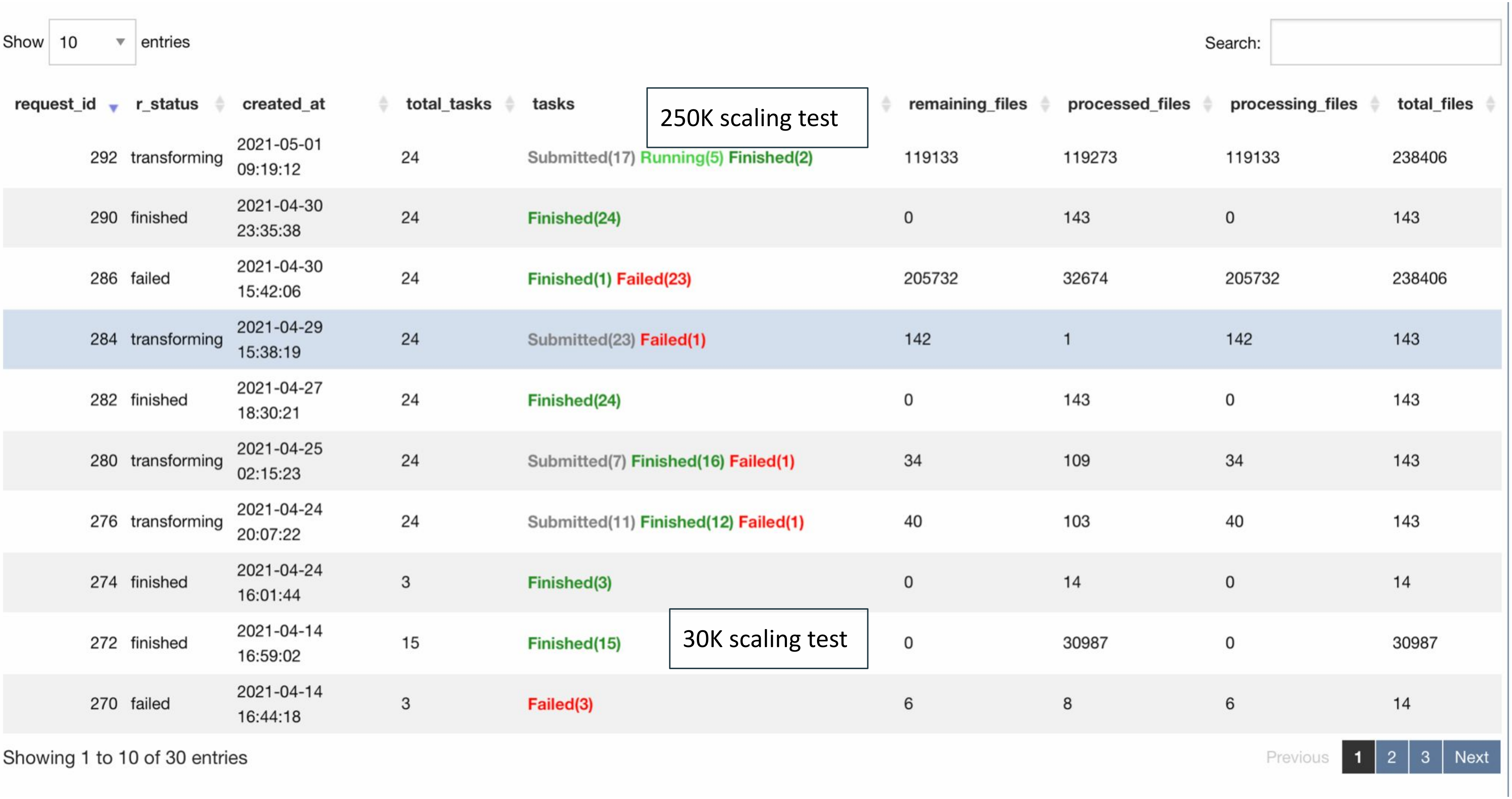
- **Data carousel**
- **HPO(ToyMC)**
- **Job level DAG(LSST exercise)**
- **Task level DAG(ActiveLearning)**

Future developments

- Main Structure improvements
- Monitor and clients improvements
- 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

Backups

iDDS Rubin scaling test status



250K scaling test

30K scaling test

request_id	r_status	created_at	total_tasks	tasks	remaining_files	processed_files	processing_files	total_files
292	transforming	2021-05-01 09:19:12	24	Submitted(17) Running(5) Finished(2)	119133	119273	119133	238406
290	finished	2021-04-30 23:35:38	24	Finished(24)	0	143	0	143
286	failed	2021-04-30 15:42:06	24	Finished(1) Failed(23)	205732	32674	205732	238406
284	transforming	2021-04-29 15:38:19	24	Submitted(23) Failed(1)	142	1	142	143
282	finished	2021-04-27 18:30:21	24	Finished(24)	0	143	0	143
280	transforming	2021-04-25 02:15:23	24	Submitted(7) Finished(16) Failed(1)	34	109	34	143
276	transforming	2021-04-24 20:07:22	24	Submitted(11) Finished(12) Failed(1)	40	103	40	143
274	finished	2021-04-24 16:01:44	3	Finished(3)	0	14	0	14
272	finished	2021-04-14 16:59:02	15	Finished(15)	0	30987	0	30987
270	failed	2021-04-14 16:44:18	3	Failed(3)	6	8	6	14

Showing 1 to 10 of 30 entries