PanDA Evolution for ATLAS and Beyond

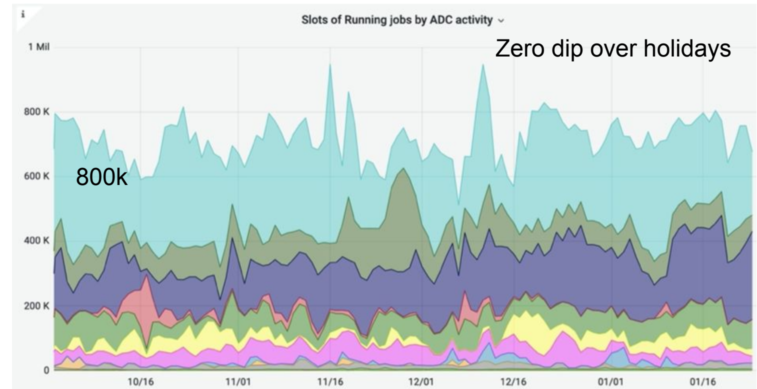
Tadashi Maeno (BNL)
on behalf of PanDA/iDDS teams

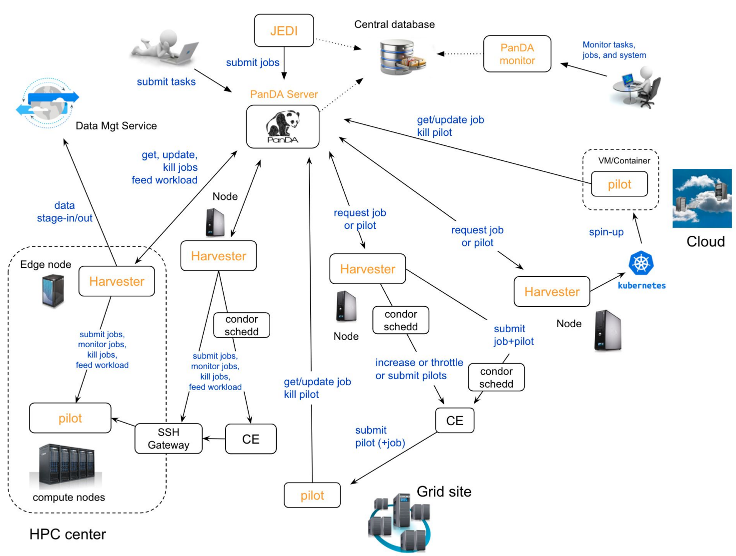
3rd May 2022
Software & Computing Round Table

@BrookhavenLab

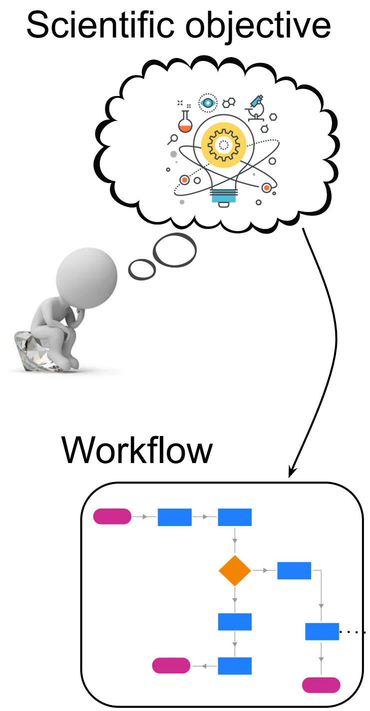
Production and Distributed Analysis System

- PanDA: the workload management system
  - Manages 24x365 processing on ~800k concurrent cores globally for ATLAS, all workloads from evgen to analysis, all resource types, ~150 computing centers, ~1500 users, ~300M jobs/yr, in tandem with Rucio for data management
  - Smooth horizontal scaling
  - Easy to use submit client, python script or Jupyter notebook
  - Highly developed monitoring and historical data reporting
  - New(ish): PostgreSQL support, Kubernetes based services, BNL-led ATLAS-Google project, Jupyter support, OIDC/OAuth2 authentication
  - PanDA documentation
- iDDS: intelligent data delivery system
  - A joint project between ATLAS and IRIS-HEP
  - Supports arbitrarily complex fine-grained workflows defined via DAGs or workflow description languages
  - iDDS documentation

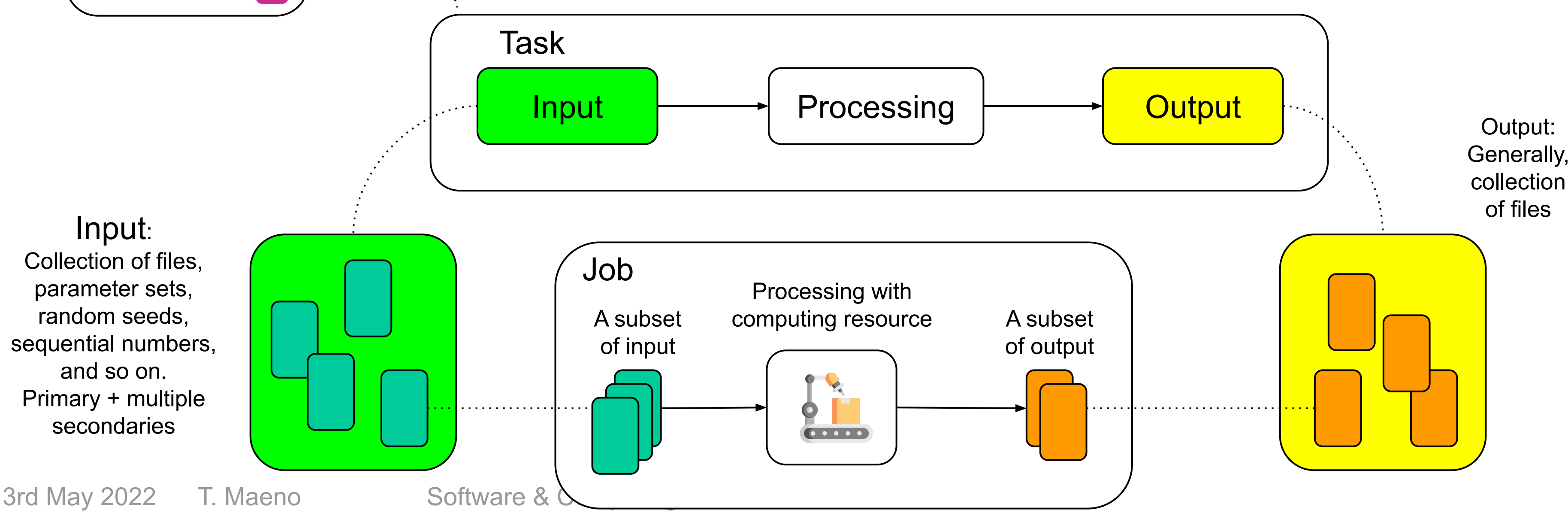




Hierarchical Workload Entities in PanDA



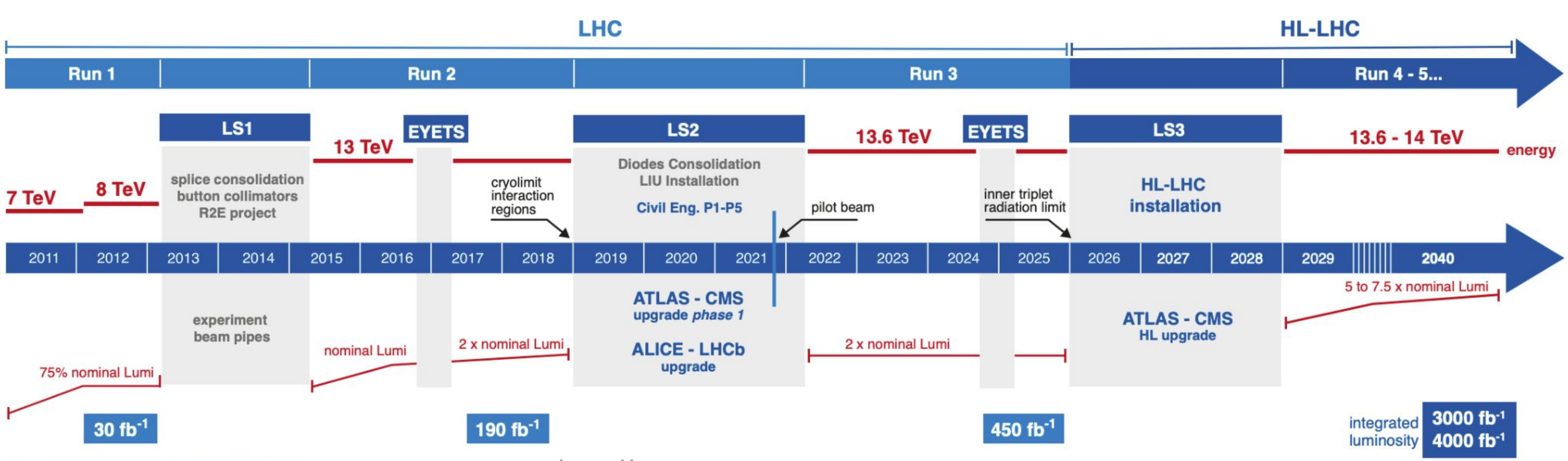
- A workflow: a top-level entity of workload to accomplish a scientific objective
- A task: a step in a workflow
  - A group of tasks = a workflow
  - A task processes input to produce output
- A job: an artificial partition in a task
  - A group of jobs = a task
  - A job processes a subset of input on a computing resource to produce a subset of output
  - On the traditional grid, a job ultimately corresponds to a batch job in a batch system
- An event (optional): a minimum unit in each constituent of input with the finest processing granularity



Evolutions for ATLAS

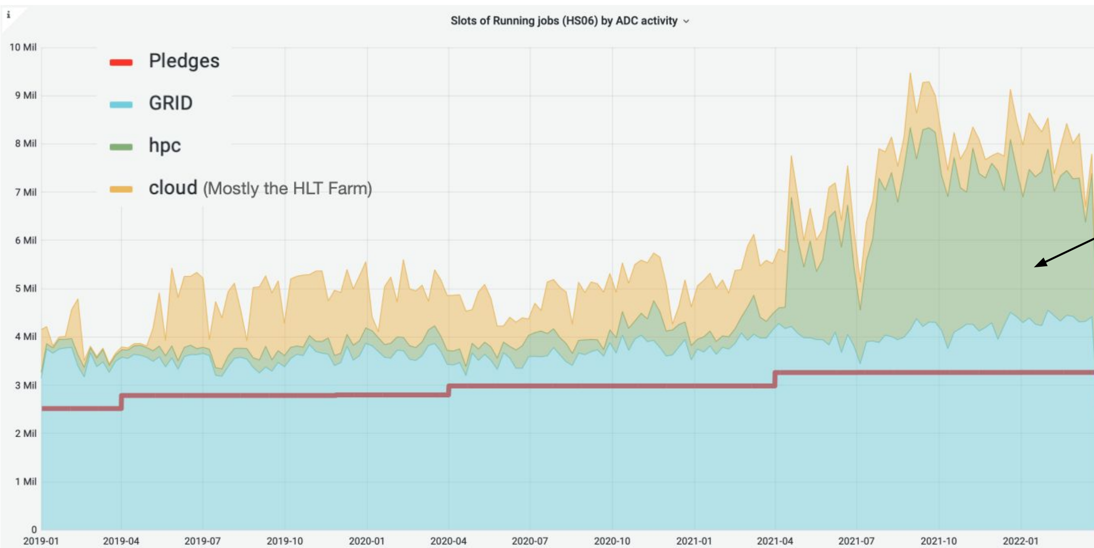
PanDA@ATLAS

- PanDA was born in 2005 and has been working for ATLAS since then
- Open organizational culture in ATLAS encouraging innovative evolutions in PanDA, while steadily running for high volume Monte Carlo production and data reprocessing, and making large-scale computing resources available for individual analysis
- Integration of heterogeneous and global resources (the grid, HLT farm, HPCs, commercial clouds, volunteer computing, GPUs), with a capability to manage workloads with various granualities
- Great opportunity in Long Shutdown 2 (LS2, 2018-2021) to develop a lot of R&D activities for Run 3 and HL-LHC
  - Automation
  - Optimal resource usage
  - New workflows
- Will highlights major evolutions in next slides



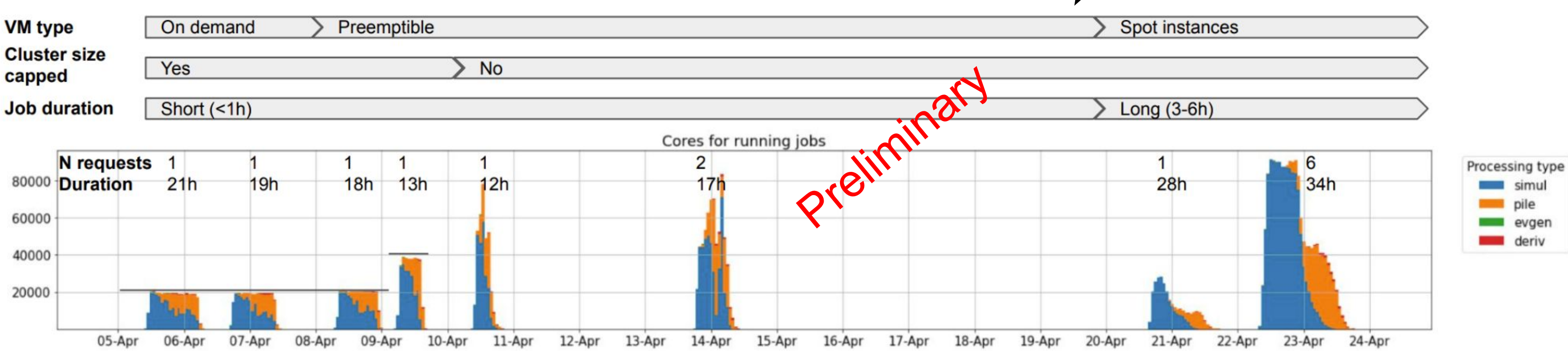
Integration of Emerging Resources

- ➢ Various plugins to access resources
  - HTCondor, Advanced Resource Connector middleware, low-level batch system API, k8s, RPC over SSH
- ➢ Flexible deployment model for the resource manager
  - Central, regional, or running on edge services



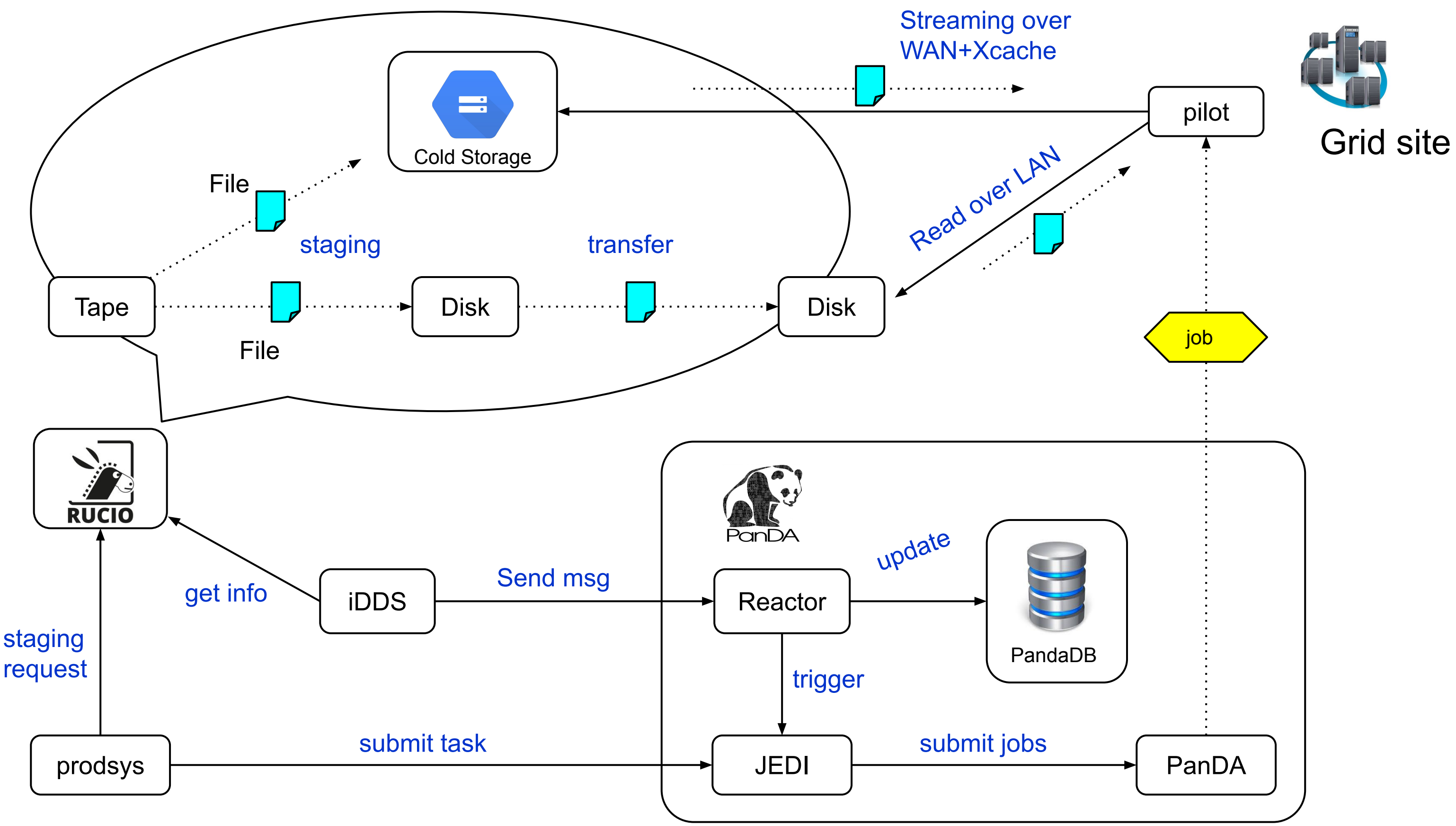
Significant contributions from EuroHPC systems

Bursty MC production exercises on Google Compute Platform with various VM types



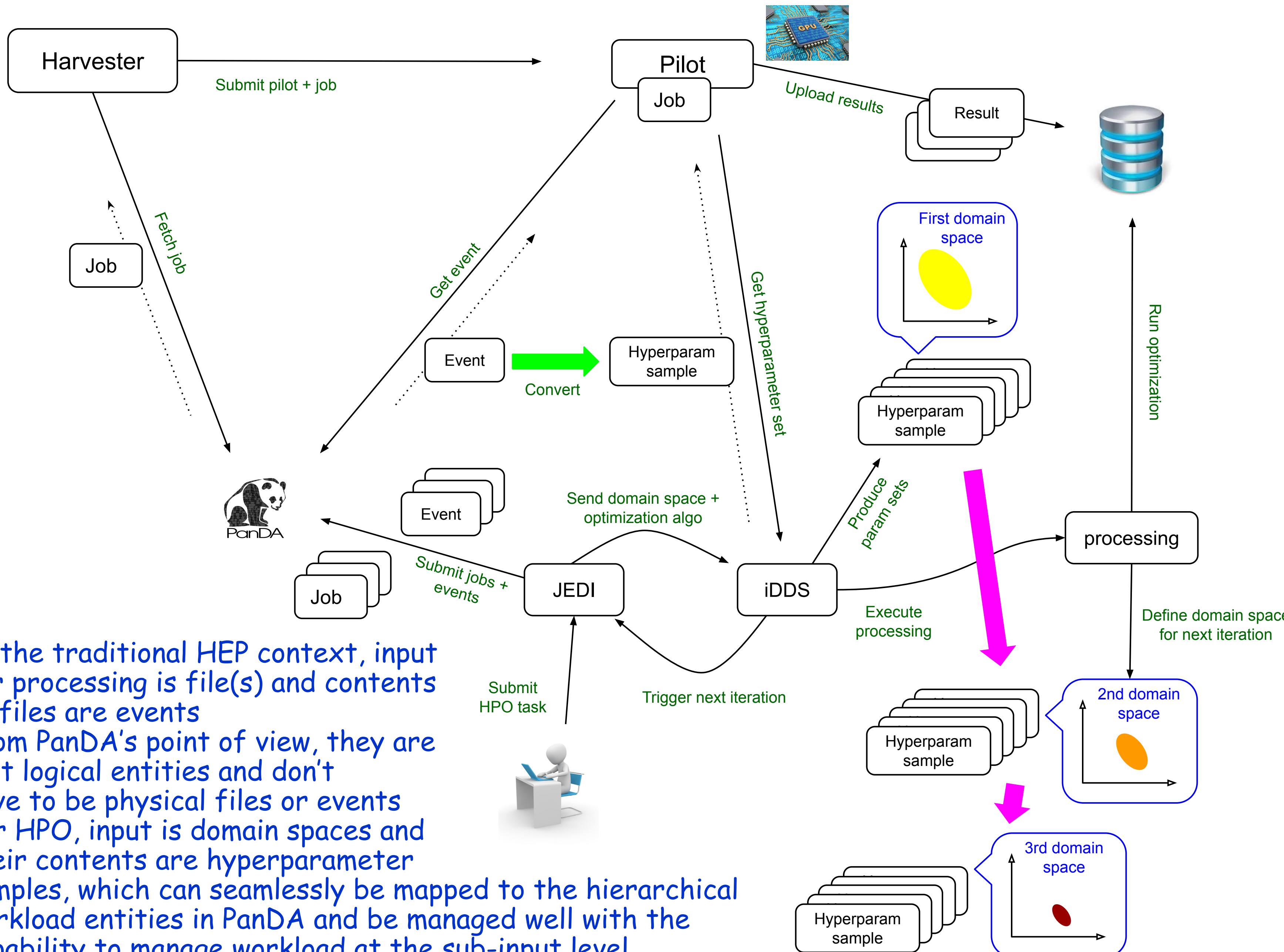
Data Carousel

- Fine-grained disk-efficient tape staging
  - On-demand disk replicas and their aggressive deletion
- The first usecase of iDDS in production, working as a high-level service to orchestrate PanDA and Rucio
- Mitigating the disk problem that is the most crucial in ATLAS



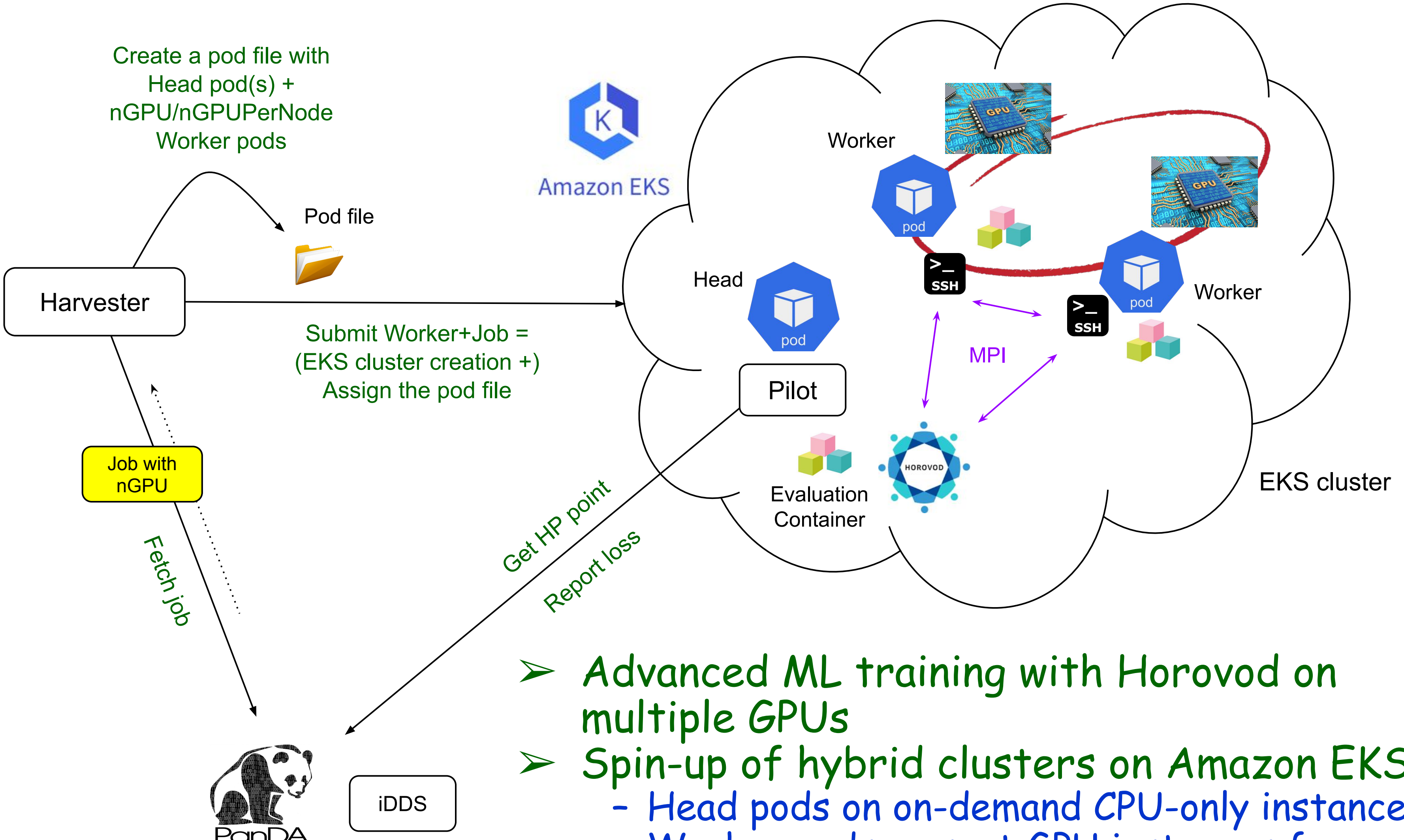
New and Evolving Workflows,
Currently applied to, but not limited to, ATLAS

Hyperparameter Optimization (HPO) Service



- In the traditional HEP context, input for processing is file(s) and contents of files are events
- From PanDA's point of view, they are just logical entities and don't have to be physical files or events
- For HPO, input is domain spaces and their contents are hyperparameter samples, which can seamlessly be mapped to the hierarchical workload entities in PanDA and be managed well with the capability to manage workload at the sub-input level

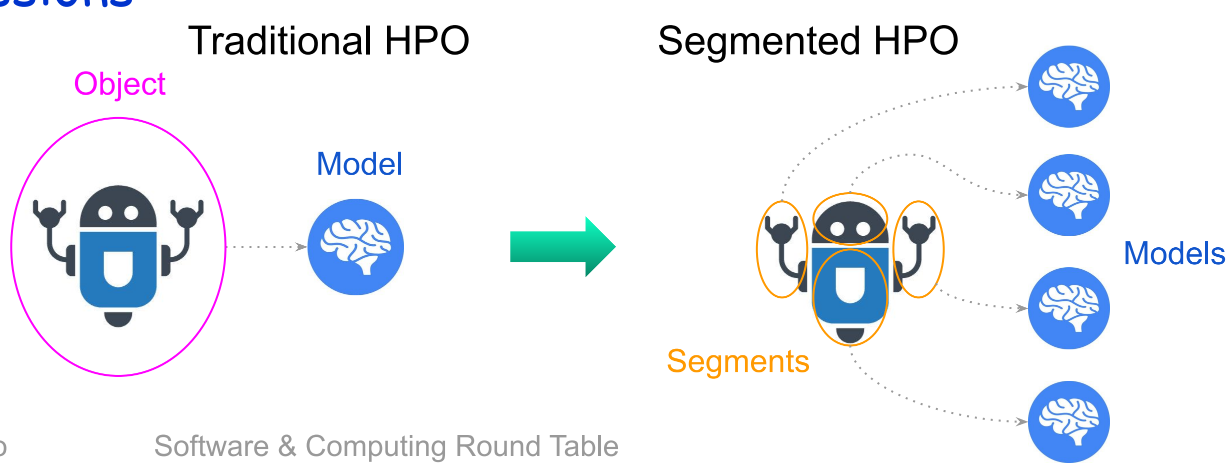
Elastic Distributed Training on Amazon EKS



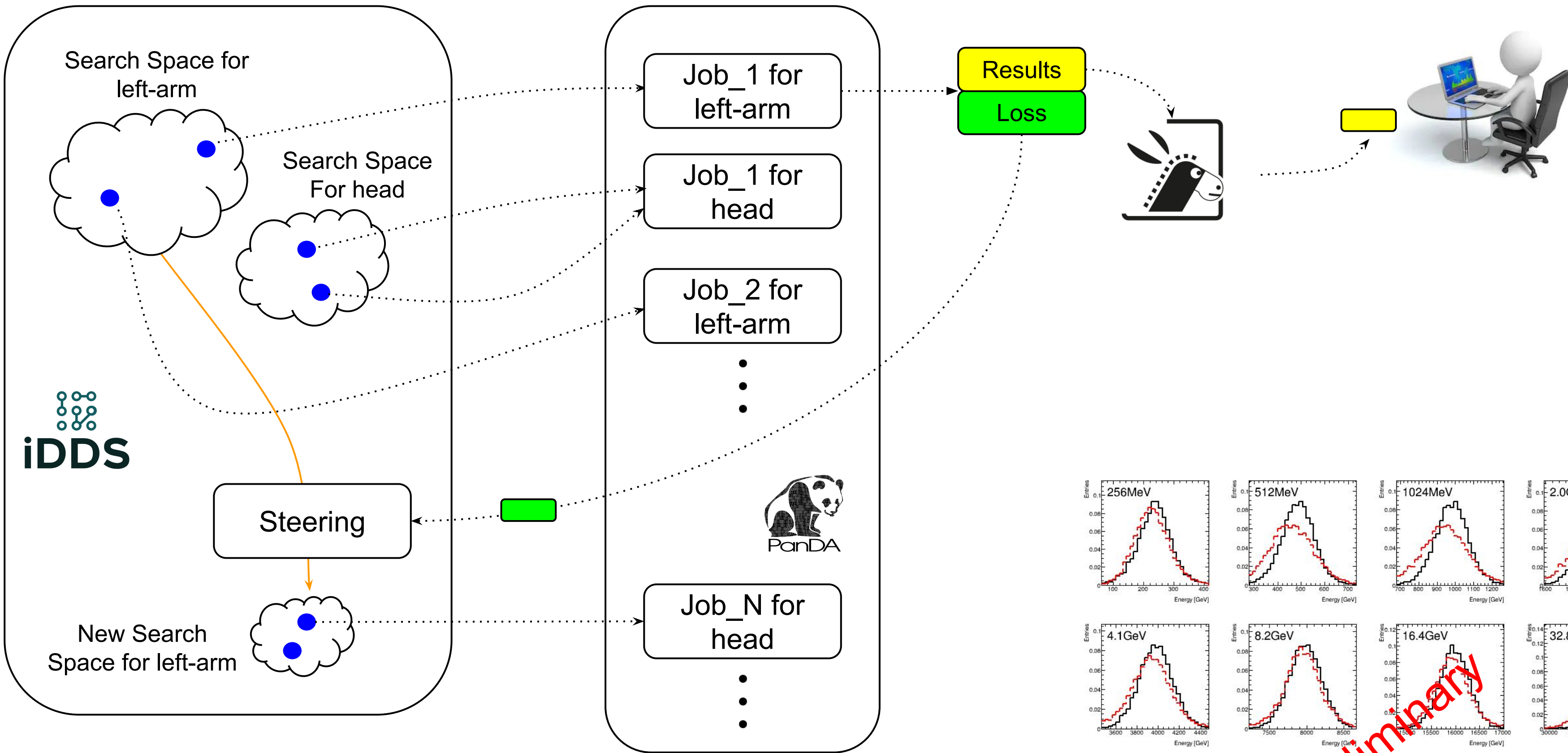
- ➢ Advanced ML training with Horovod on multiple GPUs
- ➢ Spin-up of hybrid clusters on Amazon EKS
  - – Head pods on on-demand CPU-only instances
  - – Worker pods on spot GPU instances for cost-saving + dynamic scaling-up/down of Horovod
- ➢ Capability for complex resource provisioning

Segmented HPO 1/2

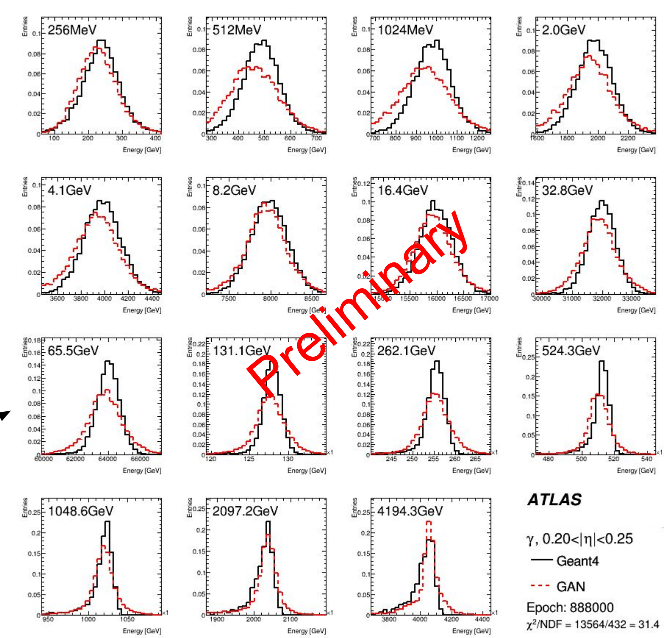
- A traditional HPO with a single ML model for a monolithic object, while in some cases objects can logically be segmented
  - Depending on nature of the object and use-case
- HPO service for a segmented object
  - One ML model per sub-object (per segment)
  - A single HPO session to optimize all ML models in one-go
  - Smaller training dataset for each model → Fast turnaround
  - Concurrent training of multiple models → More distributed workload
- FastCaloGAN : 300 GANs = 100 pT-η slices x 3 PIDs
  - 100 GPU-days to train 300 GANs → Better to be parallelized
  - GAN = Segment
  - One segmented HPO session instead of 300 individual sessions



Segmented HPO 2/2

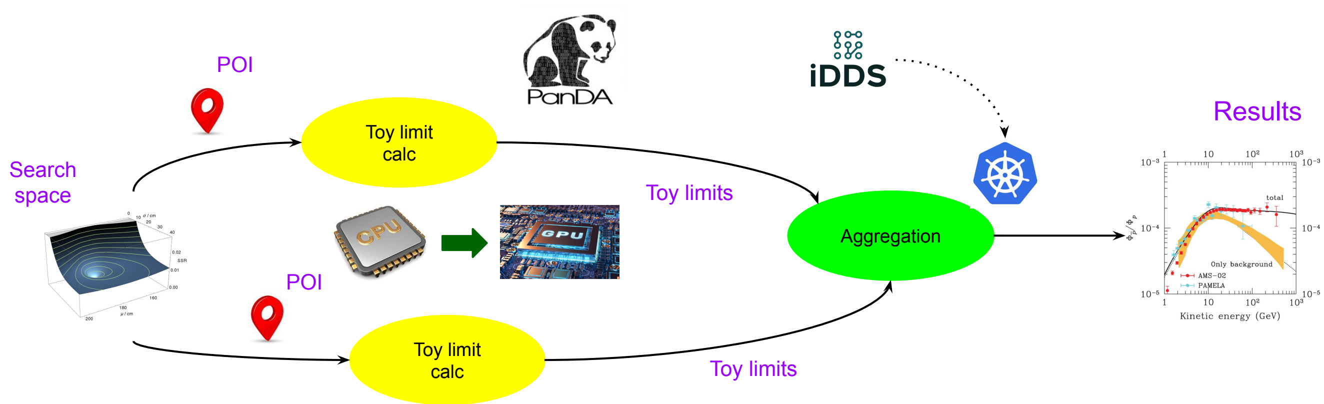


Being actively used to converge FastCaloGAN parameters and full detector training for central MC production in Run 3



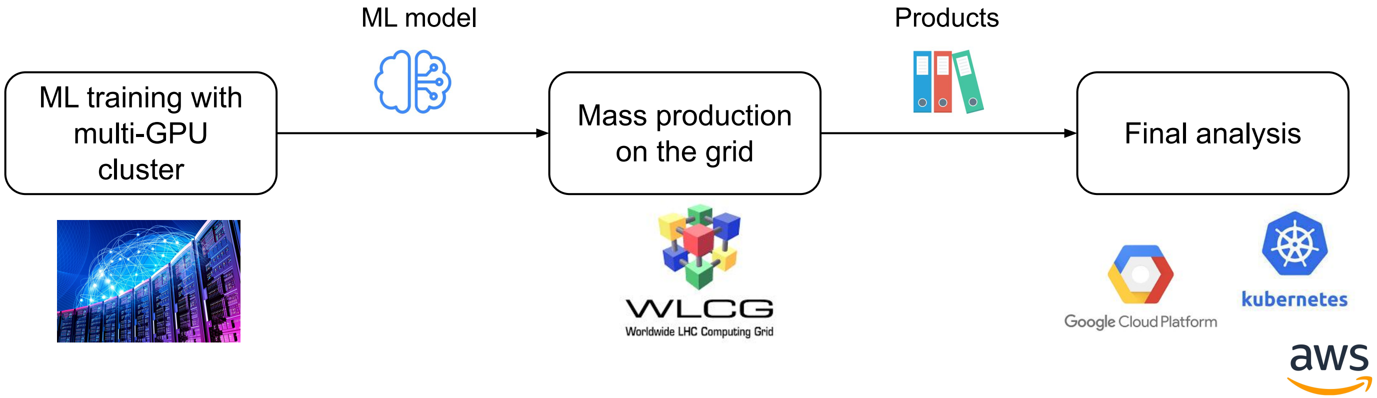
MC Toy Based Confidence Limits with iDDS

- Confidence Limits in Analyses
  - Exclude some range of relevant phase space for the considered theory / process
  - Show that obtained results is meaningfully different from what could have obtained by chance
- Independent MC toy calculations and post-processing to extract limit from sampled test statistic distribution
- A variant of HPO workflow to have a chain from toy limit calculations to post-processing
  - iDDS has a pool of pointers to MC toys
  - Each job takes a pointer to calculate the relevant toy limits and takes another pointer if the walltime is still available
  - iDDS triggers the post-processing that combines toy limits to the final results



Running Workflow 1/2

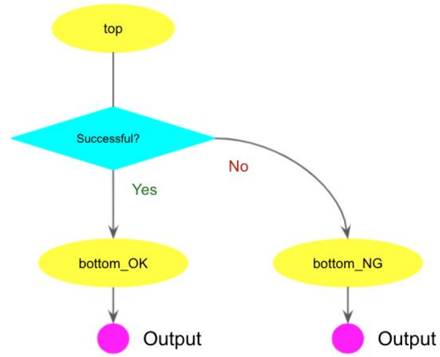
- Users can run complex workflows with
  - Conditional branching
  - Nested workflows
  - Parallelization with scatter
  - Loops
- Leverages PanDA's capability to integrate heterogeneous resources and workloads
- Workflow description with a directed acyclic graph (DAG)
  - Written in CWL
  - Python-based language and GUI to come
  - Loops in workflows if necessary, which is not natively supported in DAG



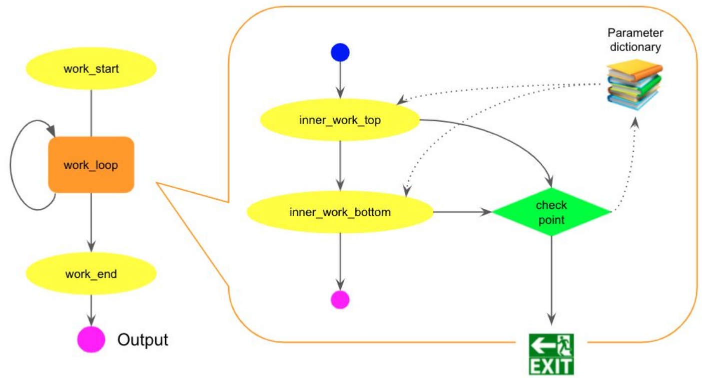
Running Workflow 2/2

➢ Workflow examples

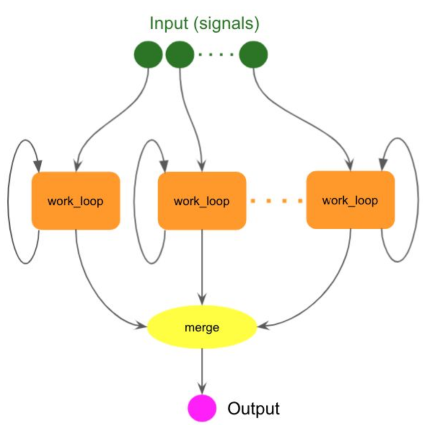
- Conditional workflow

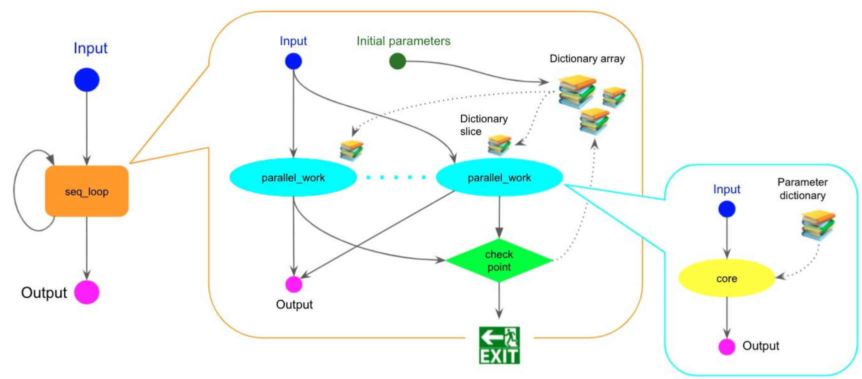


- Loop



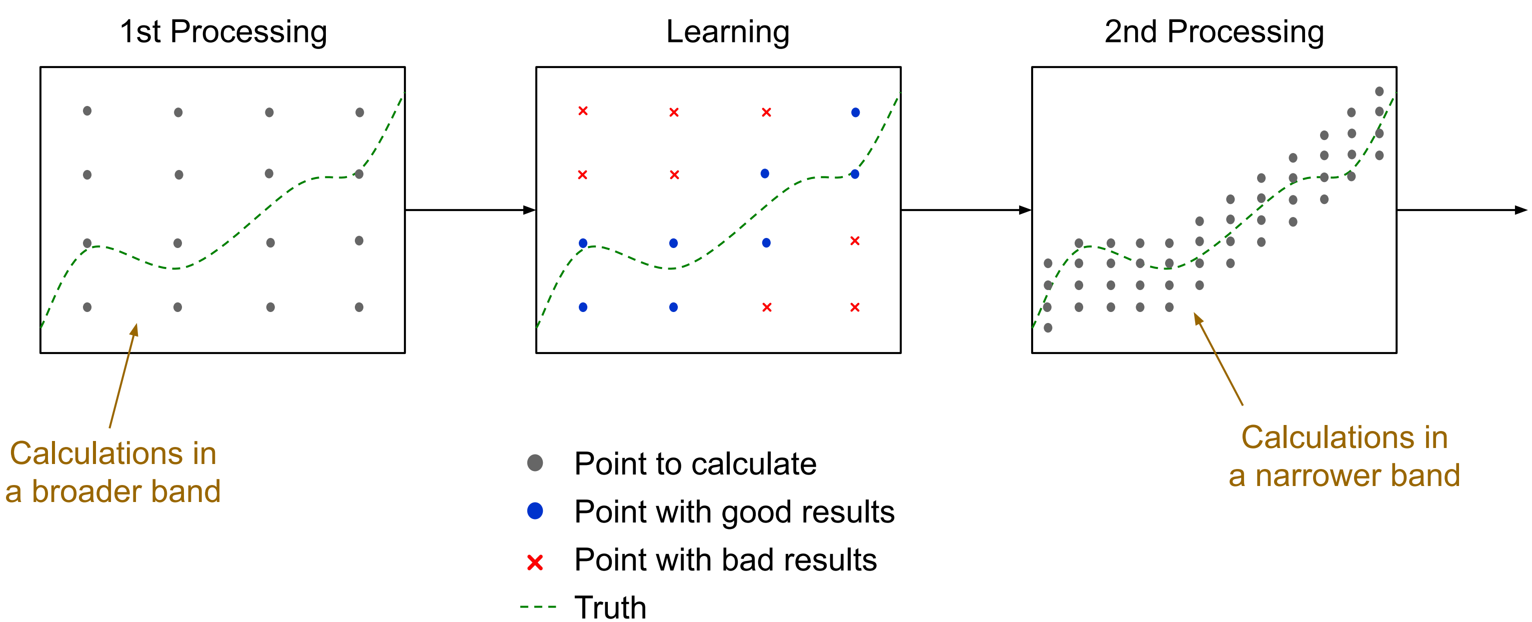
- Loop + scatter





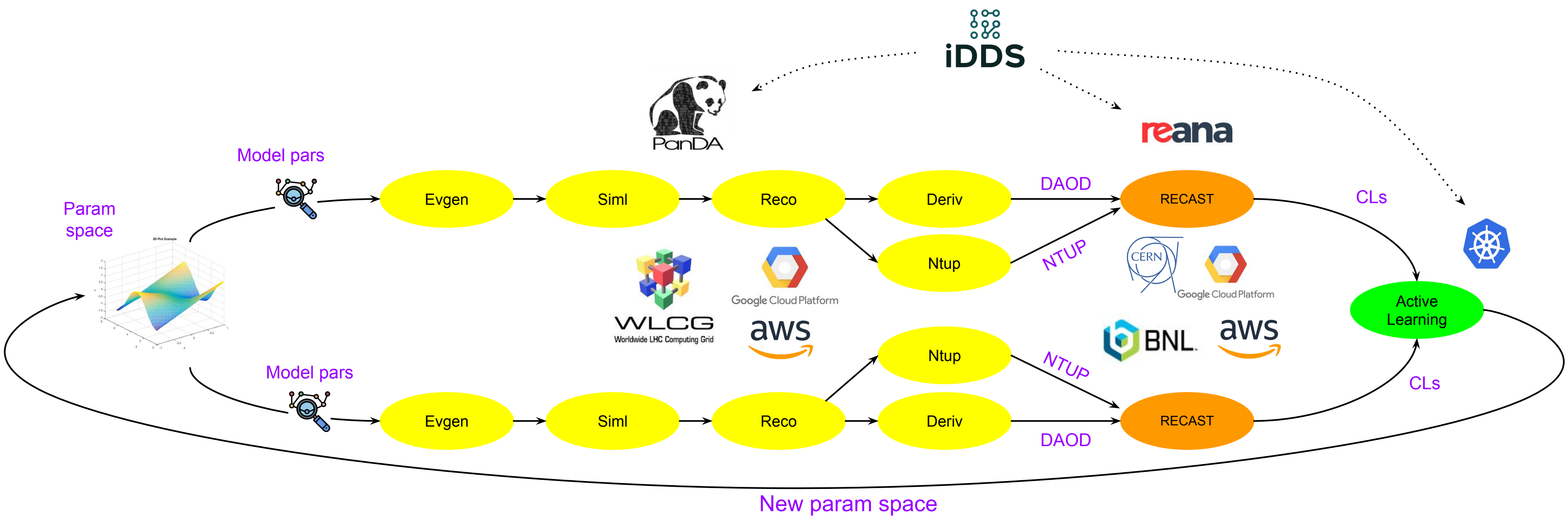
Active Learning 1/2

- Improving analysis efficiencies by using re-definition of the search space for the next iteration based on results in the previous iteration
- Reduction of total amount of computation due to pruning of "hopeless" region in the search space
- Various ML optimization functions and techniques to re-define the search space
- Feedback loops in the workflow



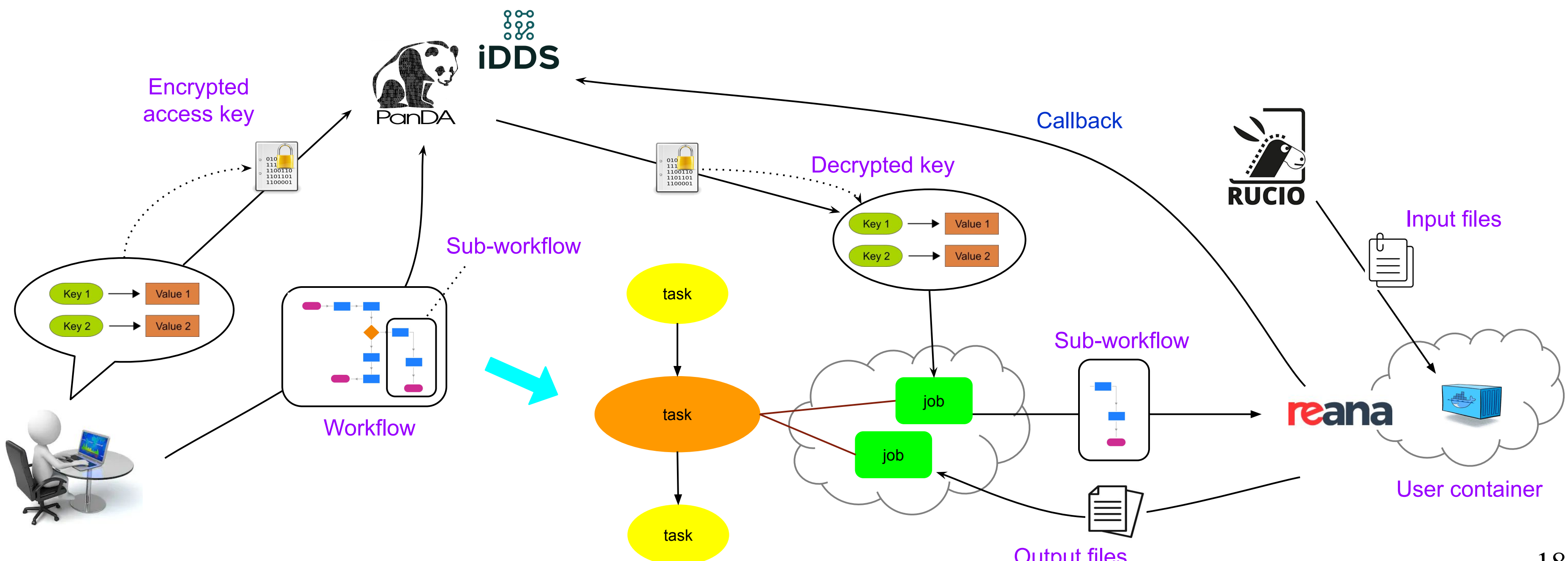
Active Learning 2/2

➢ Active Learning workflow for mono-Hbb analysis
1. Model parameter points in the original parameter space
2. Full production chain to produce Derived Analysis Object Data (DAOD) and ROOT NTuple (NTUP) for each point
3. RECAST for each point to calculate CLs
4. Re-define a new parameter space based on all CLs
5. New model parameter points in the new parameter space
6. Iterate 2-5



PanDA/REANA Integration

- Complex workflows offloading sub-workflows to external automation systems like REANA, AirFlow, Luigi, …
- The user uploads an access key to PanDA which is passed to jobs as a secret env variable or file
- Sub-workflows are internally mapped to special tasks and jobs
- Callback from REANA to trigger cascading of the special jobs
- Generalization of the PanDA's REST interface receiving callbacks from Rucio to work with any kind of external services

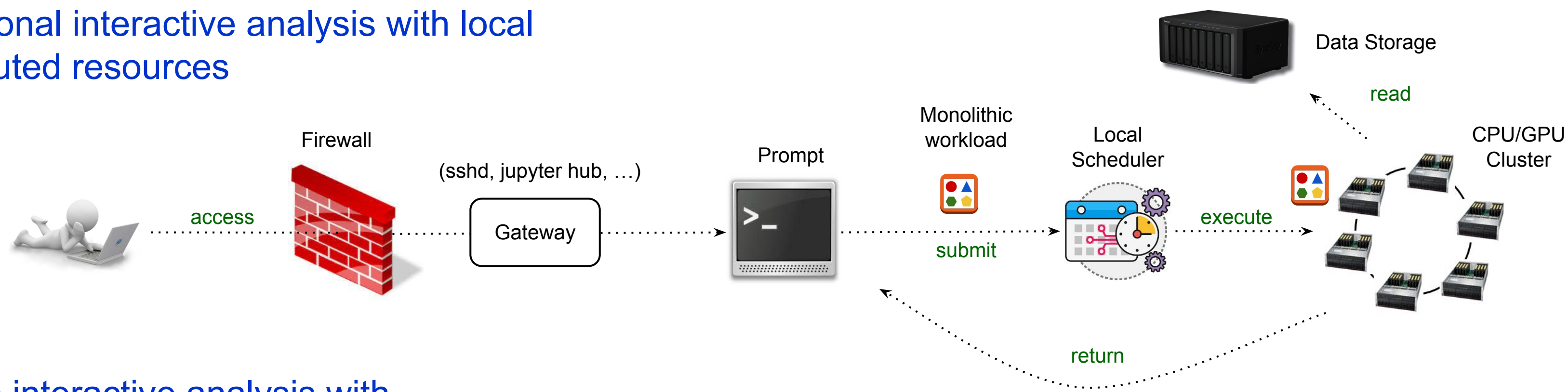


Pseudo Interactive Analysis 1/2

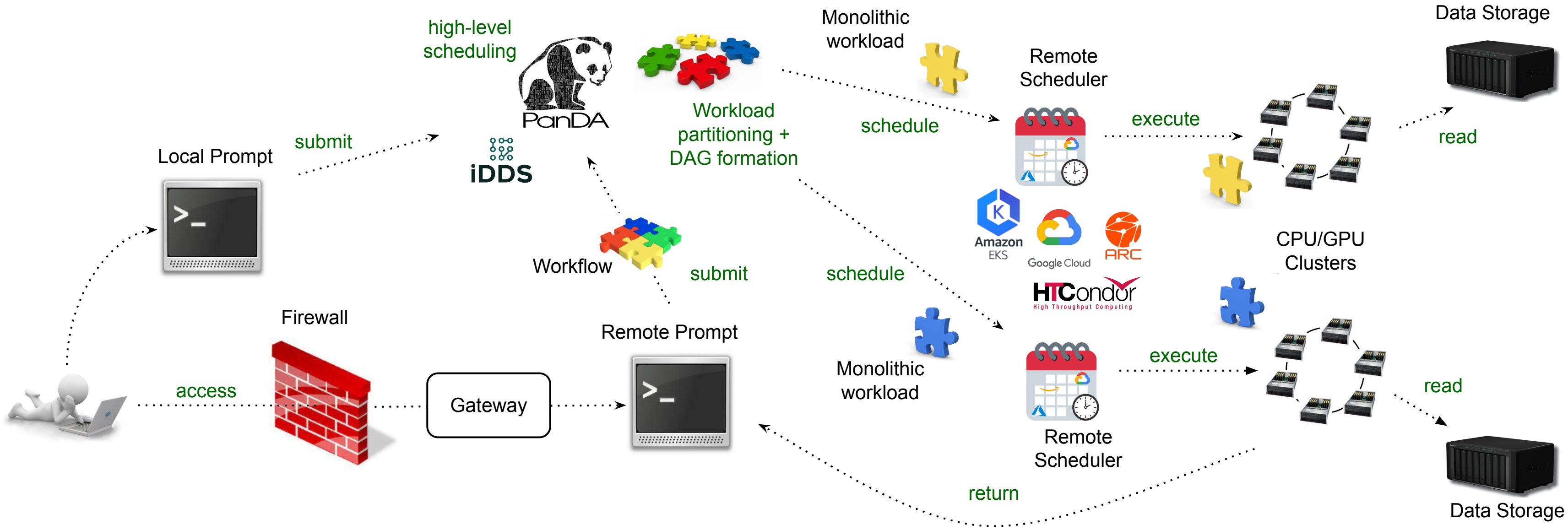
- Interactivity
  - One of hot topics in ATLAS
  - More for processing backend rather than user interface
    - No interactivity in combination of synchronous user interface and slow backend
- Pseudo interactive system
  - Interactive system on top of asynchronous resources
    - Interactive user interface
    - Geographically-distributed asynchronous processing backend
    - Reasonable response time
      - For express analysis, bulk spin-up with k8s on commercial cloud resources rather than traditional grid resources
      - Users can accept tens of minutes, a couple of / several hours, and so on, depending on data volume, intensity of computation, and their expectation
- Advantages
  - Central accounting and fairshare, one stop service isolating users from various processing backends and multiple resource types (grid, HPC, clouds, opportunistic, …), further scale-up with more computing resources across various resource providers, …
- Disadvantages
  - Latency, more development efforts for interface and monitoring due to lack of direct access to computing resources, …

Pseudo Interactive Analysis 2/2

Traditional interactive analysis with local distributed resources



Pseudo interactive analysis with geographically distributed resources



➢ Ongoing efforts to get rid of latencies from the entire system and demonstrate rapid resource provisioning

Current Status of R&D Activities

- Integration of emerging resources
  - In production
- Data carousel
  - In production
- HPO
  - In production
  - Room to improve user interface and monitoring
  - To attract more users as ML becomes popular in ATLAS/HEP
- Workflow support
  - In production
  - Being improved with the Active Learning exercise
- Integration of external services
  - Ongoing
- Pseudo-interactive analysis
  - Ongoing
- More to come …
  - E.g., writing a proposal to generalize scaled-up ML services beyond ATLAS

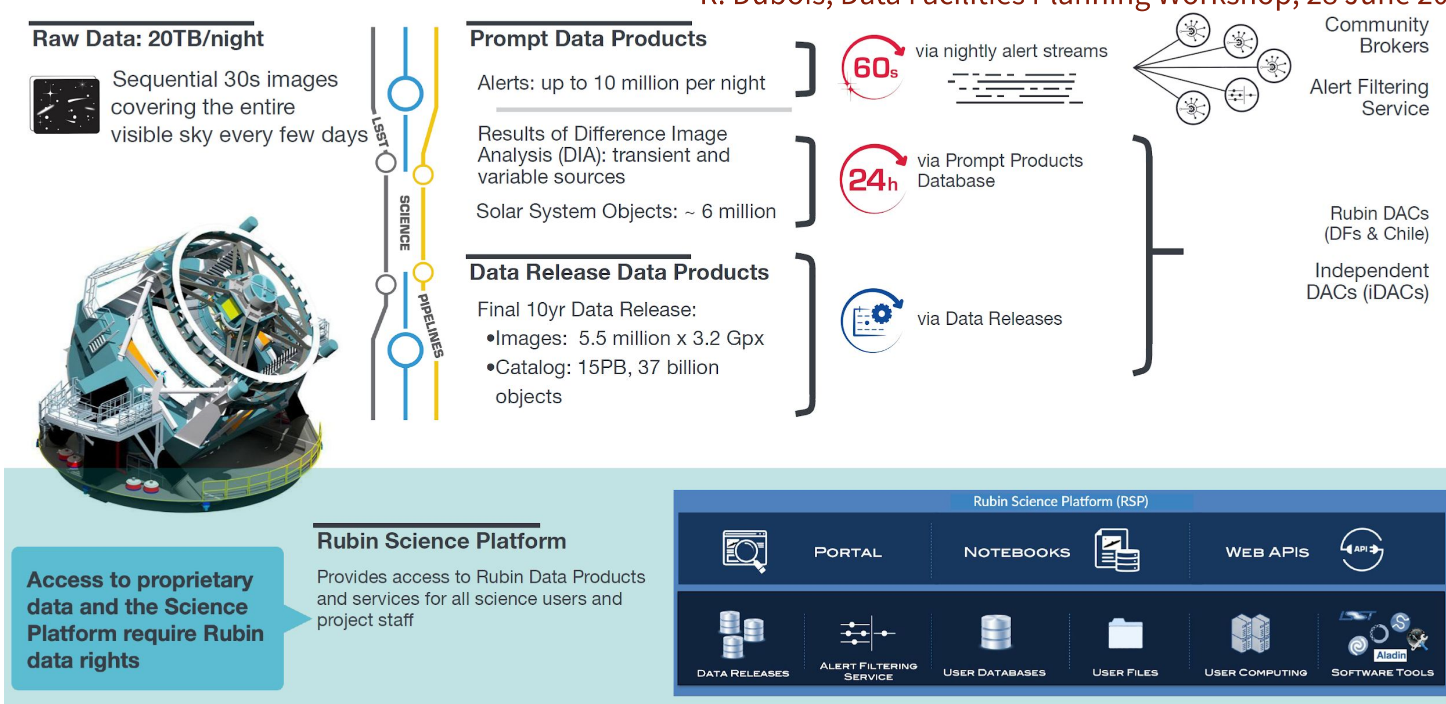
PanDA for Rubin Observatory

VERA C. RUBIN
OBSERVATORY

Vera C. Rubin Observatory

- To conduct the 10-year Legacy Survey of Space and Time (LSST)
- LSST will deliver a 500 petabyte set of images and data products that will address some of the most pressing questions about the structure and evolution of the universe and the objects in it
- Consists of an integrated system that combines an 8.4-meter primary mirror, the world's largest digital camera, a complex data processing system, and an online education platform

R. Dubois, Data Facilities Planning Workshop, 28 June 2021

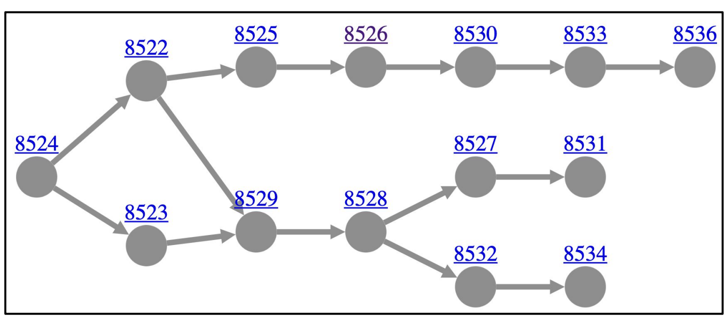


Rubin Computing

- ➢ A handful sites
  - SLAC and NCSA US
  - IDF Google
  - IRIS UK
  - IN2P3 FR
- ➢ Billion files per year
- ➢ ~200k short jobs running in parallel, 950 TFLOPS by the DR11
- ➢ Software release:
  - Docker images stream (https://hub.docker.com/r/lsstsqre/centos) (Daily, Weekly releases
  - Github repository (https://github.com/lsst/)
- ➢ Conda based package management
- ➢ User Analysis: Rubin Science Platform (Jupyter based), Data Production: PanDA based

Computing Operation

- PanDA integration done in first half of 2021
- Culminated during last spring/summer in successful scaling tests (~200k concurrent jobs) that drove the Rubin decision to adopt PanDA
- Began as a demonstrator, now used for production
- DAG-based processing
  - DAG support in PanDA originally developed for Rubin
  - Job level DAG optimized for managing DAG dependencies of big tasks (80K jobs)
  - Task level DAG optimized for triggering the final merging task
- Largely stable since Oct 2021
  - Rubin ops is routinely running PanDA workloads at large scale
- New PanDA hires for Rubin



DAG visualization

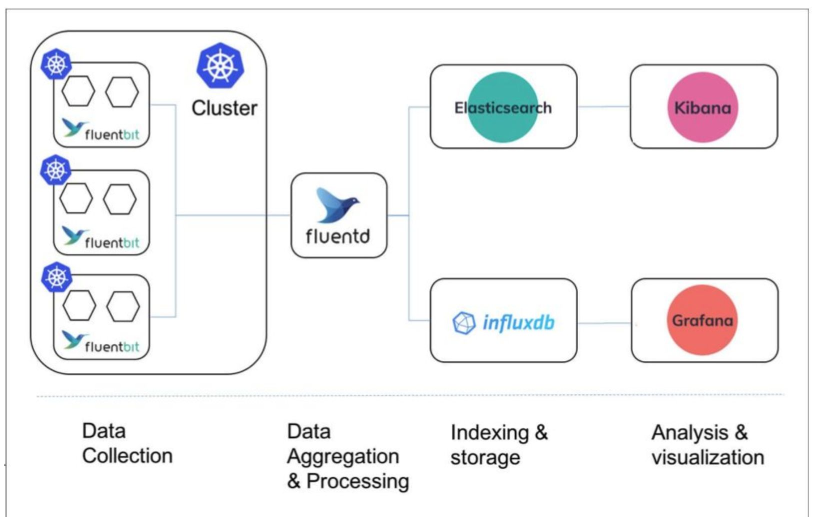
id	username	status	graph	workflow name	on (UTC)	tasks	tasks	remaining_files	released_files	total_files
784	michelle gower	finished	plot	u_michellegower_tinydc2_v23_0_1_rc1_allbackport_20220202t162205z	2022-02-02 16:22:40	7	Finished(7)	0	17	17
783	michelle gower	finished	plot	u_michellegower_tinydc2_v23_0_1_rc1_dm-32830_20220131t235841z	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

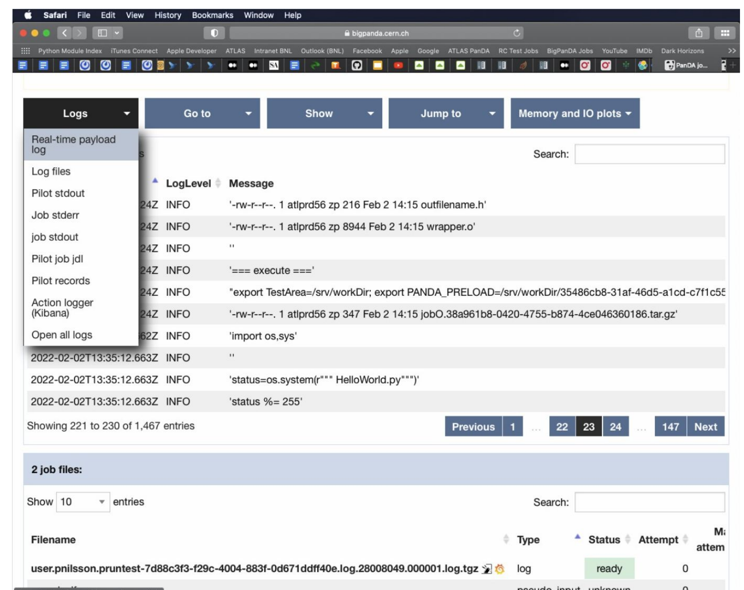
Showing 1 to 10 of 305 entries

Previous 1 2 3 4 5 … 31 Next

Quasi Real-time Log Access

- ➢ Suggestion from Rubin developers to use collectors tailor made for rapid logging of the payload
  - Watching the tail of the payload log, when it changes it adds job info and sends as json to external logging service
  - Logstash, Fluentd and Google Cloud Logging, …
- ➢ Preliminary implementation for Rubin used in production since last fall, being refactored to be experiment agnostic
  - Strong interest from ATLAS
  - Plugin structure to support various services and experiments





PanDA Services for Rubin - Current, and at SLAC

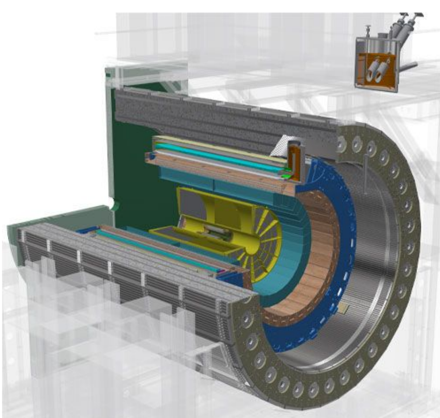
- PanDA, PanDA-IAM, Harvester, iDDS, CRIC and Monitors are currently set up and maintained at CERN, with thanks to the DOMA R&D project
  - Playground for non-ATLAS experiments to try
- Migration to SLAC when SLAC has the hardware (Spring 2022)
  - JEDI and the panda server: 4 nodes with 8 cores + 16 GB RAM + 200 GB disk on each
  - Postgres: 1 node with 8-10 cores + 128 GB RAM + 4 TB SSD disk
  - Harvester+condor: 2 nodes with 4 cores + 8 GB RAM + 200 GB disk
  - Pandamon: 1 node with 8 cores + 16 GB RAM + 200 GB disk
  - iDDS: 4 nodes with 8 cores + 16 GB RAM + 200 GB disk
- PanDA team will install the PanDA services, with differences from ATLAS
  - PostgreSQL as back end DB, not Oracle. All services have been ported to PostgreSQL
  - K8 based service hosting, rather than VMs
  - So the new approaches, PostgreSQL and K8 services, are implemented but to be exercised at scale (CERN services are the production instance until SLAC validated)
- Software migration will involve
  - Databases, JWT, other services like ActiveMQ, Monitors

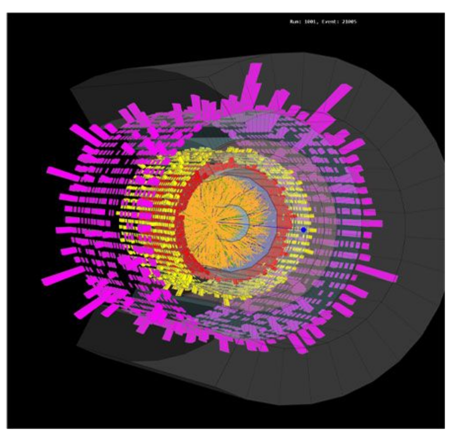
PanDA for sPHENIX

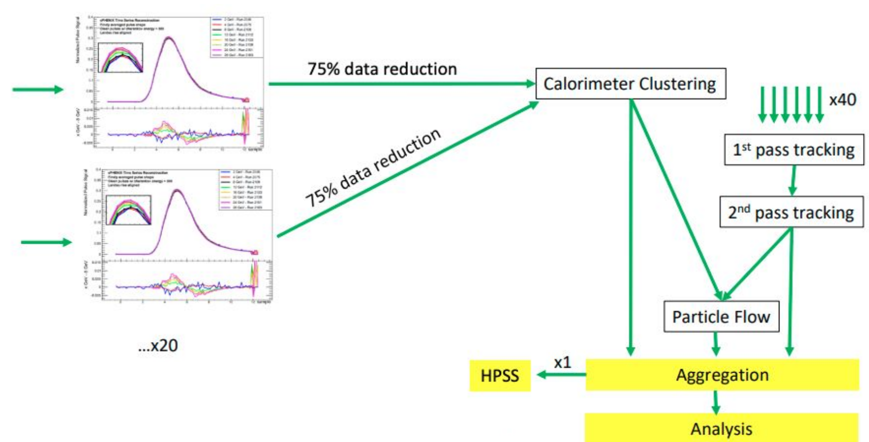
sPHENIX

sPHENIX

- A new detector under construction for the RHIC facility at BNL, with state-of-art capabilities for studies of the strongly interacting quark-gluon plasma using jet and heavy-flavor observables
- Very small SW core group and tight timeline
  - Do not reinvent the wheel, use existing modern tools → PanDA and Rucio
- A ton of minbias data in a very short period of time
  - 25sec/evt reco time, 107B AuAu events in 24 weeks, 230PB/year
  - Ongoing Mock Data Challenge for testing dataflow



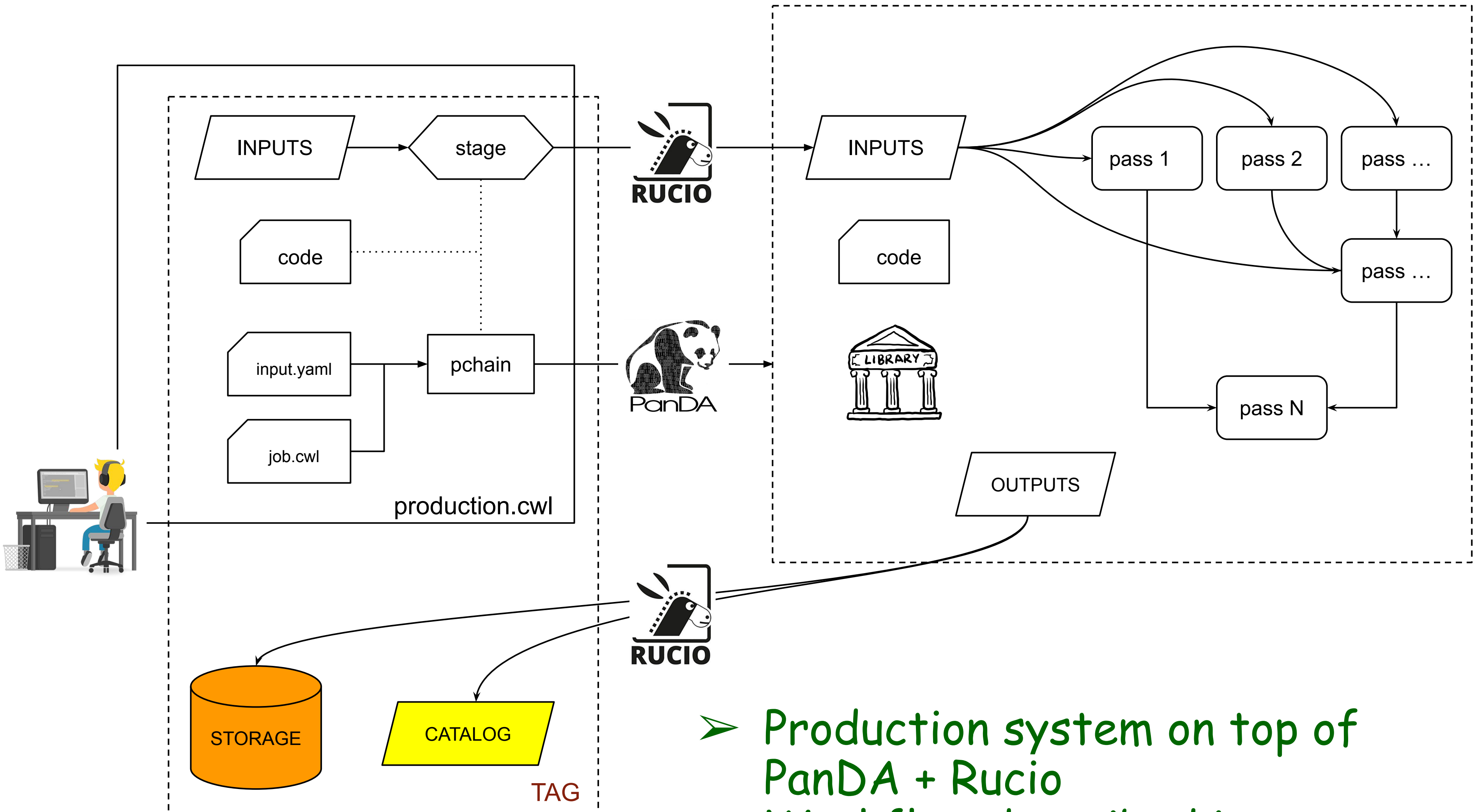




PanDA@sPHENIX

- sPHENIX adopted PanDA at about the same time and with the same timeline as Rubin
  - Evaluation and pre-production using DOMA PanDA at CERN
- Most usefully, sPHENIX and Rubin have very similar needs for PanDA service deployment
  - PostgreSQL backend
    - Newly implemented in PanDA in the last year, on the foundation of largely agnostic implementations across the PanDA services
  - k8s/OKD based deployment of services
    - Also newly implemented in the last year as an alternative to the VM based deployments used in the CERN instance
- Rucio integration from the beginning
  - Leveraging years of PanDA's experience with Rucio and Rucio expertise at BNL
- Both sPHENIX and Rubin want to have their home-based services up and running in Spring 2021
  - At BNL for sPHENIX, SLAC for Rubin
- Nice synergy between the efforts, efficient and mutually supportive
  - Also with EIC, for which requirements are similar to sPHENIX to provide a PanDA EIC demonstrator

sPHENIX Handy Remote Execution Koordinator



- ➢ Production system on top of PanDA + Rucio
- ➢ Workflow described in DAG/CWL

Conclusions

- Great opportunity in LHC LS2 to develop a lot of R&D activities in ATLAS PanDA for Run 3 and HL-LHC
  - Automation, optimal resource usage, and new workflows
  - Revealed PanDA's capabilities to manage emerging workloads as well as traditional HEP workloads
  - Some activities are still ongoing and more challenges will come
- PanDA has been expanding to Rubin (astrophysics) and sPHENIX (nuclear physics) beyond ATLAS (HEP)
  - Experiment-agnostic DOMA PanDA services and deployment with PostgreSQL backend and containers+k8s are reducing difficulties to adopt PanDA
  - Not all advanced functions in PanDA to take, depending on requirements/needs of each experiment
  - Nice synergy between the efforts, efficient and mutually supportive