PanDA

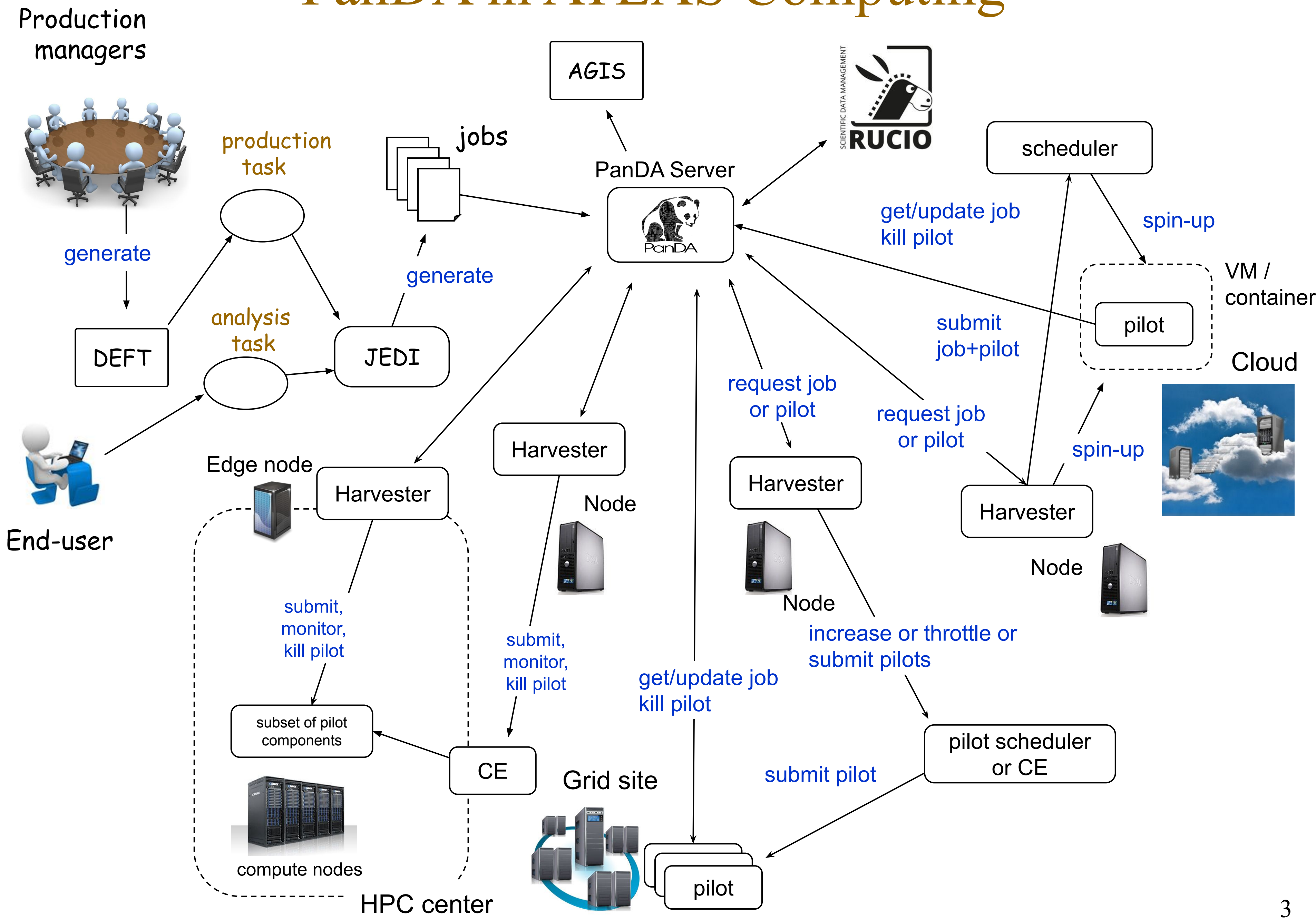
Tadashi Maeno (BNL)

NPPS meeting, Jun 19

PanDA in Nutshell

- PanDA = Production and Distributed Analysis System
  - Designed to meet ATLAS production/analysis requirements for a data-driven workload management system capable of operating at LHC data processing scale
- Continuous evolution while steadily running for ATLAS since 2005 including data taking periods
  - Significant refactoring to move to Oracle from MySQL, major system reengineering to implement new paradigm for high level workload management and fine-grained processing mechanism, migration of ATLAS DDMS to Rucio from DQ2, migration to new pilot provisioning machinery, …
- ~150k running production+analysis jobs with ~440k cores, ~32M HTTPS sessions per day, 56M transactions in Oracle per day, 1.6k individual users for analysis in 1 year
- ATLAS PanDA, BigPanDA, BigPanDA++, beyond ATLAS, Google projects, …
- Plenty of advanced and interesting functions/activities but only recent ATLAS ones to show due to limited time slot

PanDA in ATLAS Computing



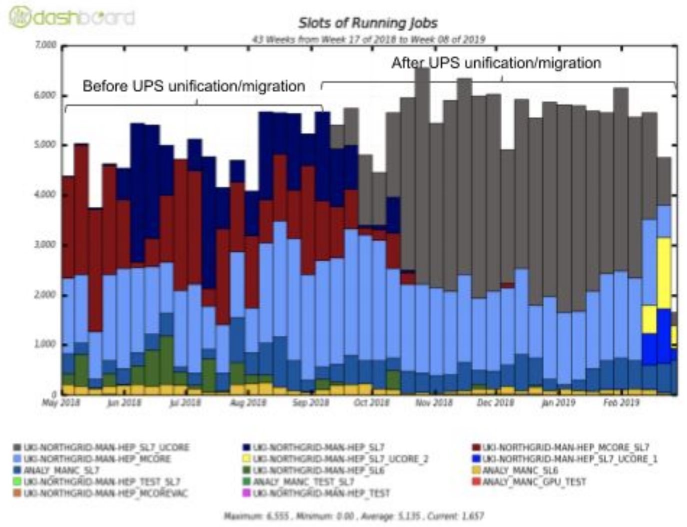
Harvester 1/2

- A resource-facing service between PanDA server and collection of pilots (workers) for pilot provisioning
- Stateless service plus database for local bookkeeping
- Flexible deployment model and modular design for various resource types and workflows
  - On HPC edge nodes with limited runtime environment
    → A single node + multi-threading + sqlite3.
    On dedicated nodes
    → Multiple nodes + multi-processing + MariaDB
  - Plugins with native API, such as SLURM, LSF, EC2, GCE, k8s, gfal, …, and plugins with 3rd party services, such as condor, ARC interface, Rucio, FTS, Globus Online, …
- Objectives
  - A common machinery for pilot provisioning on all computing resources
  - Better resource monitoring
  - Coherent implementations for HPCs
  - Timely optimization of CPU allocation among various resource types and removal of batch-level partitioning
  - Tight integration between WFMS and resources for new workflows
- The project launched in Dec 2016 with 11 developers in US (BNL, UTA, Duke U, ANL), Norway, Slovenia, Taiwan, Italy, and Russia

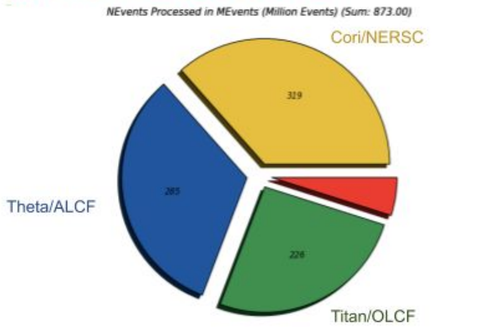
Harvester 2/2

- ➢ Entire ATLAS grid migrated by Jan 2019
- ➢ ATLAS High Level Trigger (HLT) CPU farm with 50k cores, aka Sim@P1 in production
- ➢ Successfully demonstrated GCE + GCE API + Google Storage + preemptible VMs
- ➢ All US DOE HPCs in production since Feb 2018

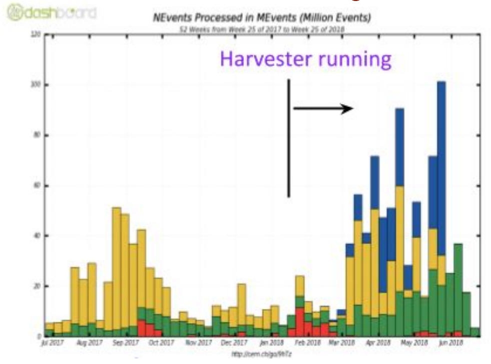
Migration of UK grid resources



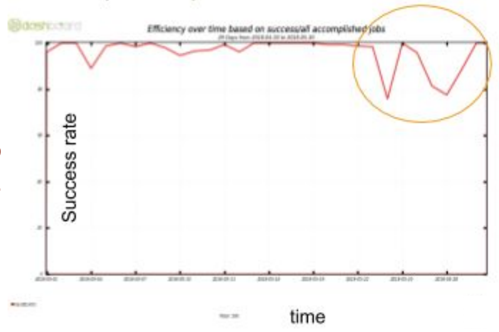
The total number of events (in M events) processed for ATLAS simulation production for last 6 months at US DOE HPCs



The number of events processed per day at US HPCs around migration

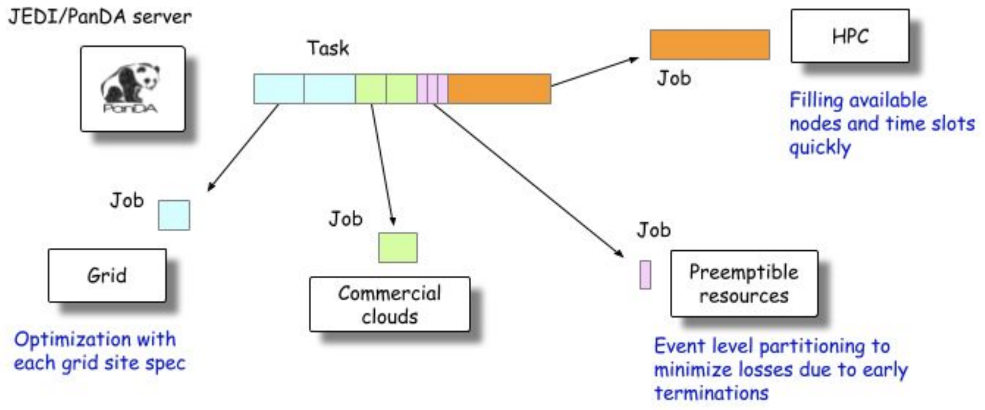


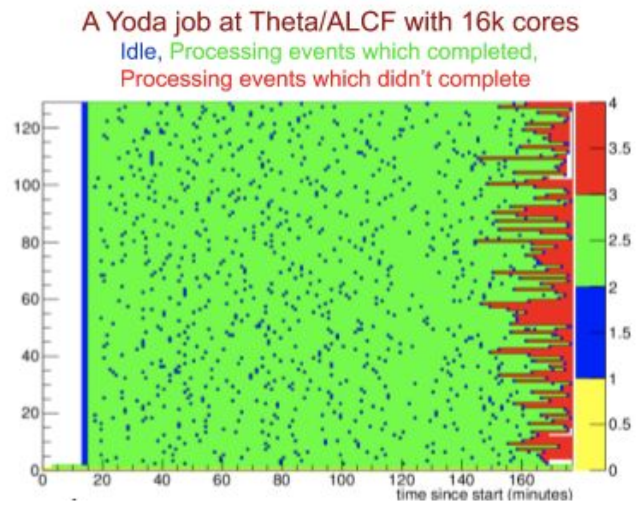
Effect of switching from normal VMS to preemptible VMs on GCE



Integration of HPCs with Jumbo Payload

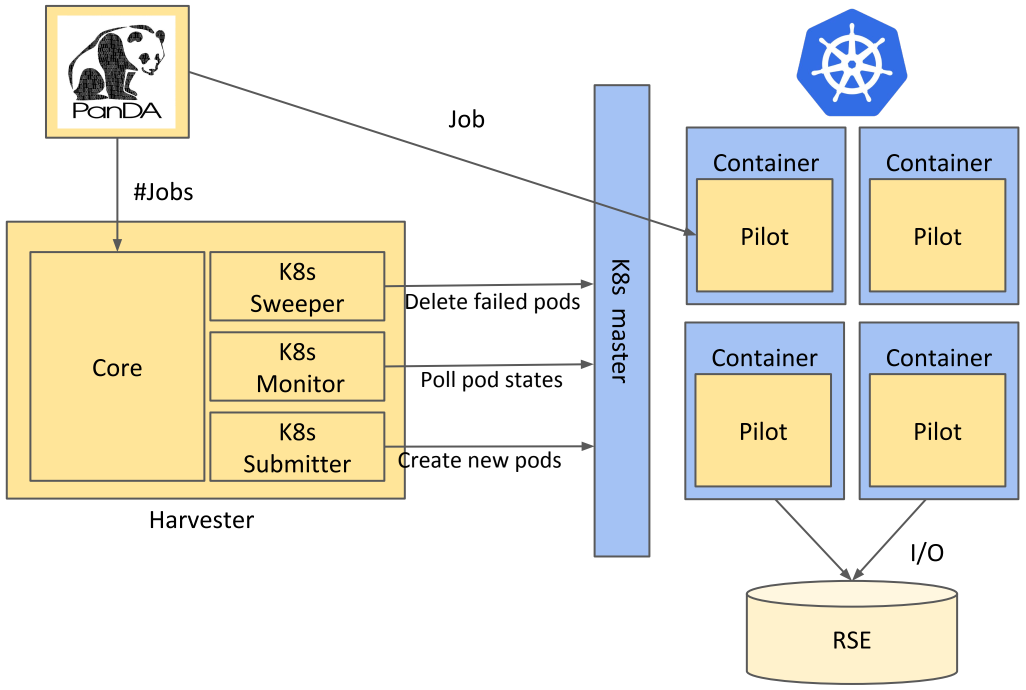
- Batch jobs are no longer atomic entities in PanDA thanks to capability of high level workload management and event-level bookkeeping
- Dynamic shaping of jobs based on real time information of available compute power and walltime for each resource
- No dedicated/custom tasks for HPCs
  - Old : Special tasks to have big jobs at HPC
  - New : Common tasks share among various resources including HPCs to have proper sizes of jobs at each resource
- In full production at Theta/ALCF and Cori/NERSC while at limited scale for Titan/OLCF due to fragile OLCF file system
- Successfully ran at MareNostrum 4 at BSC, will continue for MN5 which has been granted by EuroHPC recently

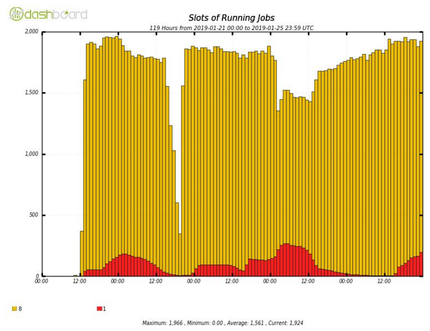

JEDI/PanDA server
PanDA
Task
Job
HPC
Filling available nodes and time slots quickly
Job
Grid
Optimization with each grid site spec
Job
Commercial clouds
Job
Preemptible resources
Event level partitioning to minimize losses due to early terminations



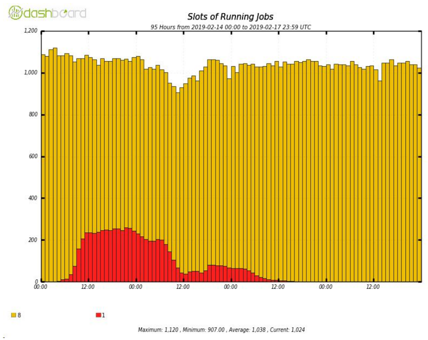
Resources via Kubernetes

- Use Kubernetes as CE + a batch system
  - Central harvester manages remote resources through kubernetes
- Based on SLC6 containers and CVMFS-csi driver
- Proxy passed through K8s Secret
- Still room for evolution, e.g. allow arbitrary container/options execution, maybe split I/O in 1-core container, improve usage of infrastructure
- Tested at scale for some weeks at CERN, being continued at UVic





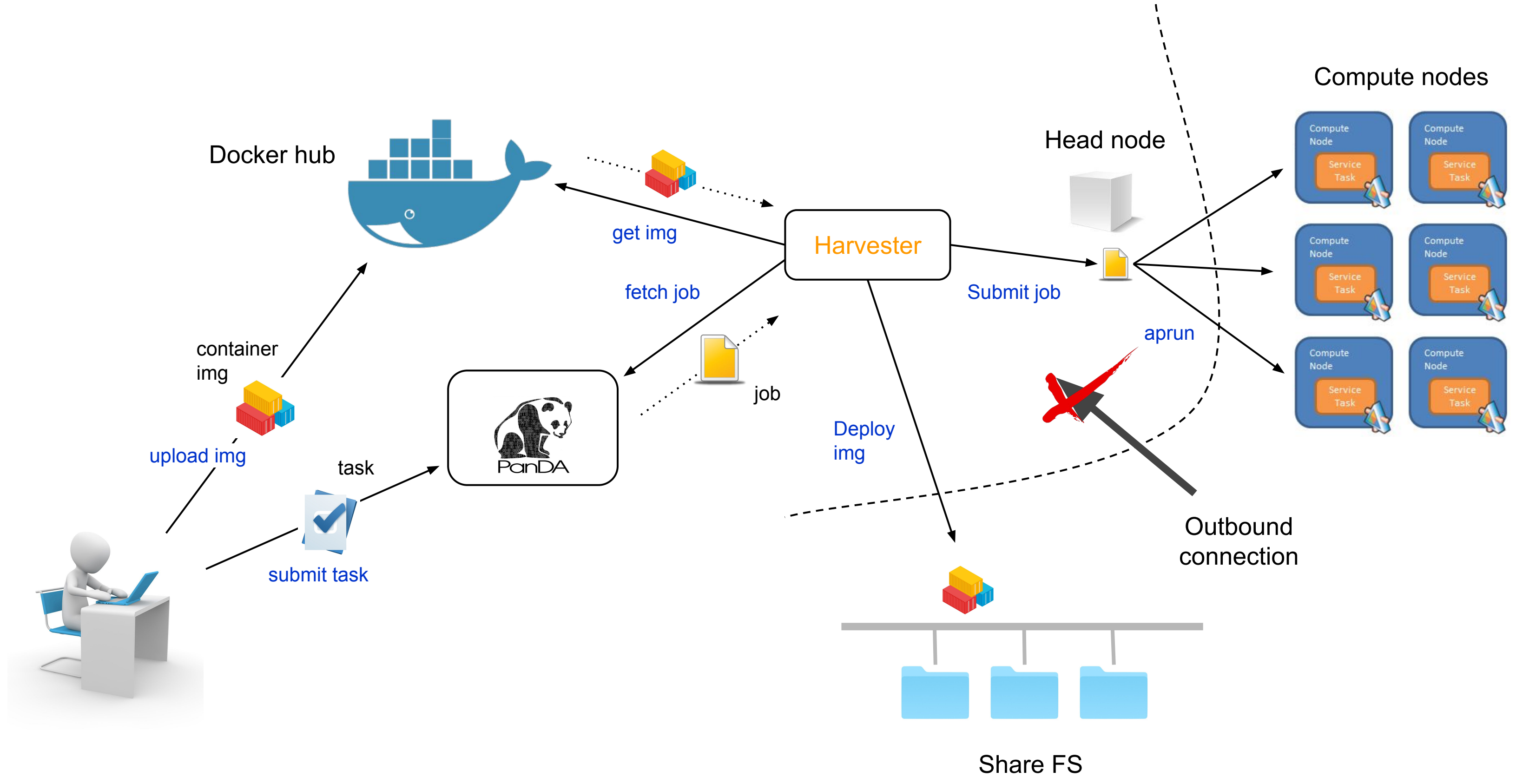
With default K8s scheduler (round robin load balance)



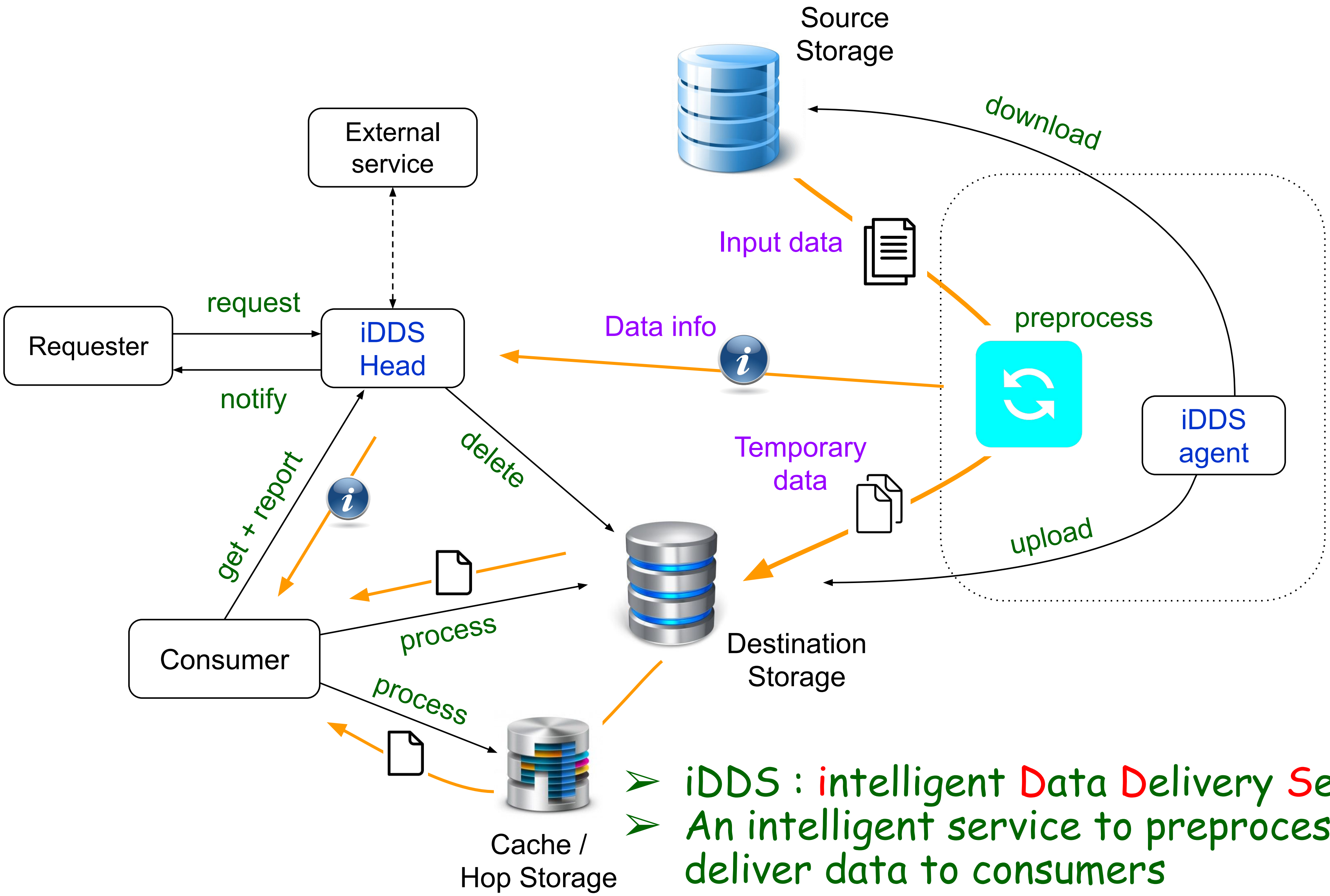
With policy tuning to pack nodes

HPC/GPU + ML + MPI

- Distributed training on HPC or GPU cluster through PanDA and Harvester
- Multi-node payload with MPI to be prepared by users for now
  - Might provide a common MPI framework in the future
- On-demand deployment for user container images
- Trying at BNL Institutional Cluster



iDDS 1/2



- iDDS : intelligent Data Delivery Service
- An intelligent service to preprocess and deliver data to consumers
  - Delivered data = files, file fragments, file information, or sets of files

iDDS 2/2

- Join project between ATLAS and IRIS-HEP
- To generalize concept/workflow of Event Streaming Service
- Not a storage, WFMS, or DDMS
  - Delegation of many functions to WFMS, DDMS and Cache
- iDDS + WFMS (as preprocessing backend) + DDMS + Cache = CDN
- Requirements
  - Experiment agnostic
  - Flexibility to support more use-cases and backend systems
  - Easy and cheaper deployment
- ATLAS usecases
  - Fine-grained processing
  - Tape carousel and dynamic data placement
  - Data delivation with WAN
  - On demand data transfers at HPC
  - Custom data transformation for hyperparameter optimization
  - …
- Potentially huge R&D but ATLAS manpower is limited for now
- Splinter meeting in S&C workshop next week in NY to reach a consensus in ATLAS before the project "officially" kicks off
  - Collaboration with other projects
  - Manpower allocation