



HL-LHC Related Software R&D in the  
BNL Nuclear and Particle Physics Software (NPPS)  
Group  
(Supplementary Slides)

Torre Wenaus (BNL)

IRIS-HEP Workshop on HL-LHC Computing R&D  
Catholic University of America  
October 23 2019

# HL-LHC software R&D in BNL NPPS



These slides summarize software R&D projects in the BNL NPPS group that aren't covered elsewhere

Nuclear and Particle Physics Software (NPPS) Group in the BNL Physics Department consolidates much of the NPP software development in the department

- Currently 19 people, 14 HEP, 5 NP
- ATLAS, Belle II, DUNE, EIC, PHENIX, sPHENIX, STAR
- More info at <https://npps.bnl.gov>

# PanDA evolution

- The PanDA workload manager has evolved continuously since it entered ATLAS wide production in 2008
- Expanding capability on a stable, scalable functional foundation
- Manages ATLAS distributed computing, analysis and production
- Global orchestration of 400-600k cores 24x7 across ~140 sites, peaking > 1.2M cores
- Uses resources of all types: grid (distributed computing clusters), HPCs & LCFs (including no outbound access), clouds, volunteer computing (ATLAS@Home)
- Key features are its *scalability* and its *close coupling with large scale data management* (the Rucio data manager of ATLAS)
- Integrated workflow and dataflow moving >1 PB, >20 GB/s, 1.5-2M files per day, > 400PB, 1B files
  - Over 1.5 Exabytes processed per year
- Powerful monitoring system for operators and analysis users
- Sophisticated allocation and prioritization among dozens of activities and over 1000 individual analysis users
  - ATLAS physics coordination drives the prioritization and sharing on a weekly and sometimes daily basis
- ATLAS aggressively harvests *opportunistic resources* because our physics is compute-limited -- more cycles, more physics
  - PanDA makes this possible and manageable
- Strong *HPC support* good for DOE's strong **"use HPCs"** mandate
- Much of the rest of these slides is concerned with how we are building on the PanDA foundation with innovative R&D ultimately directed at solving HL-LHC computing challenges

ATLAS PanDA Dash Tasks Jobs Errors Users Sites Incidents Search My BigPanDA Prodsys Services VO Help

PanDA jobs: last 12 hours Params: limit=10000 Warning: limit 20000 per job table Task has 9 jobs in total Jobsactivated, limit is set to 2000 Total jobs found = 120000

32008 jobs in this selection

Job modification times in this listing range from 2018-02-14 06:43:00 to 2018-02-14 18:43:00.  
Job current priorities in this listing range from 1000000 to -1000000. See priorityrange in the job attribute summary to see how priorities are distributed.

Job attribute summary Sort by count, alpha

actual:corecount (10)	1 (28092)	2 (1) 4 (501)	6 (235) 7 (1) 8 (2804)	12 (9) 16 (292)	24 (9) 128 (64)
Atlas-17.0.6 (1)	Atlas-17.2.13 (4)	Atlas-17.2.14 (350)	Atlas-17.2.2 (2)	Atlas-17.2.6 (2)	Atlas-19.2.4 (26)

workinggroup (32)

AP_BPHY (19)	AP_EXOT (285)	AP_HIGG (2709)	AP_HION (4678)	AP_JETM (259)	AP_MCGN (4180)	AP_REPR (3475)	AP_SOFT (112)	AP_STDM (925)	AP_SUSY (154)	AP_TALP (17)	AP_TALK (133)	AP_TOPQ (905)	AP_TRIG (1)	AP_UPCR (171)	AP_VALI (2291)	des-instel (545)	det-muon (1)	GP_EGAM (29)	GP_EXOT (365)	GP_FTAC (42)	GP_HIGG (888)	GP_JETM (1)	GP_PHYS (1130)	GP_SUSY (441)	GP_TALP (12)
--------------	---------------	----------------	----------------	---------------	----------------	----------------	---------------	---------------	---------------	--------------	---------------	---------------	-------------	---------------	----------------	------------------	--------------	--------------	---------------	--------------	---------------	-------------	----------------	---------------	--------------

Overall error summary

Category:code	Attempt list	Errors	Sample error description
transformation:85	jobs	1	Non-zero exit code from transform subprocess executor
transformation:85	jobs	2	analysis output merge crash - consult log file more information here

Click to show/hide failed jobs details Prodsys Jobs Handling

Job list Only the most recent 1000 jobs are shown. Remove the limit and sort by PanDAId, time since last state change, ascending mod time, descending mod time, priority, attemptnr, id

PanDA ID Attempt#	Owner Group	Request Task ID	Transformation	Status	Created	Time to start d:h:m:s	Duration d:h:m:s	Mod	Cloud Site	Priority
					2018-02-				2018-	

JEDI/PanDA server

Task

Job

Grid

Commercial Clouds

Job

Job

Job

HPC

Volunteer computing

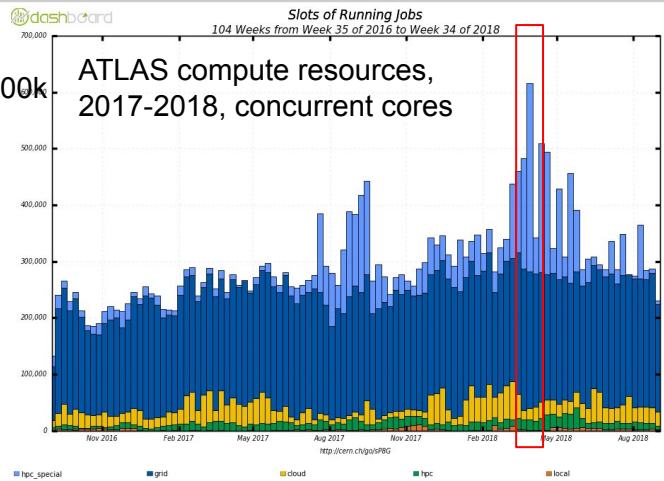
Optimization for each grid site

Economical usage on Amazon EC2 spot market

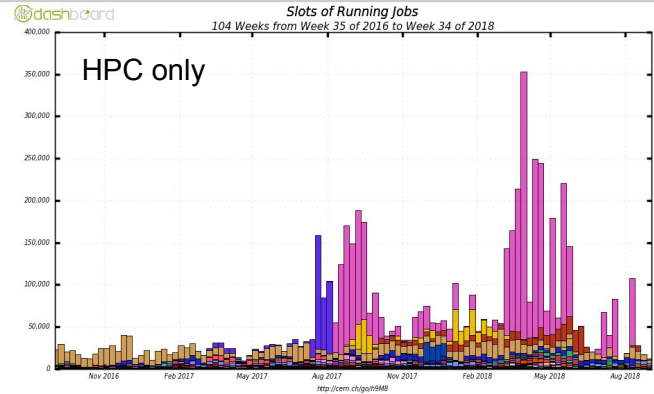
Filling available nodes and time slots quickly

Event level partitioning to minimize losses due to early terminations

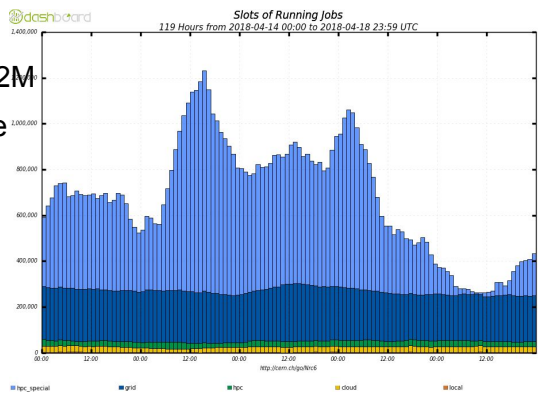
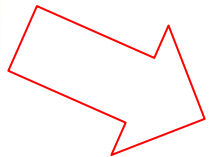
# HPCs in ATLAS via PanDA



A long history but a new era from 2018: very large facilities



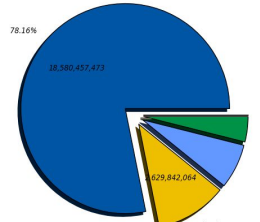
Light blue: “special” HPCs, where special means big, difficult to use, US DOE  
 Dark blue: the grid  
 Yellow: cloud resources including (dominantly) HLT  
 Green: “regular” HPCs, meaning easier to use, European or US NSF



Our workload management system is highly scalable!

- NERSC Cori p2\_mcore
- ORNL\_Titan\_MCORE
- hpc
- HPCMN\_MCORE
- NERSC\_Edison\_mcore
- NERSC\_Edison
- UIO\_MCORE
- NCS
- NERSC\_Edison\_2
- NERSC Cori p1\_ES
- CONNECT\_ES\_ODYSSEY\_MCORE
- CONNECT\_PIZDANT
- CONNECT\_ES\_ODYSSEY
- CSCS-PizDant
- CONNECT\_ES\_ODYSSEY
- CSCS-LCG2-HPC\_MCORE
- HPCMN
- UIO
- ALCF\_Theta
- CONNECT\_BLUEWATERS\_MCORE
- LRZ-LMU\_Mus1\_MCORE
- LRZ-LMU\_Mus2\_MCORE
- CONNECT\_STAMPEDE\_MCORE
- CONNECT\_UIUC\_MCORE
- CSCS-LCG2-HPC
- IRAP\_MCORE
- RRC-KI-HPC2
- Titan\_Iong\_MCORE
- LRZ-LMU\_Mus1\_MCORE1
- IRAP\_MCORE\_L0RN
- UIO\_MCORE\_L0RN
- CONNECT\_ES\_STAMPEDE\_MCORE
- LRZ-LMU\_CZPAP\_MCORE
- NCS\_MCORE

## CPU HS06 shares, 2018



Grid: 78%  
 Cloud, HLT: 11%  
 HPC special: 7%  
 HPC regular: 4%

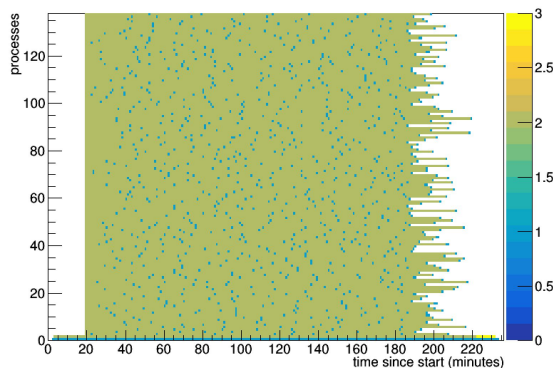


# ATLAS Event Service (AES)

Efficient processor and memory utilization  
In ATLAS production for several years

Without event service, each core processes N events. Once a core has finished its allocation, it idles (white)

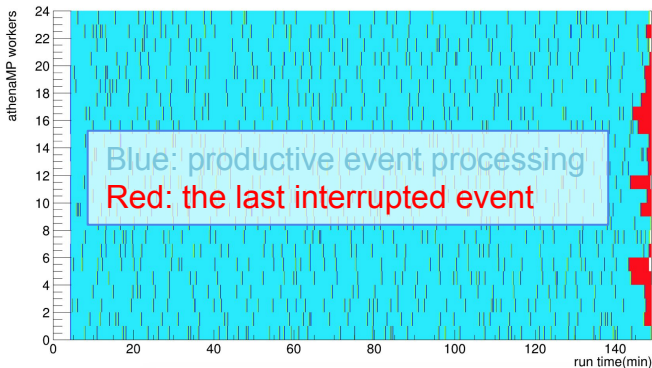
NERSC utilization per core, no AES



With event service, each core is allocated events to process until the scheduler slot ends.

If the job is suddenly killed by preemption, all processed events are preserved except the last few minutes (all are lost in a conventional job)

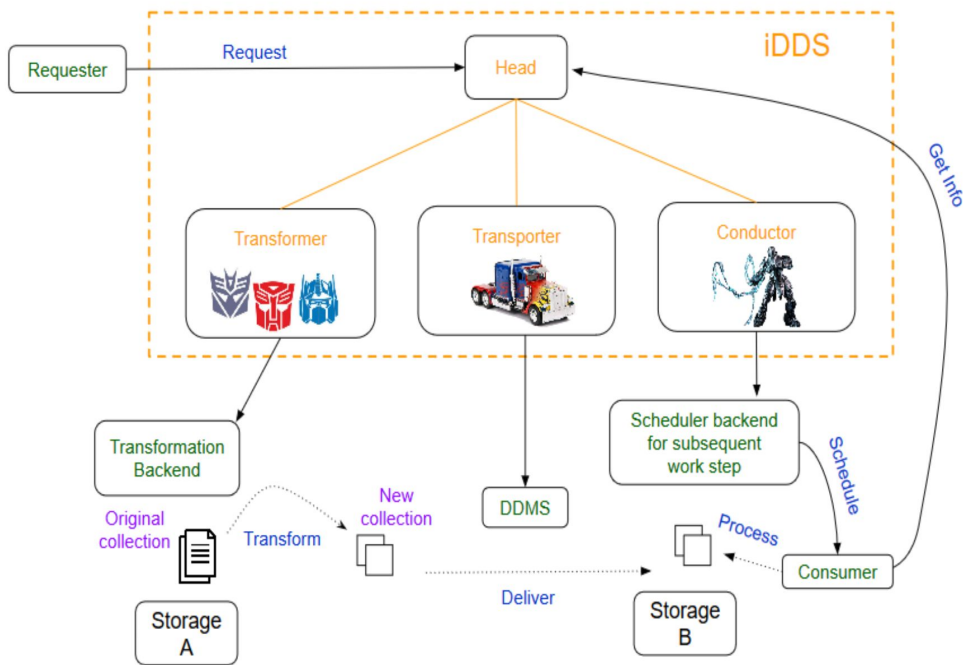
NERSC utilization per core with AES



**Next phase: fine grained event delivery via Intelligent data delivery system (iDDS)**

**AES elastically fills all cores for the full job lifetime**

# Intelligent data delivery system (iDDS)



- Experiment agnostic
- Deliver full or partial events efficiently, promptly, dynamically over the wire, leveraging caching
- Avoid up-front file replication latencies, redundant replication when work is rebrokered
- Leverage the WAN latency hiding of asynchronous fine grained data delivery
- Marshal/transform the data storage-side for efficient delivery and processing

iDDS project is joint with IRIS-HEP, with interest from WLCG-DOMA, CMS

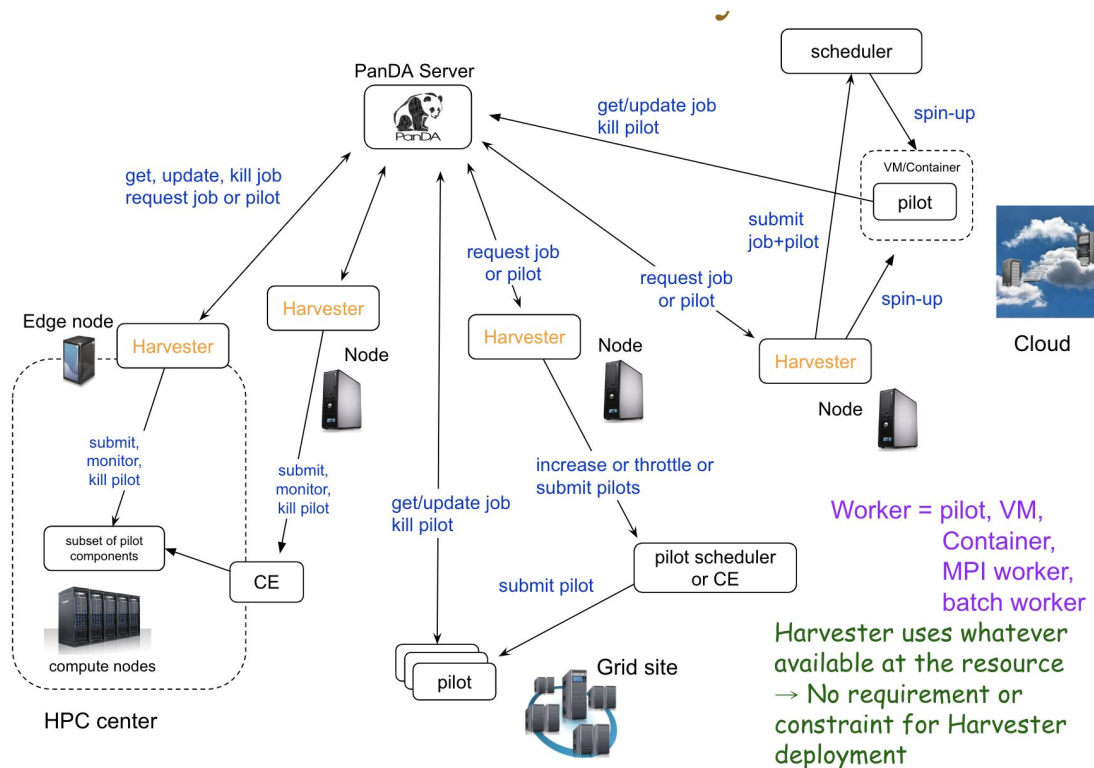
Brand new project in early prototyping, all welcome! Join up via HSF (contact me or Brian Bockelman)

# Data carousel



- Orchestration of workload management, data management and storage to feed a production campaign's inputs from tape via a sliding window of disk-resident data
  - Disk footprint is only ~5-10% (perhaps) of input data set size
  - PanDA + Rucio provide the tight orchestration required
- Promising results from several rounds of trials
- Next step: explore optimizing data placement on tape for efficient retrieval by carousel
- Next year: use iDDS to dynamically stream data to remote consumers: efficiently and rapidly use distributed processing to minimize campaign completion time
- ATLAS R&D led by BNL NPPS and SDCC, with participation of ~all ATLAS Tier 1s and WLCG DOMA

# Harvester



Worker = pilot, VM, Container, MPI worker, batch worker

Harvester uses whatever available at the resource  
→ No requirement or constraint for Harvester deployment

- Mediates interactions between a resource and workload/data management
- Encapsulates resource heterogeneity, presenting uniform interface to workload management and monitoring
- Incorporates via plugins the particular behaviour and functionality needed by a resource
- Particularly suited to complex HPCs but also used across the grid
- Experiment agnostic, with first usage beyond ATLAS recently reported (ASGC)



# Leveraging exascale + accelerators



- **Training and optimizing deep learning neural networks** looks to be the only route for ATLAS to exploit exascale early (Summit now and new machines as they arrive).
  - **Fast simulation**, and particularly **fast chain** (fast all the way to analysis outputs)
  - **Analysis groups** already using ML
- We can expect ML applications will be well supported on future machines
  - Whatever their exotic architecture, they'll be ML engines
- Working on **scaling ML applications with PanDA** to utilize large scale resources in order to minimize turnaround time in development and tuning ML apps
  - **Distributed training and hyperparameter optimization** to achieve fast turnaround
    - Spiking workloads onto an HPC and/or across the grid for rapid turnaround
      - There's a productivity premium on rapid turnaround: physicists are waiting for the result
      - Distinctly different usage mode from steady state production, leveraging HPCs to empower developers of ML applications
    - PanDA provides large scale orchestration with management of data flows and metadata
  - Support this usage at user level, not just managed production
- In development on the grid and HPCs (Summit)
- Complement to this: PanDA now supports **concurrent CPU and GPU payloads**
  - When running GPU heavy apps like ML, complement with concurrent CPU apps to more fully utilize the resource