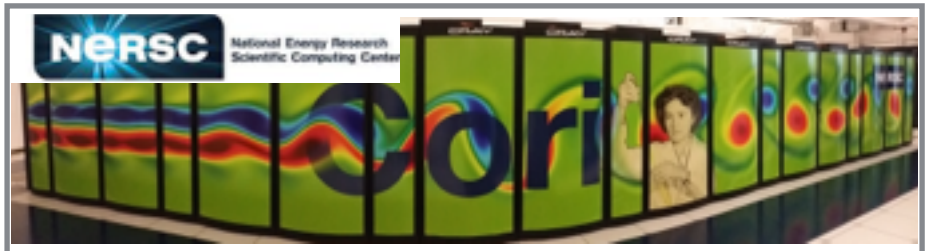
DOE HPC Integration Summary

J. Taylor Childers (Argonne)
with Doug Benjamin (DukeU)
Vakho Tsulaia (LBNL)
Wen Guan (UWisconsin)
Danila Oleynik (BNL)

US HPC Facilities

- 48k Nodes: 64 threads, 16GB each
- 1.6 GHz BlueGeneQ PowerPC
- 3.1M parallel threads possible
- 6.8B core-hours/year (Grid ~2.5B/year)

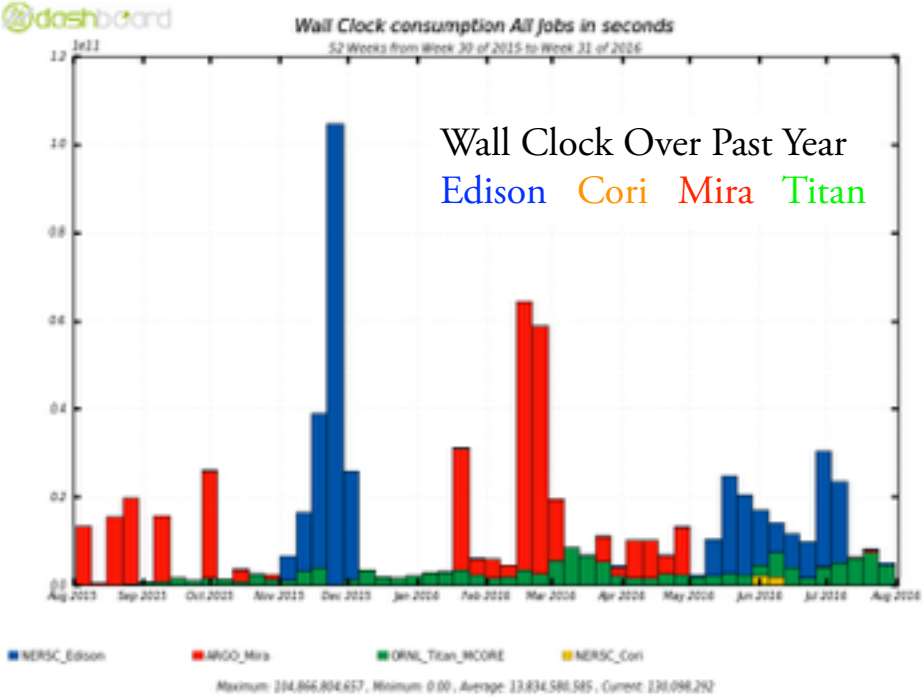


- 9,304 nodes: 68 cores x 4 HW threads (272 threads/node)
- Intel Xeon Phi (Knights Landing)
- 16GB on-chip memory
- 96 GB DDR4 2133 MHz

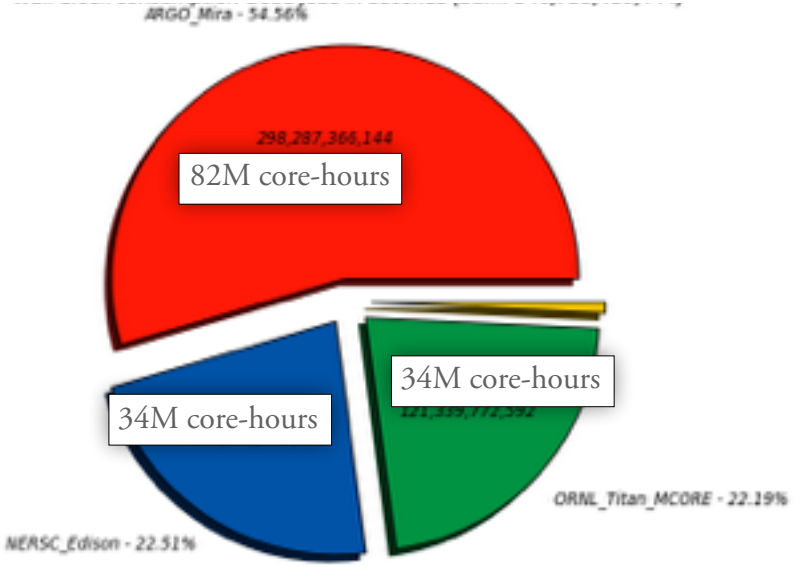
- 18,688 nodes: 16 CPU cores, 1 NVIDIA Kepler GPU
- 2.2GHz AMD Opteron with 32GB
- 6GB RAM on GPU
- 2.6B CPU-core-hours/year

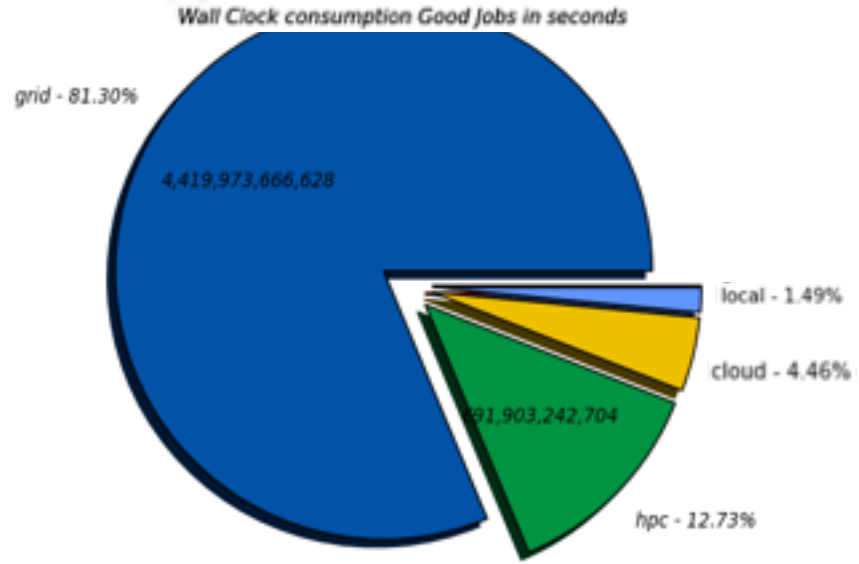
OLCF
Oak Ridge Leadership Computing Facility

US HPC Facilities: Usage 1 Aug 2015 - 1 Aug 2016

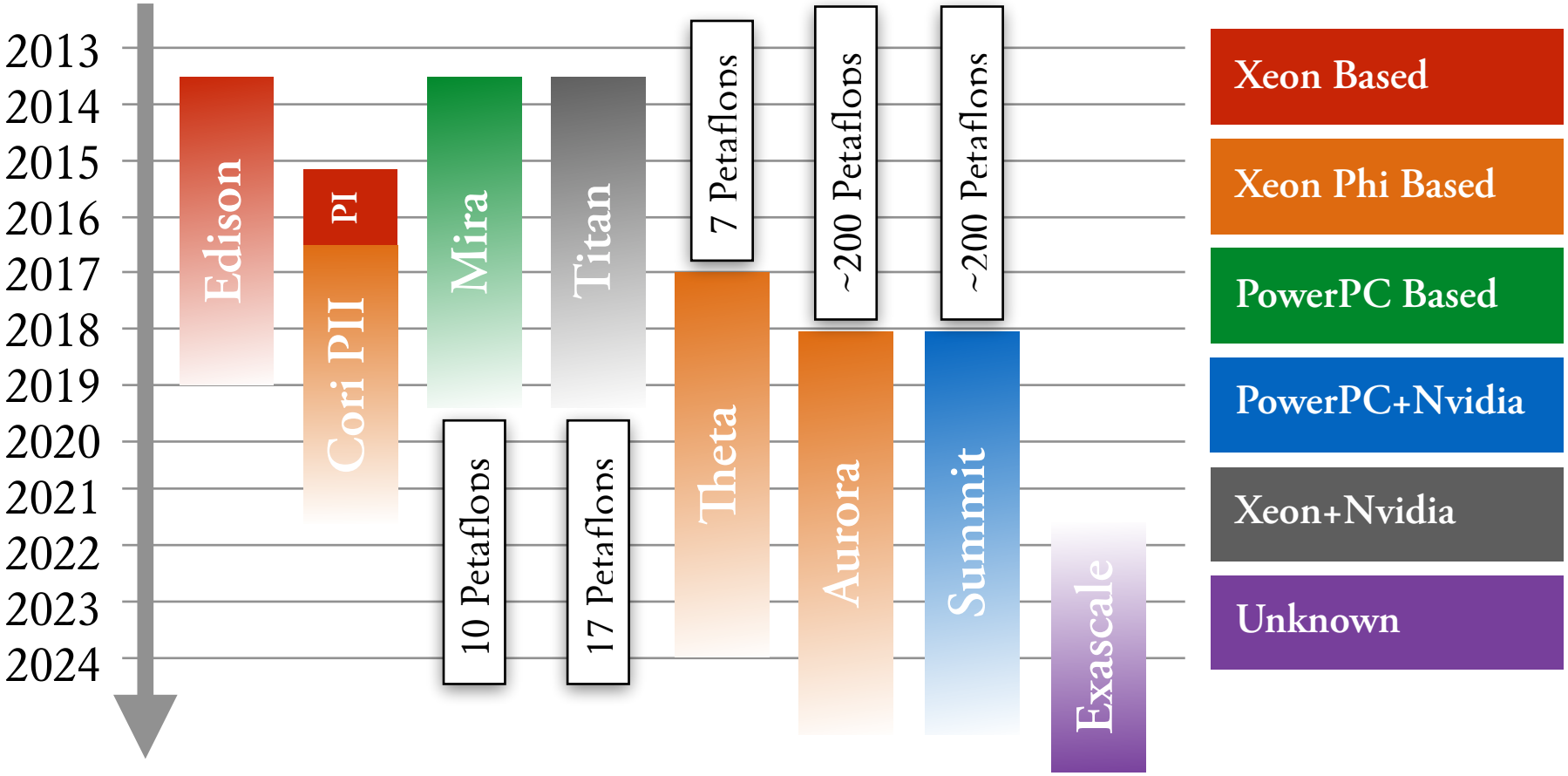


150M core-hours provided to ATLAS

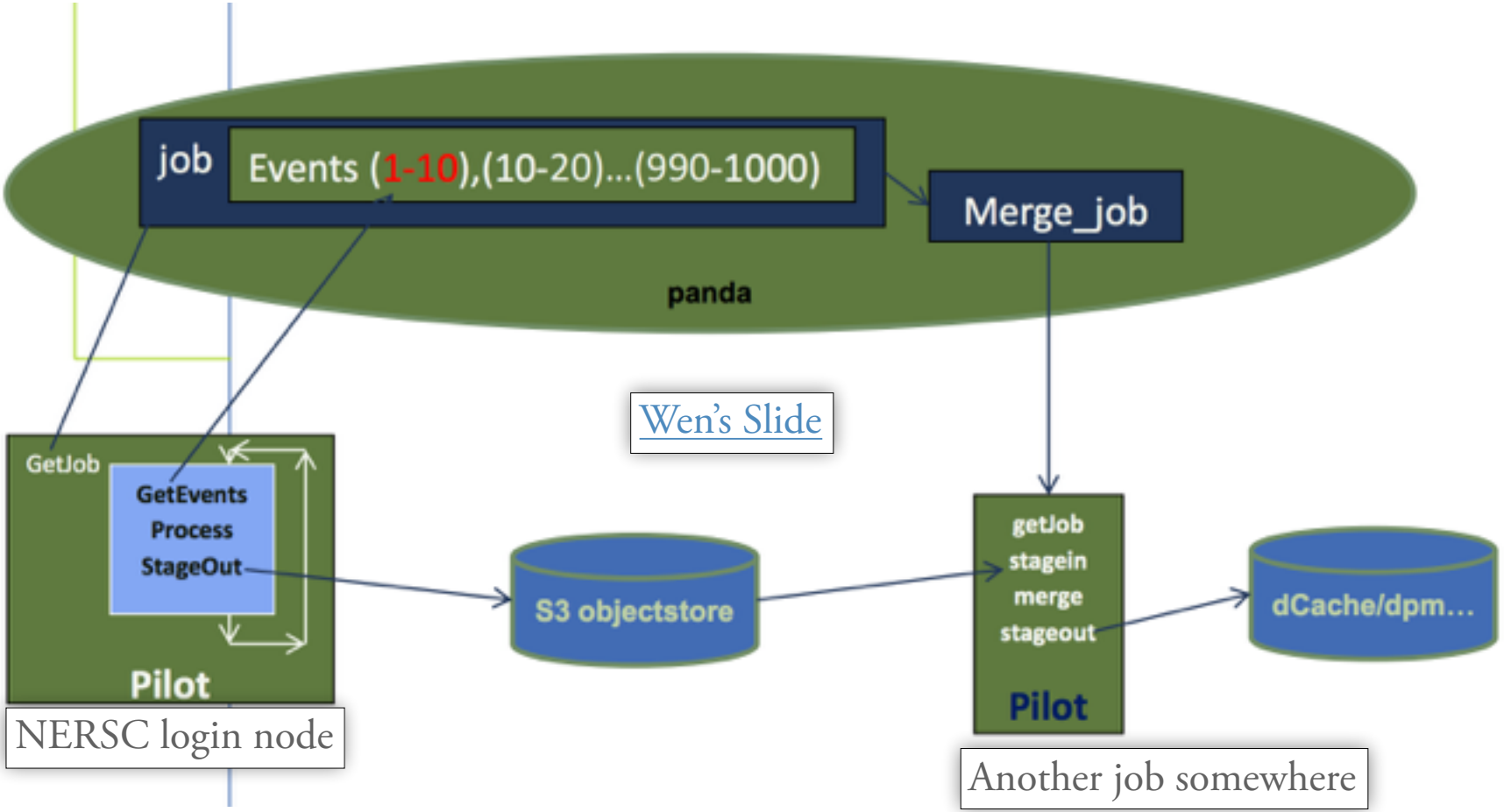




US HPC Facilities: Past & Future

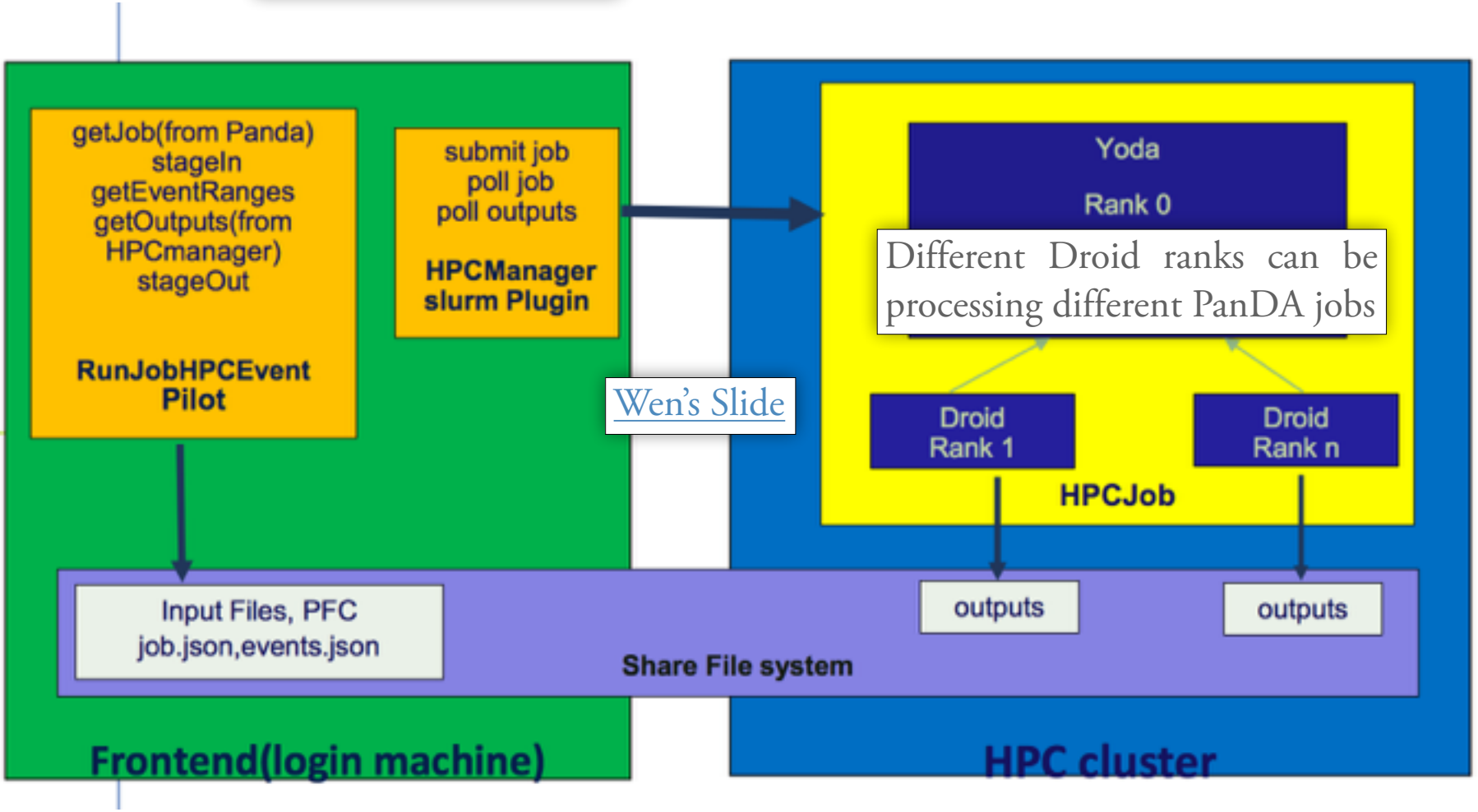


PanDA Integration: NERSC



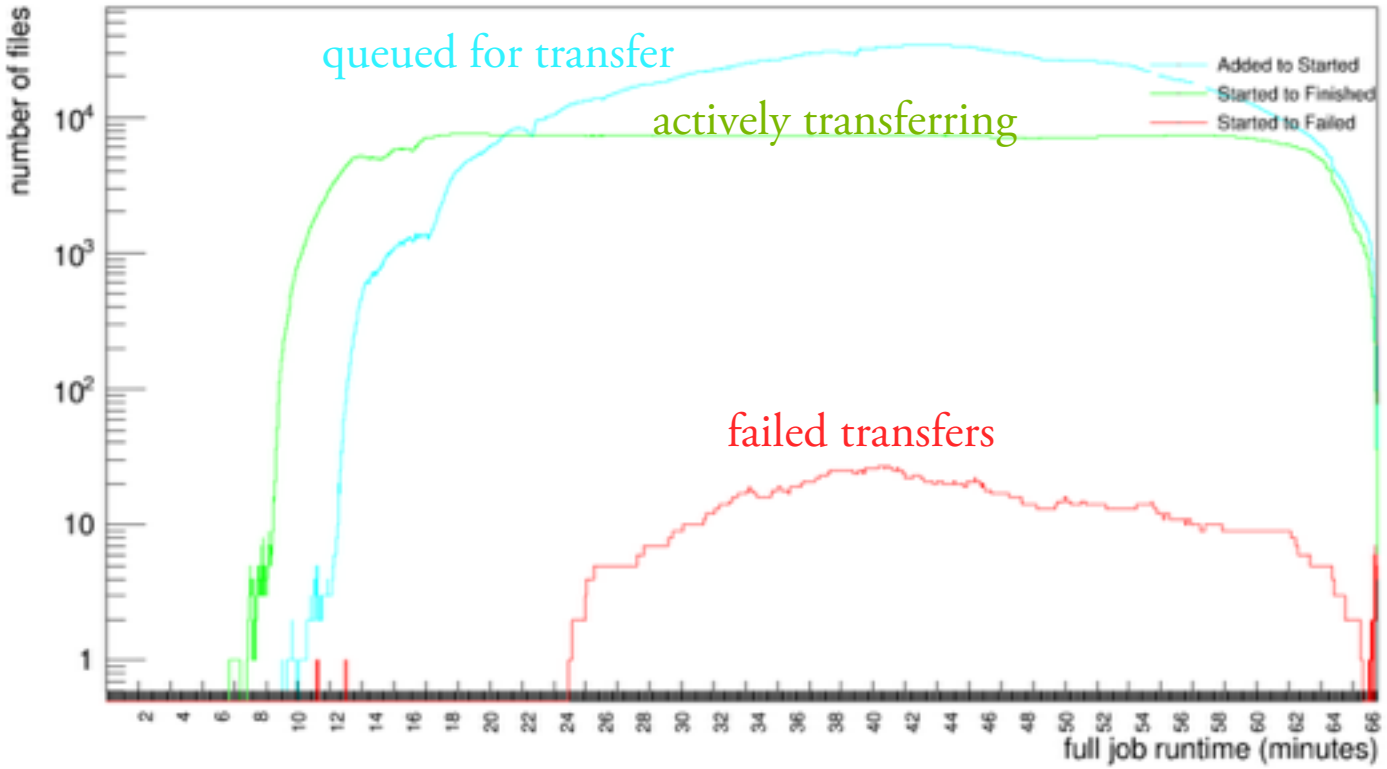
PanDA Integration: NERSC

Zoom in on HPC-side



PanDA Integration: NERSC Challenges

- We've saturated the BNL Object Store (OS)
  - 400 node job saturated OS, transfers back up, transfers fail
  - 1 transfer = 1 event
  - Failed transfers represent simulated events that are lost and must be redone



PanDA Integration: NERSC Challenges

- We've saturated the BNL Object Store (OS)
  - 400 node job saturated OS, transfers back up, transfers fail
  - 1 transfer = 1 event
  - Failed transfers represent simulated events that are lost and must be redone
- Doug found that our CPU efficiency for these processes is very low
  - Efficiencies calculated from Droid logs and reported by SLURM are similar

Doug's Talk

job 1373735 - 69.2% (697 nodes)
job 1448750 - 45.2% (700 nodes)
job 1457725 - 46.1% (699 nodes)

job 1459947 - 83.24 % (100 nodes)
job 1460498 - 69.75 % (100 nodes)

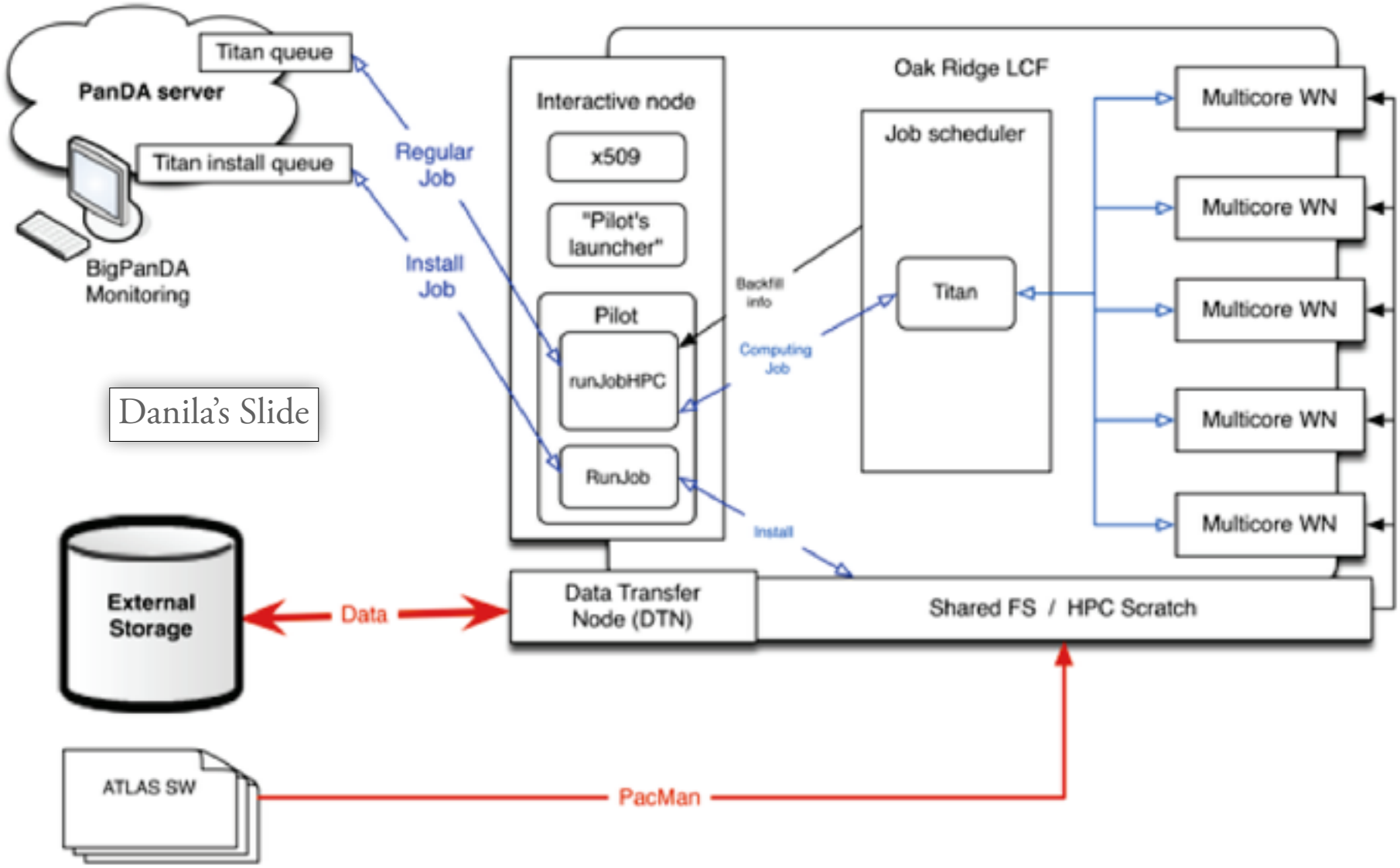
PanDA Integration: NERSC Challenges

- We've saturated the BNL Object Store (OS)
  - 400 node job saturated OS, transfers back up, transfers fail
  - 1 transfer = 1 event
  - Failed transfers represent simulated events that are lost and must be redone
- Doug found that our CPU efficiency for these processes is very low
  - Efficiencies calculated from Droid logs and reported by SLURM are similar
- Addressing these challenges by removing per-node Object Store transfers
  - run single transfer daemon on login node instead
  - handle larger files: 1 file per athena rank, 1 file per node?
  - For the moment, target tar-balling output files and gridftp to BNLT1/MWT2 for Object Store merger.
- and investigating where the CPU inefficiencies arise
  - Vakho suggested may be related to running over many PanDA jobs per node
  - Doug/Taylor trying to identify which step is inefficient, i.e. during or between event simulation

PanDA Integration: ALCF

- Mira will not be integrated to PanDA
  - There is the PowerPC compilation
  - Mira will only be around another 3-4 years
- We can still benefit from running Generators
  - Working on Sherpa optimization
  - Next up MadGraph
  - these cover the two biggest generators for ATLAS
- Theta is the Aurora test system with the same computing capacity as Mira and the same architecture as Cori Phase-II
- Benefit from the work done at NERSC to deploy Yoda/Droid on Theta as soon as we can get access
- Working with NERSC team to ensure solution is ALCF compatible
- Then deploy on Aurora when it is installed Q4-2017

PanDA Integration: OLCF



PanDA Integration: OLCF

- Similar CPU efficiencies as at NERSC, average 65% +/- 20%
- Pilot-to-wrapper workflow dependent on short queue time, such that it virtually fulfills PanDA's late binding requirement
  - This could become a problem later if backfill hours become scarce or system admin changes
- Titan has a similar lifetime as Mira and
- will be replaced with a PowerPC+Nvidia machine in 1-2 years making its future less certain within ProdSys

Commonalities - Differences

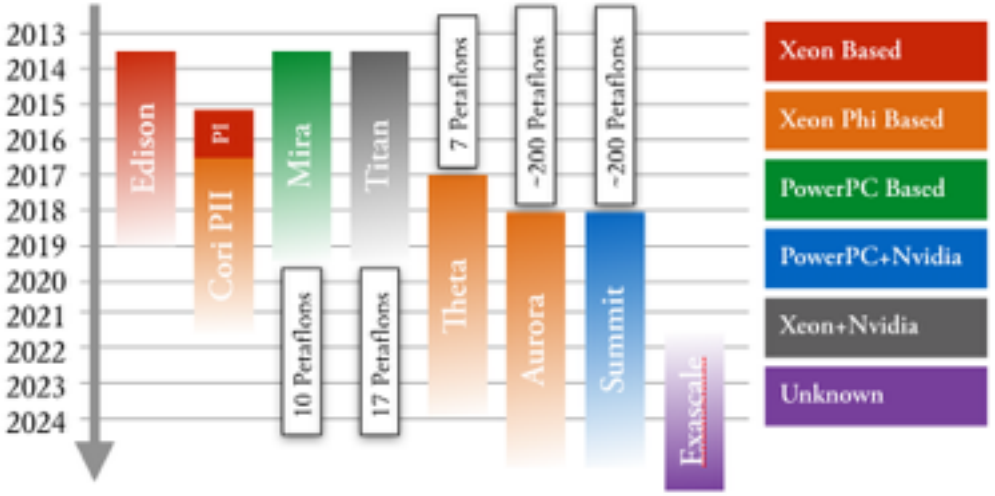
- Both solutions do the following:
  - employ some MPI wrapper launching 1 AthenaMP per node (with as many ranks as cores)
  - run a pilot independent of the HPC job payload
  - pilot retrieves multiple PanDA jobs
- Differences:
  - MPI wrappers: Yoda+Droid (python) @NERSC vs. C++ @OLCF
  - Yoda+Droid = PanDA jobs split across nodes, OLCF C++ = 1 PanDA job per node
  - Data Transfer mechanisms, Object Store vs. Pilot movers

Recommendations for Moving Forward with Common Solutions

- Common Data Transfers Tools
  - NERSC, ALCF, OLCF support Globus Online with gridftp tools to transfer data through dedicated Data Transfer Nodes (DTNs) with high performance
  - Need common API that can interface to DTNs
  - Using DTNs guarantees performance and support from local admins
- Common MPI wrappers
  - PanDA team is supporting Yoda+Droid
- Common Local Queue API
  - SAGA is a supported API that fulfills this.
  - Already part of the Pilot
  - Being harvested by Harvester?
- Common Pilot
  - Currently have one for Titan, one for NERSC

Why we need Common Solutions

- FTEs:
  - NERSC 1.5 FTEs:
    - Vakho 25%
    - Wen 50%
    - Taylor 30% (last two months)
    - Doug 50% (last two months)
  - ALCF 0.2 FTEs:
    - Taylor <15%
    - Doug <5%
  - OLCF 0.1 FTEs:
    - Danila 5-10%



- NERSC FTE is high as it is current test bed for the new Yoda+Droid solution
- LCFs are past their big development period for the current machines, but in the next year, with new machines coming online, effort will increase again.
- Using common tools means a common team can support a common solution across the sites
- Have had 3 teams supporting 3 solutions at 3 sites
- Recently consolidated ALCF team and NERSC team

Why we need Common Solutions

- FTEs:
  - NERSC 1.5 FTEs:
    - Vakho 25%
    - Wen 50%
    - Taylor 30% (last two months)
    - Doug 50% (last two months)
  - ALCF 0.2 FTEs:
    - Taylor <15%
    - Doug <5%
  - OLCF 0.1 FTEs:
    - Danila 5-10%

With FTE=$300k
Total FTEs * $300k / past year's delivered core-hours

$7k per million core-hours

$0.75k per million core-hours

$0.89k per million core-hours

- NERSC FTE is high as it is current test bed for the new Yoda+Droid solution
- LCFs are past their big development period for the current machines, but in the next year, with new machines coming online, effort will increase again.
- Using common tools means a common team can support a common solution across the sites
- Have had 3 teams supporting 3 solutions at 3 sites
- Recently consolidated ALCF team and NERSC team

Using Common Tools is Unnatural