NERSC and the ALICE Computing Grid:

Challenges for integrating NERSC resources into an existing distributed and automated data processing model

R. Jeff Porter (LBNL)

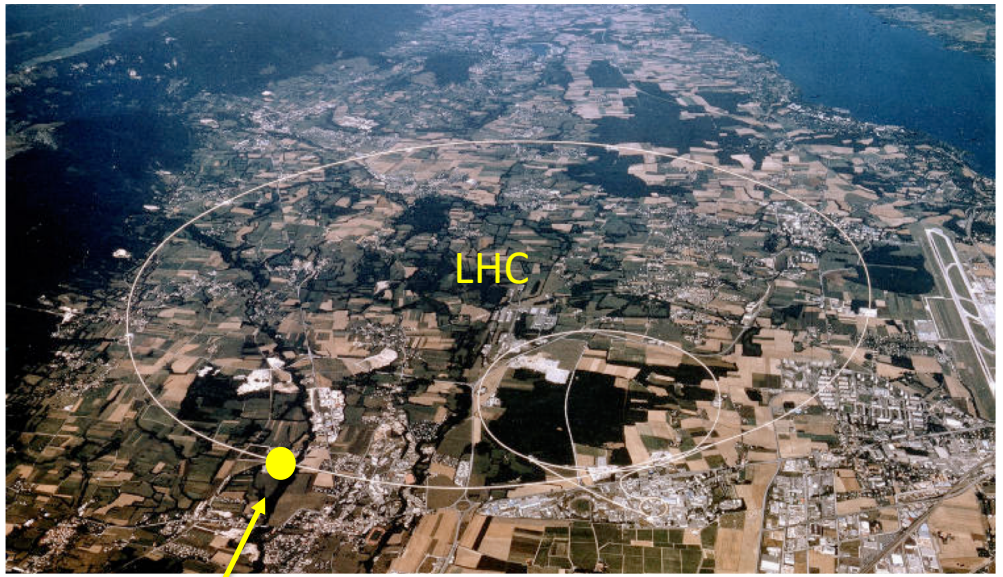
NERSC SIG on Experimental Facilities

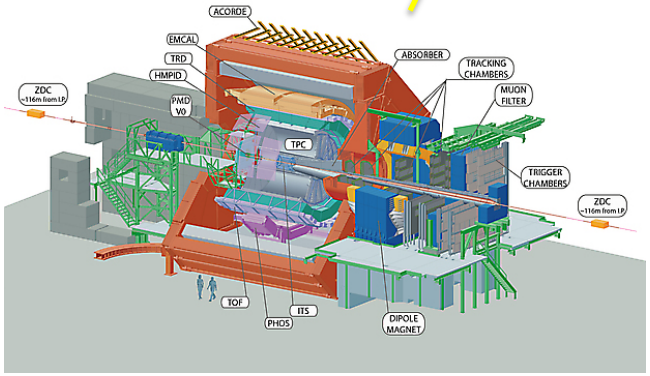
July 1, 2020

U.S. DEPARTMENT OF ENERGY | Office of Science

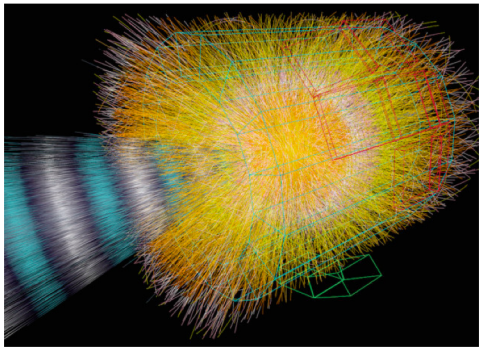
Outline

- **ALICE @ NERSC**
- **Grid Computing Model**
- **NERSC & the ALICE Grid**
  - Running jobs
  - Accessing storage
- **Outlook**



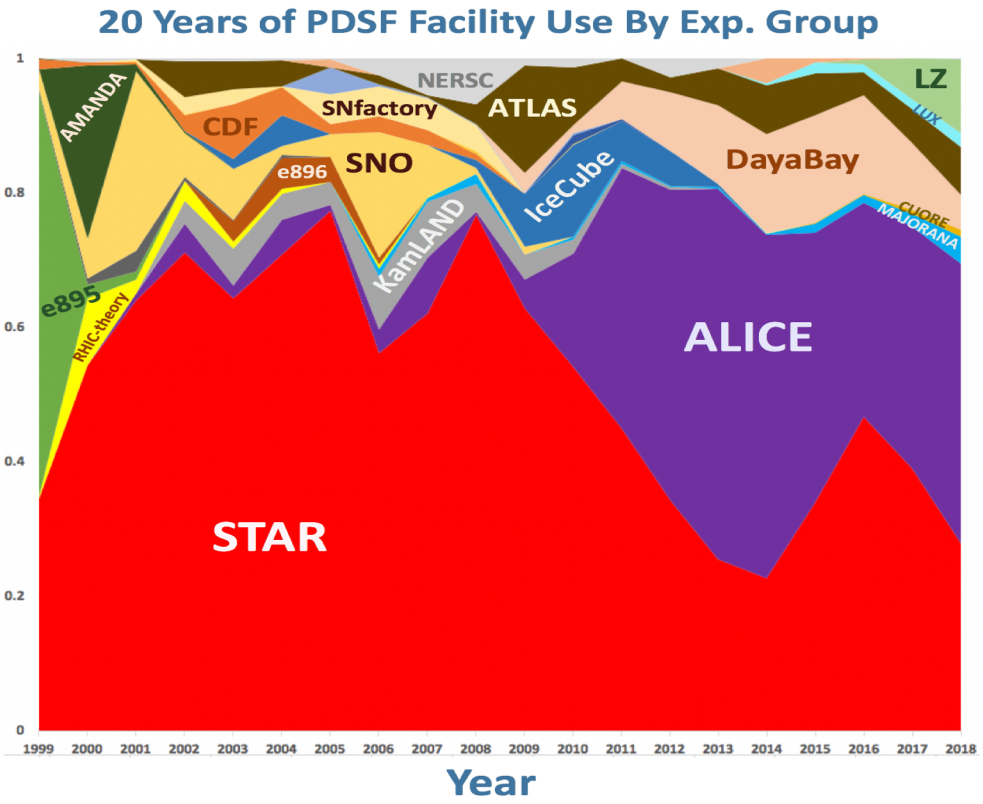


Pb+Pb collision in ALICE



ALICE history with NERSC

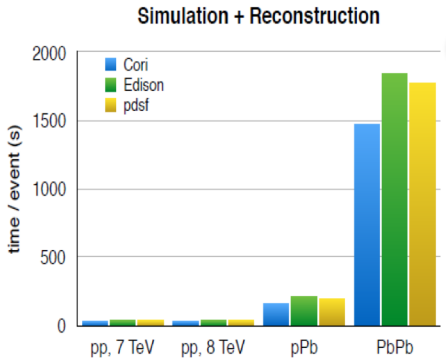
- ALICE has made use of NERSC resources for more than 15 years, mostly on PDSF
- The arrival of CORI Phase I in 2015 motivated us to attempt to migrate work on PDSF onto CORI



ALICE R&D activities on CORI

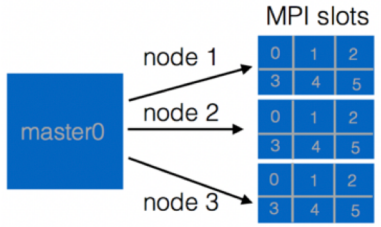
- **LBNL ALICE group proactively R&D'd different models for using CORI**

- **4 years later, CORI is lightly used by ALICE**
  - mainly by local group for one-off tasks
  - Remains an outlier resource in ALICE



M. Fasel, J. Porter
ACAT 2016

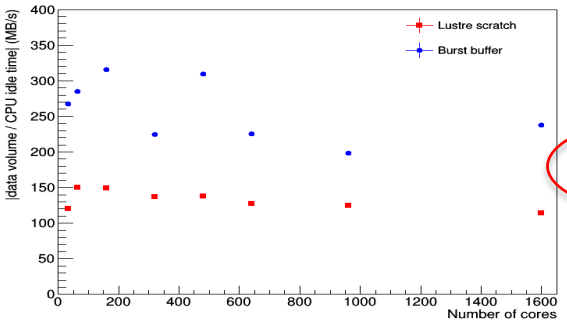
Benchmarked with real ALICE workloads



Built tool to pack serial jobs into multi-node jobs

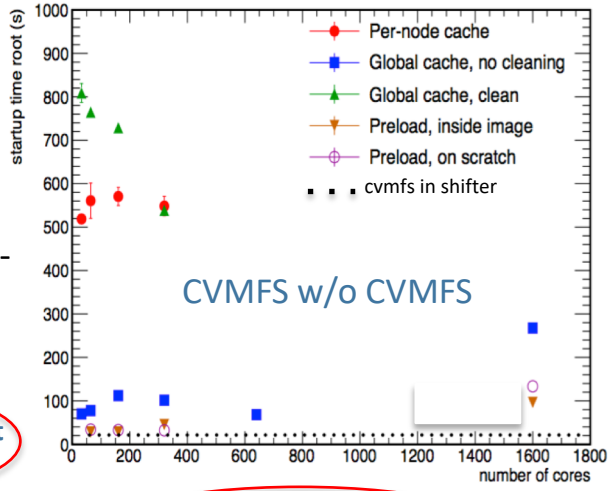
- **What about the direct integration of NERSC into the ALICE computing model?**

Scaling large I/O with burst buffer



Contrib. to W. Bhimji
CHEP 2016

Scaling with Shifter + non-standard CVMFS modes



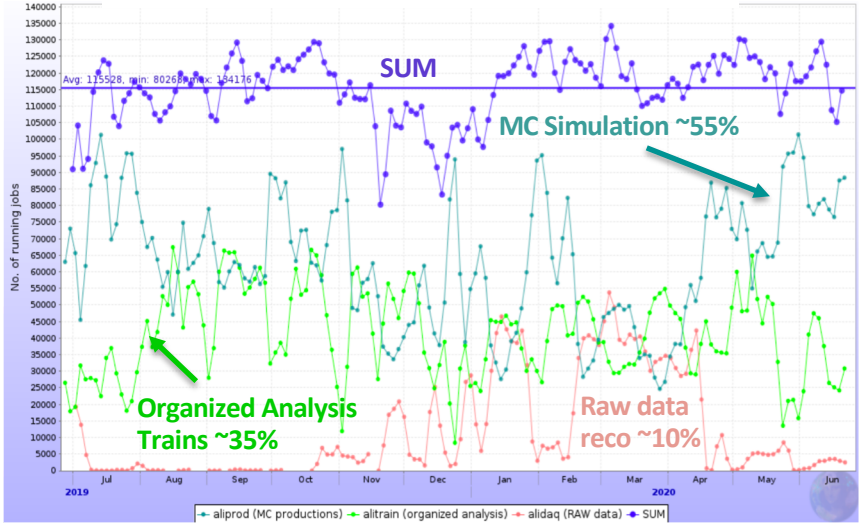
Contrib. to L. Gerhardt
CHEP 2016

M. Fasel, J. Porter
CERNVM Workshop, RAL 2016

ALICE Computing Model

- **Grid Facility & Processing model**
  - ~80 active sites
    - CERN Tier-0, 8 Tier-1s, ~70 Tier-2 (T2)
  - >120,000 serial jobs, 24 x 365
    - Fully automated, resubmit on job failures
  - 110 PB distributed disk storage
    - Data distributed to multiple sites
    - Jobs run where the data is
  - AliEn: software to connect distributed resources into single facility

Running Jobs over past 12 months

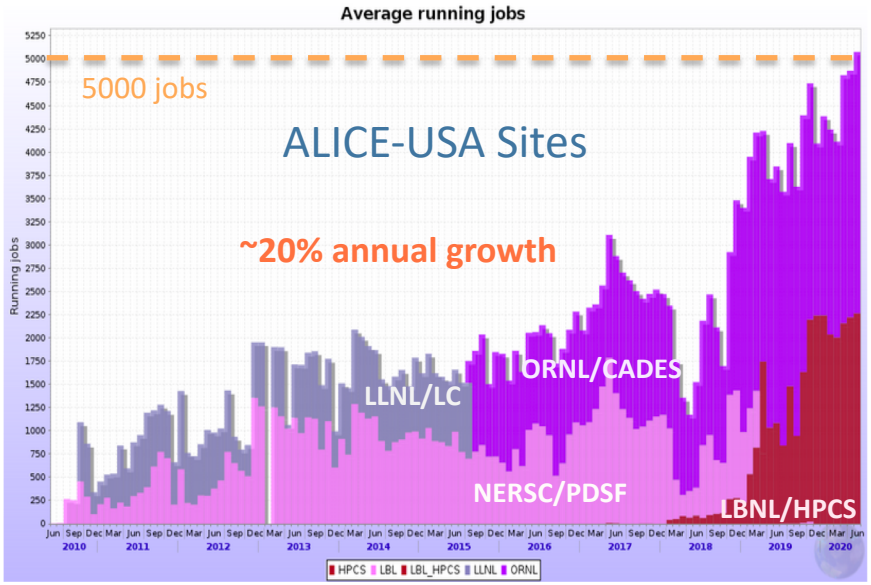


- **Every T2 site supports ~90% of job types**
  - MC simulation jobs:
  - Organized analysis trains

ALICE-USA Computing Project

- **Project Launched in 2009**
  - Supply grid-enabled resources to fulfill MoU-based US computing obligations to ALICE
  - Operate T2 facilities at 2 DOE labs
    - NERSC/PDSF @ LBNL
    - Livermore Computing @ LLNL
  - LBNL as the host institution

- **During operations since 2010**
  - Twice decommissioned & replaced T2s
  - LLNL/LC → ORNL/CADES in 2015
  - NERSC/PDSF → LBNL/HPCS in 2019



concurrent running jobs: 2010-2020

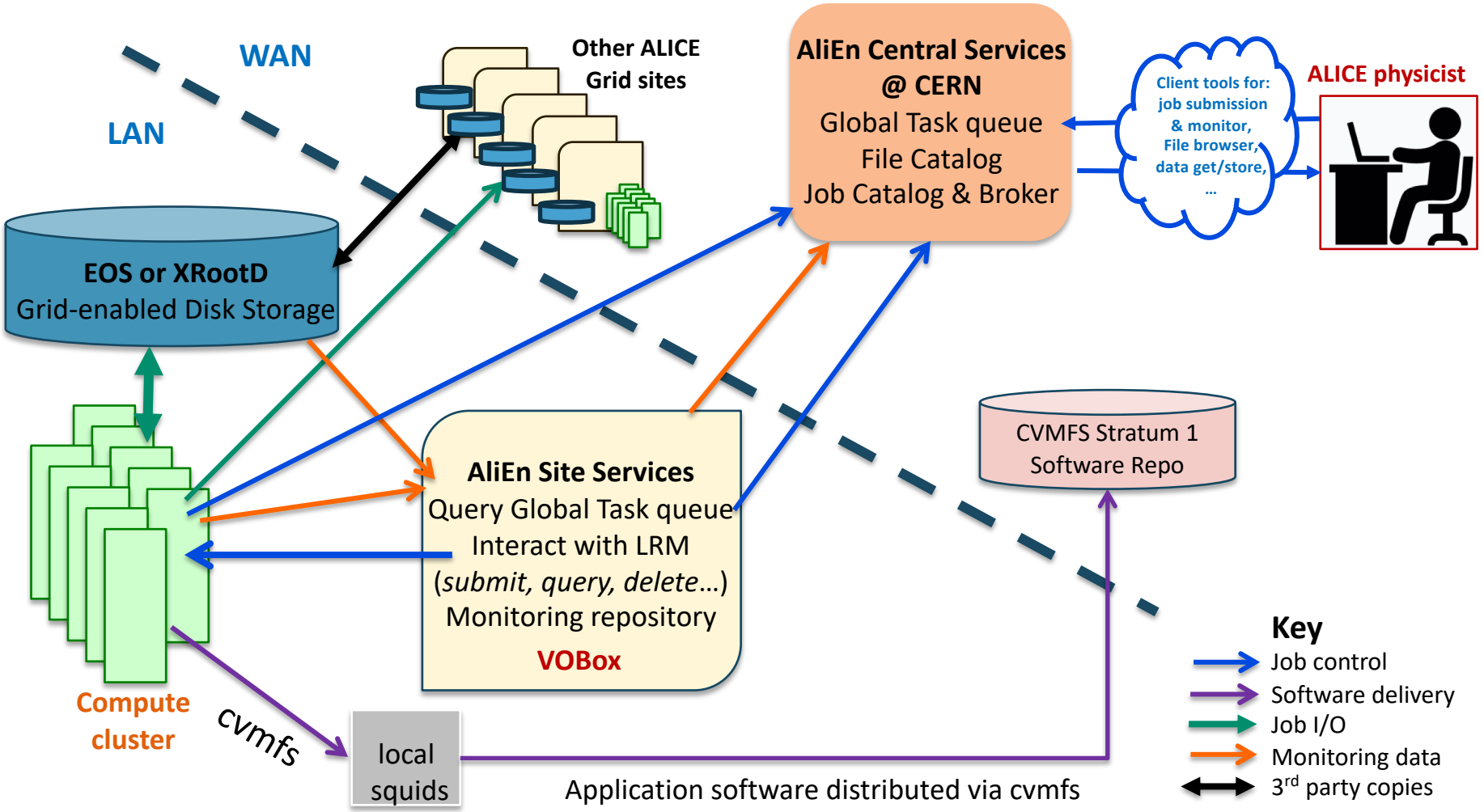
- **Current Facilities:**
  - LBNL/HPCS: 2300 CPU cores, 3.0 PB Storage Element (SE)
  - ORNL/CADES: 2800 CPU cores, 3.0 PB SE

Site requirements to plug into the Alice Grid

- **Node level**
  - any modern Linux distribution
  - memory capacity of ~2.5 GB/useable-core
    - Significant swap enabled is highly desirable
  - outgoing network connectivity from worker node
  - local disk (or performant scratch) for small block I/O
  - CVMFS for software distribution

- **Facility Level**
  - workflow node (VOBox in WLCG-speak) as site point of contact
  - most any LRM: LSF, PBS, SGE, SLURM, HTCondor, …, ARC-CE, OSG-CE, …
  - optimally configured for serial jobs
  - large long term disk storage:
    - Grid enabled with EOS or XRootD
    - Incoming network with ALICE Token AuthN

ALICE Grid & Site Topology



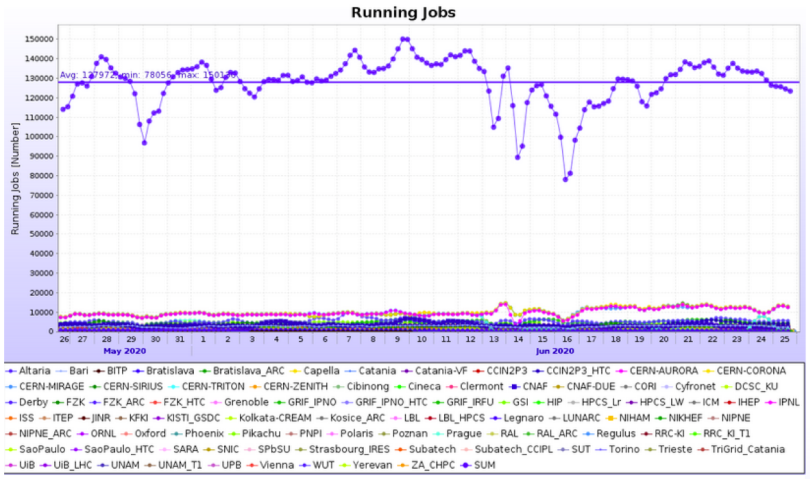
Site requirements & NERSC CORI

- **Node level**
  - ✓ any modern Linux distribution — we use a thin shifter image + CVMFS
  - ✓ memory capacity of ~2.5 GB/useable-core
    - ✗ Swap enabled is highly desirable ← this limits our full use of CPU cores
  - ✓ outgoing network connectivity from worker node
  - ✓ local disk (or performant scratch) for small block I/O — Shifter's per-node-cache
  - ✓ CVMFS for software distribution

- **Facility Level**
  - ✓ workflow node (VOBox in WLCG-speak) as site point of contact
  - ✓ most any LRM: LSF, PBS, SGE, SLURM, HTCondor, …, ARC-CE, OSG-CE, …
  - ✗ optimally configured for serial jobs
  - ✗ large long term disk storage:
    - Grid enabled with EOS or XRootD
    - Incoming network with ALICE Token AuthN

} We can try to address these

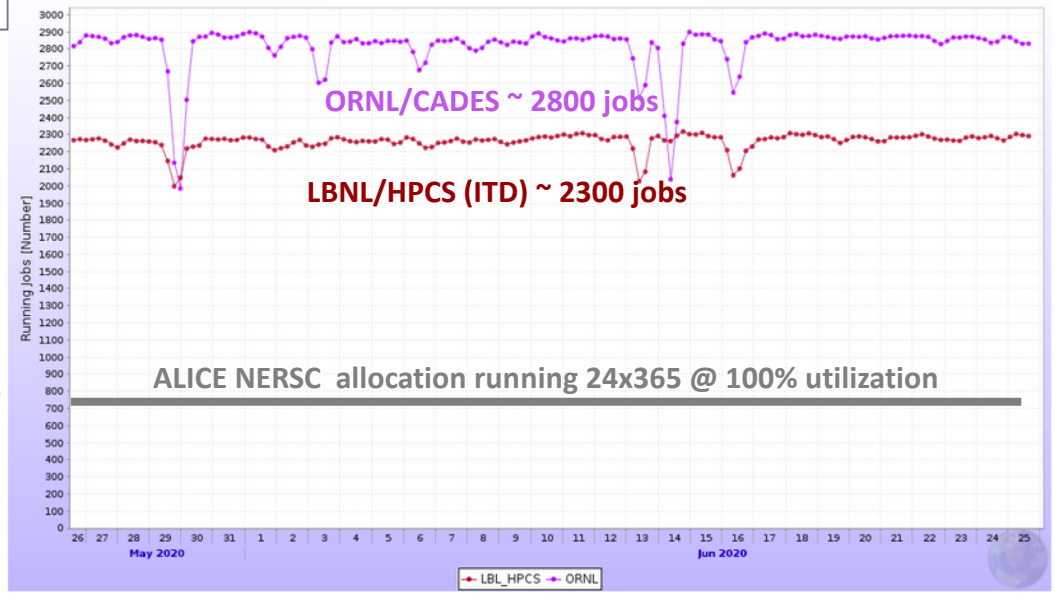
Context for changing ALICE Grid services



ALICE Grid ~ 130,000 jobs

Goal must be to extend ALICE grid sites services in a non-disruptive way (i.e. no specialized, local maintenance required), retaining fully automated workflow

ALICE USA ~5000 jobs



Scale of current allocation →

~0.5% ALICE Grid CPU

~10% ALICE-USA T2s

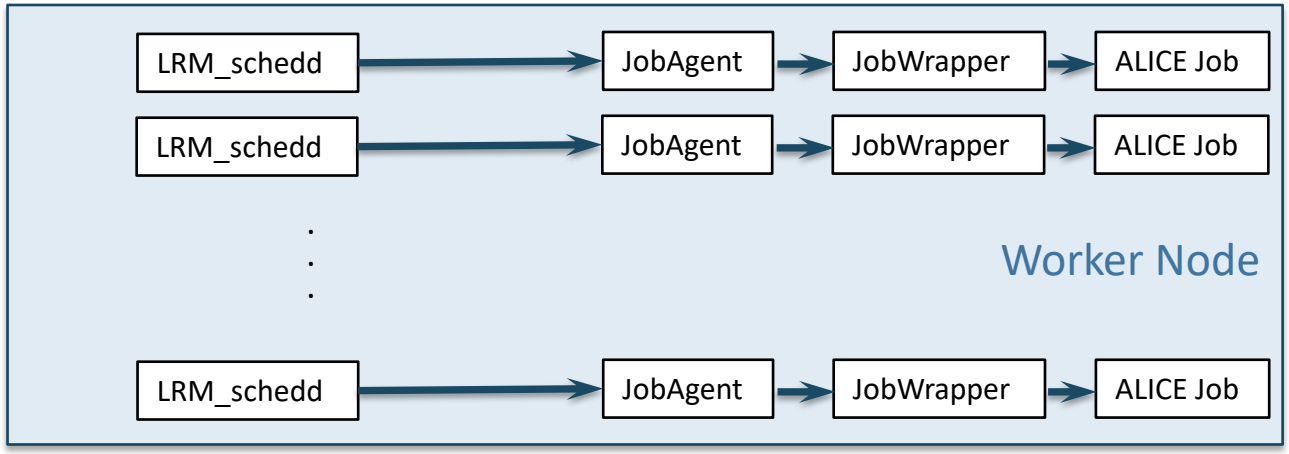
AliEn (Alice Environment) → jAliEn

- **Complete rewrite of code base began in 2017**
  - Legacy Perl, C, C++ consolidated into a set of Java packages
  - Retain AliEn functionality, remove cruft & add flexibility for the future

- **Opportunity for including HPC-friendly features**
  - Hosted* two CS graduate students at LBNL from UPB (Bucharest)
    - Sergiu Weisz: whole(multi)-node scheduling and SLURM integration
    - Mihai Popescu: enhanced monitoring and XRootD Proxy R&D
    - 3 months at LBNL + 9 months back at UPB
  - Project worked closely with ALICE jAliEn developers

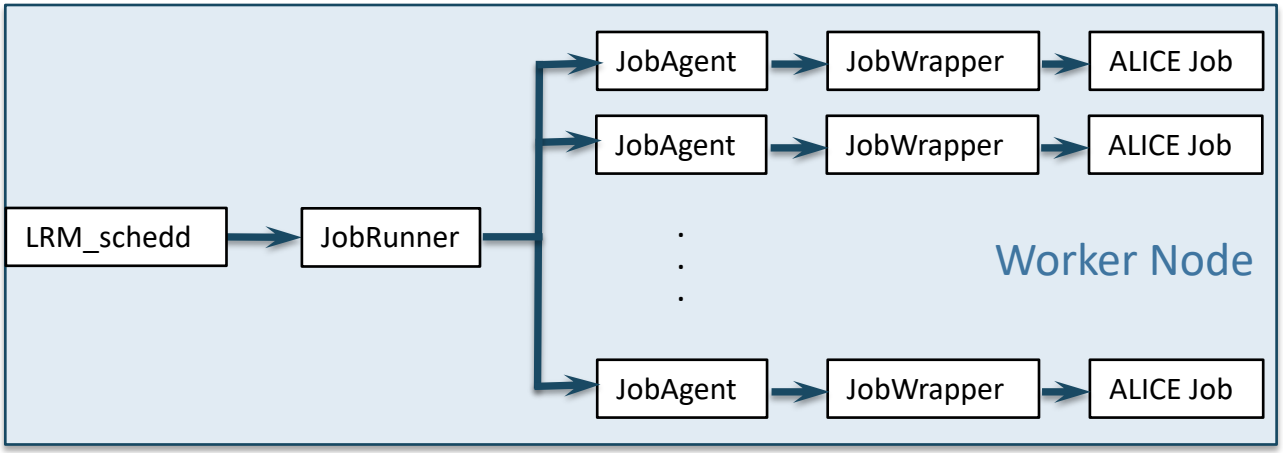
* Initial funding from LDRD with Physics (Z. Marshall)

jAliEn serial "job-level" architecture



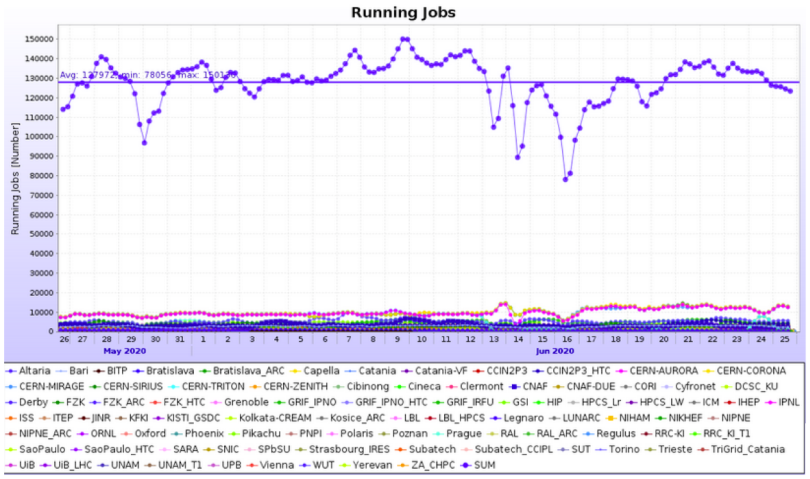
- **JobAgent:**
  - Gets full job definition from central services
  - spawns JobWrapper thread with job definition & monitors resource usage
  - Repeats when JobWrapper exits if enough time remains
- **JobWrapper**
  - prepares sandbox and launches payload (ALICE Job)
  - validates output and copies output to destination storages

Extending jAliEn: "Node-level" architecture



- **JobRunner**
  - Manages node resources and launches JobAgents as needed to optimize node usage
- **JobAgent:**
  - Gets full job definition from central services & reserves resources from JobRunner
  - spawns JobWrapper thread with job definition & monitors resource usage
  - Exits when JobWrapper exits
- **JobWrapper**
  - prepares sandbox and launches payload (ALICE Job)
  - validates output and copies output to destination storages

Initial deployment in early 2020



ALICE Grid ~ 130,000 jobs

Retained fully automated workflow

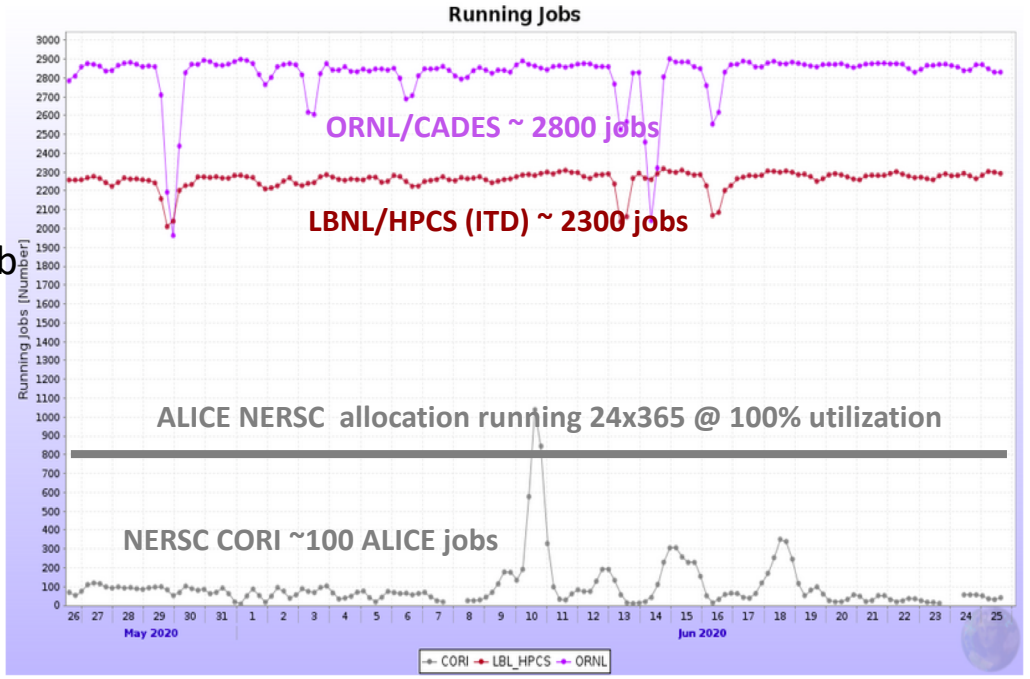
- Late-binding of job to resource
- Auto cleanup & resubmit on job failure
- Useable in serial and whole node settings

Sources of low utilization rate:

- Mem+no-swap limits use to ~45 jobs/node
- Inefficient packing into wall time of slurm job
- SLURM scheduling
  - jAliEn keeps 10 x 1-node jobs queued
  - Limited backfill with 48 hour jobs?
- High error rate (16%)

This gives us something to work with

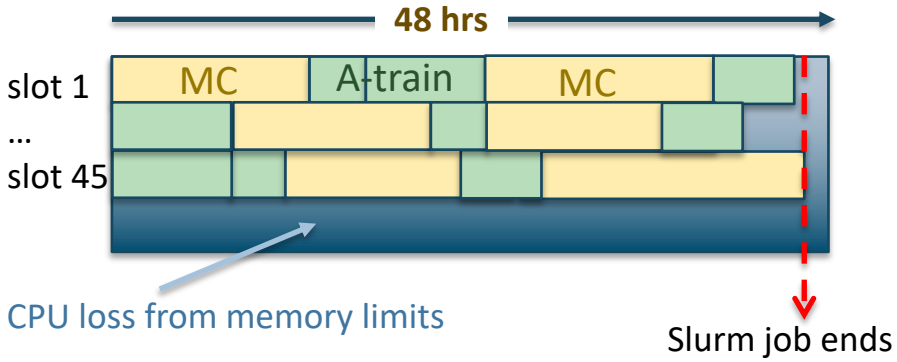
ALICE USA ~5000 jobs



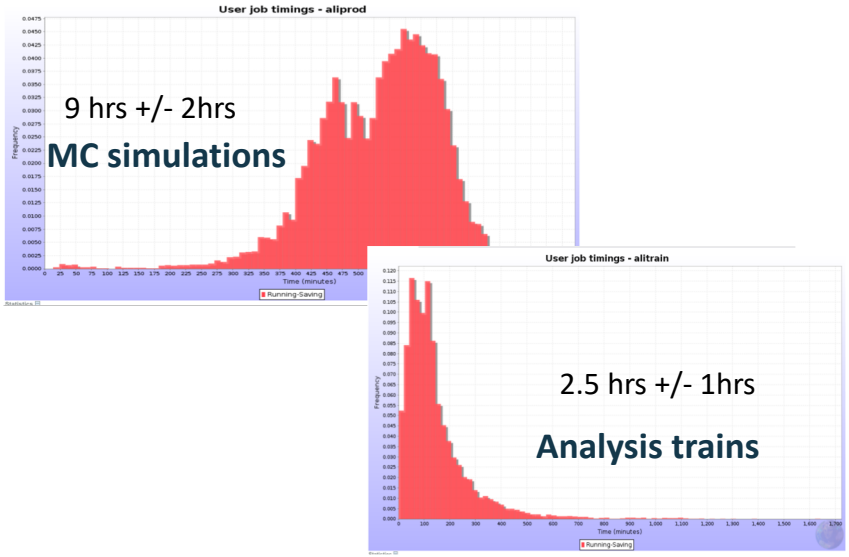
Challenges for packing SLURM job

- **2.5 GB/core memory requirement**
  - ~70% utilization is max on Haswells
- **Job mixture is dynamic**
  - ~60% MC simulations
  - ~40% analysis trains
- **Job timing estimates are hard**
  - Vary even within each category
  - Completely new jobs are run every day
  - Different configs, data sets, algorithms, train lengths
- **Job timings are routinely measured**
  - JobRunner can use that information
- **Plan to test shorter SLURM jobs (12hrs?) for enhanced backfill throughput**

JobRunner packing the node – ideal case

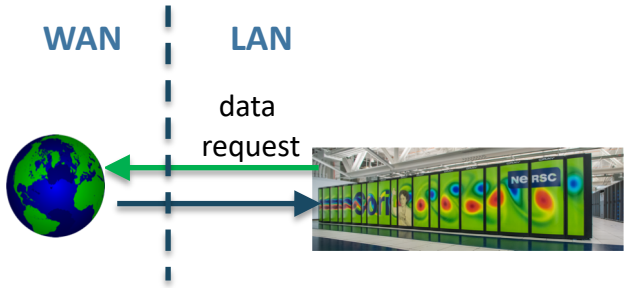


Job timing measurements



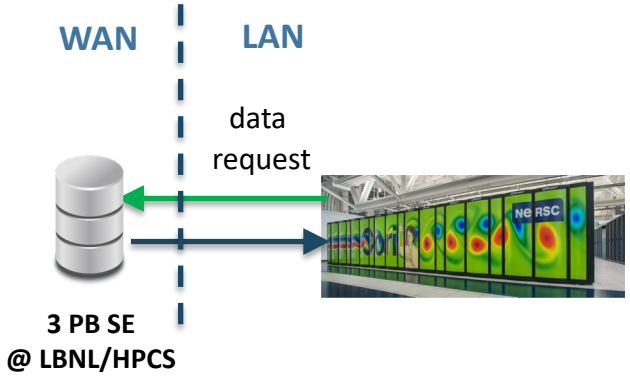
Source of high error rate

- **job timeouts pulling data from WAN**
  - No local Storage Element (SE) @ NERSC



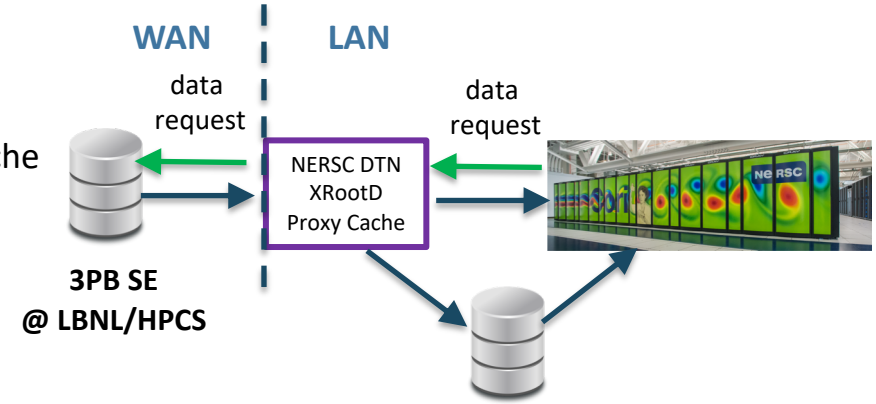
- **ALICE Storage Elements**
  - 'forever' resource: scale & grow with site CPU
  - Grid-enabled with EOS or XRootD

- **ALICE-USA T2 SE at LBNL/HPCS**
  - 3 PB SE can be preferred as 'nearby' in AliEn
  - Cori CPU becomes an extension of the LBNL T2
  - ESNet 6 may have a pairing @ LBNL

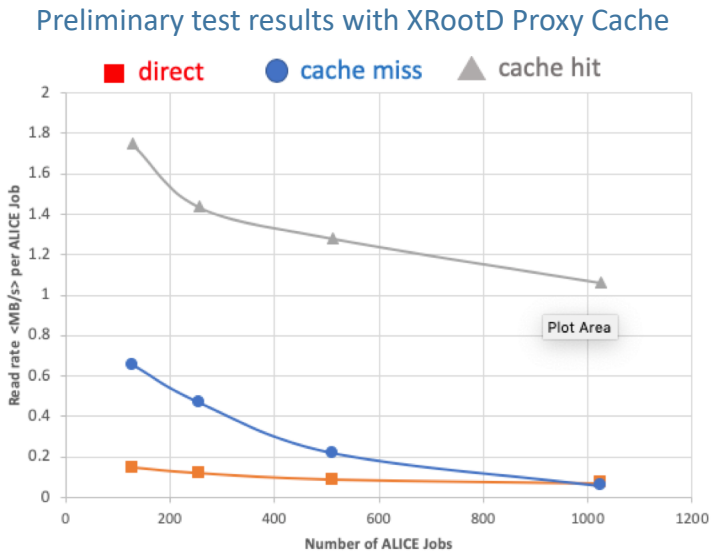


Optimize remote data access

- **XRootD Proxy Cache**
  - Deployed in user space on DTN
  - Fills request from (any) remote SE & adds data to cache
  - returns local filename to client if data is in cache



- **Initial tests use a single DTN node for XRootD Proxy Cache + (non-optimized) analysis code**
  - Cache hits show significant improvement
  - Even cache misses are improved relative to direct access



- **XRootD is highly modular and open source**
  - "Return-local-file" developed at GSI by ALICE colleagues
  - Extendable for our use case as warranted

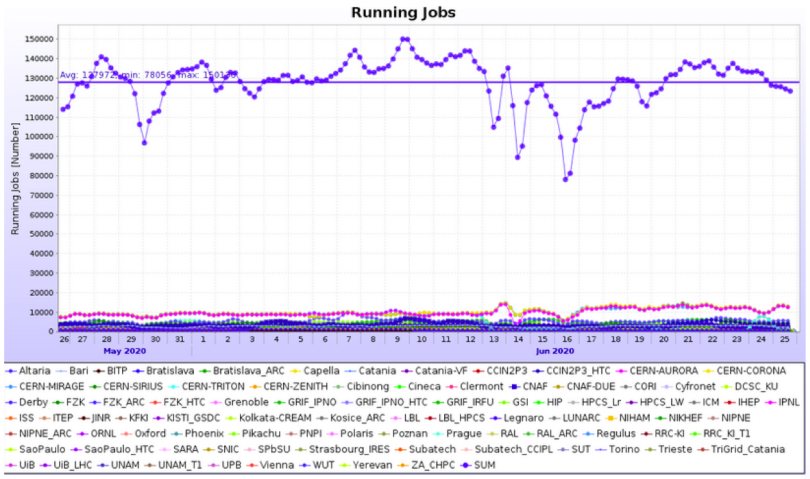
Summary & Outlook

- **Integrated NERSC resources into the ALICE Grid Facility**
  - Leveraged NERSC supplied features:
    - Outgoing network from worker nodes
    - Workflow nodes for ALICE VOBox
    - CVMFS on nodes via Shifter
  - Added whole-node scheduling to ALICE job management tools

- **Workflow retains automation though not highly efficient for using NERSC**
  - Extends without altering ALICE computing model
  - Overall throughput is low
  - Known steps may improve CPU utilization, throughput, & data access

- **Effort was ALICE development, CORI was a use case**
  - ALICE is experimenting with multi-core simulation jobs
  - Other accessible (HPC) sites have similar requirements

Features added for CORI are general: Lawrencium example

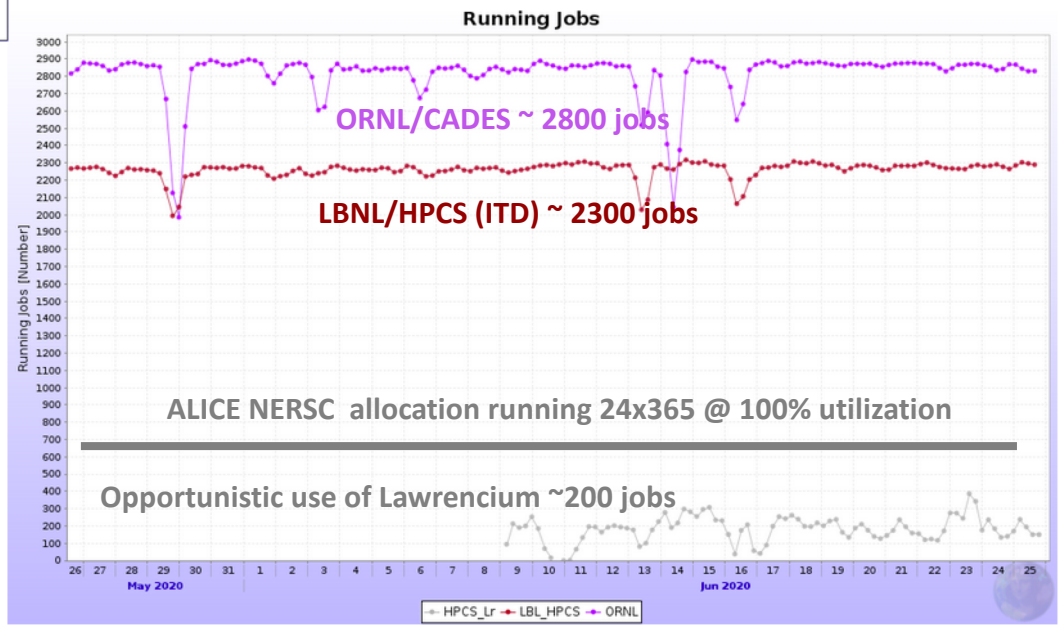


ALICE Grid ~ 130,000 jobs

Retained fully automated workflow:

- Late-binding of job to resource
- Auto cleanup & resubmit on job failure
- Useable in serial and whole node settings

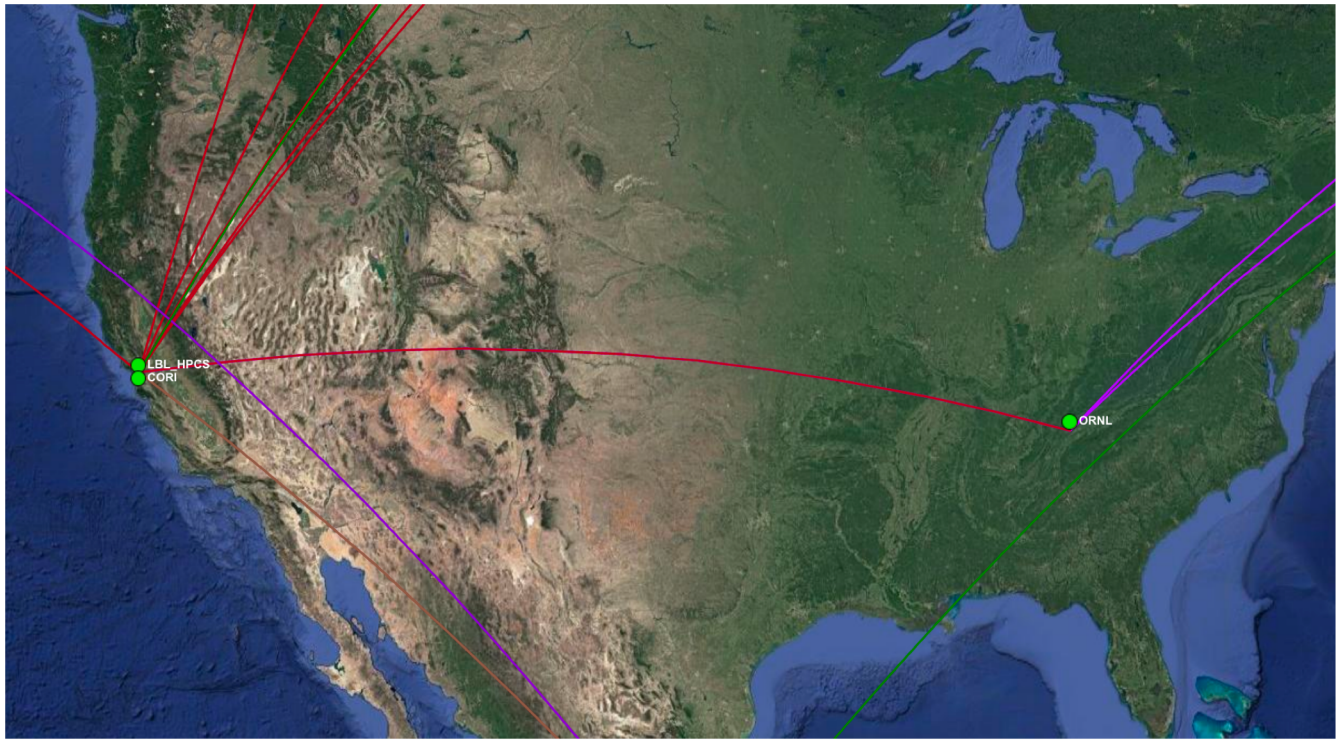
ALICE USA ~5000 jobs



Lawrencium @ LBNL

- Requires whole(multi)-node jobs
- Preemptable scavenger queue

All other site requirements are met, so we just launched it



Questions or Suggestions?