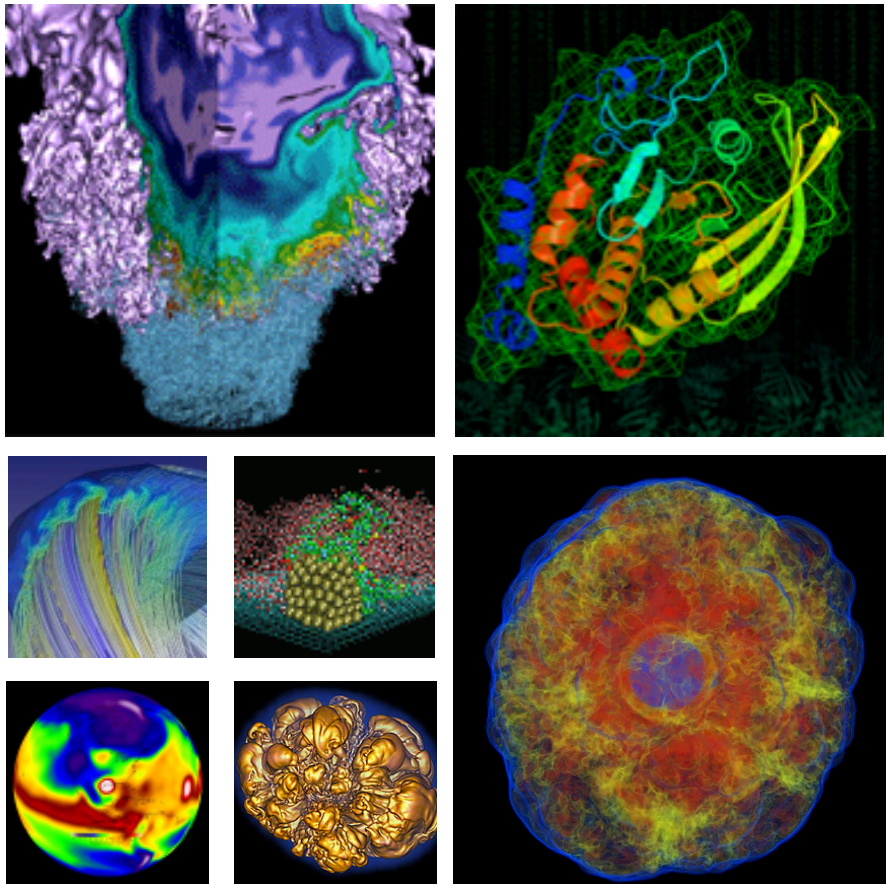
NERSC Site Report
SLUG 2017



James Botts
Douglas Jacobsen
Computational Systems Group

September 25, 2017

NERSC Vital Statistics

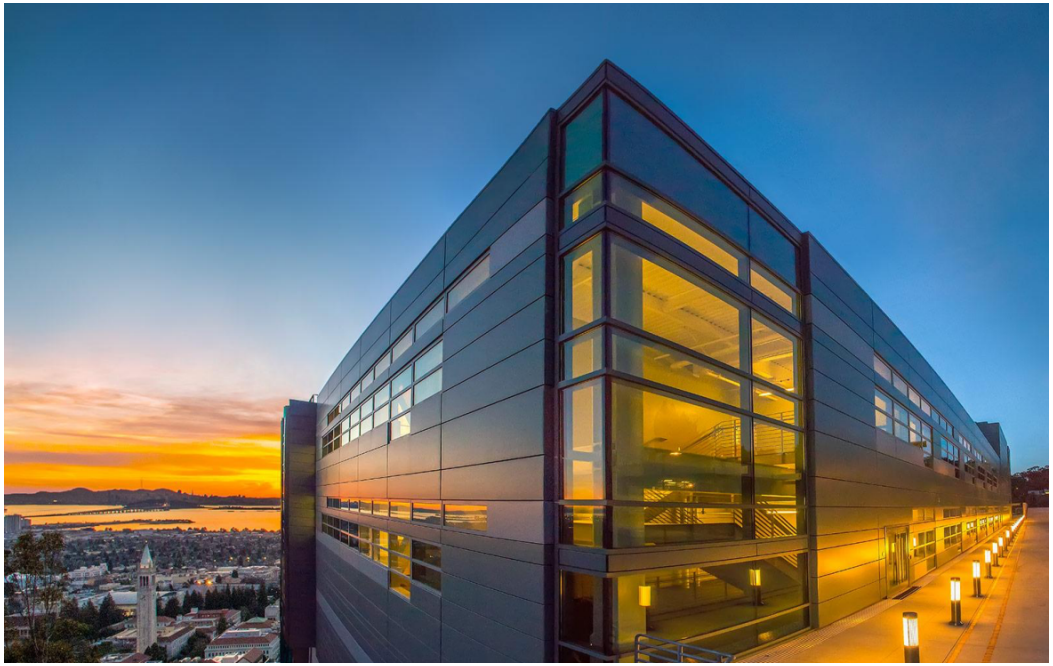
National Energy Research Scientific Computing Center
since 1974

- 860 active projects
- > 7750 active users
- 700+ codes both established and in-development
- migrated production capabitility systems to SLURM 09/2015 – 01/2016

NERSC is part of the Lawrence Berkeley National Laboratory and is located at the main LBNL campus

NERSC operates multiple supercomputers for the U. S. Department of Energy

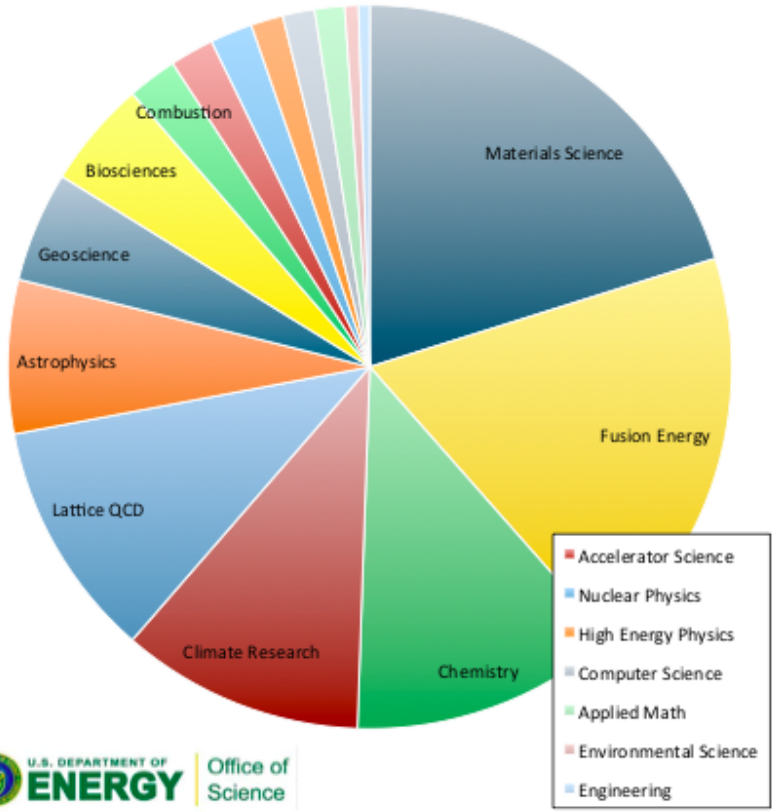
Computer time is allocate by the DOE for open science research projects funded by DOE.



edison – NERSC 7

Cray XC30
- 5,603 ivybridge compute nodes
- 89 service nodes

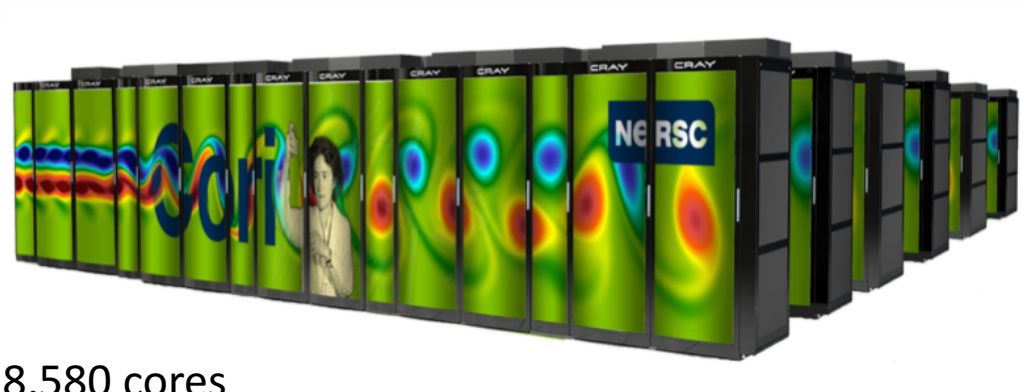
- 24 cores per node, 134,472 cores total
- 64 GB per node, 2.6 GB/core, 350 TB total
- Primarily used for large capability jobs
- Small – midrange as well
- ~ 7PB of local Lustre scratch



U.S. DEPARTMENT OF ENERGY | Office of Science

BERKELEY LAB

cori – NERSC 8



Cray XC40

12070 compute nodes

9,685 KNL	658,580 cores
2,385 Haswell	7,630 cores

- 1.16 PiB DRAM, 151 GiB MCDRAM (KNL Flat mode)
- DataWarp aka Burst Buffer (1.6 PiB)
- realtime jobs for experimental facililities
- massive quantities of serial jobs
- regular HPC workload
- shifter for Linux containers
- ~30 PB of Lustre scratch, also shared with edison

KNL

- KNL NUMA and MCDRAM modes can be set on a compute node at boot time
- users can specify the mode they want on job submission
- slurm will reboot nodes into the correct mode if needed
- reboot takes ~30 minutes
- makes reservations more complicated

NERSC, SLURM and Data Intensive Workloads

In addition to the two capability systems, Mendel (since 2012)

Generic Linux cluster (SL 6), 750+ Nodes (16-32 cores/node), FDR IB

Supports HEP, Joint Genome Institute, Materials Genome Project

separate front ends, batch systems, storage

PDSF – since 1995: HEP - serial, high throughput, fair share scheduling based on projects buy in, single core

Genepool: JGI

Matgen: 132 nodes

Strategic direction of lab

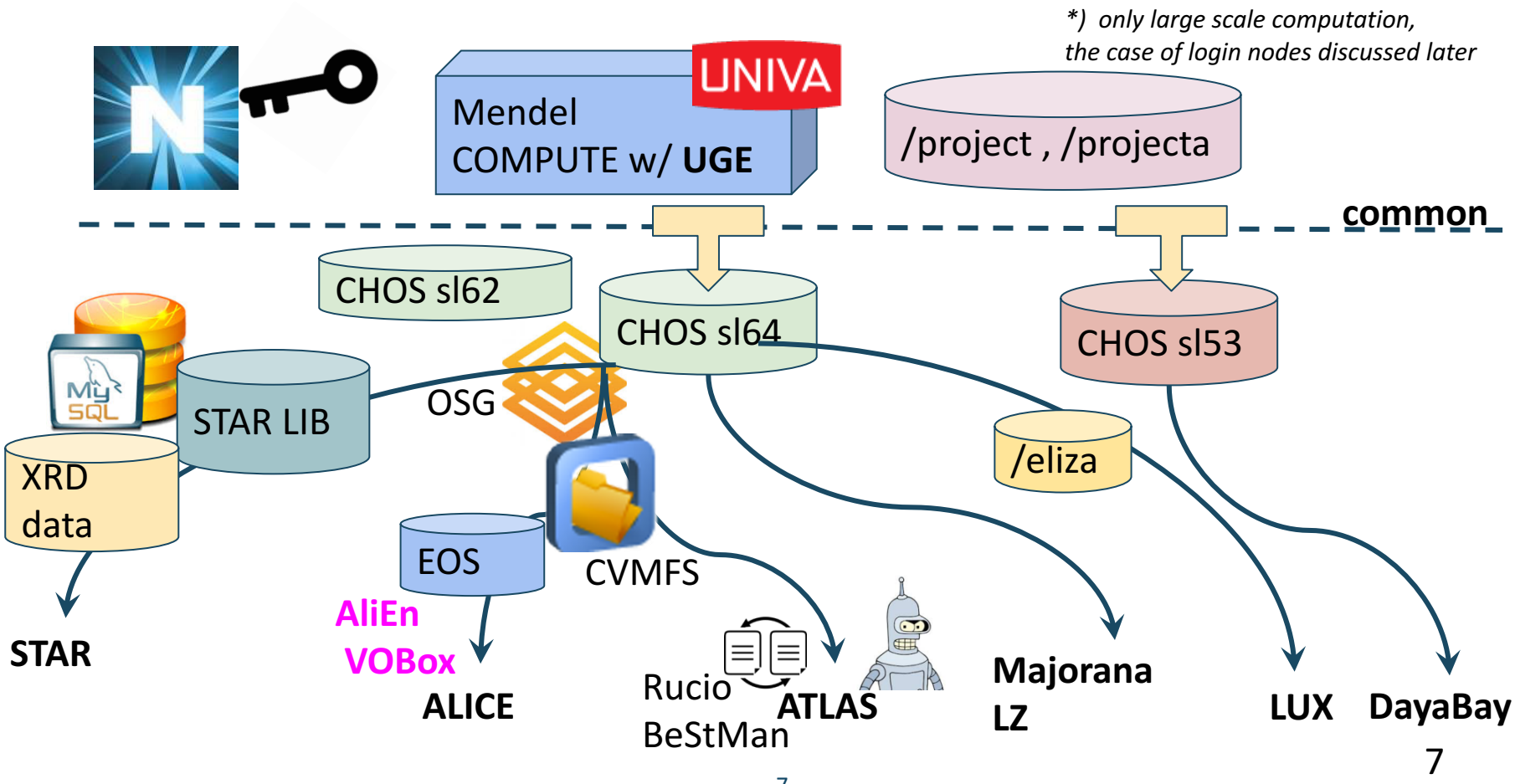
- support a few large systems (and batch systems)
- use container technology (e.g. shifter) to provide a familiar, secure and reproduceable environment to users where
- Mendel model does not scale

The Data Intensive workloads are being moved to slurm

This will facilitate the transition to the capability systems when Mendel is retired (warranties expire in 2018)

- **matgen – slurm since 2015.**
  - No integration needed with NERSC accounting.
  - Identical two node, low IO, high CPU jobs.
  - No trouble at all, very little support needed.
- **pdsf – Univa Grid Engine for many years**
  - 25% slurm now, perhaps 90% in October.
  - Configuration simple - designed to cram as many serial jobs on each node.
  - Integration with ancillary HEP services (grid, VOs) for 6 different experiments the hardest part
  - Integration with NERSC accounting a solved problem (Doug will talk about this later)
  - Trend in HEP is to use HT Condor, smaller SLURM-HEP community

Current PDSF workflow



genepool to slurm

- Has been on grid engine for many years
- Diverse workflows and needs – lots of porting, lots of user interaction needed
- Users have been, for the most part, working within a chrooted old debian environment
- Currently have 6 node slurm test cluster available
- Some of the JGI is already using cori

DOE Facilities Require Exascale Computing and Data

Astronomy

Particle Physics

Chemistry and Materials

Genomics

Fusion

Petascale to Exascale

- **Petabyte data sets today, many growing exponentially**
- **Processing requirements grow super-linearly**
- **Need to move entire DOE workload to Exascale**

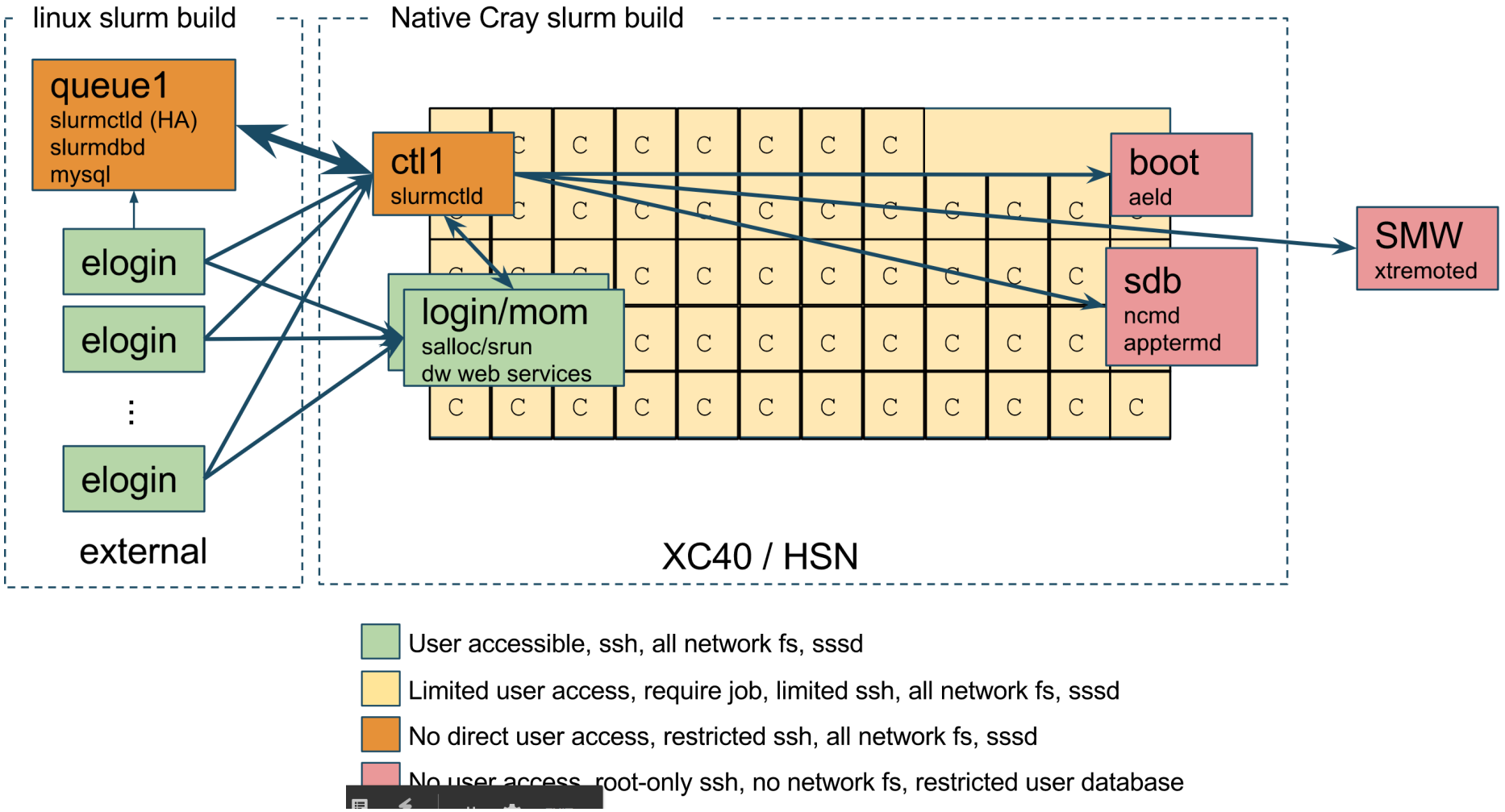
Popular features of a data intensive system and supporting them on Cori

Data Intensive Workload Need	Cori Solution
Local Disk	NVRAM 'burst buffer' and Shifter
Large memory nodes	128 GB/node on Haswell; Large memory login and service nodes
Massive serial jobs	Slurm running Natively on Cray
Complex workflows	SSH-based workflows Shifter Large Capacity of interactive resources
Communicate with databases from compute nodes	Advanced Compute Gateway Node – Software Defined Networking
Stream Data from observational facilities	Advanced Compute Gateway Node – Software Defined Networking
Easy to customize environment	Shifter Spank Plugins
Policy Flexibility	Improvements coming with Cori: Rolling upgrades, NRE Investments with vendors

Slurm enables the everything workload

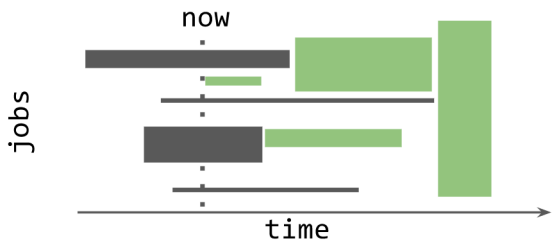
- **Flexible job prioritization**
  - Heavily customized by NERSC including an active priority management algorithm
- **Native Cray support by Slurm and Spank plugins enable user requestable features to be enabled on-node**
- **pam_slurm_adopt + Cray Linux Environment (CLE6) allows ssh-based workflows**
- **Native Cray support by Slurm enables serial job scheduling**

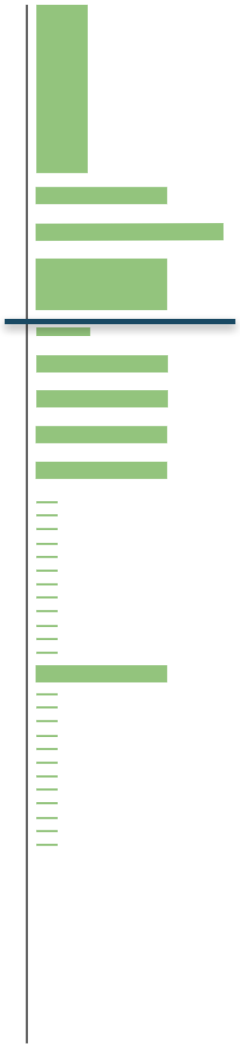
Slurm architecture on cori



Job Prioritization

- **Dominated by job age**
- **Qos priorities set starting priorities**
- **Only jobs exceeding a specific priority value (69120) reserve resources**
- **Active priority management**
  - Performed right before backfill
  - Allows each user/qos combination to have a small max number of jobs above the priority threshold
  - Prevents negative effects of queue stuffing





Upcoming work:

- Working with SchedMD to implement Priority Management algorithm to be *correctly* implemented in Slurm
- Use fairshares to inform size of aperature of jobs flowing through the priority threshold

perf/spank

- **Developer-users at NERSC require access to advanced profiling and performance tools**
- **Intel VTune is particularly popular for KNL**
- **Requires a matched userspace to kernel module version**
  - Use 2017.up2 userspace with 2017.up2 kernel driver, etc

```
sbatch --perf=vtune ...
sbatch --perf=vtune/2018.0 ...
```

- **Kernel modules loaded during job prolog and unloaded during job epilog**
  - Observed several instances where vtune collection threads do not terminate correctly. Module unload/load cycle avoids problems.
  - Allows us to support multiple versions – critical since Intel is frequently developing new features based on feedback from NERSC

perf/spank

required /usr/…/perf.so \
    base_start=/usr/…/perfbase_start.sh **base_stop**=/usr/…/perfbase_stop.sh \
    modules=vtune \
    vtune_versions=2017.up2,2018.0 **vtune_default**=2017.up2 \
    vtune_module_2017.up2=vtune/2017.up2 \
    vtune_start_2017.up2=/lib/modules/%r/extra/vtune/2017up2/vtune_start.sh \
    vtune_stop_2017.up2=/lib/modules/%r/extra/vtune/2017up2/vtune_stop.sh \
…

Allocator

Validates request
- Env mod?
- Uid/gid ACLs?
- Conflict in requests?

Prolog

dlopen() libslurm to get job record – discover if node is exclusively allocated or not

Run start scripts

Write munge-encoded status and stop-script list to local storage

Remote init()

Validates request
- Env mod?
- Uid/gid ACLs?
- Conflict in requests?

task_init()

Report errors to user if any start scripts failed

Kill task if validation failed

Epilog

Run stop scripts from munge-encoded list

If any failures occur, fail the epilog, removing node from service

future/spank

- **Building on perf/spank framework as mechanism to customize node environment reliably based on workload requests**
  - Initial coordination of software defined networking
  - Initial coordination of dynamically allocated external services

Data Intensive Workload Need	Cori Solution
Communicate with databases from compute nodes	Advanced Compute Gateway Node – Software Defined Networking
Stream Data from observational facilities	Advanced Compute Gateway Node – Software Defined Networking
Easy to customize environment	Shifter Spank Plugins

Upcoming Work and Directions

- **Bioinformatics workload moving entirely to cori in Q1 2018**
- **Working to get their job profiling and job metrics capabilities in place**
  - Slurm JobAcctGather + scalable backend
- **Planning on fairshare-based priority management in place**
- **Getting SDN and External Services Spank plugins in place for initial demonstrations**
  - Starting planning for more permanent capabilities we might want to see and support for these
- **Rolling upgrade capability**
  - Need new capabilities for Controller node based reboot command definition – will plugin into our monitoring system

Thank you!