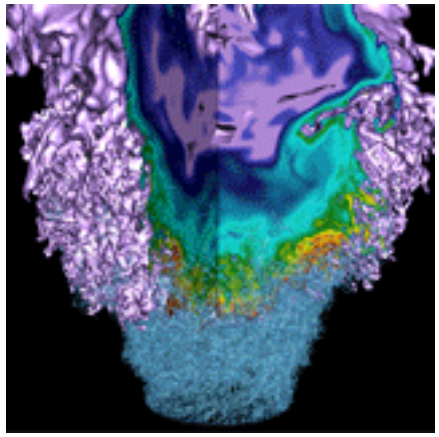
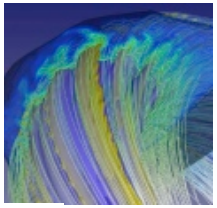
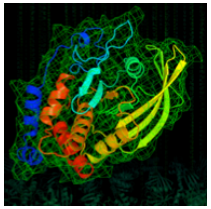
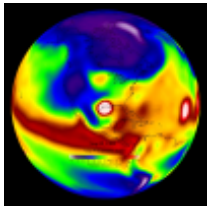
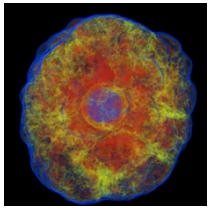
Cori: User Update

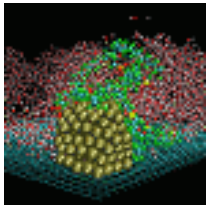












Helen He and Wahid Bhimji

NERSC 40 YEARS at the FOREFRONT

NERSC User Group Meeting
March 24, 2016

U.S. DEPARTMENT OF ENERGY | Office of Science

BERKELEY LAB
Lawrence Berkeley National Laboratory

Outline

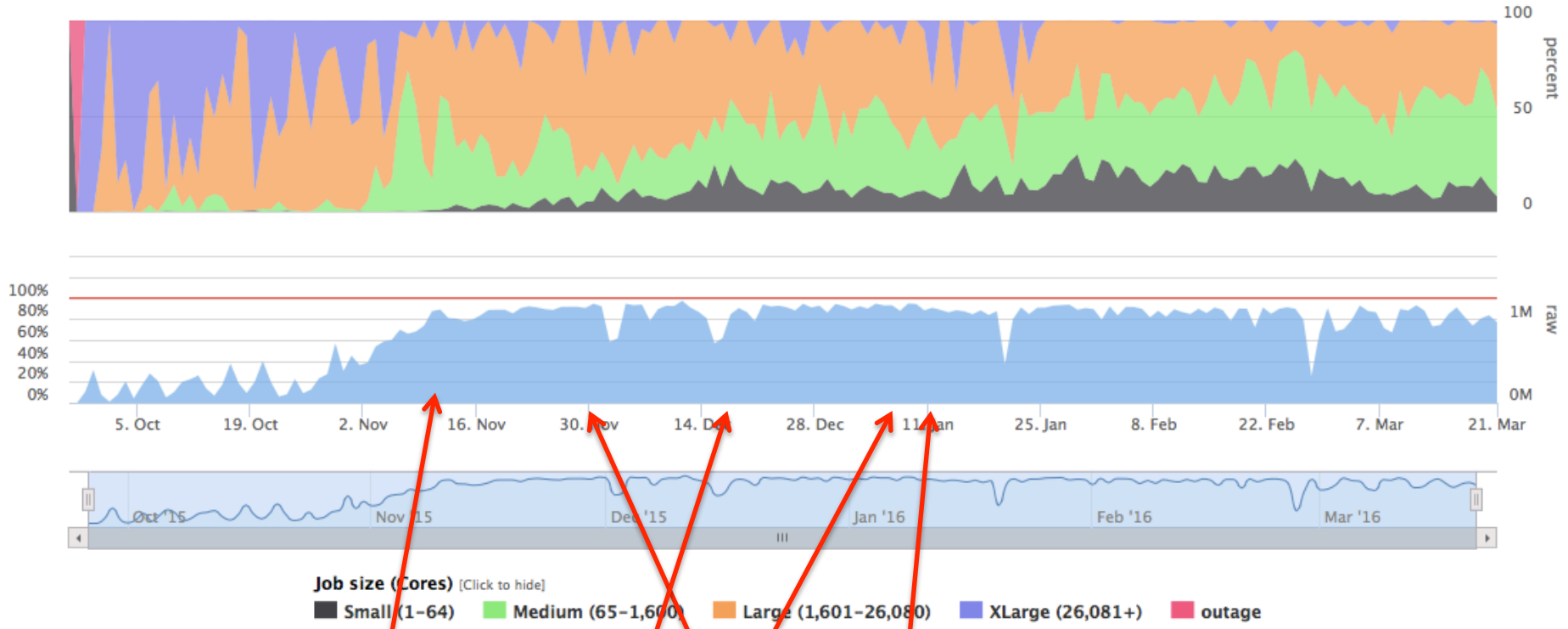
- Early User Program and Cori Usage Info
- Running Jobs and Batch Queues
- Selected User Issues
- Application SSP and Scratch IO Performance

Cori Early User Program

- **Early users were enabled in 7 phases:**
  - Allow Cori system became ready in various aspects (networking, programming environment, batch system, etc.)
- **162M MPP hours used before Cori charging started for AY16 on Jan 12, 2016.**

Category	#users	Date Enabled
Burst Buffer / Heavy Data users	50	10/29/2015
All Babbage users (covers all 3 tiers of NESAP teams)	230	11/3/2015
NUGEX members (current and past)	15	11/3/2015
Heavy Edison users	20	11/7/2015
Specific Requests (by users or staff)	70	10/29-11/7/2015
Projects that are out of time: 11/7, ~340 users	340	11/7/2015
All users	6000+ enabled, 2700 active	11/12/2015

Cori Usage Info



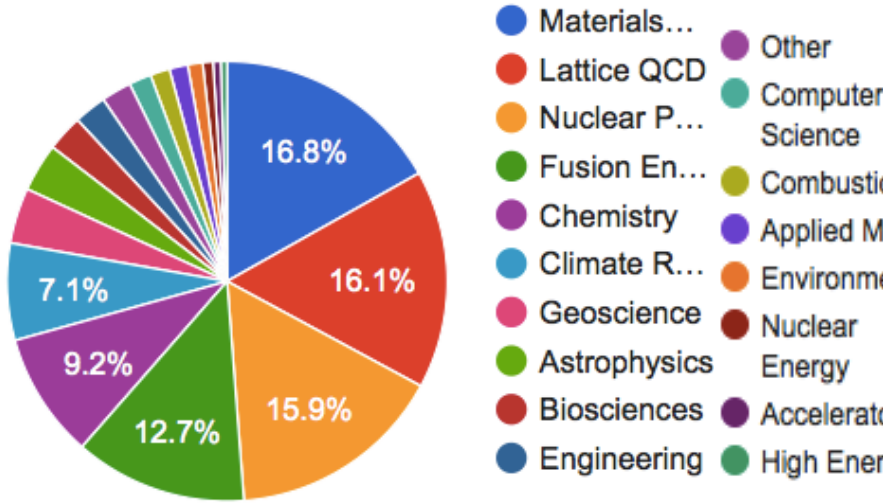
- 11/12/2015: All users enabled
- 11/30/2015 – 1/4/2016: Edison offline
- 12/15/2015: Hopper retired.
- 1/12/16: Cori started charging when AY16 began

More large jobs during free time ☺

Cori Usage Info: Free Period and AY16

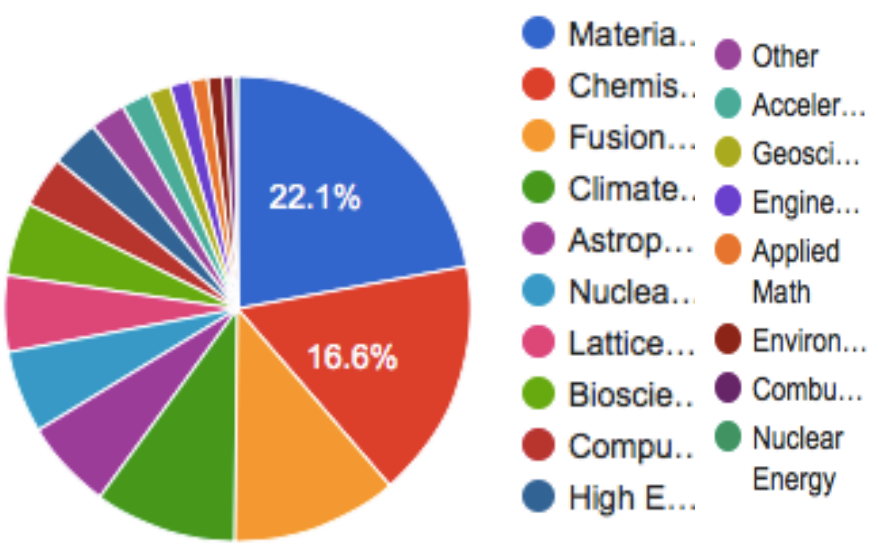
162M MPP hours used (10/29/15-1/11/16)

Raw Machine Hours by Science Area (in millions)



75.8M MPP hours used (1/12/16-3/22/16)

Raw Machine Hours by Science Area (in millions)



Cori Phase 1 Data Features

- **File Systems**
  - Burst Buffer for high bandwidth, low latency I/O
  - High performance Lustre file system: 28 PB of disk, >700 GB I/O bandwidth
  - Cross mounting of file systems (Cori scratch on Edison and DTNs) (TBA)
  - Large amount of memory per node (128 GB) as well as high memory nodes (775 GB).
- **Networking**
  - Improved outbound Internet connections (eg. to access a database in another center)
  - Software Defined Networking R&D for high bandwidth transfers in and out of the compute node (TBA)
- **On node software**
  - Improved shared library performance
  - User-defined images/Shifter

Cori Phase 1 Data Features (SLURM)

- **Cori Phase 1 also known as the "Cori Data Partition"**
- **Designed to accelerate data-intensive applications, with high throughput and "real time" need.**
  - "shared" partition. Multiple jobs on the same node. Larger submit and run limits. 40 nodes set aside
  - The 1-2 node bin in the "regular" for high throughput jobs. Large submit and run limits.
  - "realtime" partition for jobs requiring real time data analysis. Highest queue priority. Special permission only.
  - Internal sshd (CCM mode) in any queue
  - Large number of login/interactive nodes to support applications with advanced workflows
  - "burst buffer" integrated in SLURM, in early user period.
  - Encourage users to run jobs using 683+ nodes on Edison with queue priority boost and 40% charging discount there.

Transition from Hopper/Edison to Cori

- Programming environment is very similar to Hopper/Edison. Porting to Cori is straightforward in regards to software building.
- The aspect that users need to adjust the most is the transition from Torque/Moab to SLURM.
- Provided detailed documentations on SLURM transition guide, example batch scripts, and mini-tutorials.
- Worked with some specific applications for the porting. CESM is one such example. It is a new machine port, with bfb required.

SLURM Batch Scheduler Adoption

- **Overall SLURM adoption is smooth.**
- **Easy to use "premium", "ccm", good support and usage for "shared" and "realtime".**
- **A few traps (with user education):**
  - Hyperthreading is on by default
    - SLURM sees 64 CPUs per node
    - Asking nodes with "#SBATCH –n", but without "#SBATCH –N" may get half the node desired
    - Need to set OMP_NUM_THREADS=1 explicitly to run with pure MPI (for hybrid MPI/OpenMP program compiled with openmp enabled)
  - Automatic process and thread affinity is good. Can explore with advanced settings for more complicated binding options.

Batch Job Wait Time

- **Users reported about VERY LONG wait time for jobs**
- **Changes were made on Jan 15**
  - Added max number of backfill jobs per partition (on top of max number of backfill jobs per user)
  - Decreased max size of debug from 128 to 112.
  - Communicated with individual users to use the “shared” partition, job arrays, and bundling jobs.
  - Jobs do not plan to run in AY16 were deleted
- **Most debug jobs then started within 30 min instead of hours, many now start in a few min.**
- **The regular jobs wait time are significantly smaller too**
- **More tuning on Queue Configuration is undergoing.**
  - Close monitoring on job throughput and utilization
  - Changes made on 03/22 for scheduling algorithm greatly increased system utilization

NERSC Custom Queue Monitoring Script

- **"sqs" is a NERSC custom queue display script which provides basic batch job info plus the job ranking based on start time provided by the backfill scheduler.**
- **A new version of "sqs" was deployed on Jan 19 with two columns of ranking values to give users more perspective of their jobs in queue.**
  - Added job priority ranking with absolute priority value (a function of partition, QOS, job wait time, and fair share)

A Few Tips to Get Faster Job Turnaround

- Request shorter wall time, do not use allowed max wall time.
- Use “shared” partition for serial jobs or very small parallel jobs.
- Bundle jobs (multiple “srun”s in one script, sequential or simultaneously)
- Use Job Arrays (better managing jobs, not necessary faster turnaround). Each array task is considered a single job for scheduling.
- Use job dependency feature for managing workflow.

Resolved: Cray HDF5 with Intel16

- **Internal compiler error for Fortran codes when using cray-hdf5, and cray-hdf5-parallel/1.8.14 with intel/16.0.0.109**
- **Two workarounds:**
  - Use NERSC built hdf5/1.8.14 and hdf5-parallel/1.8.14 with Intel/16.0.0.109 compiler
  - Use cray-hdf5/1.8.14, but swap intel compiler version from 16.0.0.109 to 15.0.1.133.
- **cray-hdf5/1.8.16 has been installed and set to default which resolved this issue (Feb 27, 2016)**

Workaround: Node Voltage Fault

- Node voltage fault only seen with one specific application “pw.x” from Quantum Espresso.
- By default, hyperthreading is used. And the application generates a very close sequence of current spikes that may cause the Voltage Converter to self-protect and shut down.
- Workaround by user education to use 1 thread per MPI task. Also modified the NERSC provided module file to set OMP_NUM_THREADS=1. (Jan 16, 2016)

Resolved: /project IO performance

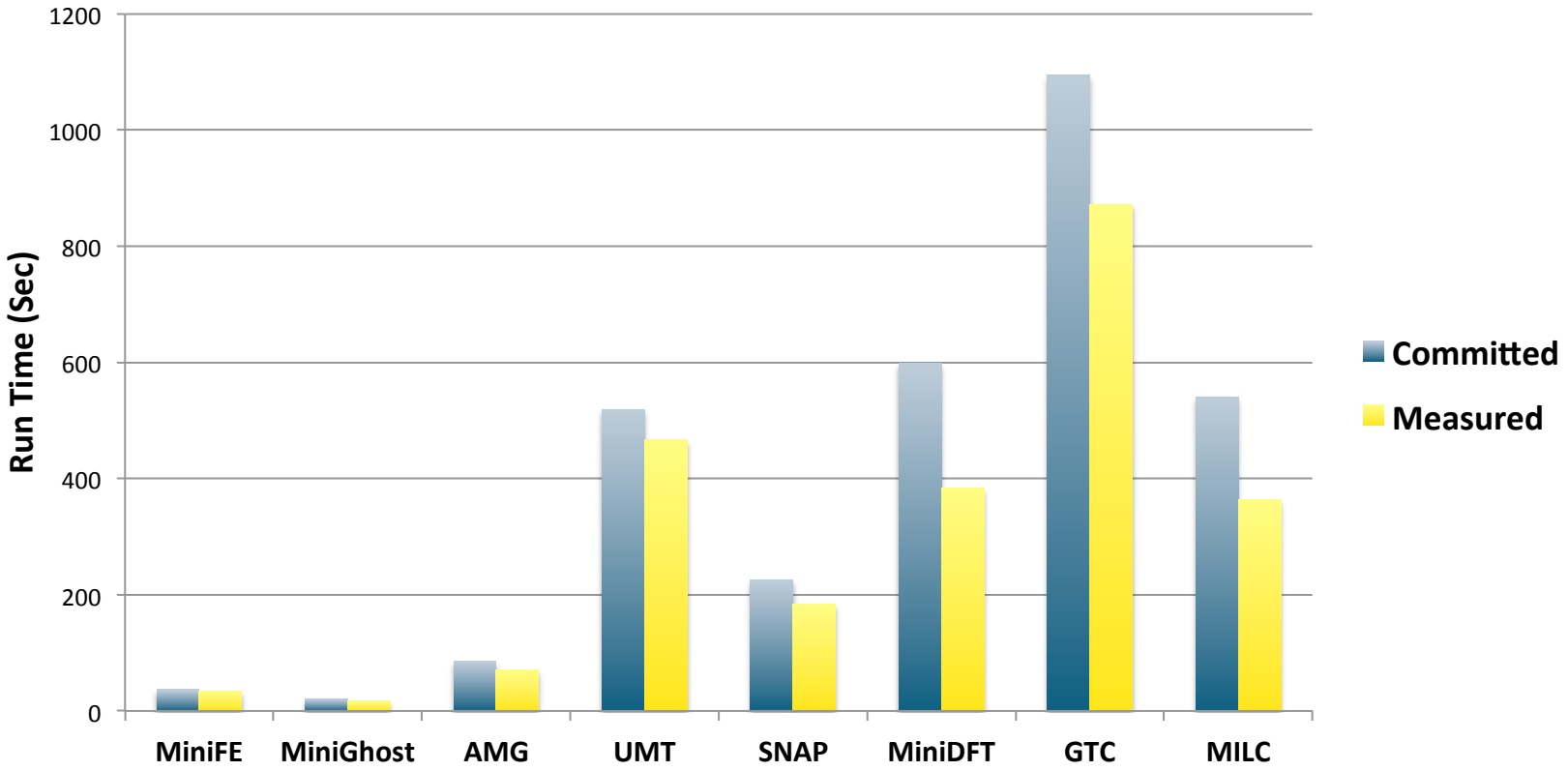
- **Two applications reported 10x parallel IO performance slowdown in /project, seen after Dec 25, 2015.**
- **Fixed during system reboot with scheduled maintenance on Jan 20, 2016.**
- **Exact cause of slowdown unknown**
  - Unlikely due to “Cori DVS nodes GPFS IB cable not used”

Current Issues

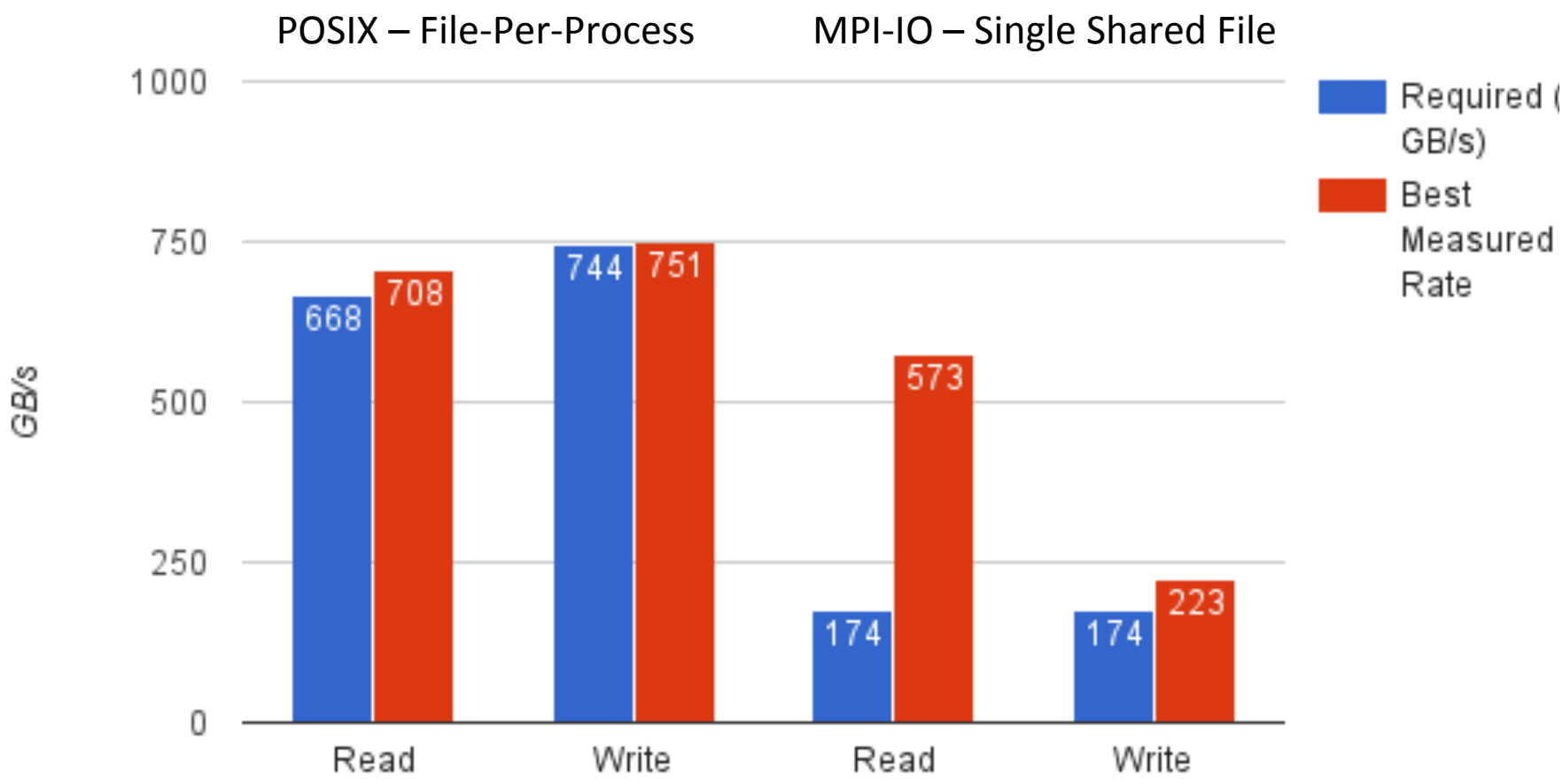
- Login nodes crash when hitting LBUG
- Compute nodes stuck in completing states
- Compute nodes OOM from applications

Cori Phase 1 SSP Performance

Committed SSP: 68.2
Measured SSP: 83.0



Peak Cori Scratch Lustre I/O Performance



Thank you.