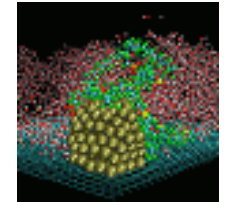
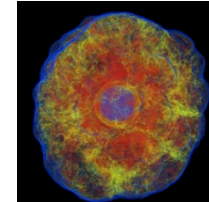
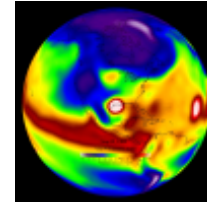
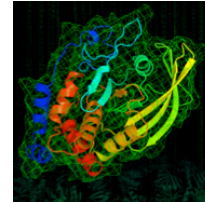
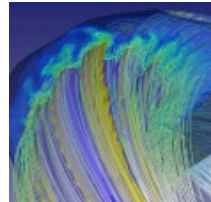
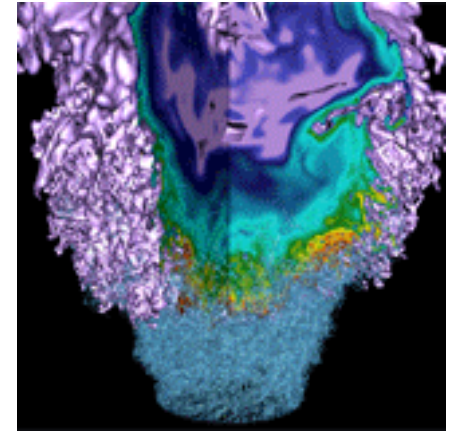


Cori: User Services Report



Helen He
NERSC/Cray Quarterly, 2/10/2016

Outline



- **Early User Program and Cori Usage Info**
- **Running Jobs**
- **Selected User Issues**
- **Application SSP Performance**

Cori Early User Program

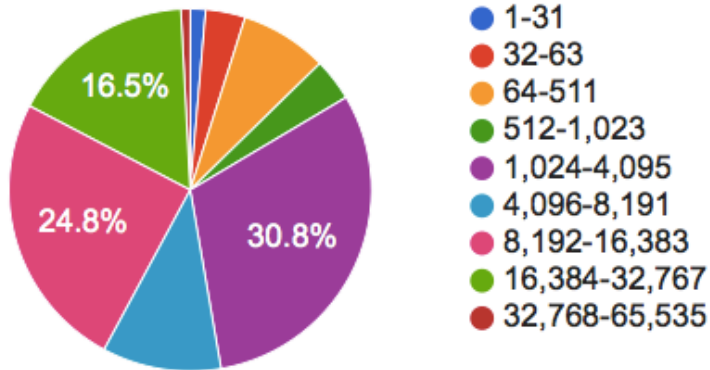
- **Early users were enabled in 7 phases:**
 - Allow burst buffer users to have more time on Cori
 - Allow Cori system became ready in various aspects (networking, programming environment, batch system, etc.)

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

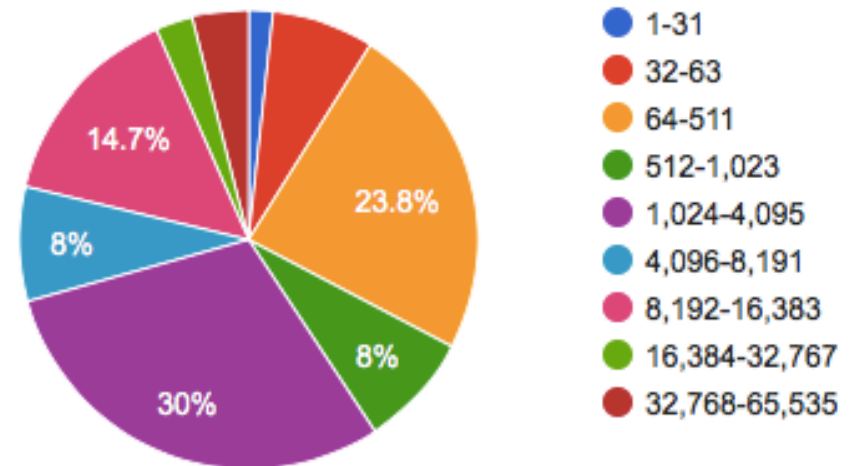
10/29-12/14/15 (free)

Raw Machine Hours by Cores Used (Percent)



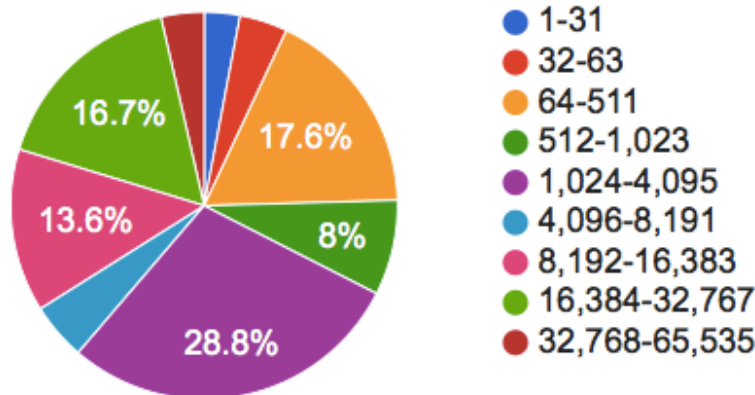
1/12-2/8/16

Raw Machine Hours by Cores Used (Percent)



12/15//15-1/11/16 (free)

Raw Machine Hours by Cores Used (Percent)



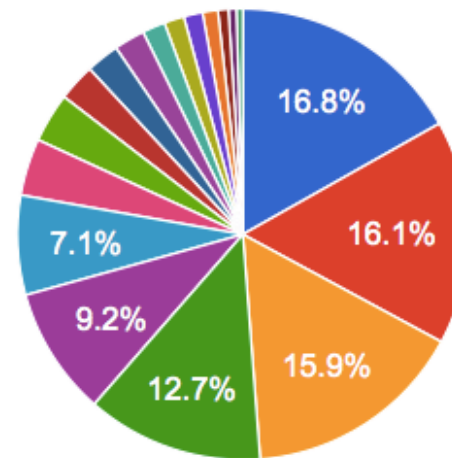
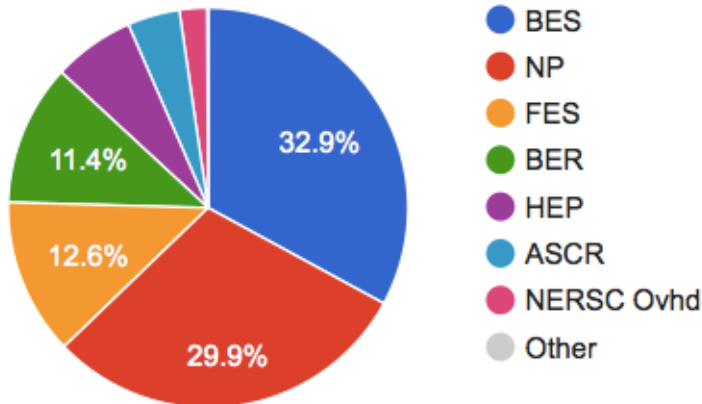
Cori Usage Info: Free Period

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

12/15/15-1/11/16

Raw Machine Hours by Science Area (in millions)

12/15/15-1/11/16
Raw Machine Hours by DOE Office (in millions)

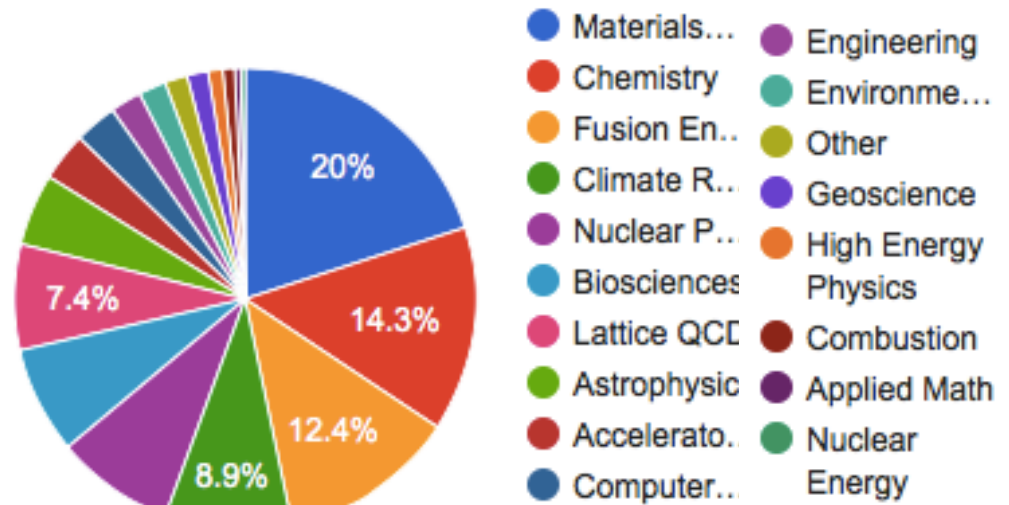


- Materials...
- Lattice QCD
- Nuclear P...
- Fusion En...
- Chemistry
- Climate R...
- Geoscience
- Astrophysics
- Biosciences
- Engineering
- Other
- Computer Science
- Combustion
- Applied Math
- Environme...
- Nuclear Energy
- Accelerato...
- High Ener...

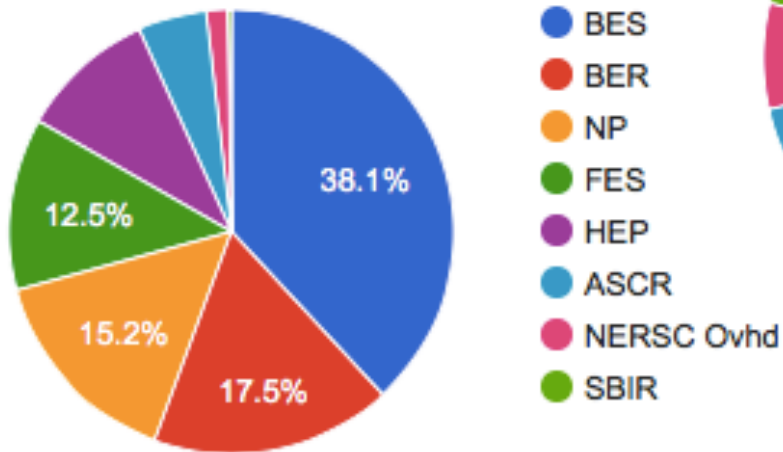
Cori Usage Info: AY 2016

**75.8M MPP hours used
(1/12/16-2/8/16)**

Raw Machine Hours by Science Area (in millions)



Raw Machine Hours by DOE Office (in millions)



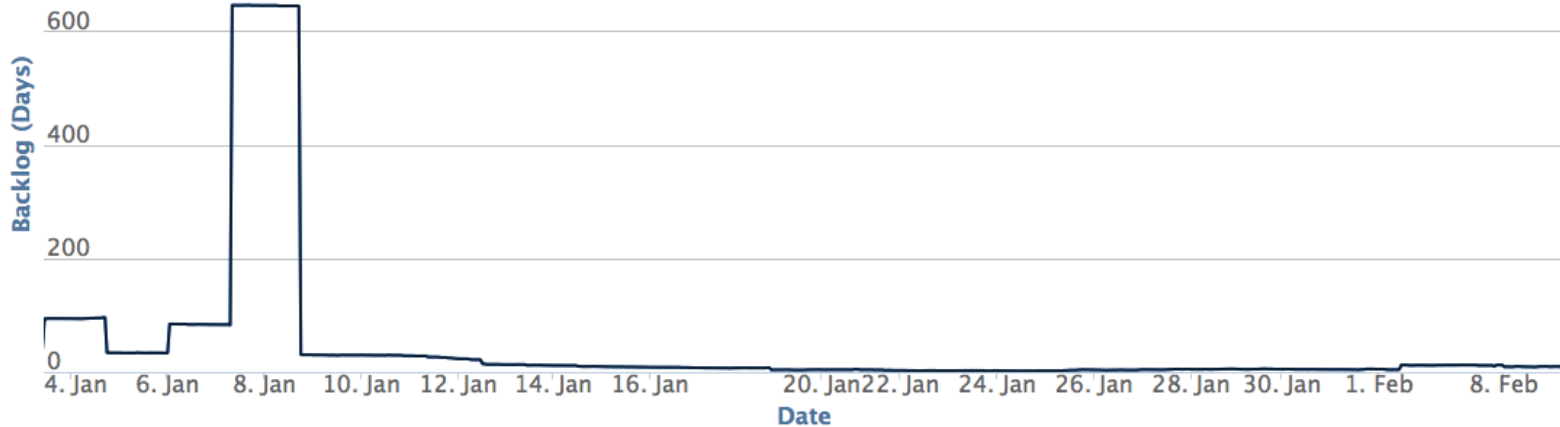
Goal for Cori Phase 1



- **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" partition (mimics "thruput" queue on Hopper). Large submit and run limits.
 - "realtime" partition. Highest queue priority. Special permission only.
 - "burst buffer" capability, in early user period.
 - Encourage users to run jobs using 683+ nodes on Edison with queue priority boost and 40% charging discount there.

Backlogs on Cori

- **Current backlog is 8.6 days.**
- **Huge submissions from 2 users increased backlogs significantly.**
 - One user submit many 512 nodes jobs, each 24 hrs. increased backlog from 40 to 92 days
 - Another user submitted a 1000-task large array job, with 1 hr wall time limit, later increased to 12 hrs time limit, increased backlog from 33 to 83 to 644 days.
 - Although backlogs caused from such submissions are shown high, they won't affect scheduling for other users jobs significantly, since the limits we have set will basically cause most of these jobs not being considered for scheduling.



Job Wait Time Improved Significantly



- **Users complained about VERY LONG wait time for jobs before AY16**
- **Changes were made from Jan 15**
 - Added max number of backfill jobs per partition (on top of max number of backfill jobs per user) significantly improved the backlog for debug jobs.
 - It allows lower priority debug jobs to run ahead of regular jobs that have higher absolute value of priority.
 - Decreased max size of debug from 128 to 112.
- **Most debug jobs now start within 30 min, many much shorter!**
- **The regular jobs wait time are significantly smaller too**
 - Additional tuning:
 - Increased max backfill interval from 30 to 150 sec
 - Tuned max backfill jobs per user, and max backfill per partition
 - Users delete more jobs submitted during free time
 - jobs with no active repo were cancelled
 - Communicated with individual users to use the “shared” partition, job arrays, and bundling jobs.

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.

New “sqs” with 2 Columns of Priority Ranking



- **A new version of “sqs” (a NERSC custom queue monitoring script) was deployed on Jan 19. Original “sqs” has one column for ranking based on start time provided by the backfill scheduler.**
- **This version provides two columns of ranking values to give users more perspective of their jobs in queue.**
 - Column RANK_P shows the ranking with absolute priority value, which is a function of partition QOS, job wait time, and fair share. Jobs with higher priority won't necessarily run earlier due to various run limits, total node limits, and backfill depth we have set.
 - Column RANK_BF shows the ranking using the best estimated start time (if available) at a backfill scheduling cycle (every 150 sec now), so the ranking is dynamic and changes frequently along with the changes in the queued jobs
- **Sample sample options:**
 - “sqs” (show user’s own jobs)
 - “sqs -a -p debug” (show only debug jobs)
 - “sqs -a -nr -np shared” (no running jobs, no shared jobs)
 - “sqs -w” (show all my jobs in wide format with more info)
 - “sqs -s” (short summary of queued jobs)

Torque/Moab Transition to SLURM



- 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
 - Need to set `OMP_NUM_THREADS=1` explicitly to run with pure MPI (for hybrid MPI/OpenMP program compiled with openmp enabled)
 - Asking nodes with “`#SBATCH -n`”, but without “`#SBATCH -N`” may get half the node desired
 - Automatic process and thread affinity is good. Can explore with advanced settings for more complicated binding options.

Workaround: hdf5 with intel16



- **cray-hdf5-parallel/1.8.14 with intel/16.0.0.109 compiler version generates internal compiler error for Fortran codes (bug 833561)**
- **Workaround: NERSC built own hdf5/1.8.14 and hdf5-parallel/1.8.14 with Intel/16 compiler**
- **Cray has provided a patch, will soon to be installed.**

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`.**

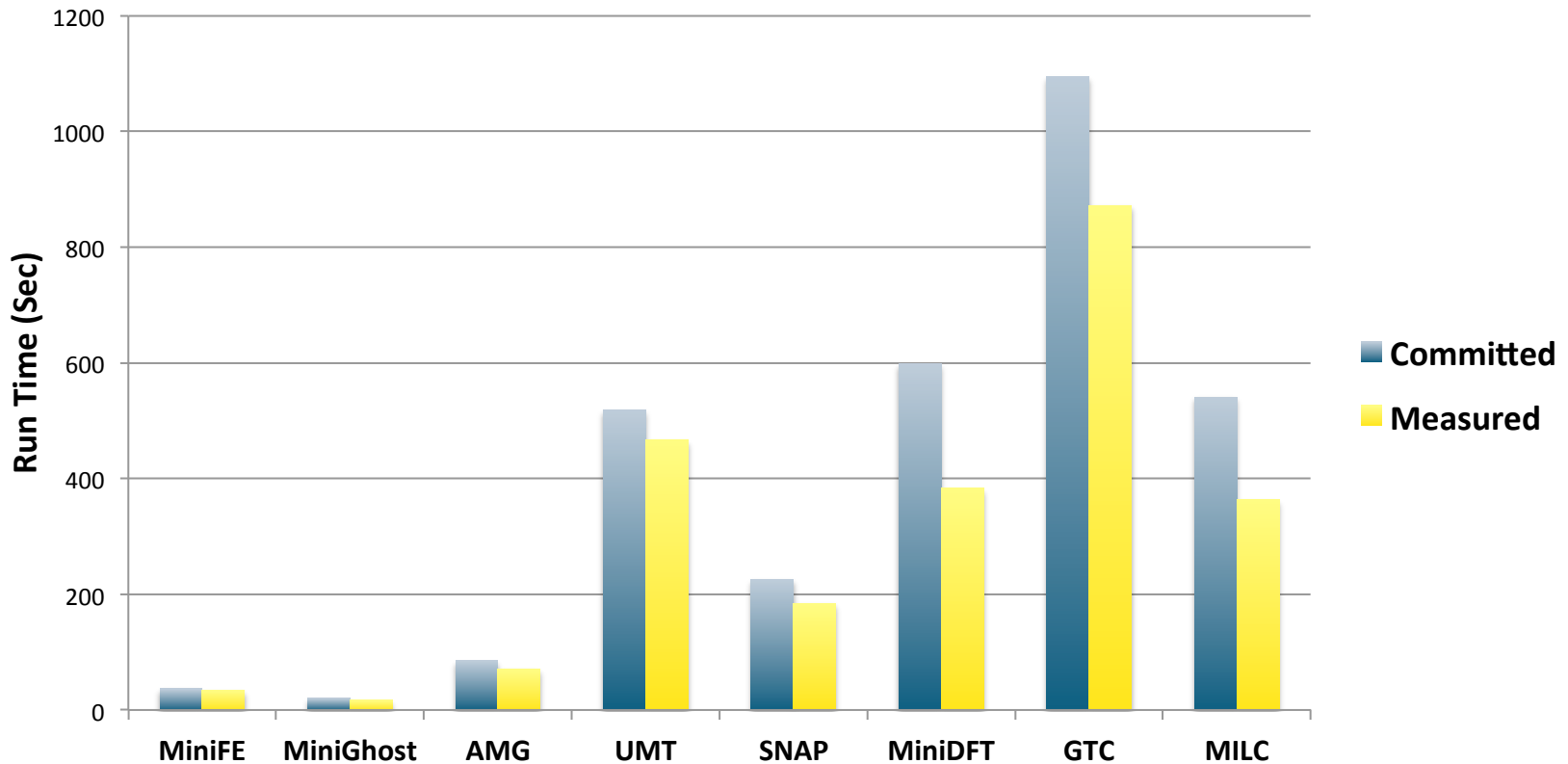
Fixed: /project IO performance



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

Cori Phase 1 SSP Performance

Committed SSP: 68.2
Measured SSP: 83.0





Thank you.