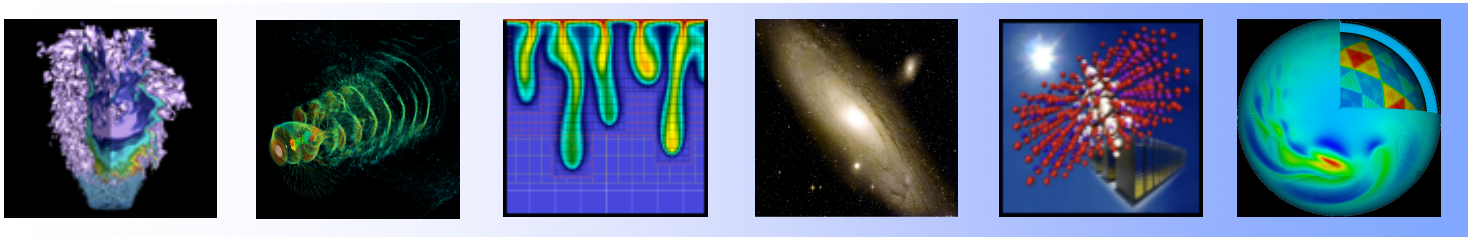


Transitioning Users from Franklin XT4 to Hopper XE6

Katie Antypas and Helen He
NERSC User Services Group
May 23-26, CUG 2011

U.S. DEPARTMENT OF ENERGY | Office of Science
NERSC National Energy Research Scientific Computing Center
BERKELEY LAB Lawrence Berkeley National Laboratory

Outline

- Introduction
- Hopper Early User Program
- Effectively Using 24 Cores Per Node
- I/O Performance
- Bugs Found and Fixed
- Error Messages
- User Feedback
- Ongoing Issues
- Summary

Franklin and Hopper

Hopper:

- **Phase 1: Cray XT5, 668 nodes, 5,344 cores**
- **Phase 2: Cray XE6, 6,384 nodes, 153,216 cores**
  - **~ 140 Tflop/s sustained**
  - **1.28 PFlop/s peak**

Franklin: Cray XT4

- **9,532 nodes, 38,128 cores**
- **~25 TFlop/s sustained**
- **356 TFlop/s peak**

Hopper's Role at NERSC

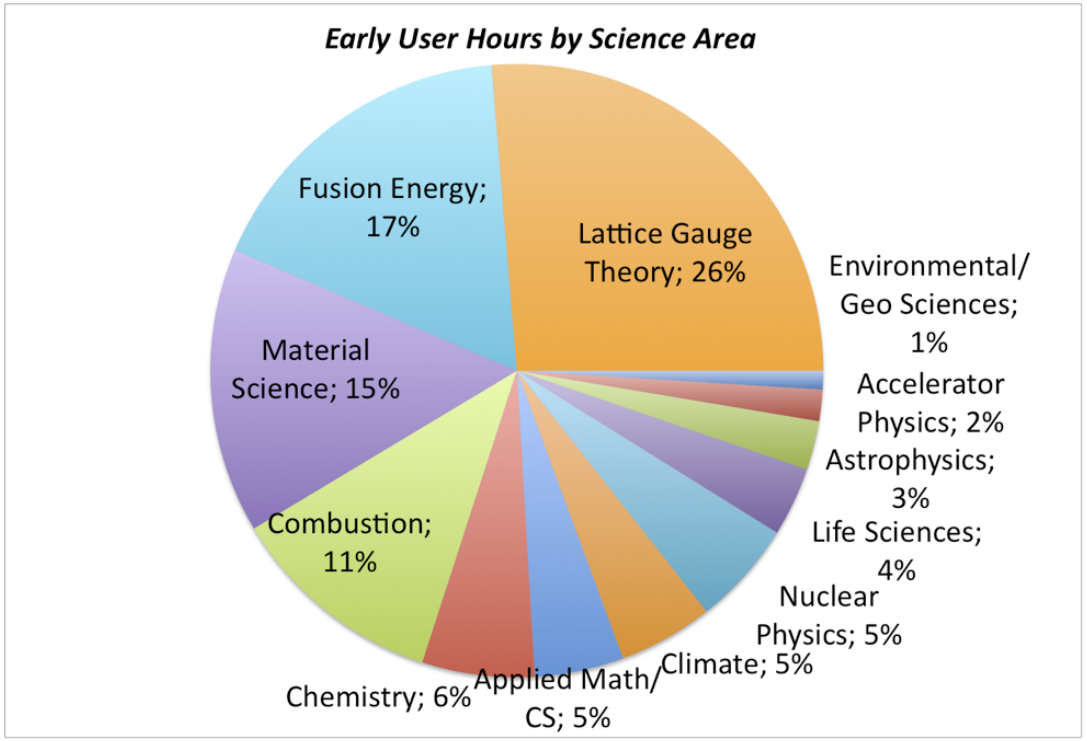
- NERSC is US DOE's primary high performance computing center
- Hopper is the new "flagship" system at NERSC after Franklin
- First time a peta-flop system is available to the general DOE research community
  - Production science runs
  - Code scalability testing
- Increases available computing time over a factor of 4 for our 4,000+ scientific users
- Serves the needs for most NERSC users from modest to extreme concurrencies

Hopper Key Dates

• Phase 2 system arrives	Jul 30 - Sept 17, 2010
• Phase 2 install complete	Sept 27, 2010
• Earliest users on system	Nov 15, 2010
• Integration complete	Nov 30, 2010
• All user accounts enabled	Dec 23, 2010
• Acceptance begins	Feb 4, 2011
• Availability test begins	Feb 5, 2011
• System accepted	Apr 19, 2011
• Account charging begins	May 1, 2011

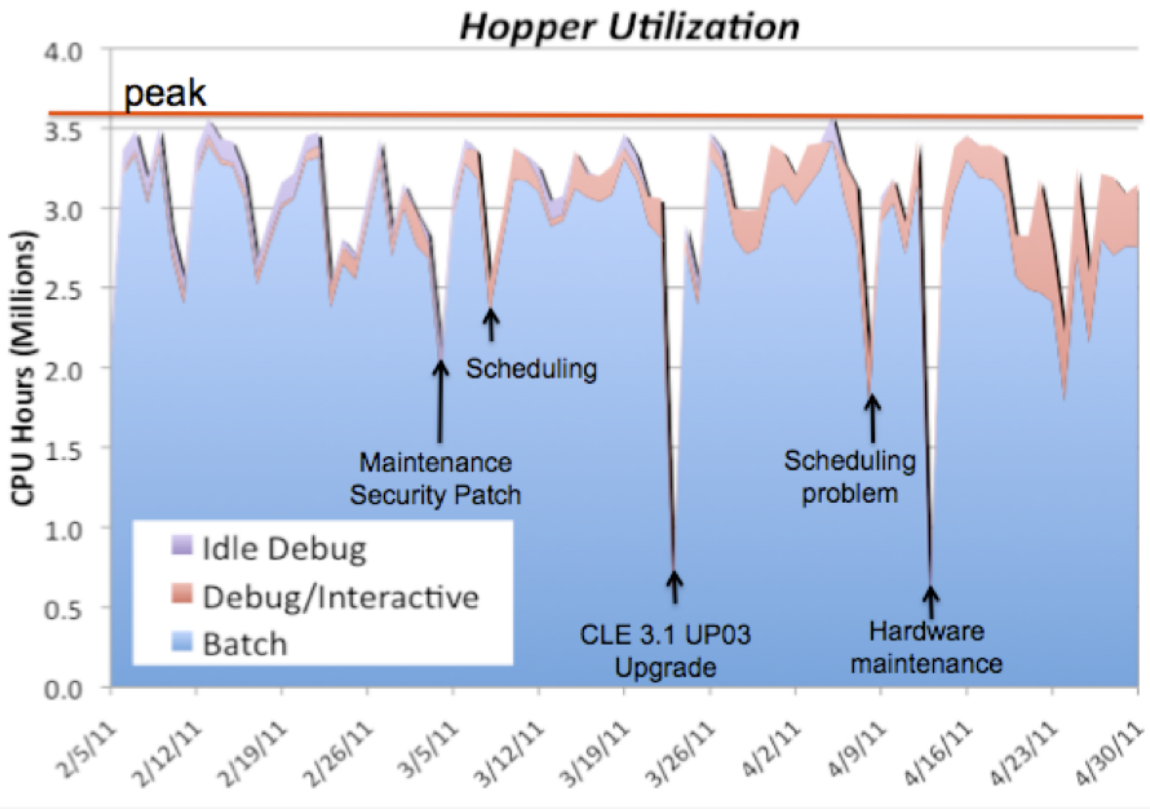
Hopper Early Hours

Breakdown of Early User Hours by Science Area
Nov 15, 2010 – Apr 30, 2011



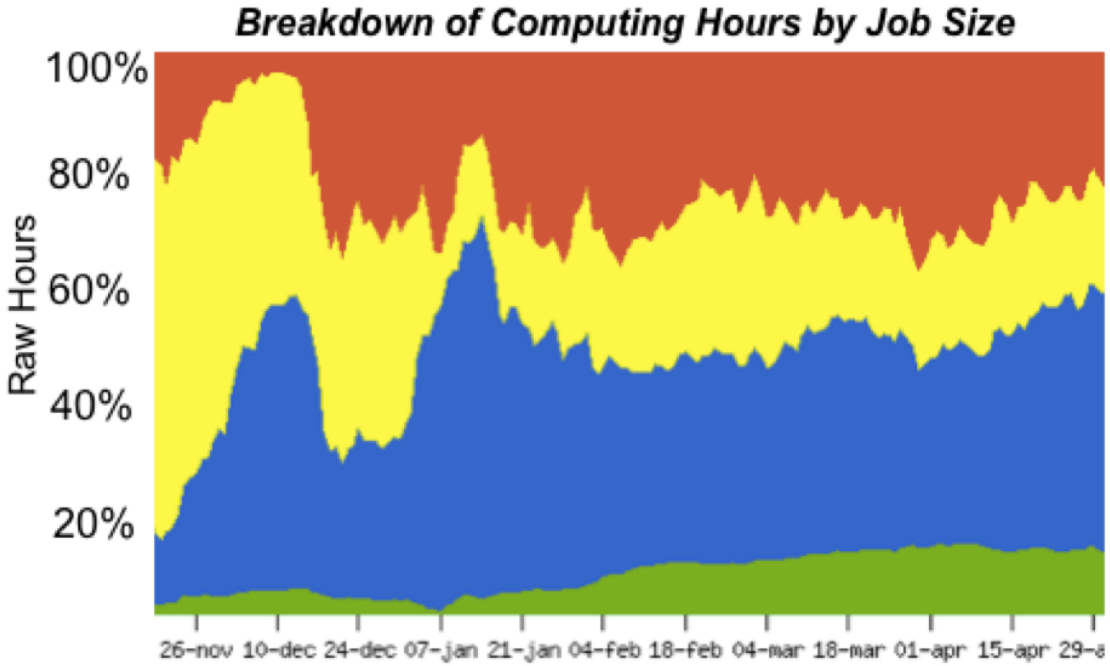
- **~350 million early hours delivered to science offices**
- **~280 projects have used time**
- **~1,000 users have accessed the system**
- **Consistently 300-400 unique users logged into system at any time**

Hopper Utilization



- Over 81% utilization in the first month 2.5 month (based on 24 hour day, including maintenances)
- System problems that would have resulted full outages on the XT4 and XT5 can be ridden through on the XE6
- Room for scheduling improvements, pack large jobs together, stabilize the system further
- Maintenances cut utilization substantially, look to minimize

Job Sizes Breakdown



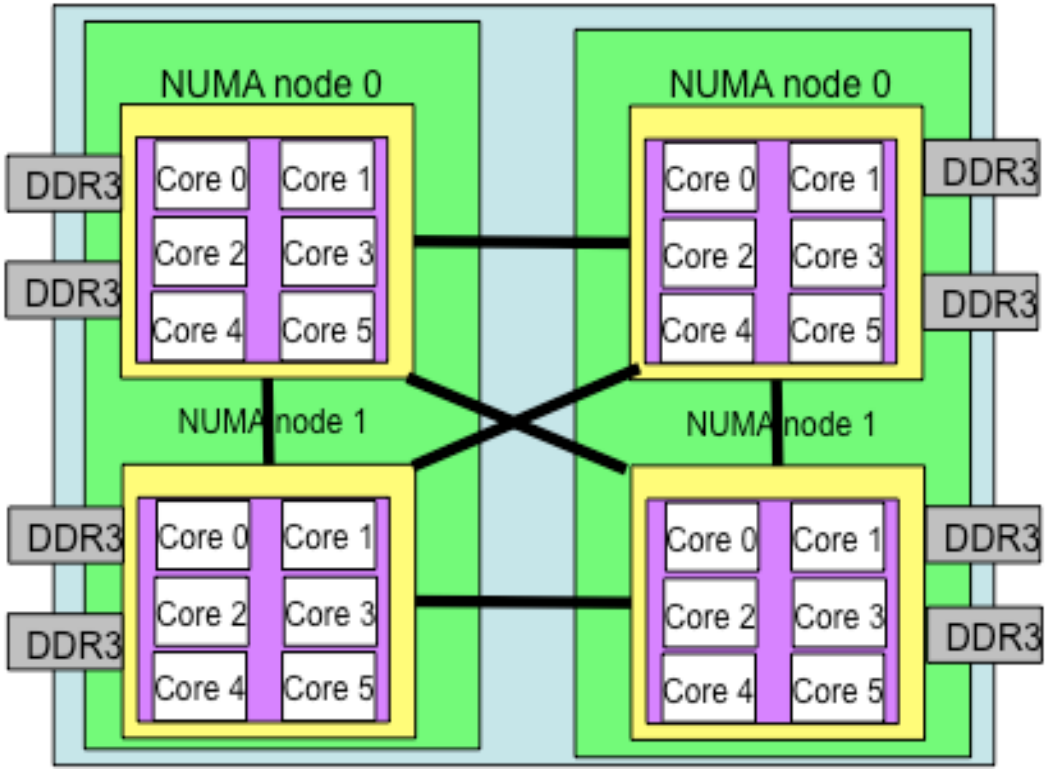
153,216 cores on Hopper

>43%		65,536+_cores
<43%		16,384-65,535_cores
<10%		1024-16,383_cores
<1%		1-1023_cores

- **Hopper is efficiently running jobs at all scales**
- **During availability period, over 50% of raw hours have been used for jobs larger than 16k cores.**

Hopper's 24 Core Compute Nodes

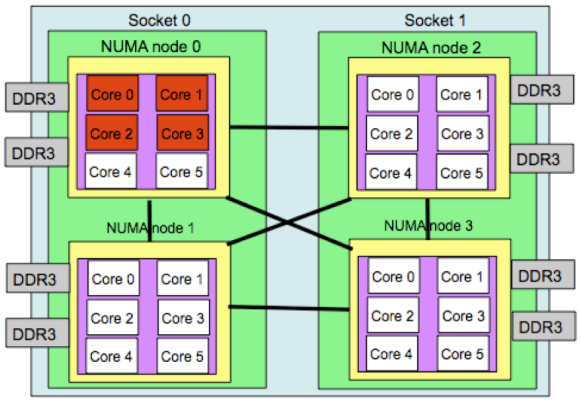
Hopper Compute Node



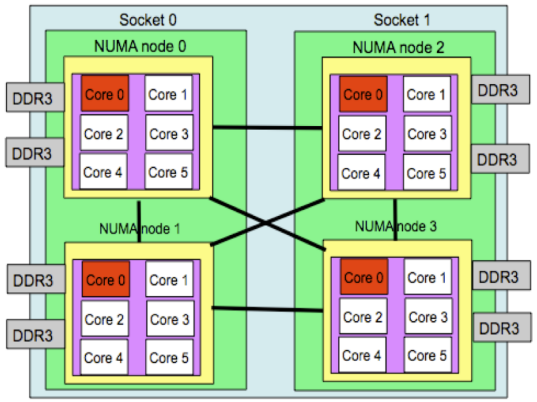
- With 32 GB of memory per node, the Hopper system gives users more addressable memory per node
- However, this is only 1.33 GB/ core, a challenge for some applications
- Longer time to access memory on a remote NUMA node
- Most users still running 24 MPI tasks per node
- More are trying OpenMP
- Some are running nodes "unpacked"

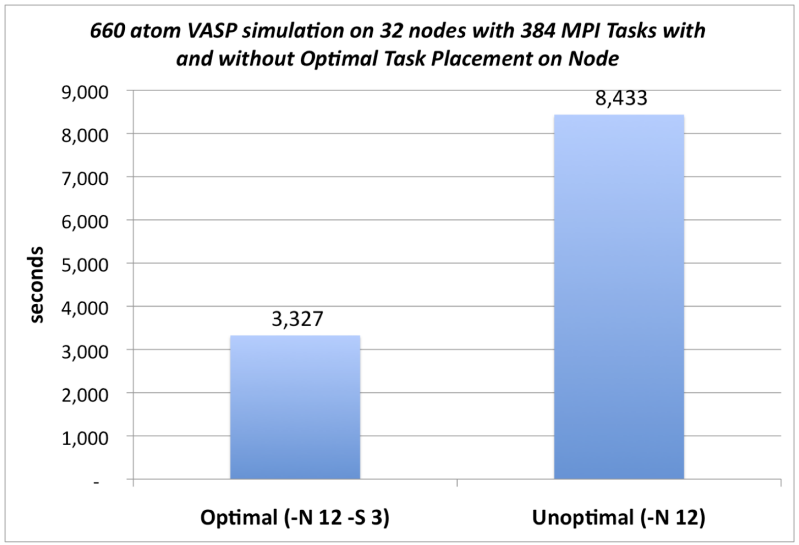
The Challenge of Distributing Tasks

Without "–S 1"



With "–S 1"



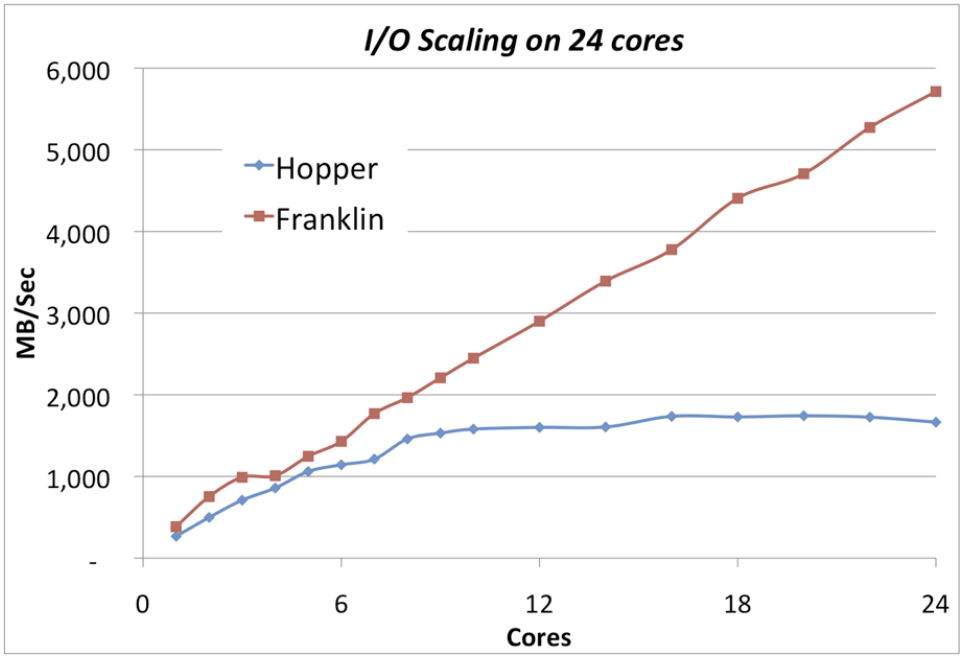


- Default aprun options assign tasks on first NUMA node before moving onto next.
- Need to use "-S" option to specify how many tasks per NUMA node to maximize memory bandwidth.
- VASP code observed 2.5x performance improvement with "-S 3" option using 12 cores per node.
- Advanced options "-sn", "-ss", "-cc" are introduced to the users.

Hybrid MPI/OpenMP Encouraged

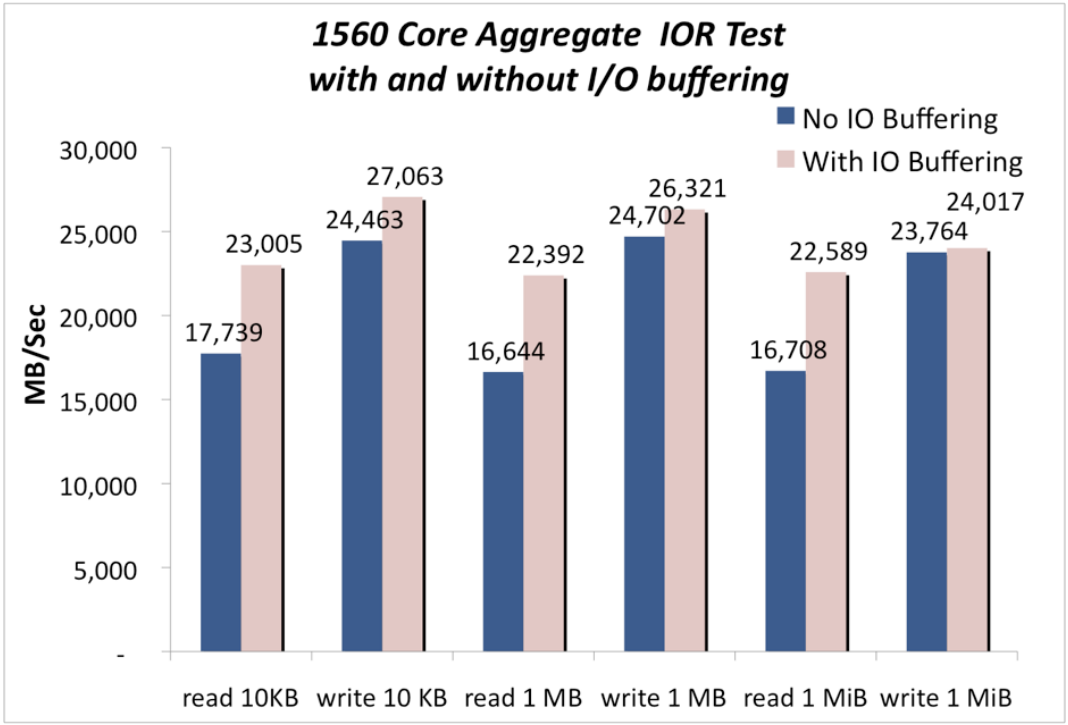
- Most MPI codes running successfully on Franklin will probably still run on Hopper.
- Some codes may get OOM error
  - Have to use fewer cores
- Hybrid MPI/OpenMP Advantages
  - Reduces memory footprint: Fewer copies of executables, fewer MPI buffers, fewer ghost cells.
  - Smaller amount of MPI messages with larger sizes
  - MPI across nodes, OpenMP within nodes is natural
- Recommend users to use max of 6 threads due to "First Touch" memory allocation policy
  - So that each thread only needs to access memory within the NUMA node it is binded.

I/O Performance



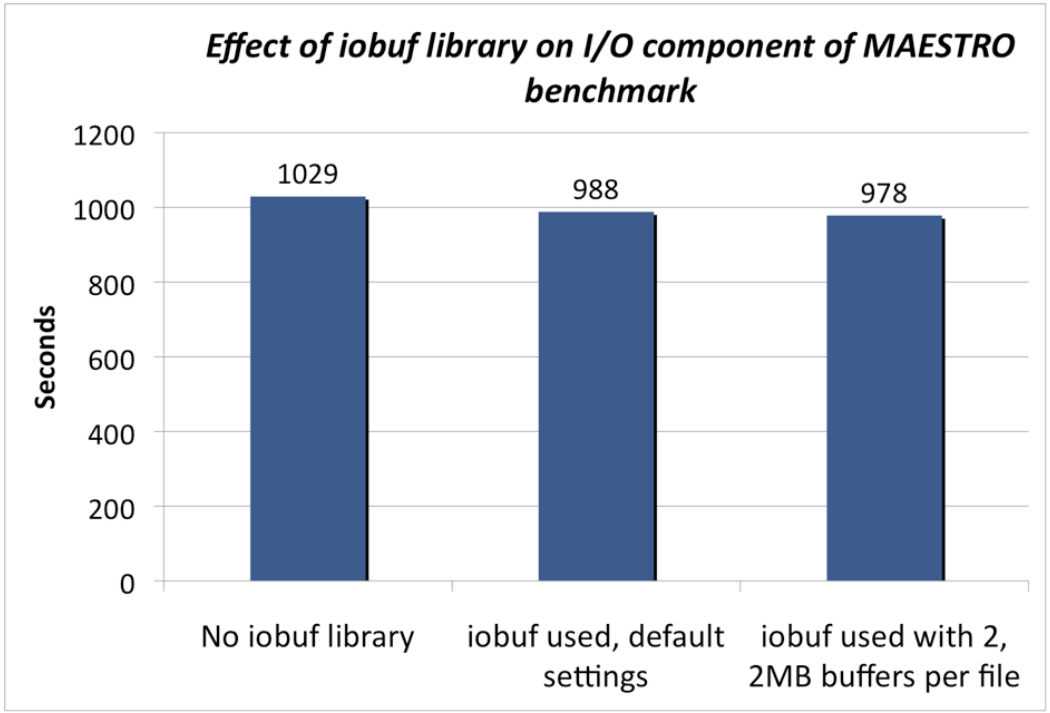
- IOR benchmark with 24 MPI tasks, each writes 2 MB of data.
- Uses 1 node on Hopper, 6 nodes on Franklin.
- Franklin performance increase linearly, reaches 2/3 of Seastar2 network's injection bandwidth of 1.6 GB/s.
- Hopper performance levels at 1700 MB/s after 8 cores, only reaches 28% of Gemini's injection bandwidth of 6 GB/s.
- Working with Cray to understand I/O performance limitation on Hopper.

IOBUF Module with IOR



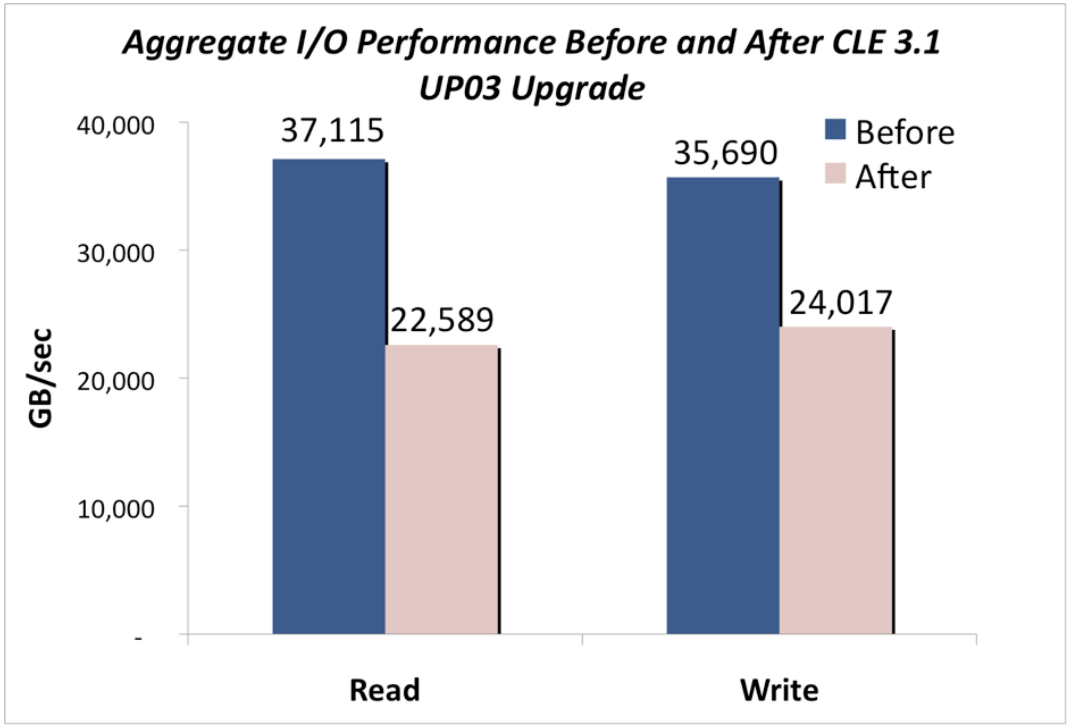
- IOBUF module to buffer I/O requests so that fewer, larger I/O operations are performed.
- Users only need to load the module and re-link applications.
- Runtime environment variable IOBUF_PARAM could be adjusted. Default: 4 buffers per file of 1 MB size.
- 1,560 core test, each writes 2 GB and then reads back.
- Read is 30-40% better with IOBUF. Write is only 1-11% better.
- Improves most for smallest transfer size.

IOBUF Module with MAESTRO



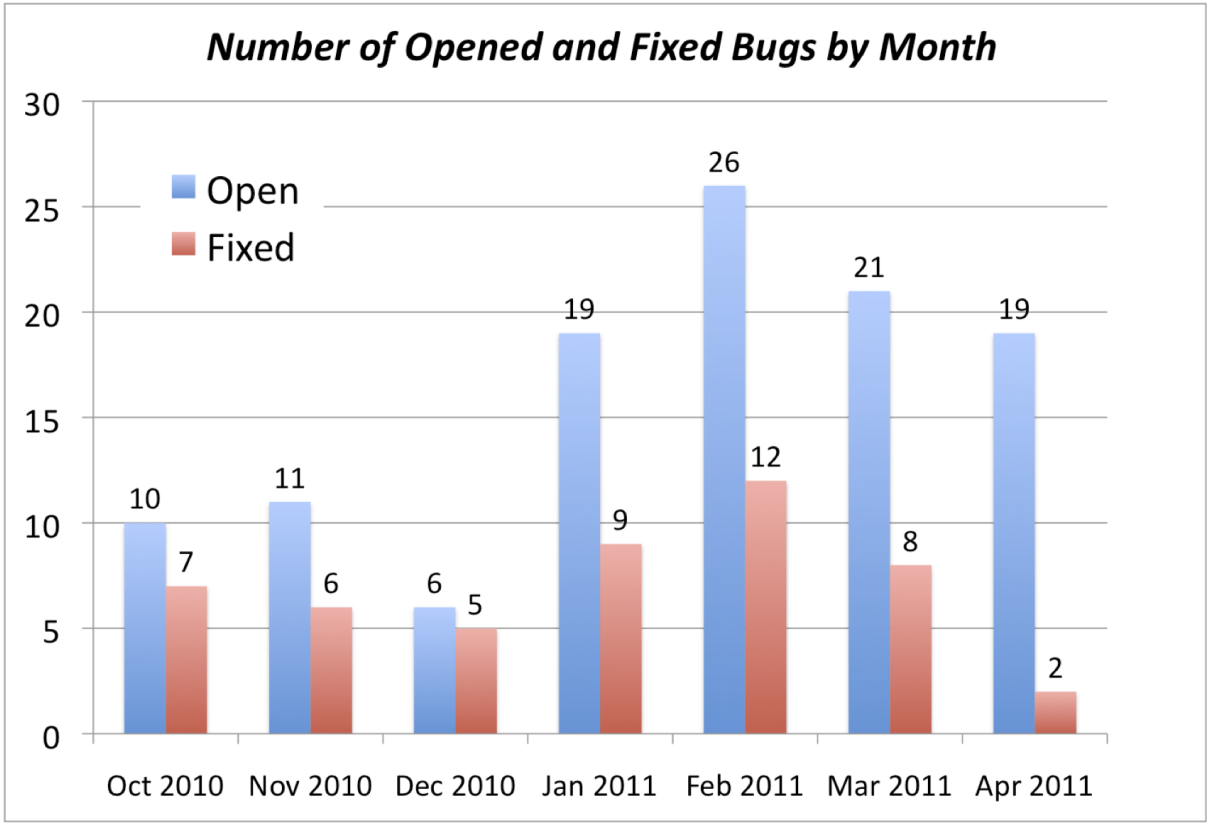
- I/O component represents small bursty I/O patterns.
- 2,048 core test, writes 3 set of restart files. One-file-per-proc.
- Each restart file set has 10,240 files, most files of 10 MB size, total of 153 GB.
- 5% performance gain via default IOR parameters.
- Another 5% gain using 2 buffers of 2 MB each instead.
- Consistent with IOR result.

I/O Degradation



- I/O performance degradation observed after CLE3.1UP03 upgrade.
- 1.560 core IOR test. Each writes 2GB. Using all 156 OSTs on one file system to measure aggregate IO performance.
- Performance dropped 30-40%.
- Cray has identified the cause. Working on a patch for us.

Bugs Reported from NERSC

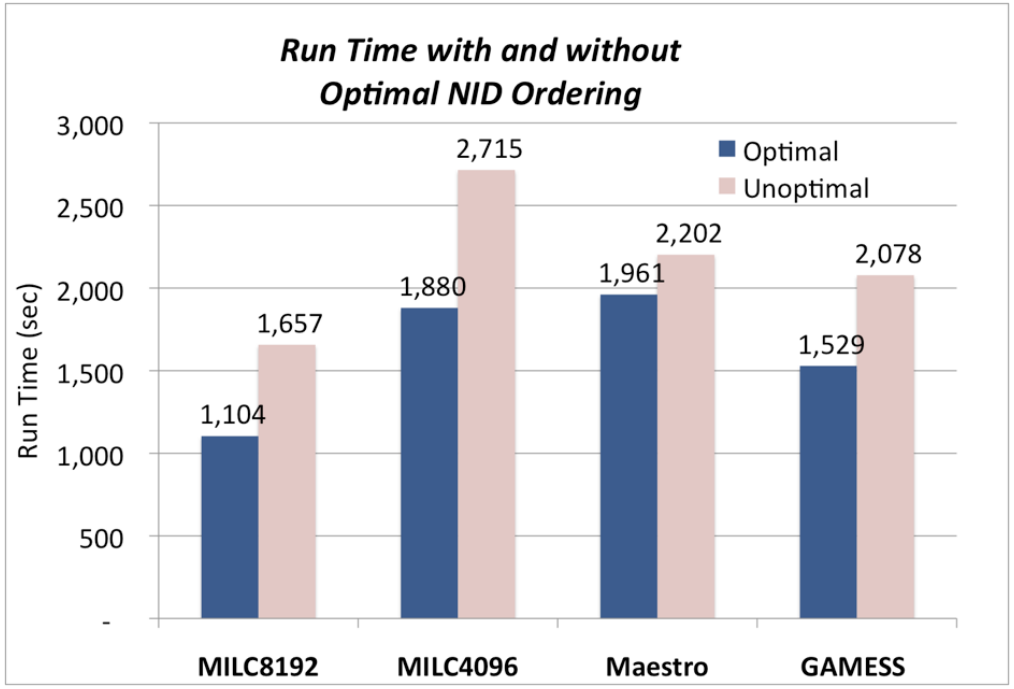


NERSC has filed more bug reports percentage-wise than the total of XT/XE peak flops we own since we have big number of users and wide variety of applications.

Problems Reported and Fixed

- Low MPI bandwidth when small pages are used
  - 3.5 GB/s for small pages, 6 GB/s for large pages.
- Scheduling problems – orphaned reservations
  - Related to many jobs submitted via job arrays overwhelming ALPS
- Mixing C++ and Fortran segfaults in PrgEnv-gnu
  - Symbol from libgfortran.a not resolved
- OpenMP compiler flags
  - PGI wrapper has no OpenMP
  - Options to –mp lost In PGI wrapper
- Libsci dynamic linking
  - Dependency on FFTW3
  - Introduced CRAY_LIBSCI_FFTW_PATH
- Dynamic libraries slowness
  - DVS layer not configured to read shared objects in parallel
  - Working with Cray (not fixed yet)

NID Ordering



- Original node allocation, is based on physical location of nodes.
- Improved node allocation, xyz ordering, is aware of interleaving topology. Still 2-D.
- Optimal "xyz-by2" ordering, 3-D. Takes advantage of full torus bisection bandwidth.
- Performance degradation noticed after CLE3.1UP03 upgrade. Discovered the optimal NID ordering got lost.

Complications between xt-mpich2 and xt-shmem

- We used to load xt-mpt by default for users, but it becomes deprecated.
- 3 options: We choose C since it affects least number of users.
  - A: load xt-mpt
    - All dynamic linking fail.
  - B: load xt-mpich2 only
    - Need to contact users who load a specific xt-mpt version
    - All shmem codes compilation fail
  - C: load both xt-mpich2 and xt-shmem
    - Need to contact users who load a specific xt-mpt version
    - Both static and dynamic linking successful
    - *Some* dynamically linked executable has run time error due to "undesirable dependency issues" between libmpich2 and libsma.
      - dmapp_dreg.c:391: _dmappi_dreg_register: Assertion `reg_cache_initialized' failed
      - Tell users to unload xt-shmem

Mysterious Error Messages

- **ERROR - nem_gni_error_handler(): a transaction error was detected, error category 0x4 error code 0xb2e**

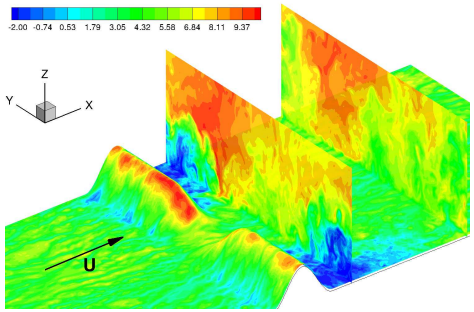
  Rank 0 [Mon Mar 7 03:46:10 2011] [c6-3c1s5n1] GNI transaction error detected
  - **Found to be accompanied by a wide variety of other error messages, such as Fatal MPI error, ALPS error, PGFIO/stdio error, segmentation fault, which are better indication of true causes for job failures.**

- **ERROR - MPID_nem_gni_check_localCQ(): Replaying failed network transaction**
  - **Many of these error messages in one job is usually followed by:**
  - **[NID 03782] 2011-04-20 18:45:43 Apid 1925046 killed. Received node failed or halted event for nid xxxx.**
  - **Indicating the failed node is the cause.**

Helpful Error Messages

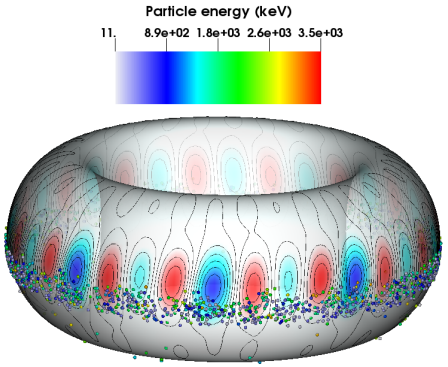
- PtlNIInitfailed: PTL_NOT_REGISTERED
  - Franklin executable submitted to Hopper
- error while loading shared libraries: libxxxx.so not found
  - CRAY_ROOTFS is not set
  - LD_LIBRARY_PATH is not updated with user's own shared objects.
- OOM killer terminated this process
  - User needs to reduce memory or use fewer cores per node
- node count exceeds reservation claim
  - ALPS tries to use more than the number of node requested via Torque keywords.
  - Check aprun command options carefully

Selected User Feedbacks



"The best part of Hopper is the ability to put previously unavailable computing resources towards investigations that would otherwise be unapproachable." – Hopper User

" During the "free" period Hopper provided very good turnaround for my jobs,, which were in the 5,000 – 10,000 processor range. This was very important for finding errors, scaling up my code and generating new results." – Hopper User



What Users Like About Hopper

- Available software – great asset to have libsci, fftw, hdf5, netcdf, petsc, Craypat, etc
- Programming environments – module files which pick up correct software based on compiler
- Huge resource – opens up new computing and research possibilities
- Shared libraries support on compute nodes – able to run more types of applications
- Scalability – solid scaling results on Gemini network
  - significant improvement in MPI latency and Bandwidth
- Stability – Hopper is a more resilient system
  - Component failures are more easily isolated
  - Survives problems that cause full crashes on XT4 and XT5

Ongoing Issues

- I/O Issues
  - Aggregate I/O performance dropped after CLE 3.1 UP03
  - Real regression in MPI-IO capabilities we worked so hard with Cray to implement on Franklin
- DVS Slowness
  - I/O performance on GPFS file systems
  - Shared libraries slowness
- Lustre Meta Data Server (MDS) hang
  - File system hang. Affects running jobs and user logins
- Consistency between external and internal login nodes environment

Summary

- Successful early user period on Hopper
- Researchers appreciate the big resource and stability of the system and they want more time.

- NERSC will continue to work with Cray to improve the system.
  - Test and submit bugs on a young MPI software stack
  - Tune DVS performance on GPFS file system
  - Examine queuing structure to improve job throughput and utilization by grouping large jobs together
  - Synchronize software releases on external login nodes and internal nodes

Acknowledgement

- Cray support teams (on-site and remote)
- NERSC User Services and Systems groups staff
- Joint NERSC/Cray Center of Excellence staff
- Authors are supported by the Director, Office of Science, Advanced Scientific Computing Research, U.S. Department of Energy.