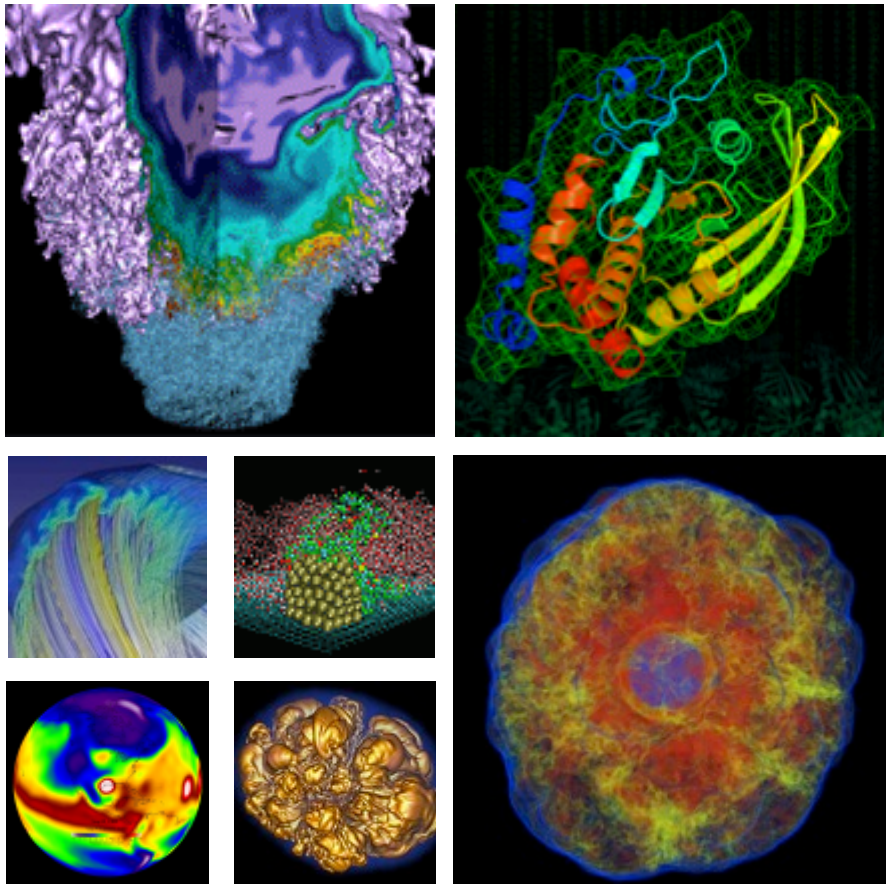
NUG Teleconference June 2013



Richard Gerber
NERSC User Services

June 6, 2013

Connection Info

Connection Info
Topic: NUG Web Conference
Date and Time:
Thursday, June 6, 2013 11:00 am, Pacific Daylight Time (San Francisco, GMT-07:00)
Event number: 662 377 626
Event password: edison

https://nersc-training.webex.com/ and chose from the list of events.

Teleconference information

1-866-740-1260
PIN: 4866820

Agenda

- Edison Phase II
- NERSC 8 Project Update
- New on the Web (My NERSC and status)
- NGF Usage in NIM
- Hyperthreading on Edison

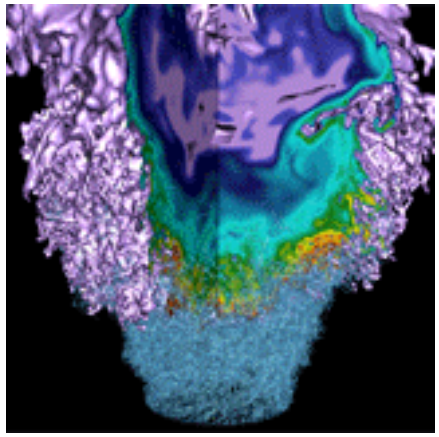
Edison Phase II

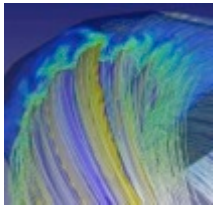
- **The full Edison Phase II system arrives June 24**
- **24 additional racks + 4 existing = 28 total**
- **3 scratch file systems**
  - Two 1.6 PB @ 35 GB/s, one 3.2 PB @ 70 GB/s
- **Installation and upgrade of existing 4 racks will take ~ 1 month**
- **Phase I system will be decommissioned on June 24**
  - Scratch data is expected to be preserved, but back up important files to HPSS (just in case)
  - Precise details will be announced soon on Edison web page

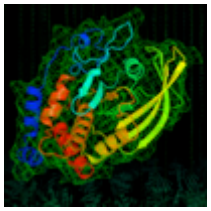
NERSC 8 Project Update

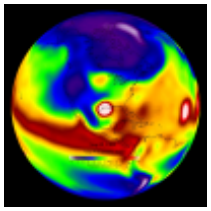
- **Preparing to release the NERSC-8 RFP within the month.**
  - We've passed DOE reviews
  - Formal RFP is being reviewed now by procurement officials.
- **RFP evaluation later this summer**
  - Vendors will have about a month to reply to the RFP.
- **We'll have an internal selection at the end of the summer and will begin to negotiate a contract with the selected vendor**
- **We'll announce the selection to the user community after contract award, approximately January 2014**
- **Estimated system delivery in late 2015**

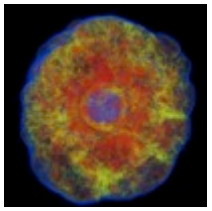
New on the Web

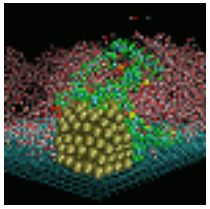




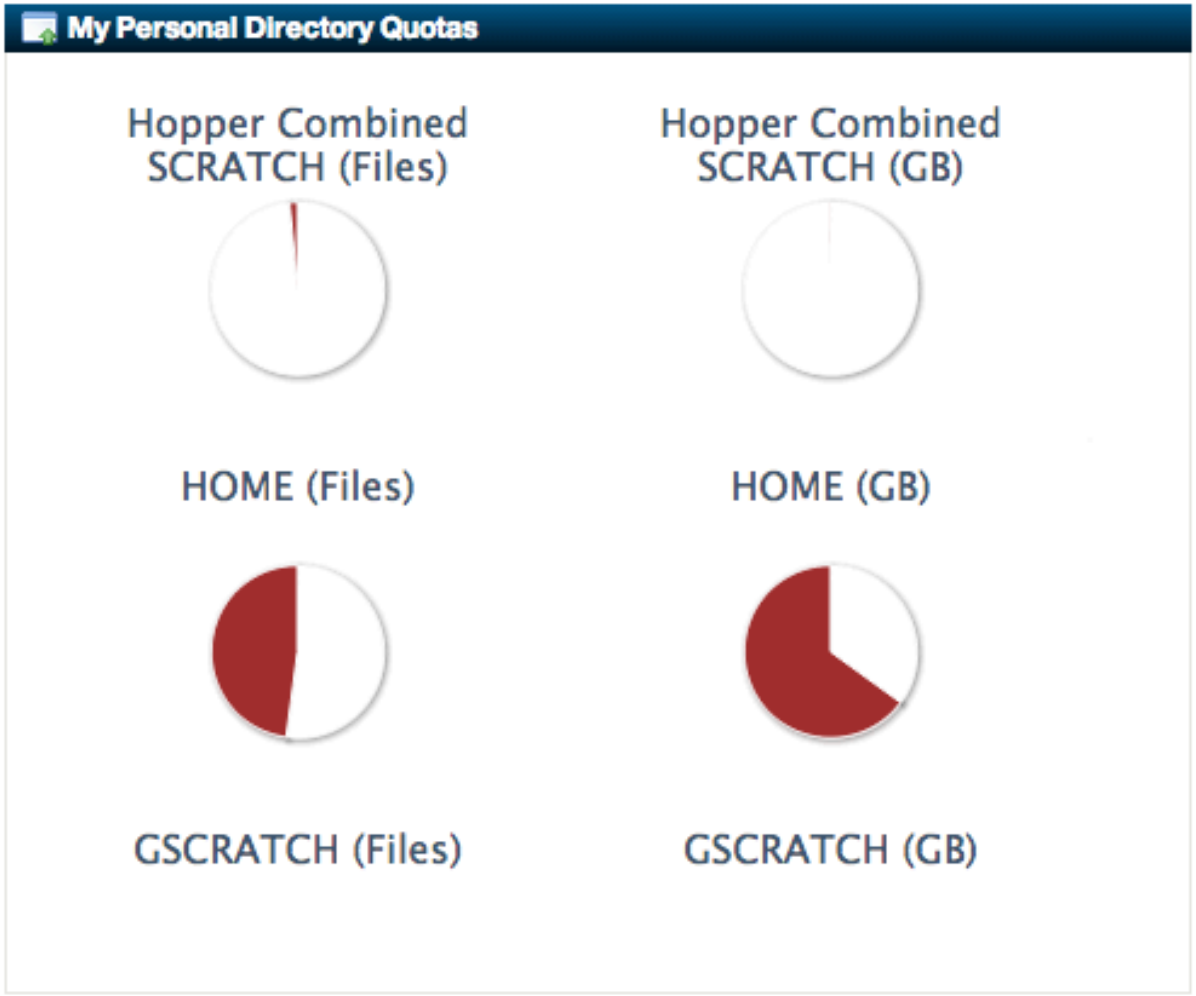




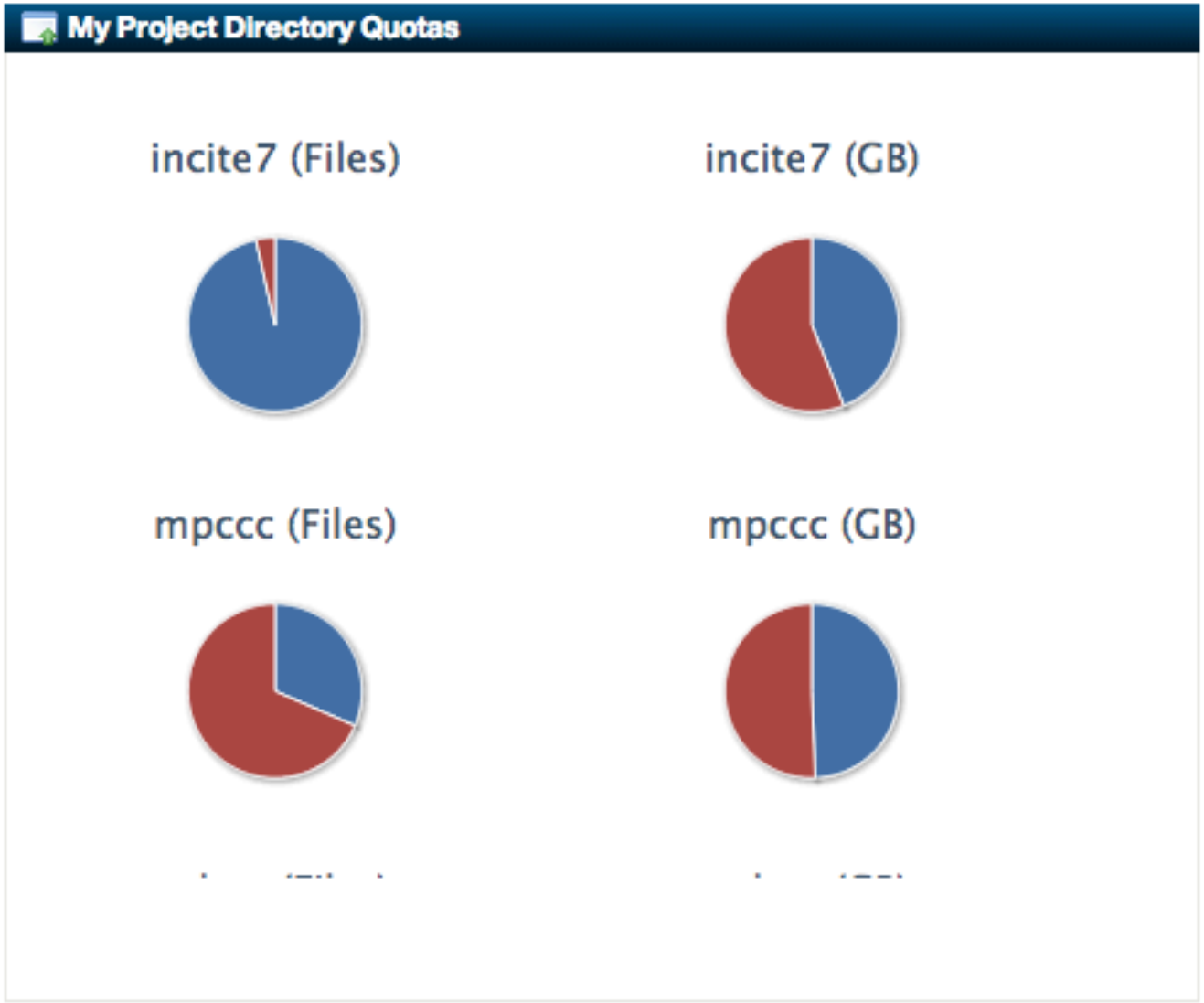




My NERSC https://www.nersc.gov/users/my-nersc/



My NERSC https://www.nersc.gov/users/my-nersc/

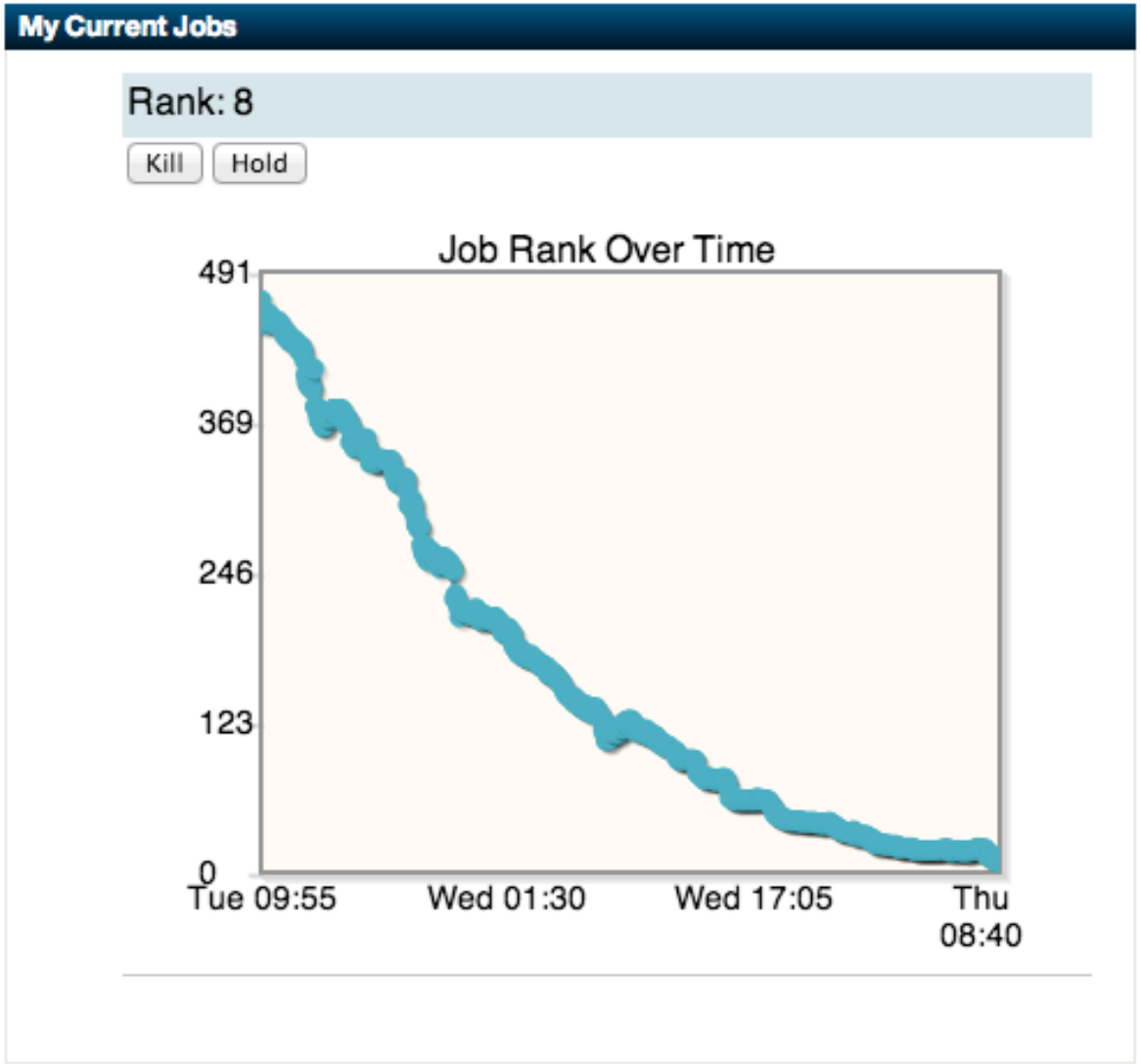


My NERSC https://www.nersc.gov/users/my-nersc/

My Current Jobs

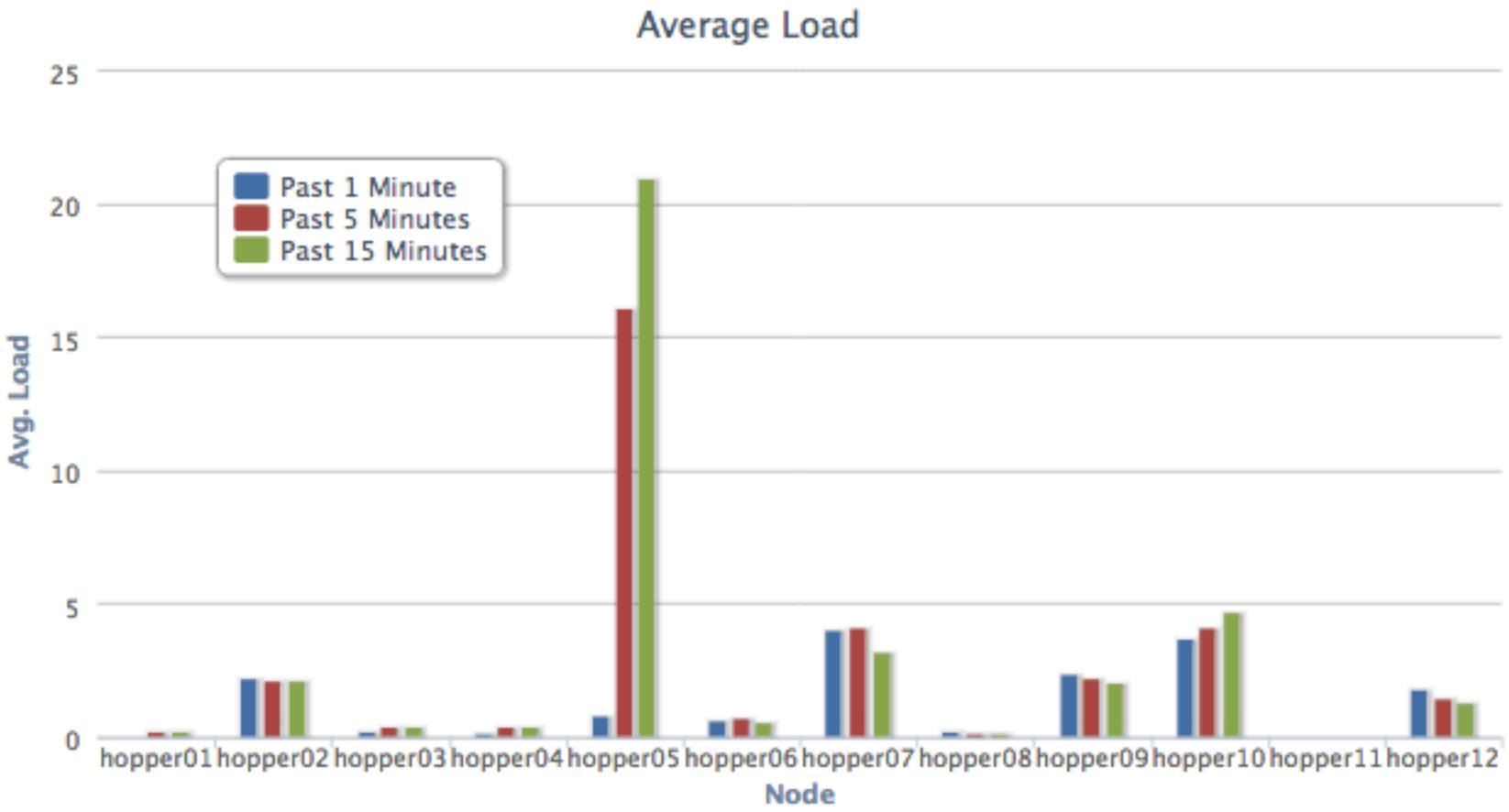
Job ID	Status/Rank
hopper:	
3260443.sdb	8
3262816.sdb	60
3265084.sdb	234
3265093.sdb	235
3265096.sdb	236
3267157.sdb	335
3267159.sdb	336
3267170.sdb	339

My NERSC https://www.nersc.gov/users/my-nersc/



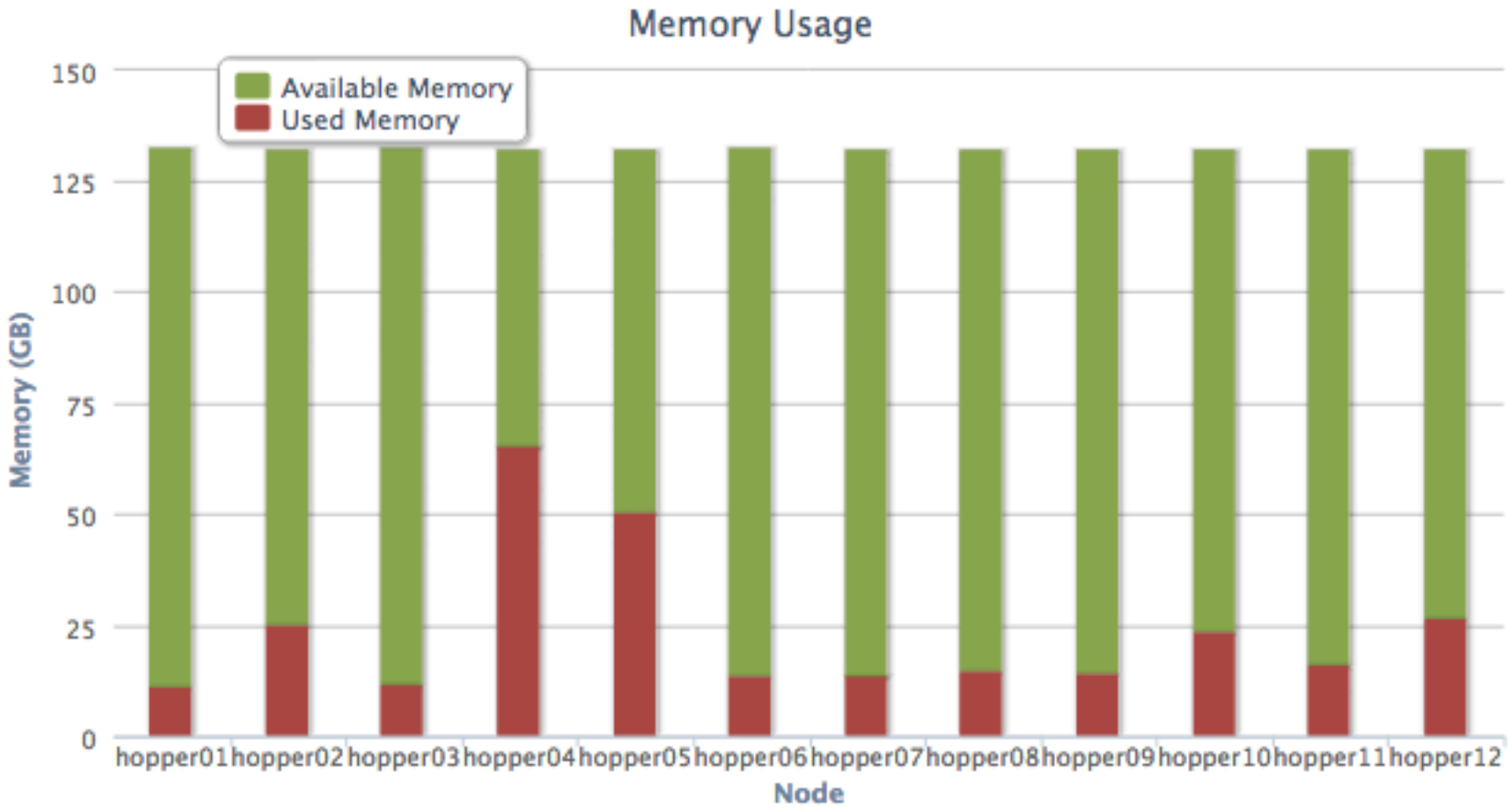
Live Status https://www.nersc.gov/users/live-status/

Login Node Status (Hopper shown here)



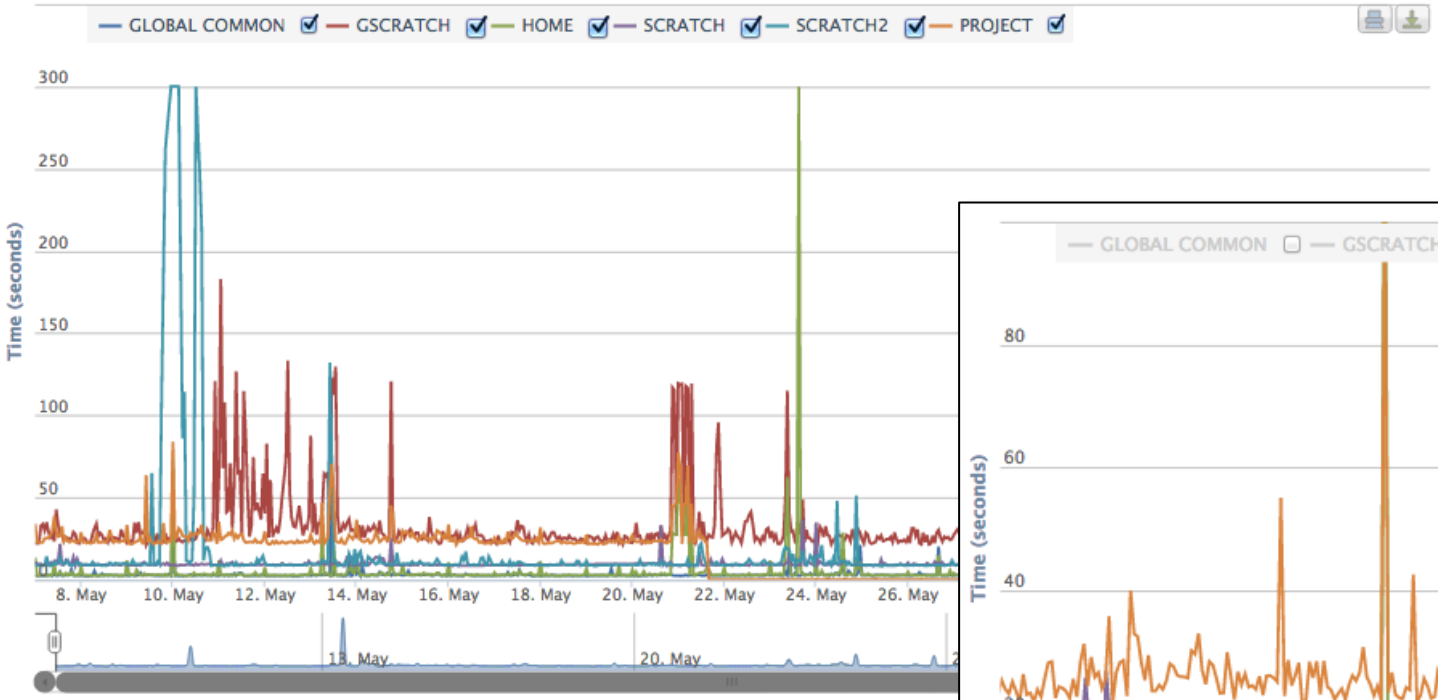
Live Status https://www.nersc.gov/users/live-status/

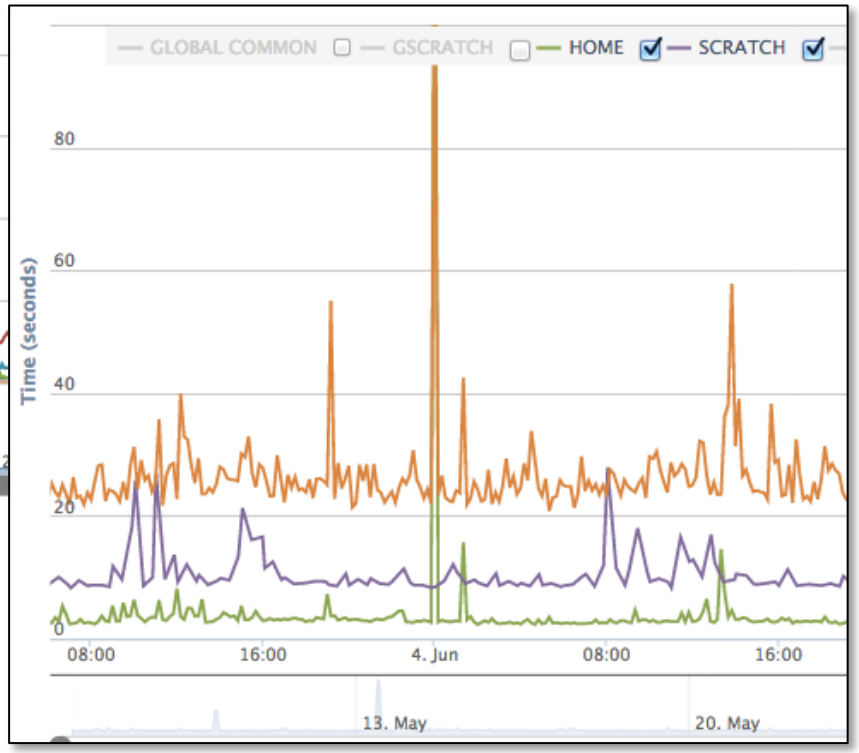
Login Node Status (Hopper shown here)



Live Status https://www.nersc.gov/users/live-status/

Time needed to execute typical interactive task on Hopper (small compile shown)





Pan and zoom

NGF Usage in NIM https://nim.nersc.gov/

Usage for My Project Directories

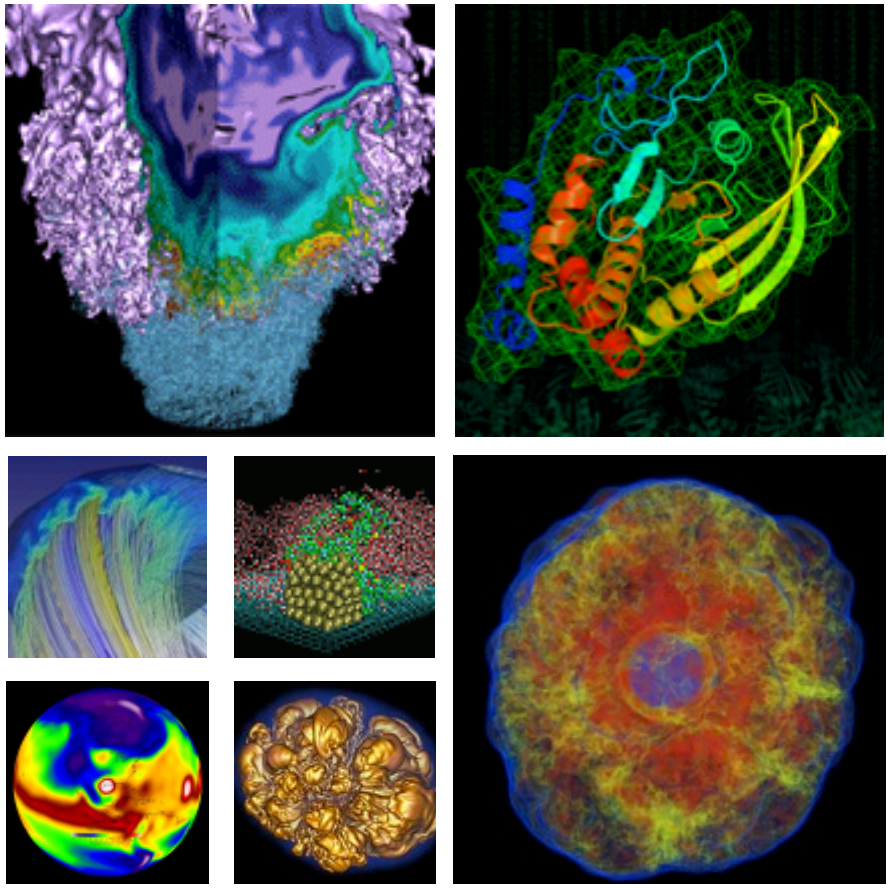
Project Directory	Owner	Group Name	ERCAP Project	Space Usage (GiB)	Space Quota (TiB)	Default Space Quota?	Space%	Inode Usage	Inode Quota
AFRDrepo	jlvay	AFRDrepo	incite7	0.1	4	Y	0	398	4,000,000
acceldac	ryne	acceldac			4	Y			4,000,000
gc2	llee	gc2			4	Y			4,000,000
incite7	geddes	incite7	incite7	2,294	4	Y	56	141,212	4,000,000
vacet	wes	vacet	HPCvis	27,105	32	N	83	2,420,108	4,000,000
vorpal	cary	vpusers	incite7	447	4	Y	11	1,716,365	4,000,000

Open Issues

- **System area for shared software – no update**
- **Debug / Interactive node reservations**
  - TBA soon after implementation details worked out
  - NUGEX queue committee has had valuable input
- **NUG 2014 Planning**
  - Committee forming
  - Looking for a year-long theme in celebration of NERSC's 40th Anniversary
- **Others?**

Other Issues or Questions?

Effects of Hyper-Threading on the NERSC workload on Edison



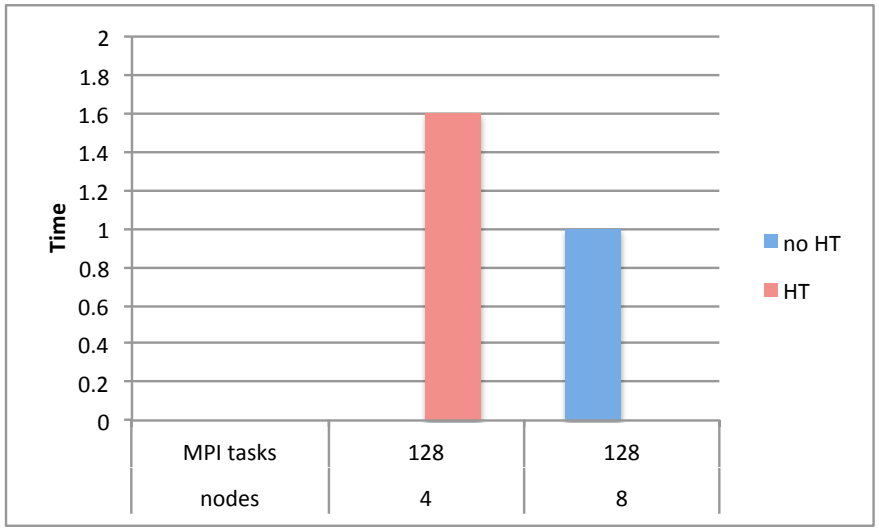
Zhengji Zhao, Nicholas Wright, and Katie Antypas
NERSC

NUG monthly meeting, June 6, 2013

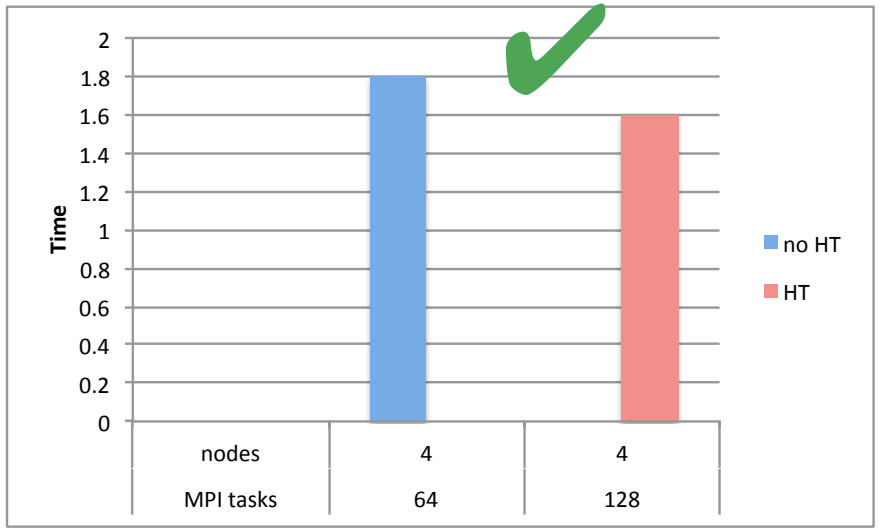
What is Hyper-Threading (HT)?

- HT is Intel's term for its simultaneous multithreading implementation. HT makes each physical processor core appears as two logical cores, which share the physical execution resources.
- HT increases the processor resource utilization by allowing OS to schedule two tasks/threads simultaneously, so that when an interruption, such as cache miss, branch misprediction, or data dependency, occurs with one task/ thread execution, the processor executes another scheduled task/thread.
- According to Intel the first implementation only used 5% more die area than the comparable non-hyperthreaded processor, but the performance was 15–30% better.

What performance benefit from HT can we expect and how should the benefit be measured?



Using a fixed number of MPI tasks and half the number of nodes, if time with HT < 2X the time with no HT, then throughput is increased. You get charged less.



Using a fixed number of nodes, HT *may reduce the run time* using 2X the number of MPI tasks – a clear win for HT.

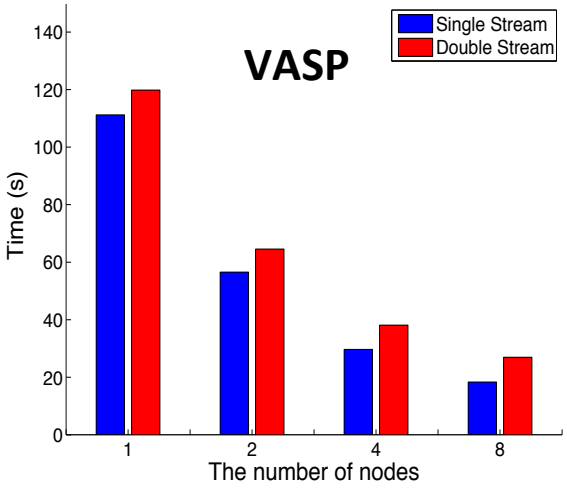
But it *may increase the run time* for other codes.

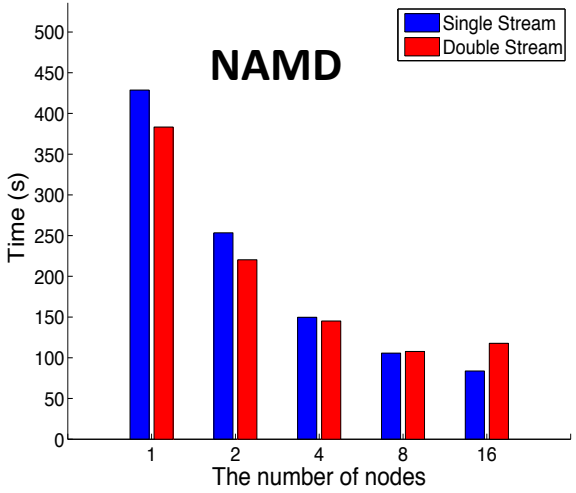
Using a fixed number of nodes is the most relevant way to make comparisons.

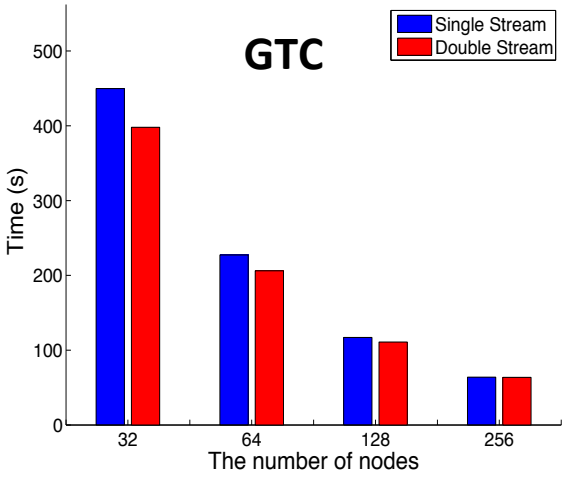
Our methods and tests

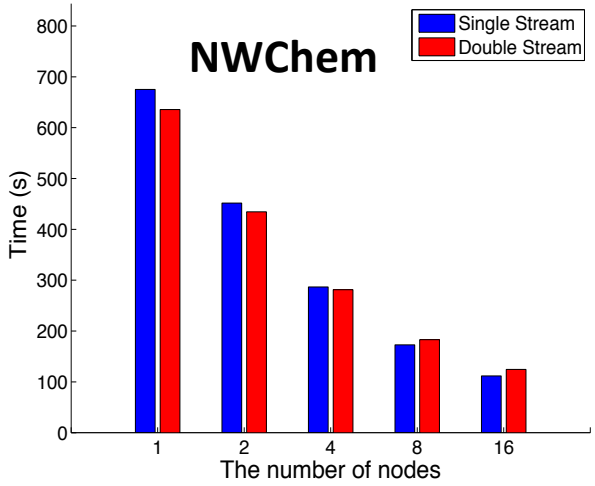
- **We measured the run time of a number of top NERSC codes using a fixed number of nodes.**
- **We used the NERSC profiling tool IPM (with PAPI) to measure hardware performance events. We measured**
  - Communication overheads, memory usage
  - Total instructions completed, and the cycles used per instruction retired (CPI).
  - L3 cache misses, branch mis-predictions, TLB data misses
  - Sandy Bridge counter configuration does not allow an easy measure of floating point operations

Runtime comparisons with and without HT



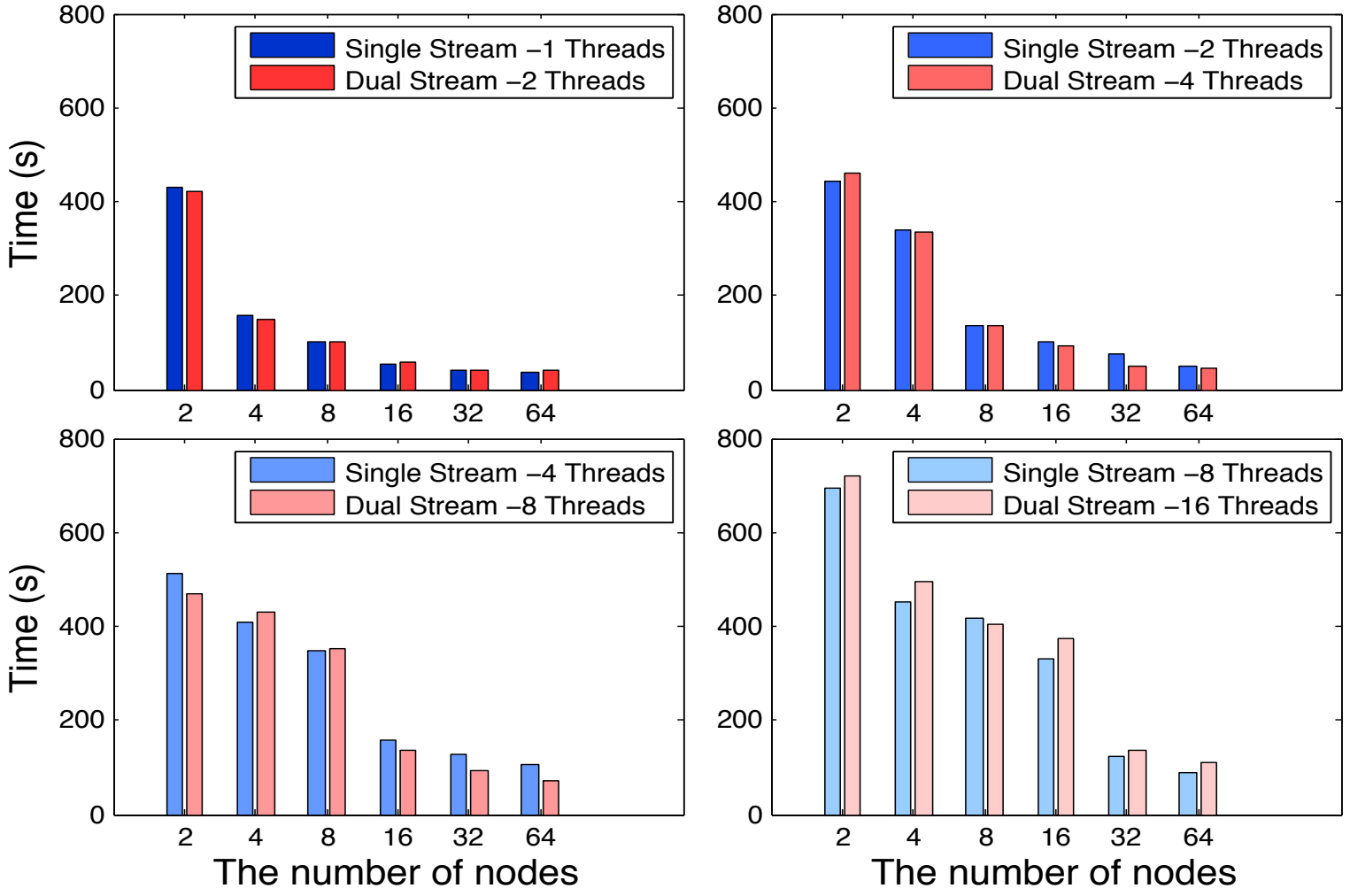






- **The HT does not help VASP performance at all node counts. The slowdown is 10-50%.**
- **HT benefits NAMD, NWCHem and GTC codes at smaller node counts by upto 6-13%.**
- **The HT benefit regions do not overlap with the parallel sweet spots of the codes.**

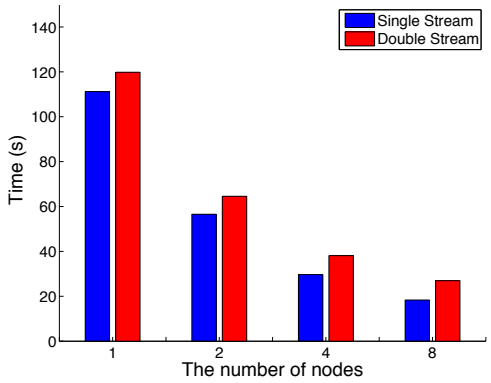
Quantum Espresso runtime with and without HT

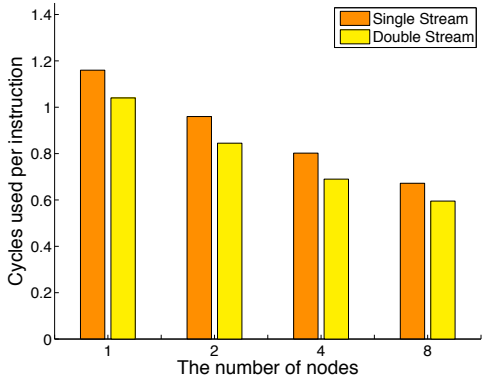


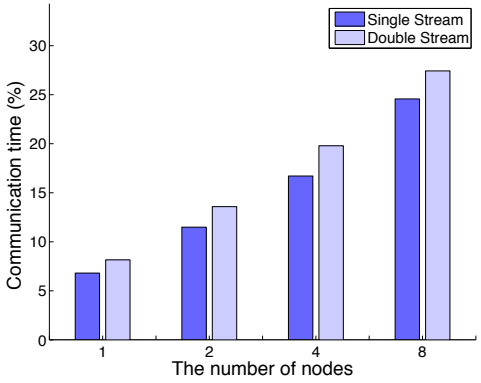
For the most of the MPI task and thread combinations, HT doesn't help the performance.

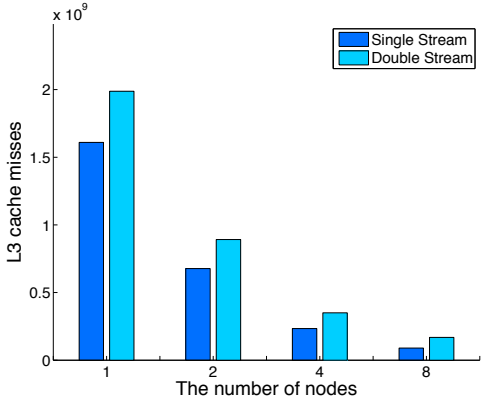
VASP:

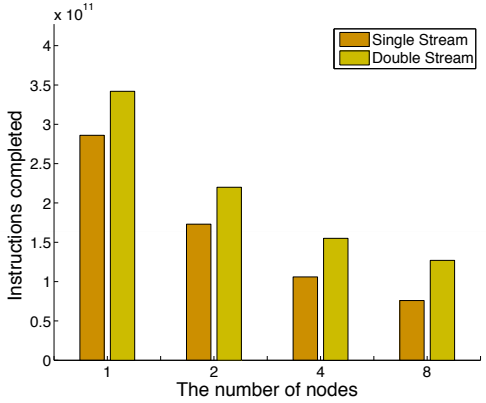
Runtime, communication overhead, instructions completed, Cycles used per instruction, L3 cache misses, and branch mis-predictions per physical core

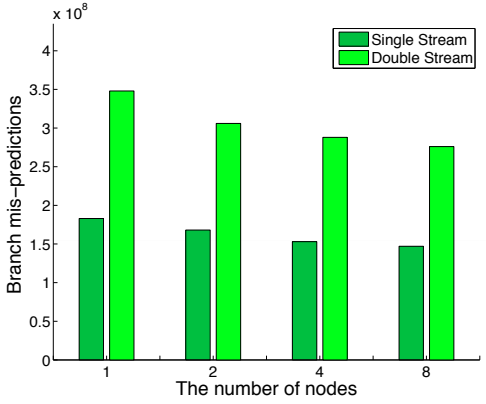












Higher CPI value may indicate an HT opportunity.

HT does not address the stalls due to data dependencies across multiple physical cores, eg., during MPI collective calls.

Values for physical cores are the sum of that on the two logical cores.

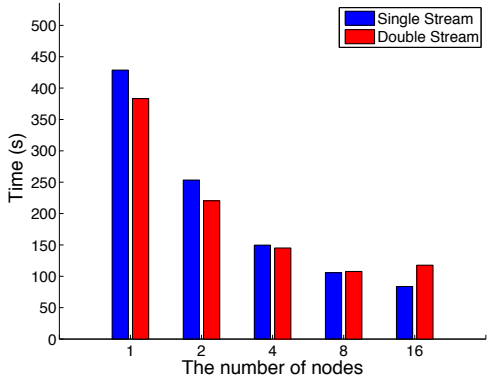
Instructions completed per physical core without HT:
$P + S$

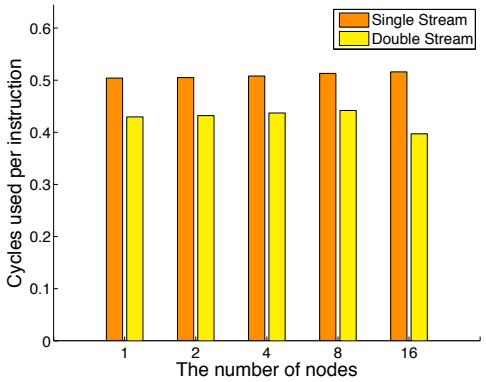
Instructions completed per physical core with HT:
$(P/2 + S) + (P/2 + S) = P + 2S$

where P and S denote the parallel and the sequential portions of the workload, respectively.

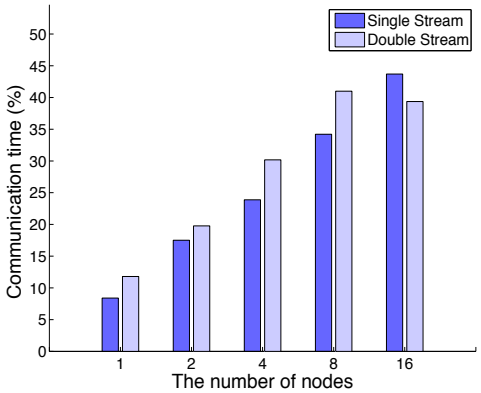
NAMD:

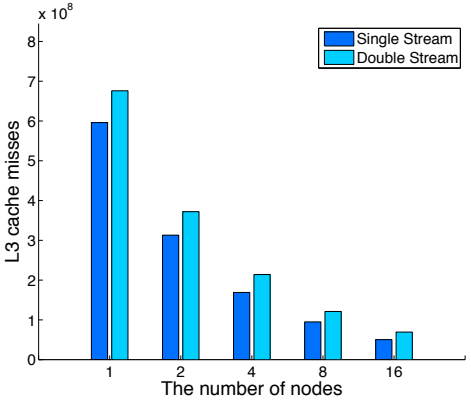
Runtime, communication overhead, instructions completed, Cycles used per instruction, L3 cache misses, and branch mis-predictions per physical core



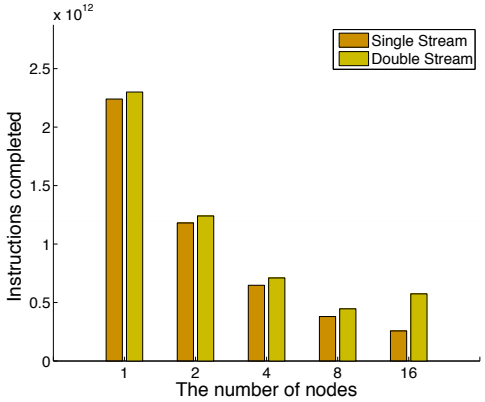


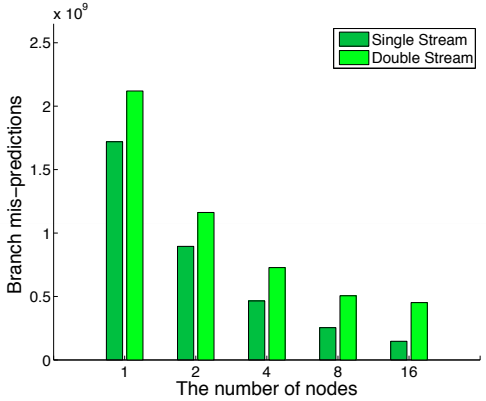
Reduced cycles per instructions with HT indicates higher resource utilizations from using HT.





HT increases the L3 cache misses and the branch mis-prediction for NAMD.

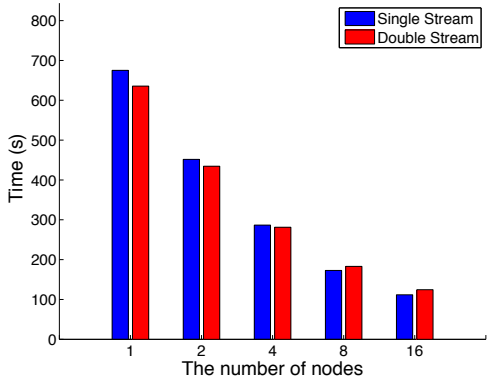


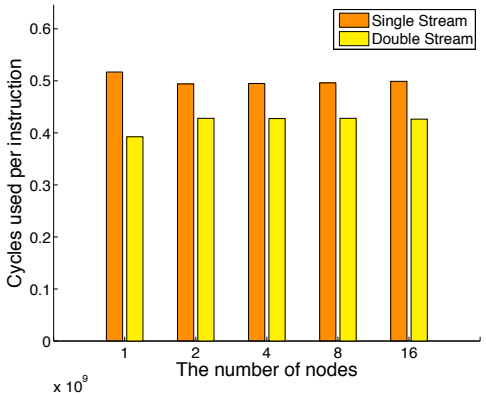


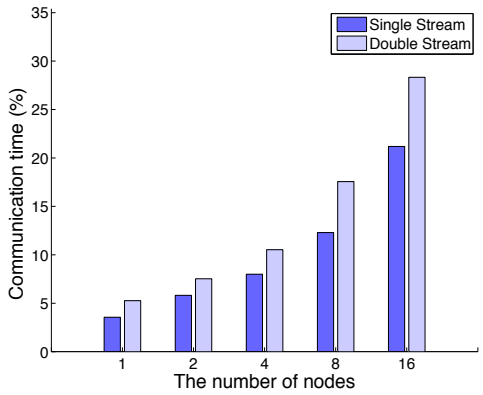
HT benefits occur at low node counts where the communication overhead is lower than that at larger node counts.

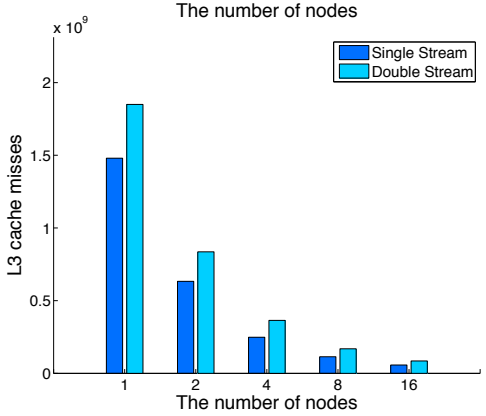
NWChem:

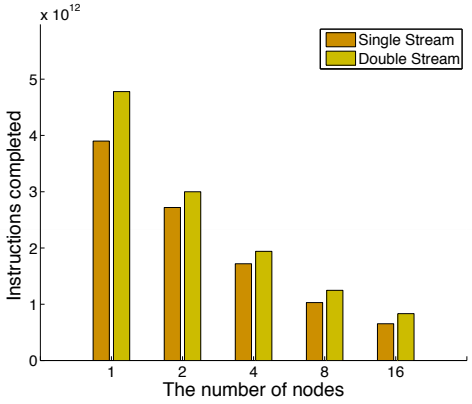
Runtime, communication overhead, instructions completed, Cycles used per instruction, L3 cache misses, and branch mis-predictions per physical core

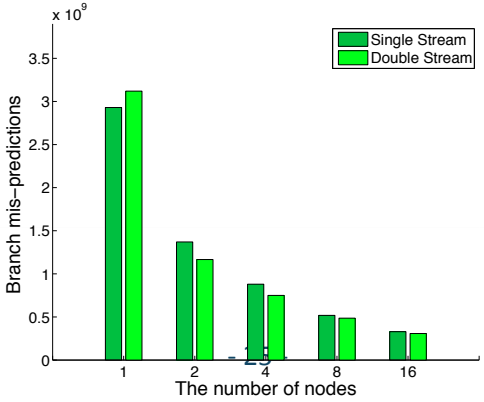








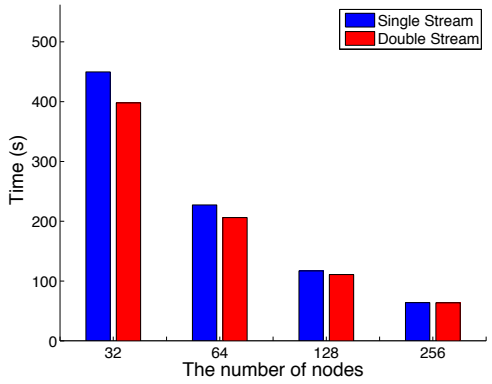


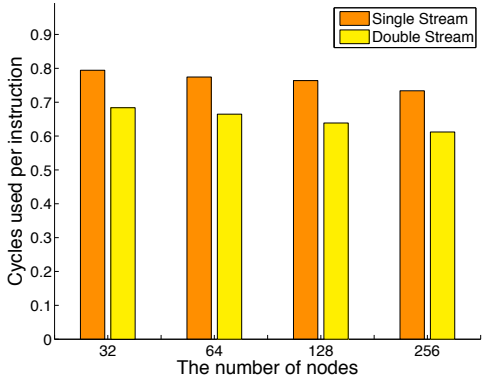


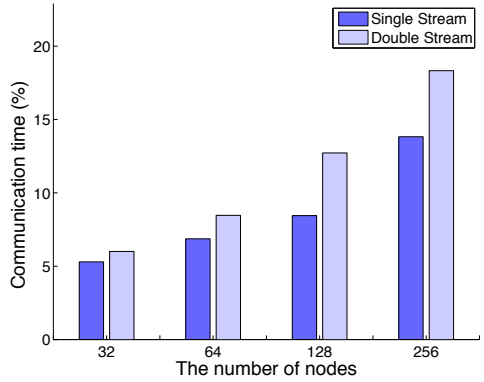
HT increases the L3 cache misses but the branch mis-prediction is similar with and without HT with HT.

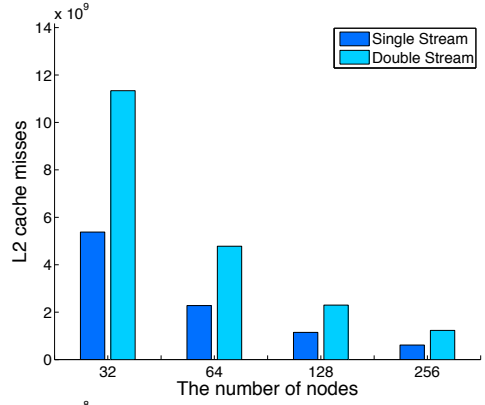
GTC:

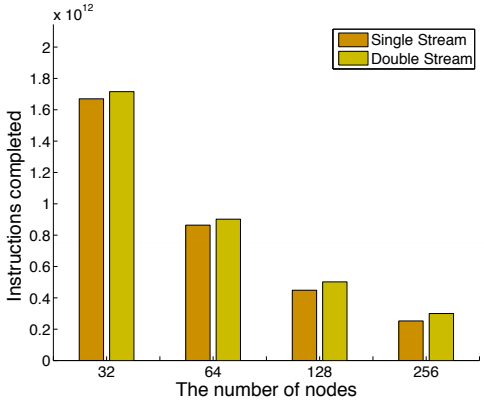
Runtime, communication overhead, instructions completed, Cycles used per instruction, L3 cache misses, and branch mis-predictions per physical core

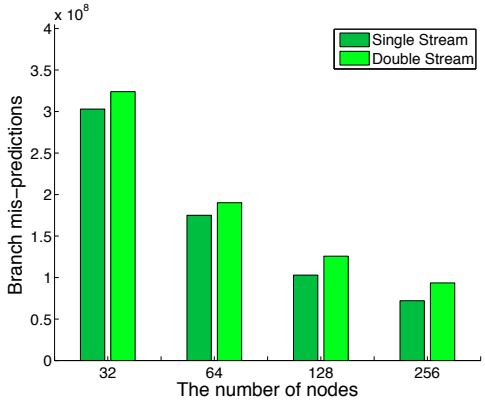












HT performance gain decreases when communication overhead increases.

The similar number of instructions completed with and without HT may indicate that the code is a highly efficient parallel code with a small sequential portion in the code.

What application codes could get performance benefit from HT?

- The cycles/instruction (CPI) value needs be sufficiently large so that HT has enough work to do to help, although HT may not address all the interruptions.
- The communication overhead should to be sufficiently small. In particular, the extra communication overhead from using twice as many MPI tasks to use HT should be small so that the amount of the interruptions that HT cannot address do not dominate the HT effect. This indicates that HT benefits are likely to happen at relatively smaller node counts in the parallel scaling region of applications except for embarrassingly parallel codes.
- We observed that codes get HT benefit when the number of instructions completed per physical core with and without HT are similar, indicating that highly efficient parallel codes (small sequential portion in the code) likely to get HT benefit.

How to collect CPI and other performance data with IPM?

```
#!/bin/bash -l
#PBS -l mppwidth=64
#PBS -l mppnppn=32
#PBS -l walltime=00:30:00
#PBS -j oe
#PBS -N test

cd $PBS_O_WORKDIR
export IPM_LOGDIR=.
export IPM_LOG=full
export IPM_HPM="PAPI_TOT_CYC,PAPI_TOT_INS"
export IPM_REPORT=full

aprun  -j 1 -n 1024 ./a.out >& output.noHT
aprun  -j 2 -n 2048 ./a.out >& output.HT
```

```
# To instrument the code with IPM,
# do
    module load ipm
    ftn mycode.f90 $IPM
```

```
# To get profiling data, do
    mdoule load ipm
    ipm_parse -full <the .xml file
    generated by the instrumented
    code>
```

Conclusion

- The HT performance benefit is not only application dependent, but also concurrency dependent, occurring at the smaller node counts. HT benefits NAMD, NACHEM, GTC codes by 6-13% at smaller node counts while it slows down VASP by 10-50%.
- Whether an application can get performance benefit from using HT is determined by the competing result between the higher resource utilization that HT enables and the overheads that HT introduces to the program execution due to resource sharing, communication overhead and other negative contributions.
- Highly efficient parallel codes (less serial portion in the codes) with high CPI values are more likely to get HT performance benefit at lower core counts in the parallel scaling region where the communication overhead is relatively low.

Conclusion - continued

- HT may help users who are limited by the allocation and have to run at lower node counts. In addition, given the fact that no code optimization (no effort) is required to get HT performance benefits, the 6-13% performance gain is significant. Users are recommended to explore the ways to make use of it, however, with caution to avoid large performance penalty from using HT if running beyond the HT benefit region.
- The large gap between the HT speedup region and the parallel sweet spot of the five major codes we examed suggests that HT may have limited effect on NERSC workload at this time.
- HT as a complementary approach (thread level parallel) to the existing modern technique (instruction level parallel) to improve processor speed, has a big potential in processor design. We expect to see more HT roles in the HPC workload in the future with continuously improving HT implementations.

Acknowledgement

- Larry Pezzaglia at NERSC for his support and help with early HT study on Mendel clusters at NERSC.
- Hongzhang Shan at LBNL, and Haihang You at NICS and Harvey Wasserman at NERSC for insightful discussion and help.
- Richard Gerber and other members of the User Services Group at NERSC for the useful discussions.

Thank you!

Next NUG Teleconference

- **Next scheduled: Thu. July 11, 2013**
- **Send suggested topics and comments to ragerber@lbl.gov**

National Energy Research Scientific Computing Center