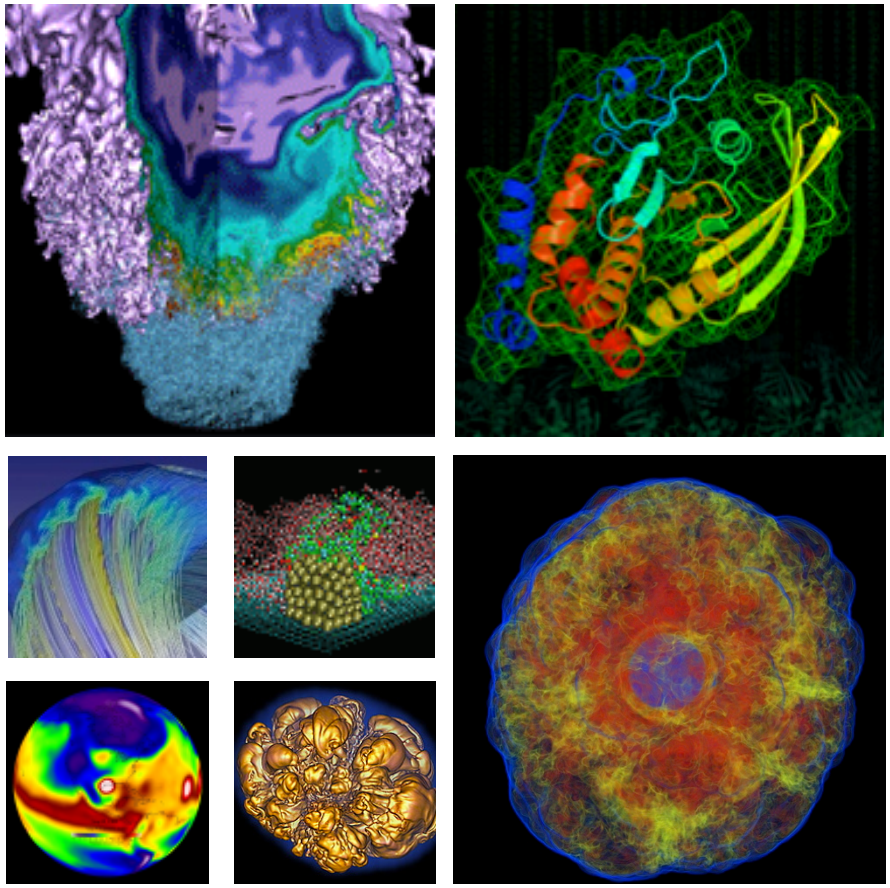
# Performance Measurements of the NERSC Cray Cascade System



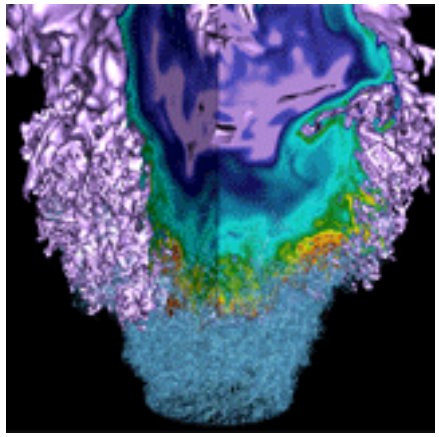
Brian Austin, Matthew J. Cordery, Harvey J. Wasserman, Nicholas J. Wright

CUG 2013, May 10, 2013, Napa, CA

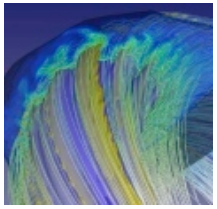
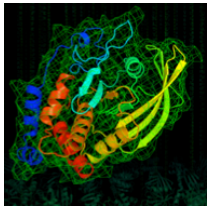
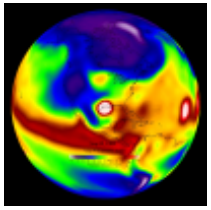
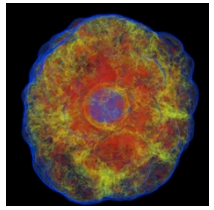
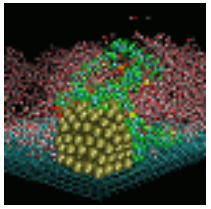
# Outline

- **Computational Systems**
- **Network Micro-benchmarks**
- **Application Benchmark Studies**
  - CAM, GAMESS, GTC, IMPACT-T, MAESTRO, MILC, PARATEC, MiniDFT
  - Performance effects of hyper-threading, core specialization and hybrid MPI+OpenMP parallelism
- **Cross-Platform Performance Comparison**
- **Conclusion**

# Computational Systems



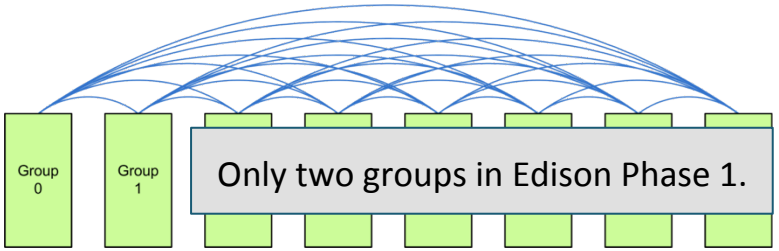
NERSC

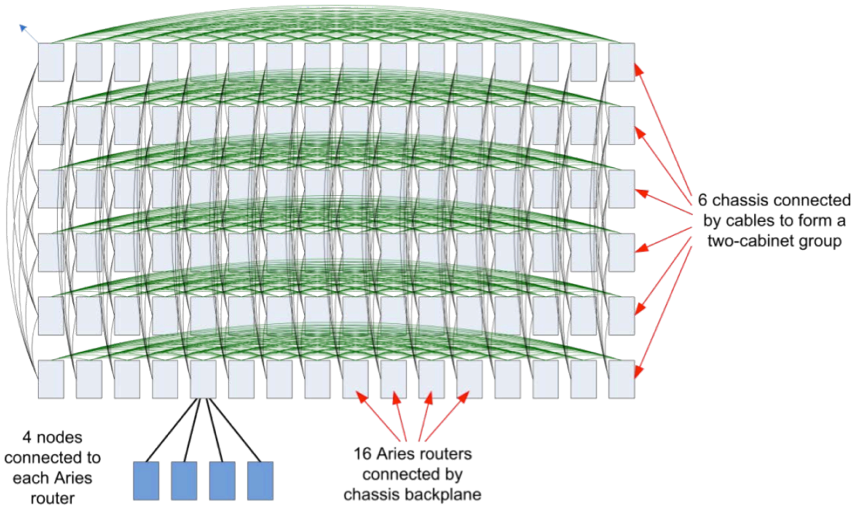






# Edison: Anatomy of a Cray XC30

- Phase 1 delivered Q4 2013
- Cray Aries interconnect
- Two 2-cabinet "groups"
- 664 compute nodes
- Two 8-core Intel 2.6 GHz "Sandy Bridge" processors per node
- 64 GB 1866 MHz DDR3 memory per node.

## Aries' Dragonfly Topology




Images: Cray XC Series Network Manual

# Edison: Technologies Explored

- **Aries Interconnect**
  - Low latency, high point-point and global bandwidth
- **Hyper-Threading**
  - Two 'virtual cores' per physical core
  - Interleaving instruction streams hides latency due to single-stream stalls.
  - Shared physical resources
- **Core Specialization**
  - Reserves one core per node for the operating system
  - MPI progress engine runs on specialized cores.

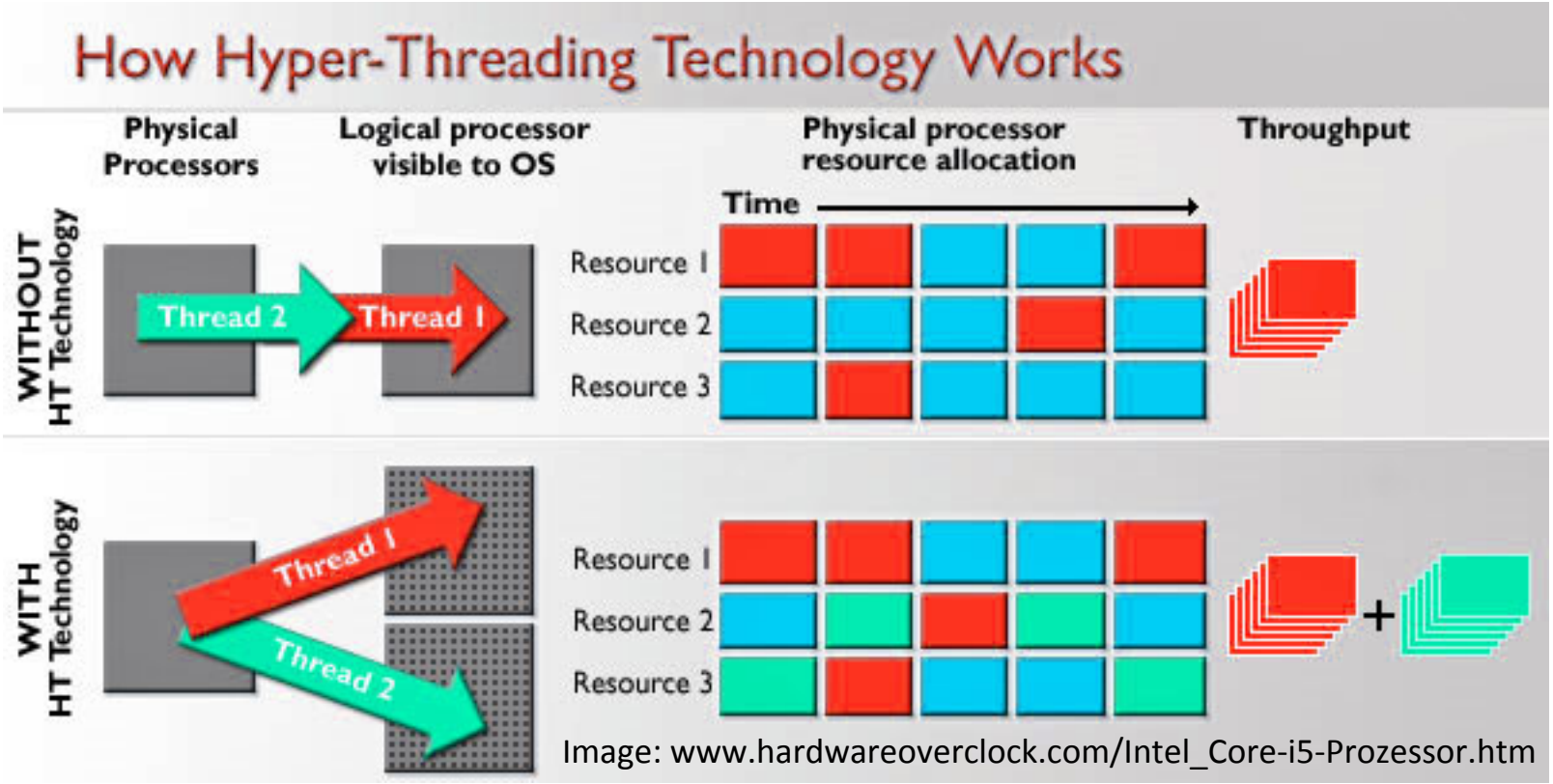


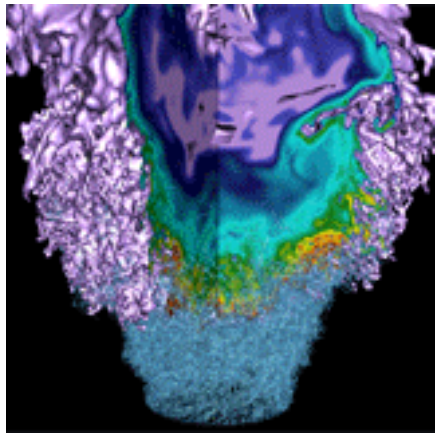

Image: www.hardwareoverclock.com/Intel_Core-i5-Prozessor.htm

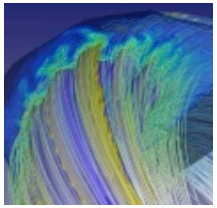
- **Resource sharing can be competitive or non-competitive.**
- **Well tuned codes with few instruction stalls typically compete for resources.**

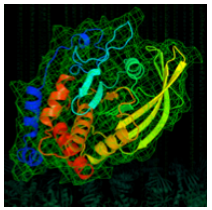
# Hopper: Cray XE6

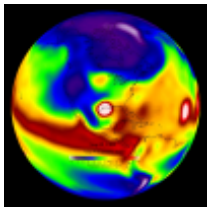
- Installed 2010
- Cray Gemini interconnect
- 3-D Torus topology
- 6,384 compute nodes
- Two 12-core AMD 2.1 GHz “Magny-Cours” processors per node
- 32 GB 1333-MHz DDR3 memory per node.

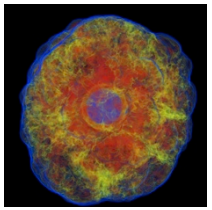
# Network Micro-benchmarks

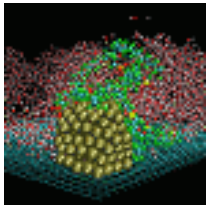






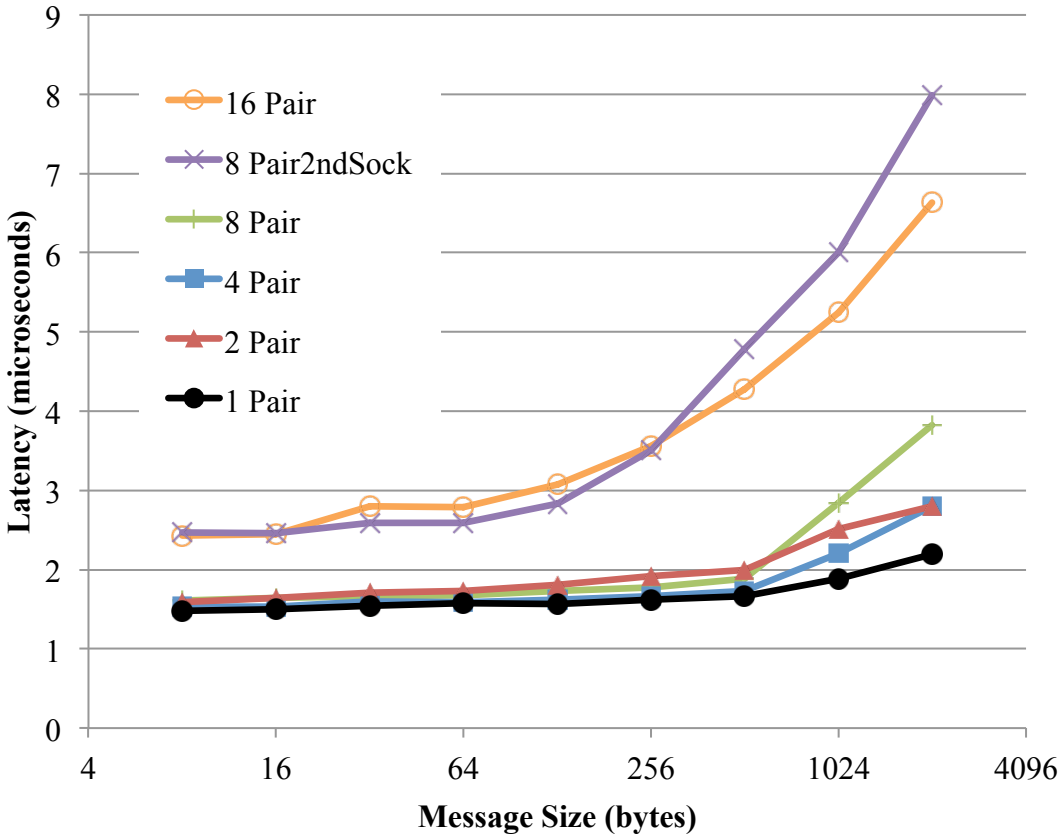






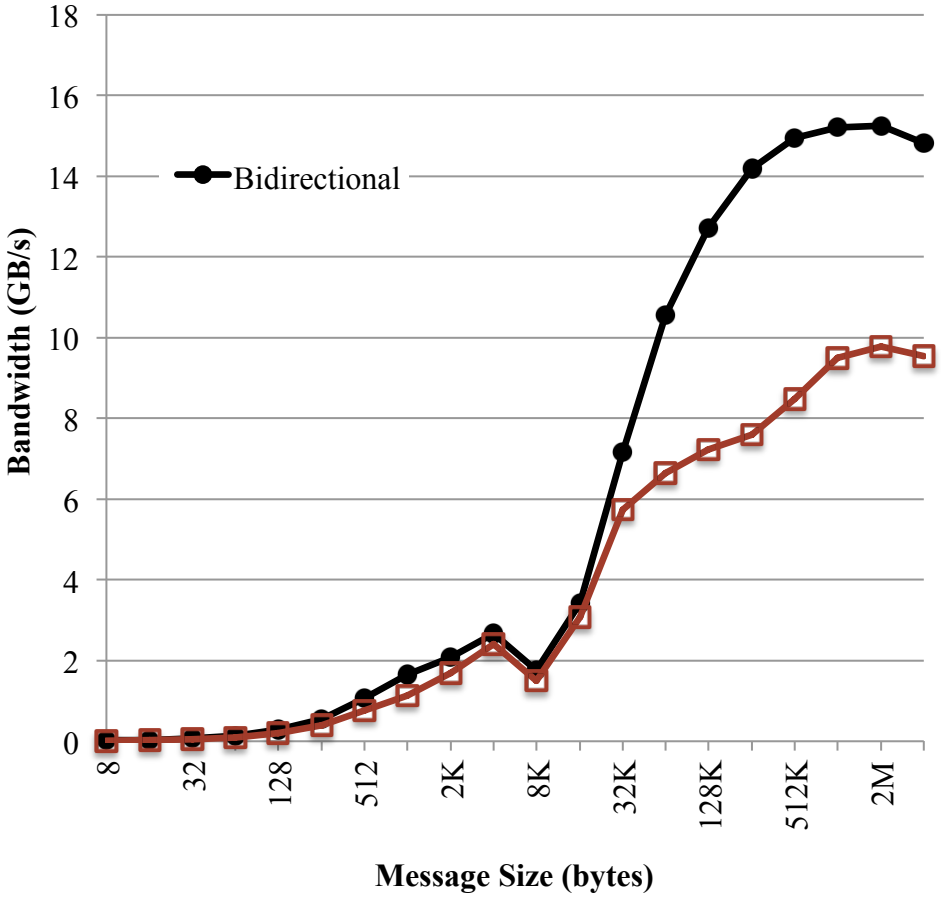
# Edison Point-to-Point Latency

- **Measured with OSU MPI multi-latency benchmark**
- **8-byte latency for 1-8 pairs: 1.5 μs**
  - Independent of location of location of nodes within group.
  - Compare to 1.8 μs for the busy Hopper system.
- **Higher 8-byte latency for 8-pairs on second socket: 2.4 μs**
- **16-pair latency comparable to 8-pair second socket.**



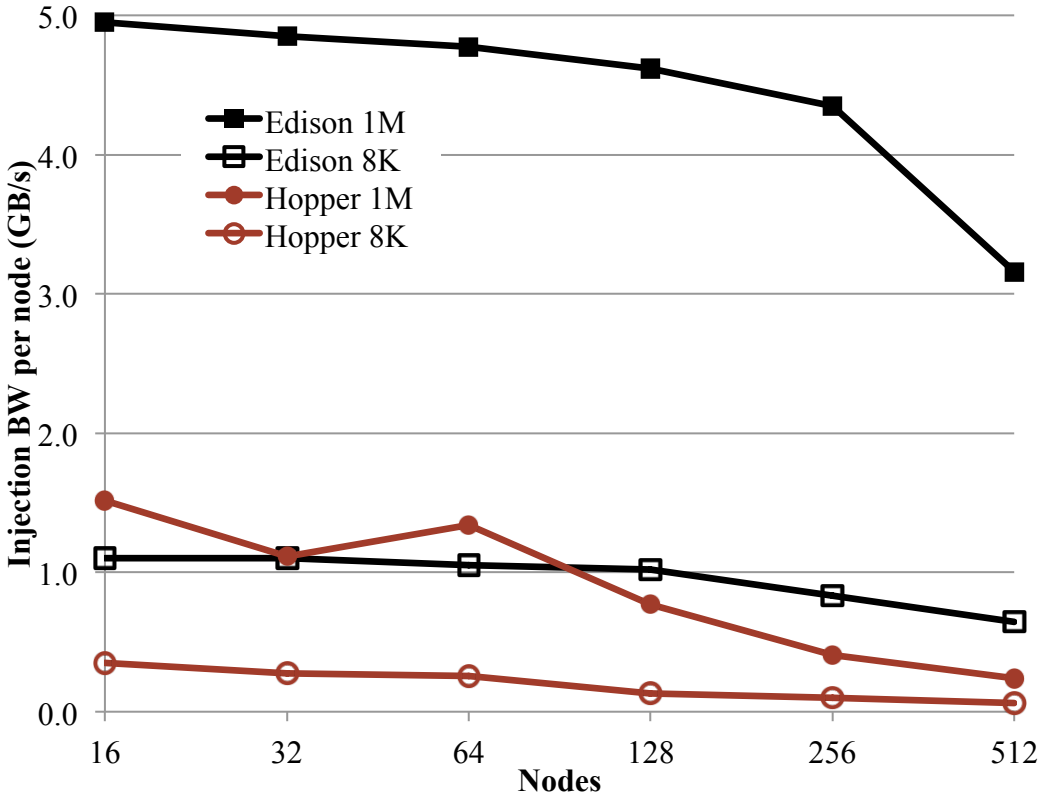
# Edison Point-to-Point Bandwidth

- **Measured with OSU MPI benchmarks**
- **Peak Bidirectional Bandwidth: 15 GB/s**
  - Compare to Hopper Z-direction: 8 GB/s
- **Peak Unidirectional Bandwidth: 10 GB/s**
  - Compare to Hopper Z-direction: 6 GB/s
- **Transition between Fast Memory Access (FMA) and Block Transfer Engine (BTE) at 8 KB.**



# All-to-all Performance

- Measured with OSU MPI all-to-all benchmark; 1 MPI process per node.
- Edison's maximum injection bandwidth per node is 3x Hopper's.
- Edison maintains high injection bandwidth as concurrency increases.
- Edison provides high global bandwidth within a 2-cabinet group.
- Edison's injection bandwidth per node drops 3 GB/s when rank-3 network is used (>384 nodes).
- The rank-3 network has an all-to-all topology; all-to-all performance *shouldn't* decrease for larger systems.



# Core Specialization (CS)

**Core specialization increases communication-computation overlap only for messages larger than 2MB.**

**Below 8KB**

- Fast Memory Access mechanism
- MPI Progress Engine not used
- Core specialization has no effect
- Overlap fraction: 0.5 – 0.75

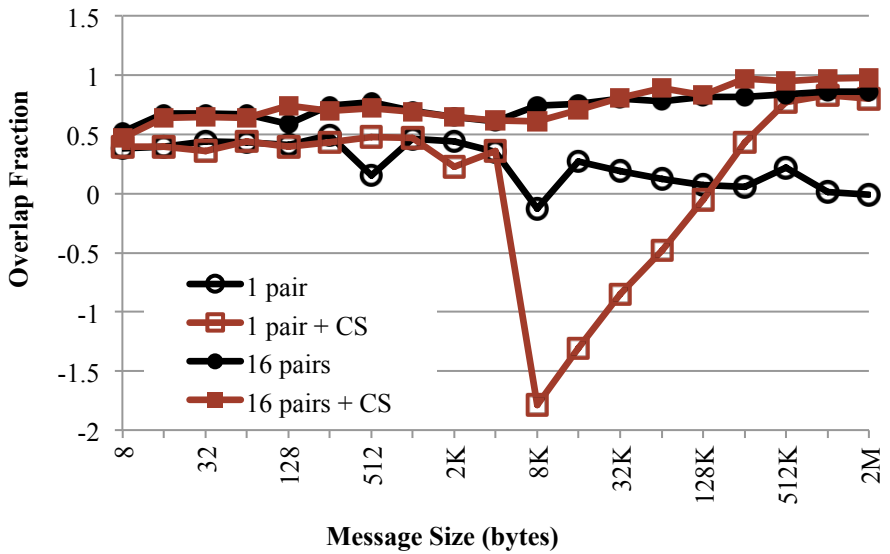
**Above 8KB**

- Block Transfer Engine mechanism
- MPI Progress Engine runs on specialized core.
- Core specialization is essential for 1-pair overlap,
- But causes negative overlap (-2) for 8 KB messages.
- Sixteen pairs obtain 0.9 overlap without CS.

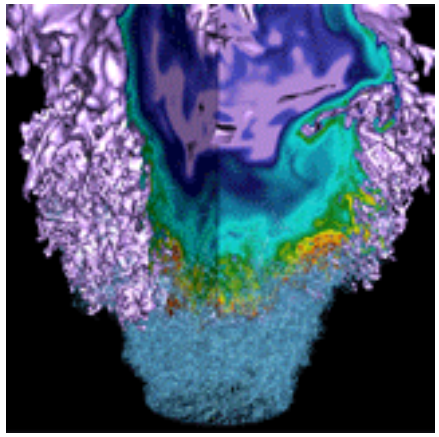
```
//CommTime
MPI_Irecv(…);
MPI_Isend(…);
MPI_Waitall(…);

//WorkTime
for(i=0; i<Nwork; i++)
     dgemm(…);
```

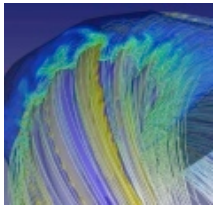
```
//NonblockingTime
MPI_Irecv(…);
MPI_Isend(…);
for(i=0; i<Nwork; i++)
     dgemm(…);
MPI_Waitall(…);
```

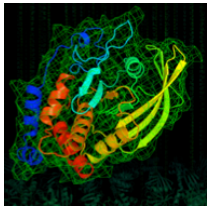
$$OverlapFraction = 1 - \frac{NonblockingTime - CommTime}{WorkTime}$$

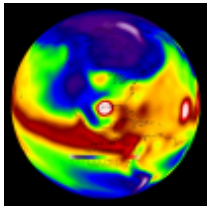


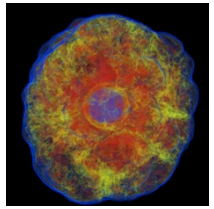
# Application Benchmark Studies

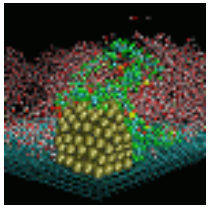












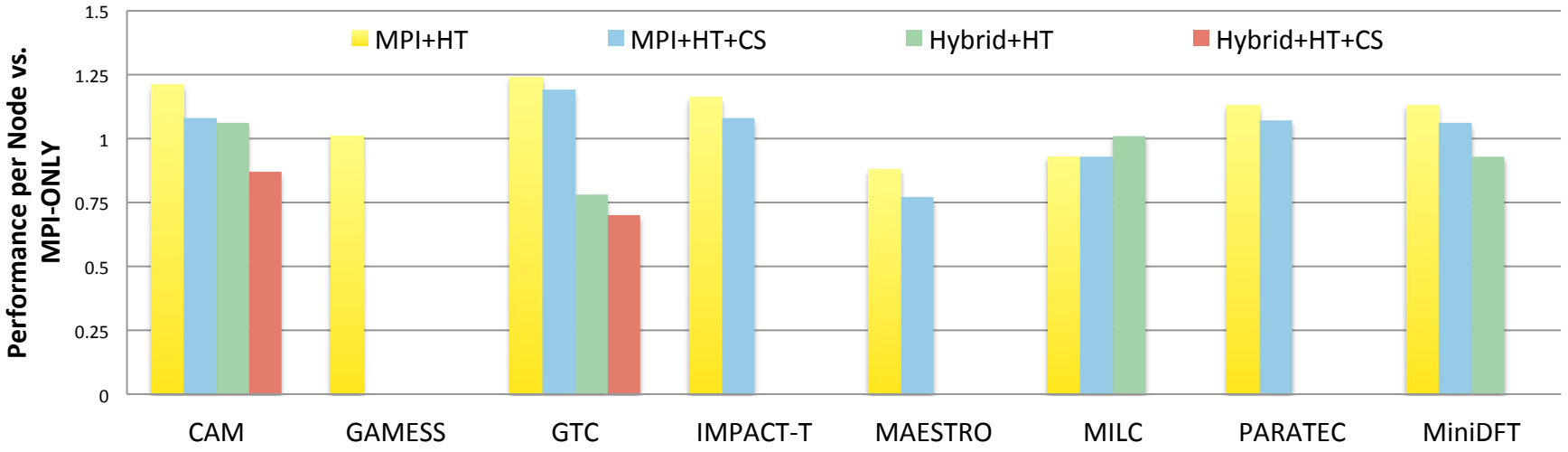
# What combination of HT and CS maximizes Edison's Sustained System Performance?

- **NERSC-6 SSP benchmark suite + MiniDFT**
  - CAM, GAMESS, GTC, IMPACT-T, MAESTRO, MILC, PARATEC
- **Default programming environment (Intel + MKL) used for all codes except PARATEC (Cray + LibSci)**
- **Fixed MPI concurrency for each code**

| Experiment | Aprun options |
|---|---|
| MPI-Only | -N16 |
| MPI+HT | -N32 –j2 |
| MPI+HT+CS | -N32 –r1 –j2 |
| Hybrid+HT | -N16 –d2 –j2 –cc numa_node |
| Hybrid+HT+CS | -N15 –r1 –d2 –j2 –cc numa_node |

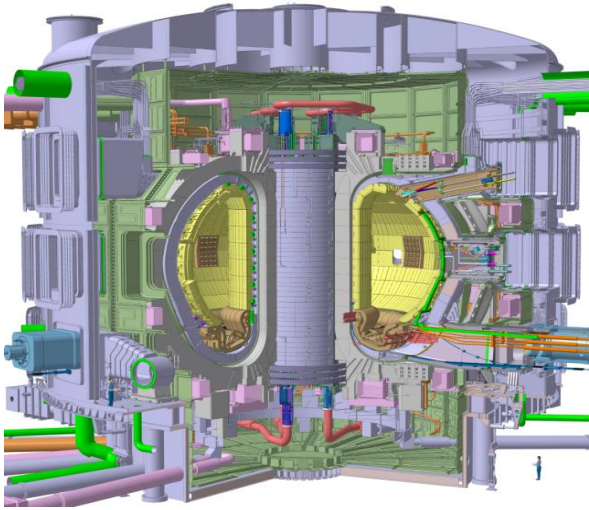
# Application Benchmark Results

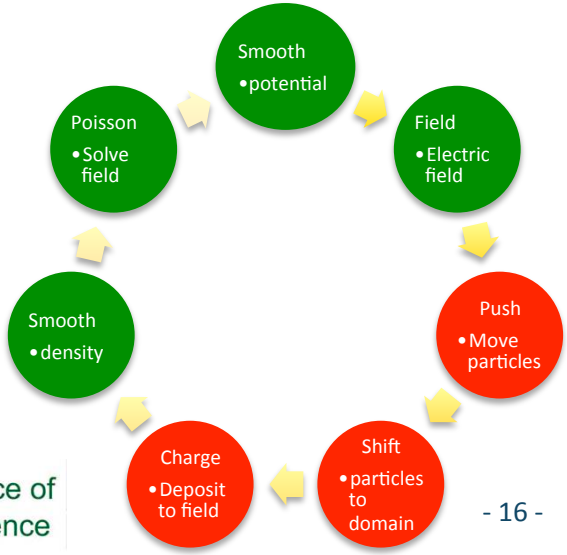
- **Compare performance per node (not runtime).**
  - MPI+HT roughly doubles MPI-Only runtime, but uses half as many nodes.
- **MPI+HT has best performance per node for 6 of 8 codes.**
- **Core specialization decreases performance per node for all but MILC.**
- **Much of MPI+HT improvement is from indirect effects on communication.**
  - (Compare MPI+HT to Hybrid+HT.)

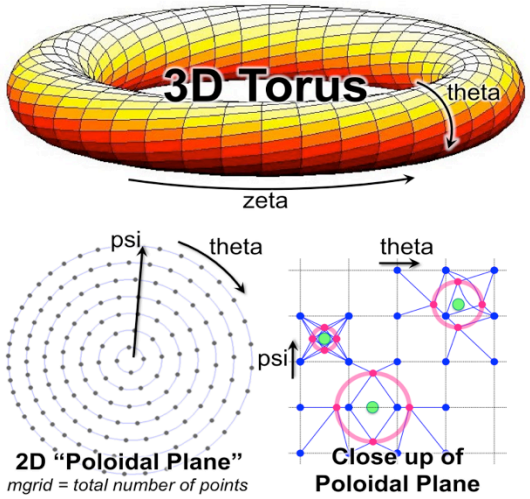


# Gyrokinetic Toroidal Code (GTC)

- **GTC simulates burning plasmas Tokomak fusion reactors**
- **Particle-in-cell algorithm**
- **Two-level MPI parallel decomposition**
  - 64 toroidal domains x 32 particle domains
  - Charge phase emphasizes RAM latency; includes reduction over particle domains.
  - Shift phase is network bandwidth limited.

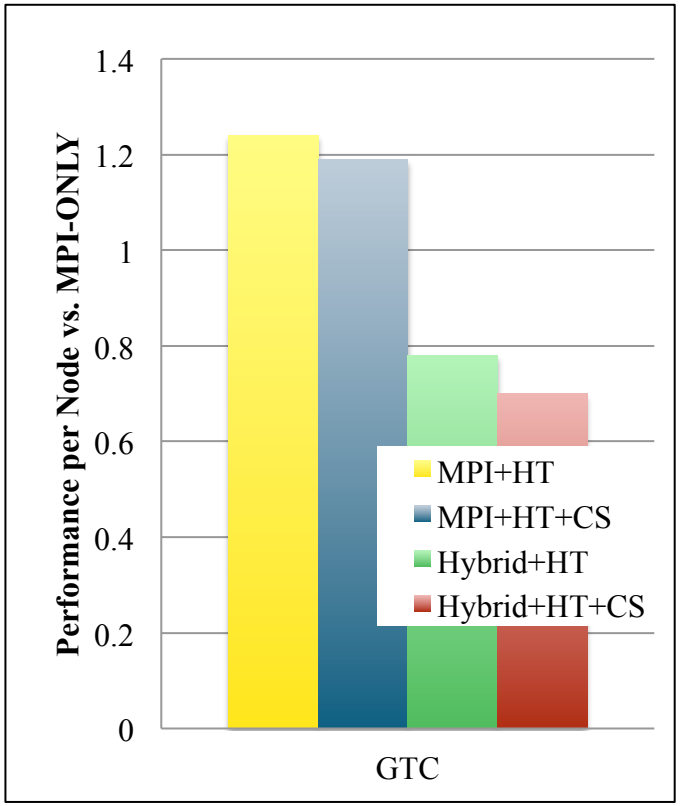




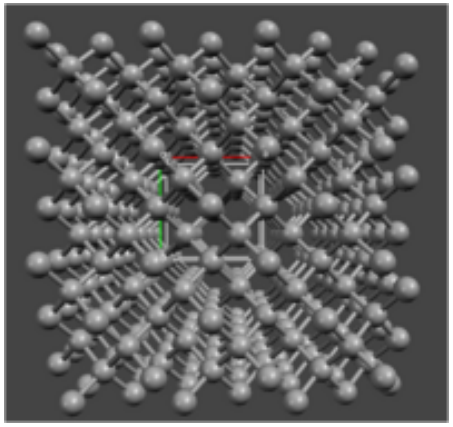



# GTC Performance Analysis

- **MPI+HT has highest performance per node**
  - HT hides memory latency of charge deposition.
  - HT increases communication locality; MPI_Allreduce time decreases when all particle domains fit on one node.
- **CS improves upon MPI+HT runtime, but hurts performance per node.**
- **HT performance does node compensate for thread synchronization.**



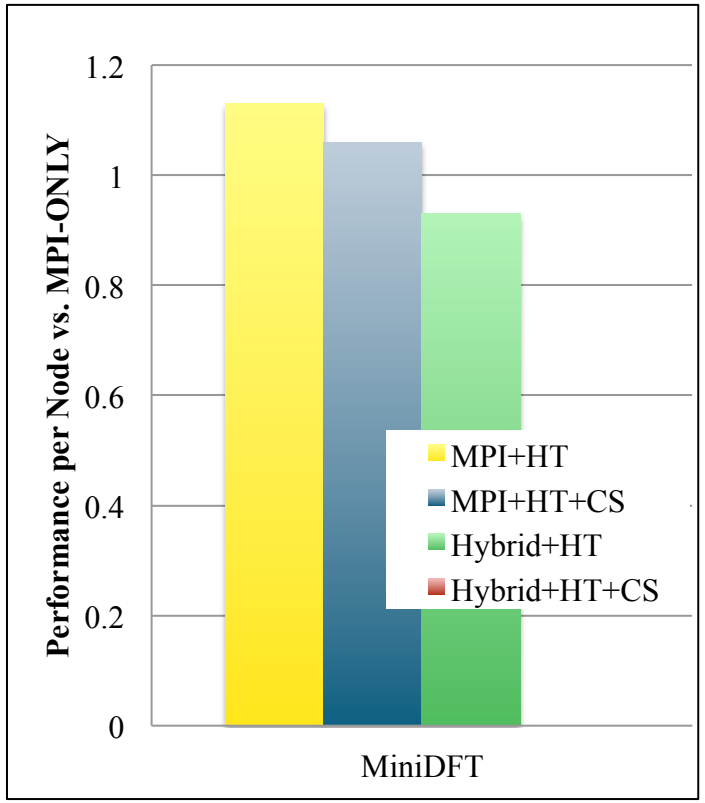
# Mini-DFT



- **Materials modeling using plane-wave density functional theory.**
- **Mini-app based on Quantum Espresso**
- **Runtime dominated by linear algebra (ScaLAPACK) and 3-D FFT**
  - Linear algebra phase is parallel over bands.
  - FFT phase is parallel over bands and g-vectors.
- **Hybrid parallelism via threaded libraries.**

# Mini-DFT Performance Analysis

- **MPI+HT has highest performance per node**
  - Excluding MPI time, HT does not improve performance. ZGEMM and FFT saturate CPU resources with a single stream
  - HT "fixes" load imbalance. Only half of allocated processes are active during ScaLAPACK phase. Inactive processes cede their share of CPU resources.
- **Core specialization slightly degrades performance.**
  - Mini-DFT does not use non-blocking communication.
- **Hybrid code does not benefit from HT**
  - During ScaLAPACK phase, active and inactive processes do not share resources (different physical cores).



# MIMD Lattice Computation (MILC)

- **Lattice quantum chromodynamics**
- **Models strong interaction between quarks and gluons.**
- **4-D (x,y,z,t) computational domain**
- **Conjugate gradient algorithm**
  - Large, sparse matrix vector multiplication
  - Sensitive to memory bandwidth
- **Parallelized using 4-D domain decomposition**
  - Global bandwidth stressed by 4-D halo exchange.
  - Conjugate gradient algorithm uses all-reduce.

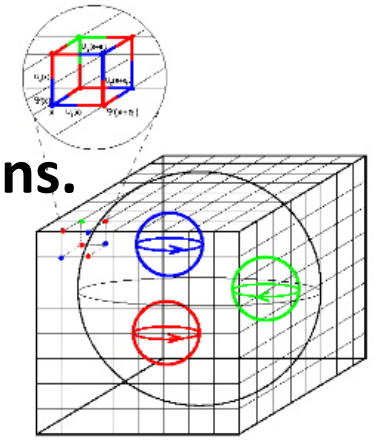
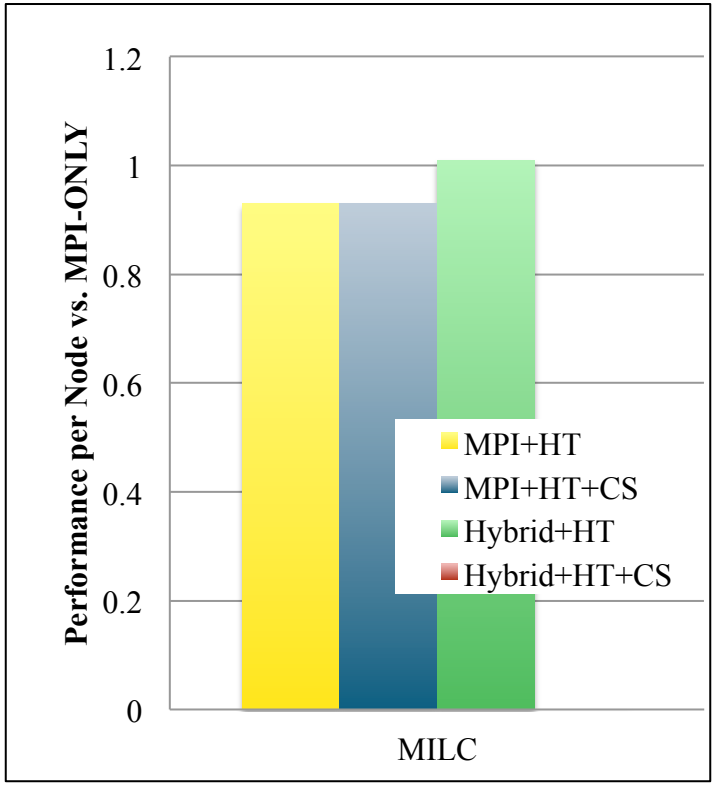


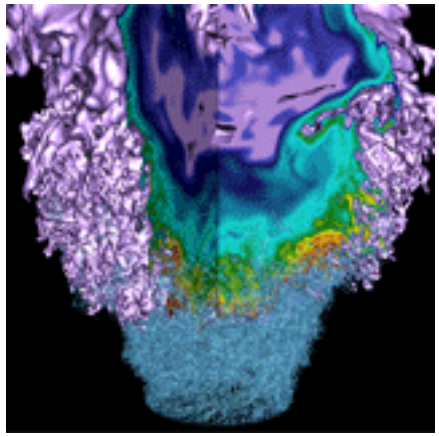
Image: usqcd.org

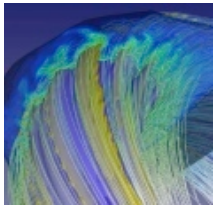
# MILC Performance Analysis

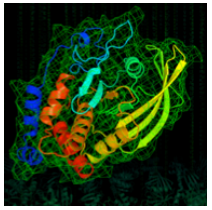
- **MPI-Only and Hybrid+HT have highest performance per node.**
  - Regular computational pattern- single stream saturates CPU resources.
- **MPI+HT performance per node is less than MPI-Only**
  - Contention for memory or cache resources.
  - Hyper-threading does not improve locality of MPI communication
- **Core specialization has no effect on performance per node vs. MPI +HT.**

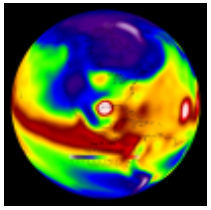


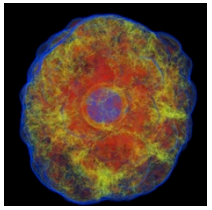
# Cross-Platform Performance Comparison

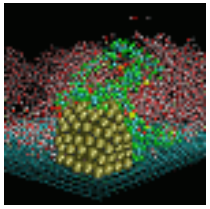






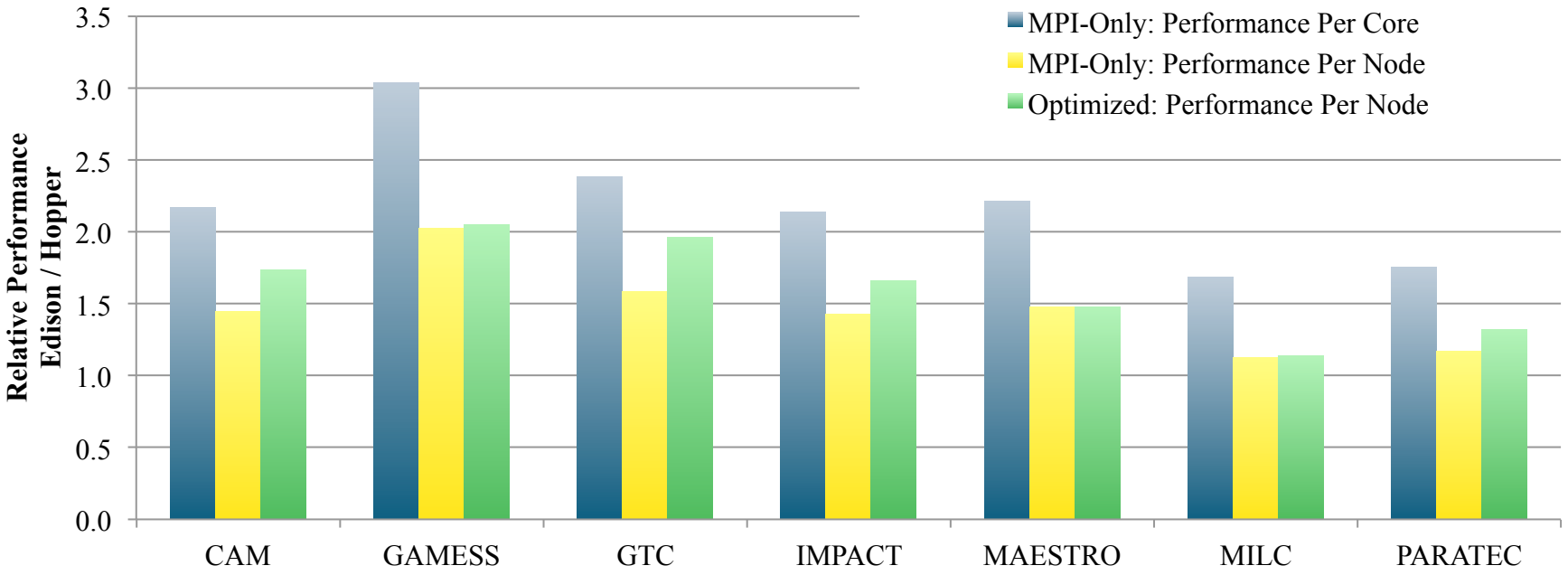






# Edison (XC30) vs. Hopper (XE6)

- **2.2x performance per core (without HT or CS)**
  - Same input runs twice as fast
- **1.4x performance per node (without HT or CS)**
  - Equals 16/24 performance per core; includes rough accounting for power.
- **1.6x throughput per node**
  - Selecting 'best' use of HT and CS



# Conclusion

- **Aries' network latency, bandwidth and scalability significantly improves upon Gemini.**
- **Hyper-threading increases throughput for 6 of 8 application benchmarks.**
- **Core specialization hurts performance for all but one code.**
- **Applications run 2.2x faster on Edison than Hopper**
  - Performance per node is 1.4x greater
  - With hyper-threading, throughput per node is 1.6x greater.


# National Energy Research Scientific Computing Center