From Quarks to the Cosmos: Enabling Scientific Breakthroughs With PSC's Cray XT3

SOS11 June 2007
Key West FL

Nick Nystrom

Pittsburgh Supercomputing Center

PITTSBURGH SUPERCOMPUTING CENTER

PSC:

- In Pittsburgh, PA
- Cooperative exercise of Carnegie Mellon University and the University of Pittsburgh
- Resource Provider for the US-NSF TeraGrid
  - Also: TACC, ORNL, SDSC, NCSA at SOS-11
  - Cover all disciplines supported by NSF
    - Excepting clinical bio-med
- (Recently decommissioned TCS: AlphaServer SC)
- Major machine now a Cray XT3, recently upgraded

General PSC Status:

- PSC's XT3:
  - 1st XT3 (SC'04, running on the floor)
  - At ~22 Tf on dual core processors
- Pending proposals
  - NSF Track 1 and Track 2 pending review
- Current PSC emphasis:
  - Improving system functionality, flexibility and performance (potentially available to other Cray systems!)
  - Ever broadening the already very wide base of scientific application
  - Ever increasing application scalability and performance
- Within the PSC and NSF Teragrid, resource limited.
  - We have the researchers
  - We have the codes
  - We have the scientific problems
  - [Question Rick's scaling anxiety (Matlab → Petascale). Were there more Terascale apps than Petascale apps at the equivalent dates?]

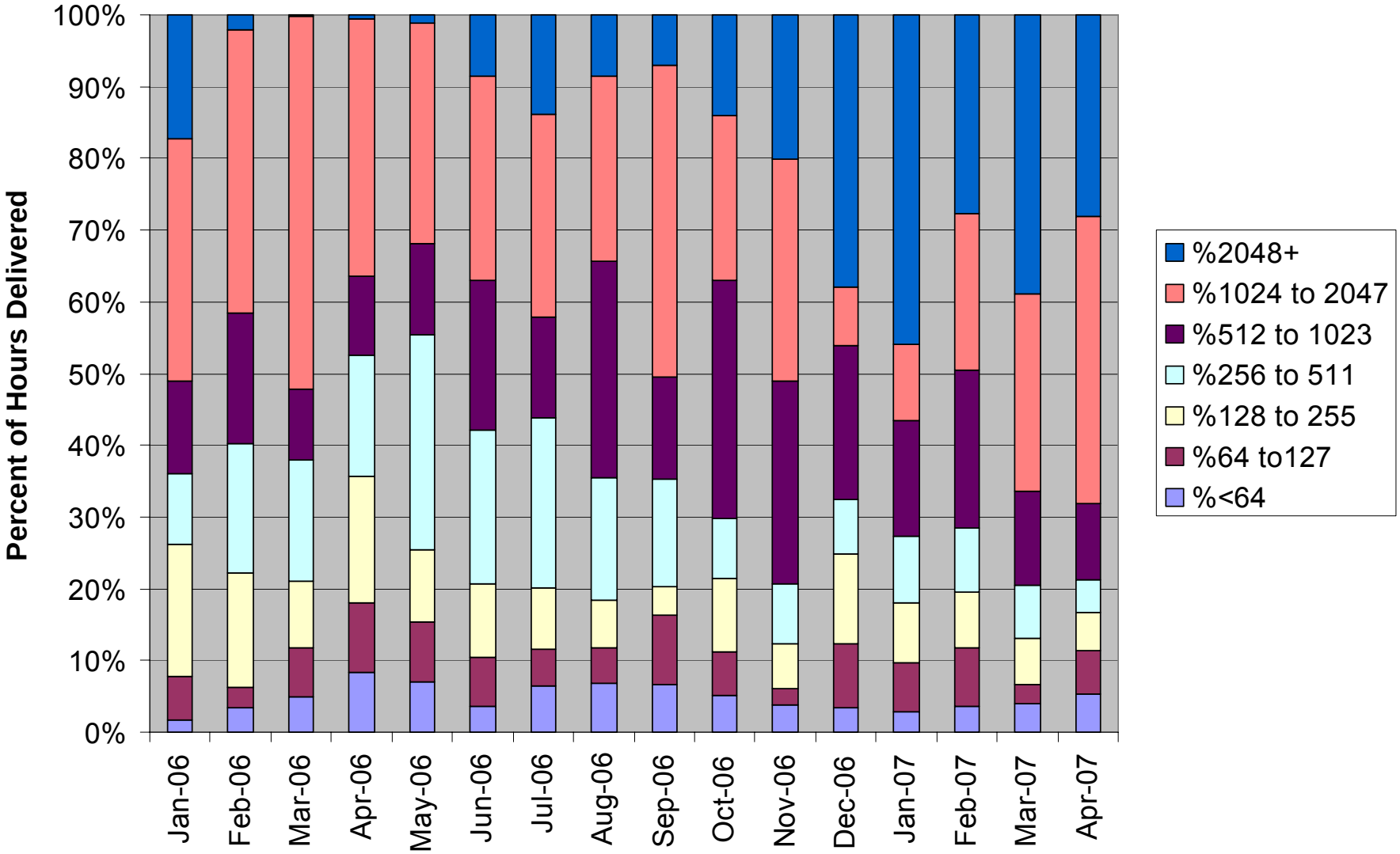
BigBen: PSC’s Cray XT3

PSC's Recent Upgrades to BigBen

- December 2006 — upgrade Opterons from 2.4GHz SC to 2.6GHz DC; double memory to 2GB/node
- February 2007 — install UPS
- March 2007 — install 2 additional "phat" (8GB) nodes (4 total) to support applications where PE 0 requires more memory; *available on demand by virtue of work PSC's Systems and Operations group did to extend the scheduler (NAMD)*
- March 2007 — malleable wall time in scheduling *(min & max times to facilitate draining and filling)*

68% of BigBen Utilization Requires 1024-4114 Cores

Retain emphasis on *wide* jobs

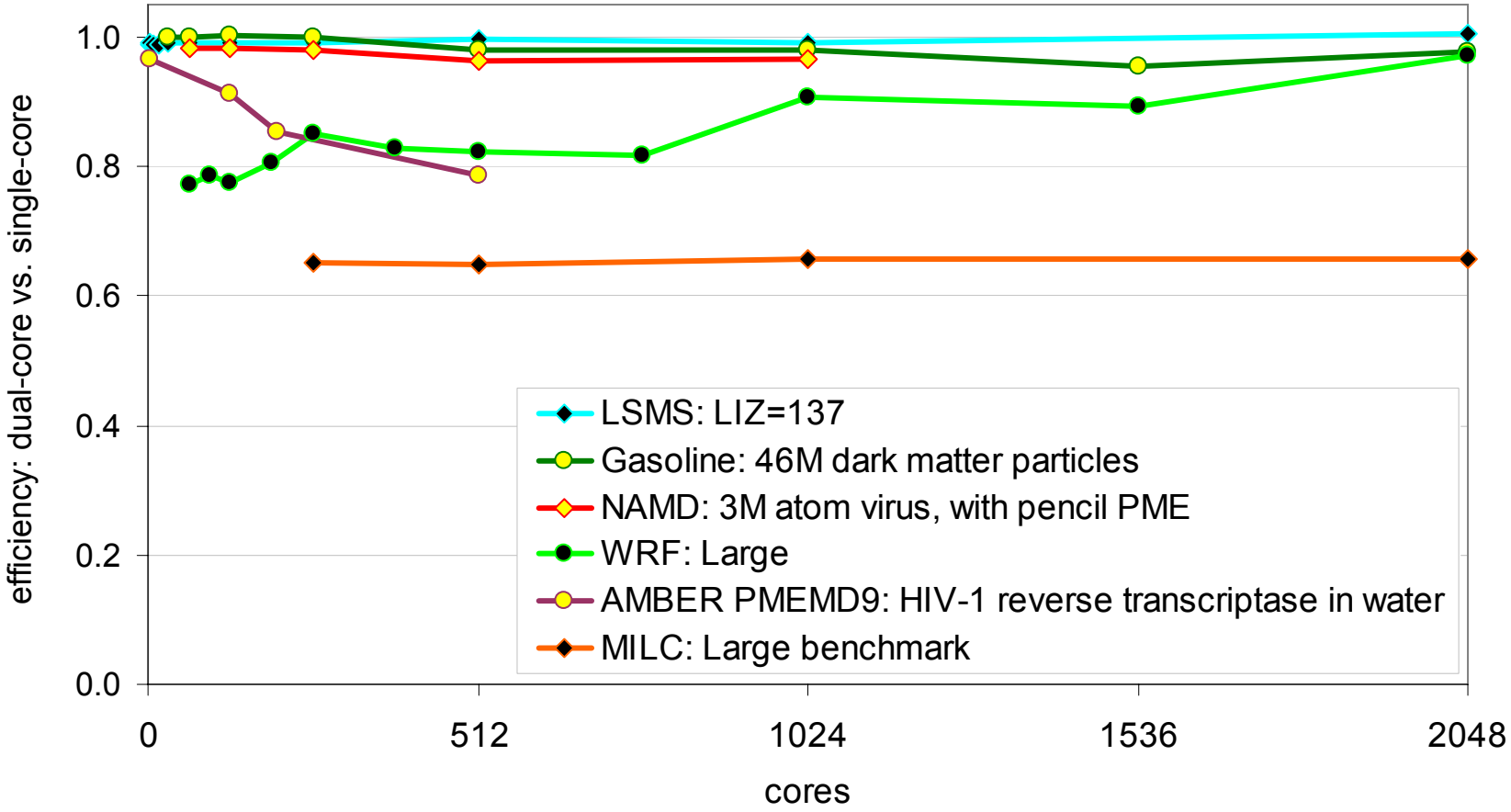


BigBen: Recent Production Accomplishments

- High scheduler throughput
  - 748 job starts/day
- MTBF
  - 1 week: 207 hrs
  - 4 week avg: 129 hrs
  - [Red Storm SOW: 50 hrs @ 10k nodes → 250 hrs @ 2k nodes; good, still a way to go!]
- Utilization
  - Routinely achieving 96-98% daily utilization
  - Monthly average of 92+% utilization

Efficiency of dual- vs. single-core Opterons (Cray XT3)

- December 2006: BigBen upgraded from 2.4 GHz single-core Opterons to 2.6 GHz dual-core Opterons

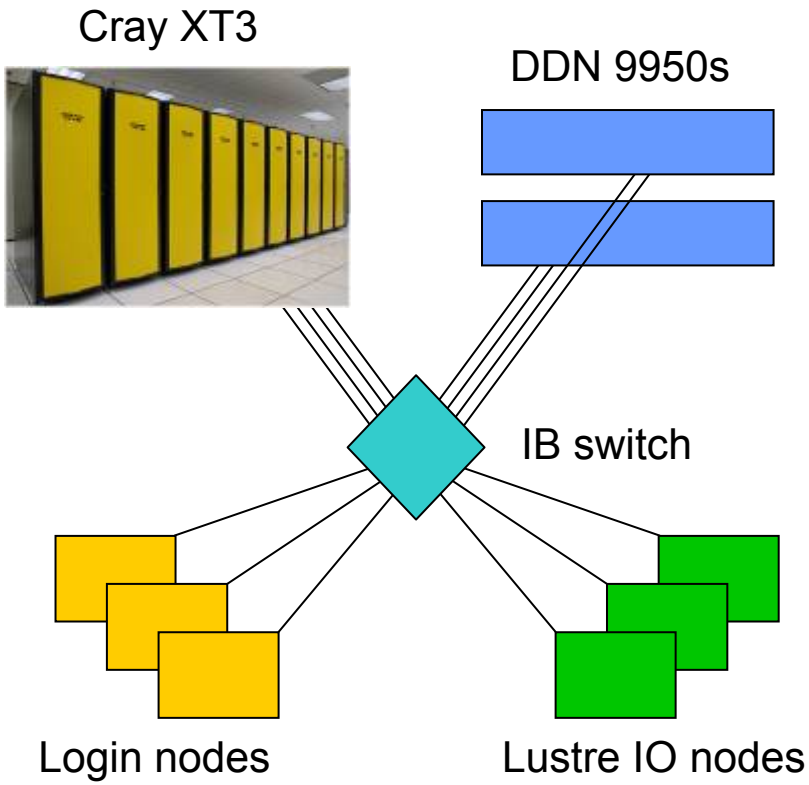


Maximizing Performance

- Getting more Science out of the same Hardware
- In cooperation with Cray in many cases
- Available to Cray for other XT3 systems

Externalizing Cray XT3 I/O

- **SIO nodes as Lustre routers**
- Routing over IB to external white-box Lustre servers
- Using DDN 9550s
- Rebuilt Linux kernel to add IB support
- Rebuilt liblustre from CFS sources to add IB support
- Tested and working on development XT3 cabinet
- **External Login Nodes**
- In service >1yr
- GigE connected
- Superior stability & performance!
- **Increased performance and stability!!**
- **Goal: All *ancillary* services → External!**



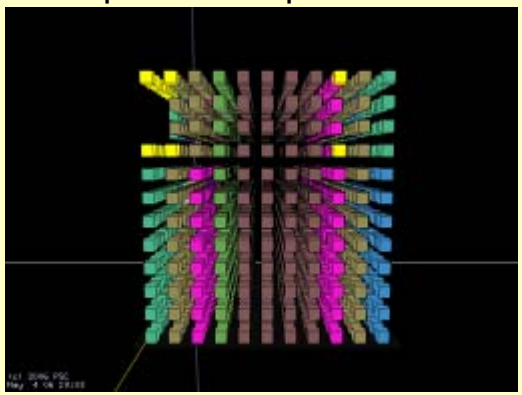
Optimizing Job Placement (1)

- The Cray XT3's SeaStar interconnect, configured at PSC as a 3D torus, offers exceptional bandwidth (SeaStar 1: sustains 2.2 GB/s injection bandwidth, 6.5 GB/s link bandwidth, both expressed as bidirectional).
- However, the default scheduling policy assigns nodes based on numbering nodes consecutively along rows, whereas physical cabling alternates cabinets to avoid inordinately long cables.
- This scheduling policy results in interconnect congestion that undermines otherwise scalable applications.
- Empirical observations indicated that production jobs were being badly fragmented, seriously degrading their performance.
- **PSC's customized job placement algorithm reduces communication contention between concurrently running jobs by maintaining compactness of processor layout and minimizing fragmentation over time.**

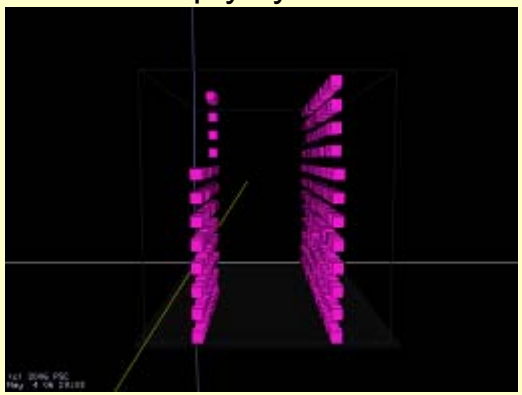
- *Research presented at CUG 2006, Lugano, May 2006. Now in production.*

Optimizing Job Placement (2): Recovering Bandwidth

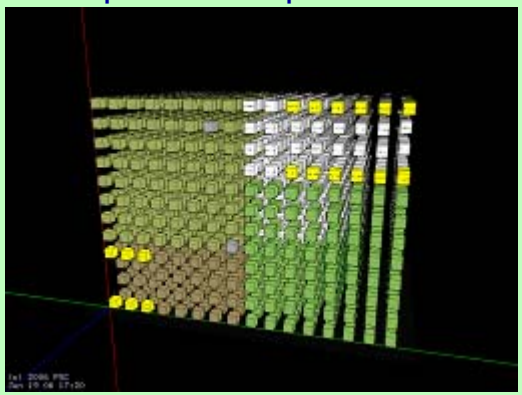
Job connectivity:
XT3 default job
placement protocol



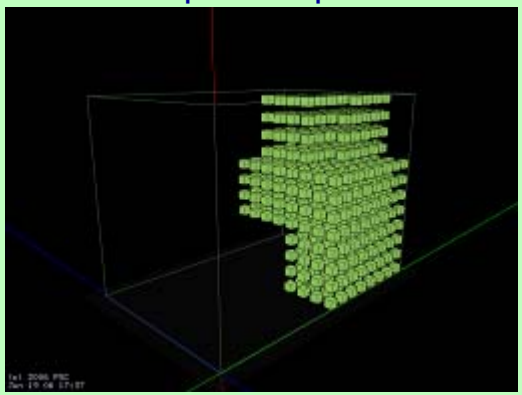
Jobs are fragmented,
even if allocated on an
empty system



Job connectivity:
PSC job
placement protocol



Processors allocated to each
job are now contiguous and
as compact as possible.



Default job placement:

- Fragmentation causes congestion as messages between processors must pass through network paths that other jobs are using.
- Incompactness increases congestion even within each job's set of nodes.

PSC job placement:

- Avoiding fragmentation reduces interconnect contention between jobs.
- Increasing compactness provides more network paths within each job.
- Both factors increase effective bandwidth, improving efficiency of communication-intensive applications.

Optimizing Job Placement (3): Results

- Experiments approximating realistic production conditions were run using DNS and NAMD and using a 1024-node PTRANS job to simulate workload.
- Broad application mix reflects different types of communication-intensive jobs.

Application	p	Placed [1]: default scheduling algorithm	placed [1]: optimized for adjacency	production average [2]	production standard deviation	improvement optimized vs. production
PTRANS	1024	129.3 GB/s	146.5 GB/s	131.2 GB/s	21.2 GB/s	**11.7%**
DNS	512	316.5 s	296.0 s	310.5 s	9.2 s	**4.7%**
DNS	192	198.0 s	163.0 s	181.7 s	23.5 s	**10.3%**
NAMD	512	161.4 s	150.1 s	167.1 s	13.1 s	**9.8%**
NAMD	32	252.7 s	228.3 s	252.0 s	12.2 s	**9.4%**

1. controlled experiment where the job was placed on processors and run with a 1024-processor PTRANS job
2. **average performance over several runs while the machine was in production, with the default scheduling algorithm**

Optimizing Job Placement (4): Status and Directions

Current Status

- Performance of communication-intensive applications such as NAMD and DNS showed improvement of up to ~10% under realistic workloads.
- PSC's processor layout algorithm is now the default in BigBen's scheduler, transparently providing efficiency gains to all applications.
- Analogous connected, compact MPI process mapping onto processors is available via **`pbsyod.new --optimize-order`**.

Ongoing and Future Work

- Support for user-specified processor and process layout, provided as either a requirement or a hint
  - e.g. specifying that an $8 \times 8 \times 8$ processor topology is either required or merely optimal for a given run, rather than, say, $16 \times 16 \times 2$ or some arbitrary layout
- Automatic generation of optimized layouts from communication profiling data

SeaStar Optimization

- Maximizing bandwidth and minimizing latency requires **efficient arbitration** at SeaStar routers along each packet's path through the 3D torus.
- The SeaStar interconnect implements performance counters and configurable parameters to tune performance.
- Meaningful tuning requires a realistic workload.
- *Exceptional collaboration between performance specialist (Weisser, PSC) and hardware architect (Abts, Cray).*
- *Results to be presented at CUG 2007.*

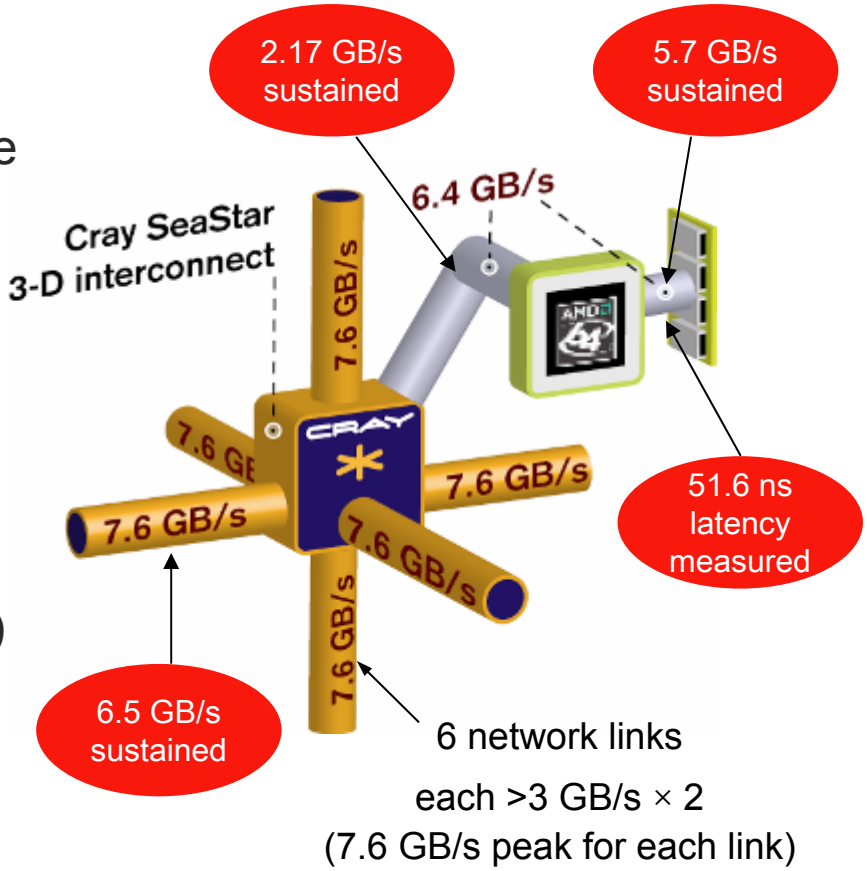


Image courtesy Jeff Brooks, Cray Inc.

Age-Based Packet Arbitration

- Age-based packet arbitration reduces maximum packet latency by routing oldest packets first at output ports.
- Parameters affecting packet routing include:
  - AGE_RR_SELECT: packet routing policy; combination of round-robin and age-based packet routing
  - AGE_CLK_PERIOD: the rate at which a packet's age increases
  - AGE_BIAS: the amount by which a packet's age increases per hop, which can vary by port
- SeaStar performance counters were used in conjunction with synthetic benchmarks to quantify performance and the effects of age-based routing

AGE_RR_SELECT: Packet Routing Policy

- **AGE_RR_SELECT: Packet Routing Policy**
  - Can be a combination of age-based and round-robin
  - Default is 25% age-based, 75% round-robin
- **AGE_CLK_PERIOD**
  - A packet's age is incremented every AGE_CLK_PERIOD cycles on the SeaStar (500MHz → 2ns per cycle).
  - Default is 2^{12} cycles per tick (8.192 μs), which is much too high.
  - Packet age is stored in an 8-bit counter.
  - If tick is too short, too many packets will max out and be indistinguishable.
- **AGE_BIAS**
  - We can set the number of ticks a packet ages per hop to be vary by port, reflecting the different number of expected hops per dimension.
  - Default AGE_BIAS is 1 on all ports.

Quantifying Age-based Routing: Average Stalled Cycles

- Infer average number of cycles a packet is stalled at the head of the input queue by reading SeaStar performance registers on each port
  - RTR_PERF_STALL
  - RTR_PERF_VC0_PKTS
  - RTR_PERF_VC1_PKTS

- Indicates reduction in injection bandwidth and overall throughput

Routing Policy Comparison

Average number of stalled cycles per packet per port

	x+	x-	y+	y-	z+	z-
round robin	42.0	42.1	17.6	18.1	12.8	11.9
age-based tick = 0x0004	31.4	31.6	17.4	17.8	11.7	11.5
age-based tick = 0x0008	24.6	24.7	15.8	15.7	11.6	11.4

Routing Policy Improvements: Results (1)

- Changing the packet arbitration policy to age-based routing and reducing the age clock tick from 2^{12} to 2^{3} reduces average message latency by 1.1μs (14%) and significantly boosts performance of benchmarks that reflect important applications:

	PTRANS	MPIFFT
Round robin	467.448 GB/s	451.327 Gf/s
Age-based	488.882 GB/s (4.6% improvement)	507.449 Gf/s (12% improvement)

Routing Policy Improvements: Results (2)

- Changing the packet arbitration policy to age-based routing and reducing the age clock tick from 2^{12} to 2^{3} reduces average message latency by 1.1μs (14%) and significantly boosts performance of benchmarks that reflect important applications:

	Allreduce	**Alltoall**
	32,768 bytes on 2048 single-core processors	
Default	607.46 μs	142.31 MB/s
Age-based	532.04 μs (12.4% improvement)	193.91 MB/s (36.3% improvement)

Virtual Channels

- In a 3D torus, 2 virtual channels are required to define a dateline in order to avoid deadlock.
- **Virtual channels can also be used to distribute bandwidth more evenly between packets.** Overall bandwidth per link remains the same, but the variance between maximum and minimum packet latency is reduced.
- SeaStar hardware supports up to 4 virtual channels, but the firmware only used 2.
- Results of new firmware using 4 virtual channels, balancing the load across channels, is compelling:
  - ***PTRANS improves by 10-25%***
  - ***MPIFFT improves by 25%***
  - ***RandomRing bandwidth improves by >40%***

Null-job System Integrity Checking.

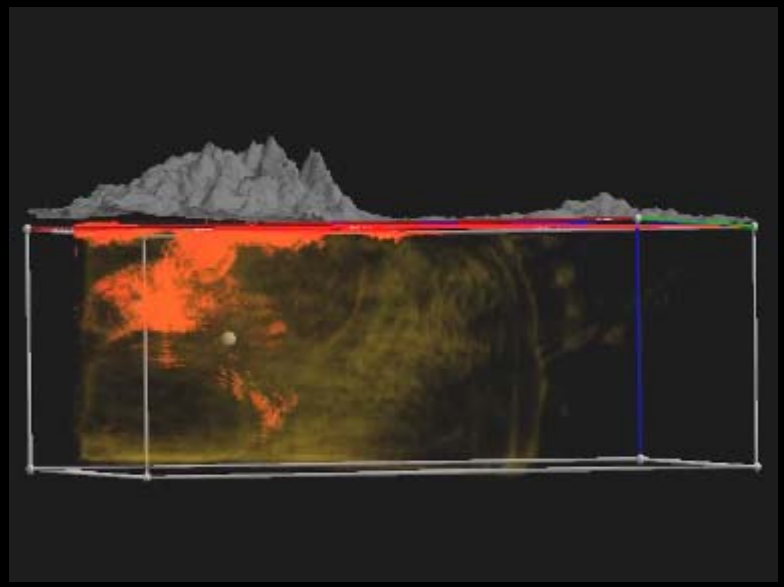
- Cray used to produce and test all system components
  - Even then, PSC routinely ran *Dave Slowinski's Prime-Finder* as a null-job on all idle processors to check system integrity.
  - It DID find at least 1, very serious system error able to give bad answers on any job.
- Cray now takes its processors from a *commodity supplier*, untested!
  - At least one other vendor has had system problems (not clear whom to blame)
- **PSC has reinstituted null-job system testing**
  - ~10% of wall-clock time available for this purpose
  - Nothing found, yet.
  - **We suggest that HPC vendors make this a system feature!**

Scientific Impact

(with emphasis on performance enhancements)

(For broad summary & specific details: www.psc.edu)

The Quake Project (CMU, UT Austin, UC Davis)

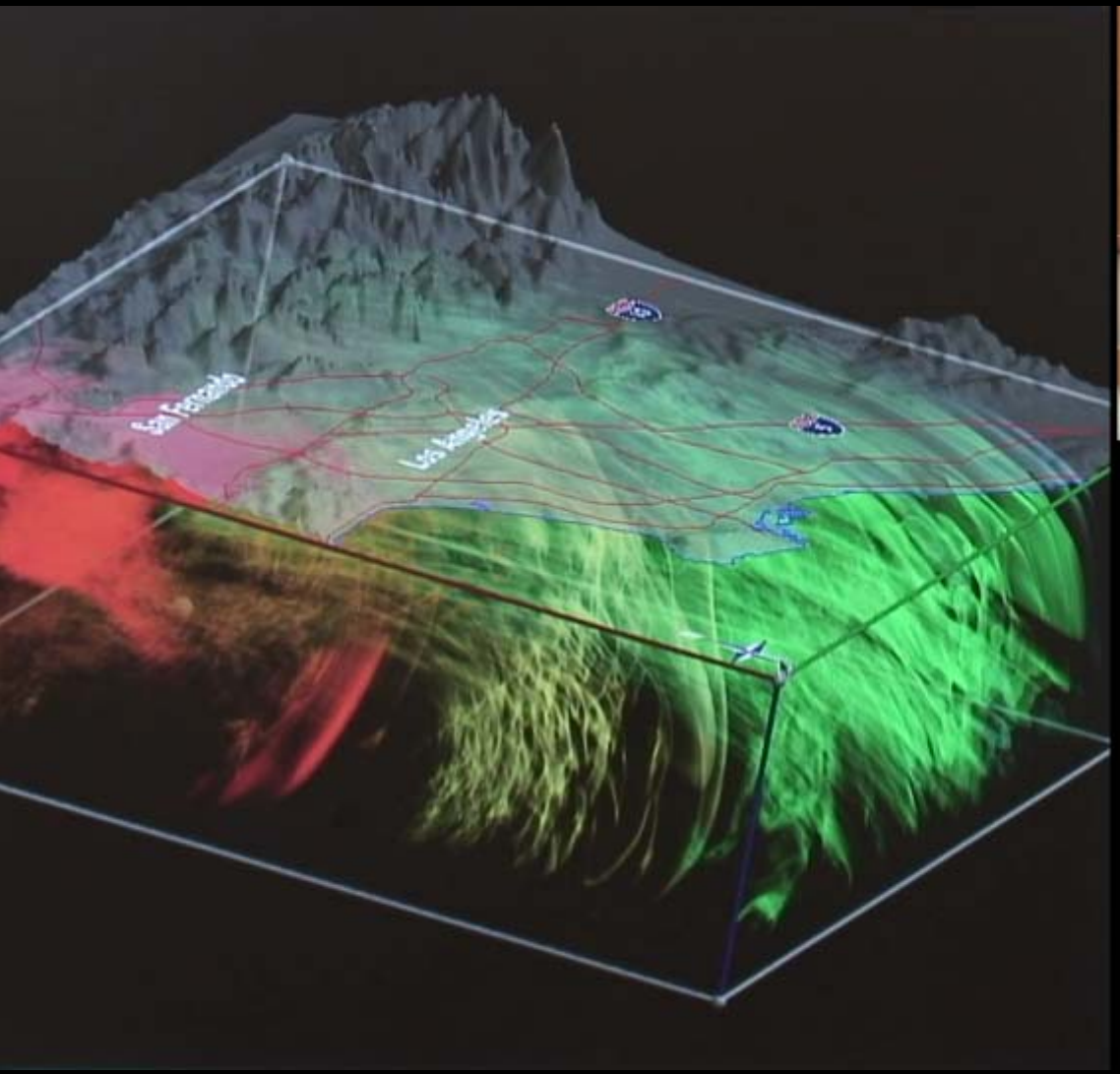


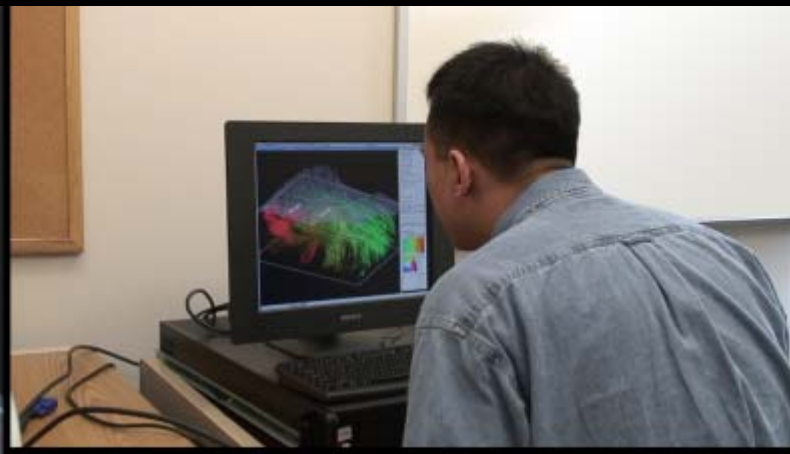
- Goal: Assess seismic hazard to large populated basins by simulating the ground motion generated by postulated future earthquakes by modeling seismic wave propagation from fault rupture to local site effects in sedimentary basins
- **[PDIO, Steering]**

Hercules: End-to-End Earthquake Simulation

J. Bielak, D. O'Hallaron, O. Ghattas, K.-L. Ma

SC|06 Analytics Challenge Winner
SC|06 Best Student Paper Finalist



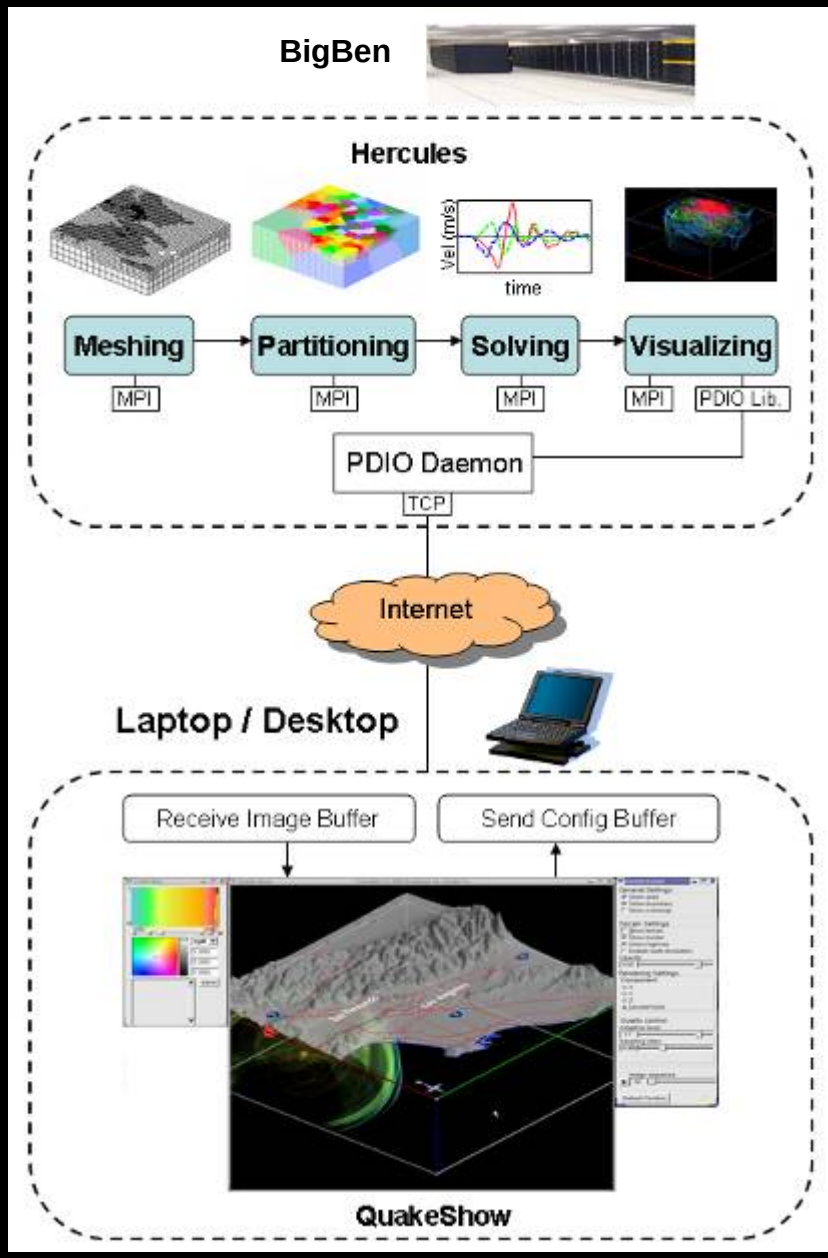


Real-time visualization and steering of earthquake simulations allows scientists to obtain high-resolution solutions that otherwise would not be possible.

Above and left, Tiankai Tu (CMU) manipulates the Terashake earthquake simulation as it runs on 2050 processors of PSC's Cray XT3.

Hercules Sustains >1TFlop/s Interactively on the XT3

- **HPC Analytics Challenge**
  - Hercules runs on BigBen (2050 nodes)
  - PDIO daemon runs on a TeraGrid ftp server
  - QuakeShow runs on a laptop with a 1.7G Hz Pentium M processor
- **Batch mode**
  - 570M node simulation on 2050 nodes
  - ***1.73 teraflops (17.6% of peak)***
- **Steering mode**
  - 11M node simulation on 2050 nodes
  - ***1.01 teraflops (10.3% of peak)***

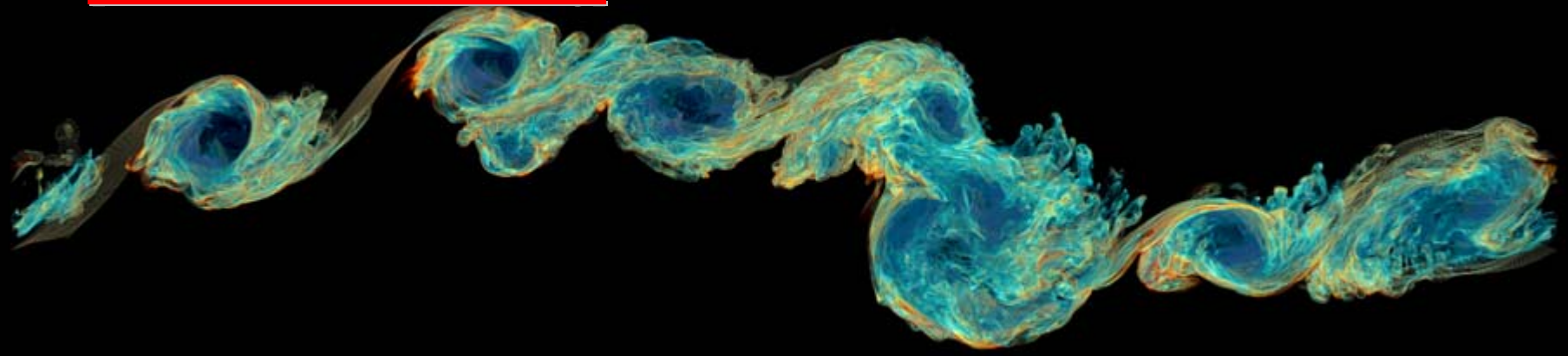


[Aside on PDIO]

- PSC developed software modules for both compute and SIO nodes
- Based on standard Liblustre modules.
- Support R/W to *external* file systems
  - Anywhere on Internet
  - No application recoding. Selected based on pathname.
  - Originally for Paul Woodward's project (PPM, below)
- In many applications, higher performance than native Lustre support.
  - Unlike standard Lustre support, PDIO provides for combining and reordering individual IO requests to vastly improve efficiency (8-9x, best case to date)

Strong Scaling of Multifluid PPM: Sustaining 44% of Theoretical Peak Efficiency

- Using 4096 compute cores + 8 I/O nodes, Paul Woodward and David Porter observed sustained performance of 2.32 Gflop/s per compute core, totaling 9.5 TFlop/s.
- **[PDIO, ifc, icc, steering]**



- *"The run on the 4104 cores and the 576**3 grid was 90% efficient. And that's pretty damned amazing. It is a testimonial to the effectiveness of your Cray XT3 interconnect and the SeaStar chips offloading so much of the communications processing."*
  – Paul Woodward, 1/7/2007

From Undergrad to Scientist

- Paul Woodward suspected strongly that PPM would run faster with Intel's compilers. This was a problem because Intel's compilers were reported to not be usable on the XT3.
- Jordan Soyke, an undergraduate who through working at PSC part-time has mastered low-level programming for the XT3, developed procedures for **running Intel Fortran and Intel C on BigBen.**

+ 20%

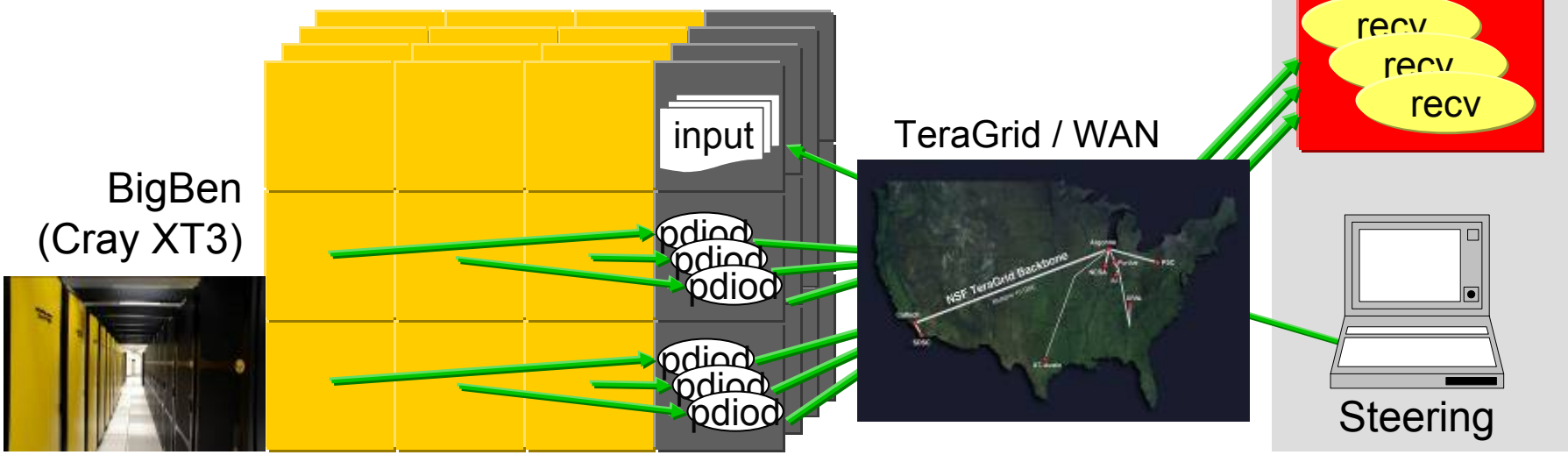
cubebench performance (MFlop/s per core)			
		bigben 2.4 GHz	bigben 2.6 GHz
PGI 6.1.1	-fastsse	1226	**1332**
IFC 9.1	-fast -funroll-loops	1470	**1590**
IFC 9.1+ IVDEP	-fast -funroll-loops	2780	3032

- Gary Glatzmaier and other researchers are now trying ifc and icc on BigBen.
- Jordan is now developing performance tools to help with optimizing applications' use of memory bandwidth.

Steering CFD: Compressible Turbulent Astrophysical Flows

Hercules & NekTar are also now using PDIO

- Interactive steering and visualization of parallel PPM runs: Paul Woodward (U. Minnesota)
- PDIO: Portals-to-TCP routing. PDIO daemons aggregate hundreds of portals data streams into a configurable number of outgoing TCP streams. Heterogeneous Portals: both QK + Linux nodes.
- Utilizing the ETF network: 10GigE end-to-end. Benchmarked >1Gbps.



Unprecedented Storm Forecast Experiments

- 2007 NOAA and University of Oklahoma Hazardous Weather Testbed (HWT) Spring Experiment **[*reservations, real-time* forecasting, improve performance]**
  - **Major goal: assess how well ensemble forecasting works to predict thunderstorms, including the supercells that spawn tornados.**
  - It is the first time ensemble forecasts are being carried out at the spatial resolution at which storms occur
  - It is also the first time ensemble forecasts are being carried out in real time in an operational forecast environment.
  - Requires >100× more computing daily than the most sophisticated National Weather Service operational forecasts. To meet this need, PSC's Cray XT3 (2,068 2.6 GHz dual-core processors, 21 teraflops peak) is the most powerful "tightly-coupled" system (designed to optimize inter-processor communication) available via the TeraGrid.
  - *"This is unique - both in terms of the forecast methodology and the enormous amount of computing. The technological logistics to make this happen are nothing short of amazing."* –Steven Weiss, Science and Operations Officer of the NOAA Storm Prediction Center (SPC)
- Collaborators: NOAA National Severe Storms Laboratory, Center for Analysis and Prediction of Storms (CAPS), LEAD (Linked Environments for Atmospheric Discovery), SPC, PSC

http://www.psc.edu/publicinfo/news/2007/2007-05-10-storm.php
http://www.caps.ou.edu/wx/spc/

Enabling CAPS Daily Forecasts at PSC

- PSC optimized WRF for the Cray XT3, gaining a 3× speedup in I/O, substantially improving overall performance. PSC also optimized the I/O for post-processing routines used to visualize and analyze the forecast output, achieving 100× speedup.
- PSC automated the daily runs and coordinated a dedicated high-bandwidth link.
- PSC modified the reservation and job-processing logic of its job-scheduling software to implement auto-scheduling of WRF runs and related post-processing, 760 separate jobs each day, demonstrating the TeraGrid's ability to use the Cray XT3, a very large capability resource, on a scheduled, real-time basis.
- PSC networking staff coordinated with OneNet, a regional network of the State of Oklahoma, and National Lambda Rail (NLR), a network initiative of U.S. universities, and with Cisco Systems, who contributed use of a dedicated lambda for up to a 12-month period.
- PSC implemented the lambda at its end in January, using existing equipment in the Pittsburgh metro and local-area network. The backbone is provided by NLR and OneNet provides the link from Tulsa to Norman, Oklahoma.
- This dedicated link from the Cray XT3 to OneNet in Tulsa to a supercomputer at the University of Oklahoma (which ingests and post-processes the data) makes possible the transfer of 2.6 TB of data per forecast day.

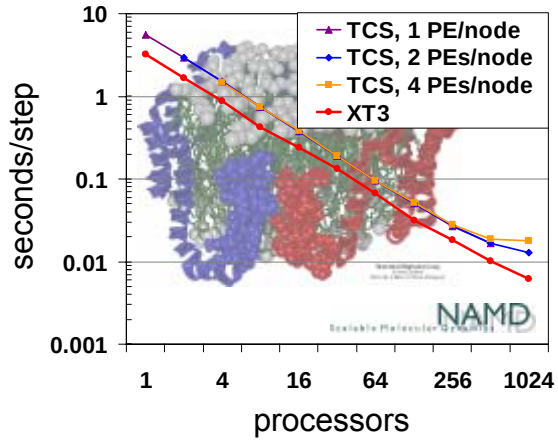
Storm Forecast Methodology

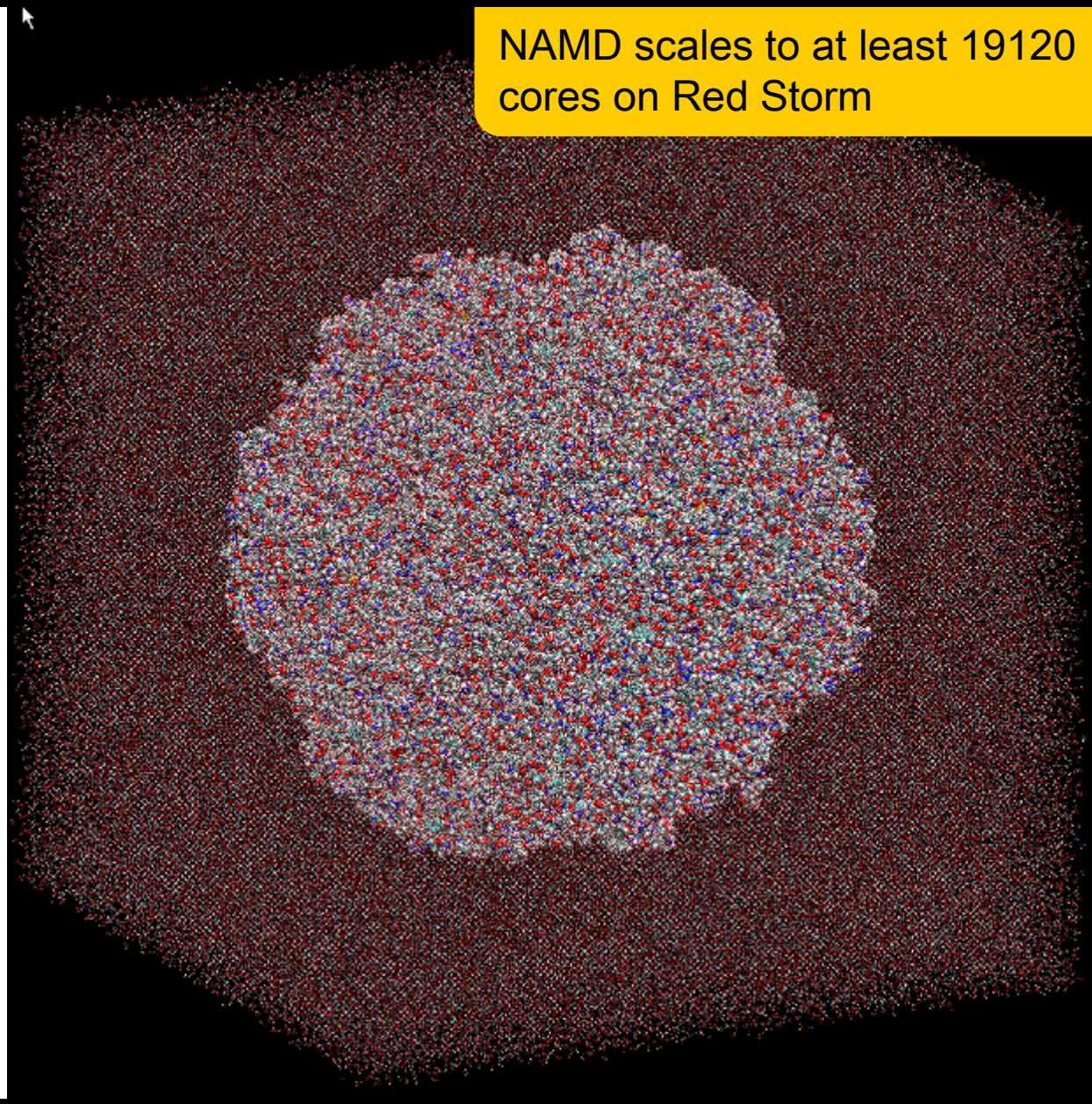
- Each night, from April 15 until June 1, CAPS transmits weather data to the Cray XT3, which runs a 10-member ensemble (10 runs of the model) in addition to a single higher-resolution WRF run, in time to produce a forecast for the next day by morning.
- The forecast domain extends from the Rockies to the east coast: ⅔ of the continental United States.
- Horizontal resolution: 4km for ensemble runs, 2km for single WRF forecasts. A scientific objective is to assess the value of ensemble forecasts in relation to the higher-resolution forecast.
- The Cray XT3 and the high-bandwidth link to Oklahoma make it possible to do both of these demanding runs daily under real-time constraints.
- ***"This experiment represents an enormous leap forward. Ensembles open up a new array of interpretative capabilities to forecasters analyzing how good the forecast is. With ensembles, you're not only forecasting the weather, you're forecasting the accuracy of the forecast."*** —Kelvin Droegemeier, LEAD Director, University of Oklahoma.

Scalable Million-Atom Molecular Dynamics with NAMD

- NAMD: Scalable simulation of large biomolecular systems; 2002 Gordon Bell performance prize
- 1,066,628 atoms (167,063 protein atoms + 299,855 waters)
- Each timestep at 19120 cores takes only 9 ms. *Requires high-bandwidth interconnect and jitter-free OS (i.e. Catamount)*

NAMD 2.6b1 ApoA1 (PME): 92,224 atoms





Enabling Breakthrough NAMD Simulations at PSC

- **Problem:** In NAMD2, PE0 holds a persistent PDB data structure and a map of the distributed data structures and the Cartesian coordinates and atom types for entire molecular system.
  - Its memory requirement increases with problem size, prohibiting certain simulations of interest on a system with only 1GB of memory per node.
- **Solution:** 2 nodes on bigben are now configured with 4GB of memory and accessed through the *phat* queue.
  - The *phat* queue is being used for production calculations by the Schulten and Voth groups to study dynamics of ribosomes and N-BAR domains, respectively.
  - ***3M atom ribosome calculations sustain >2.08 Tflop/s on 2048 processors.***
- (**The real solution:** Distribute the non-scaling data structures. A pencil decomposition of the Particle-Mesh Ewald (PME) procedure is already in place, and additional redesign is underway.)