BLUE WATERS

SUSTAINED PETASCALE COMPUTING

Blue Waters and the Future of @Scale Computing and Analysis

Dr. William Kramer

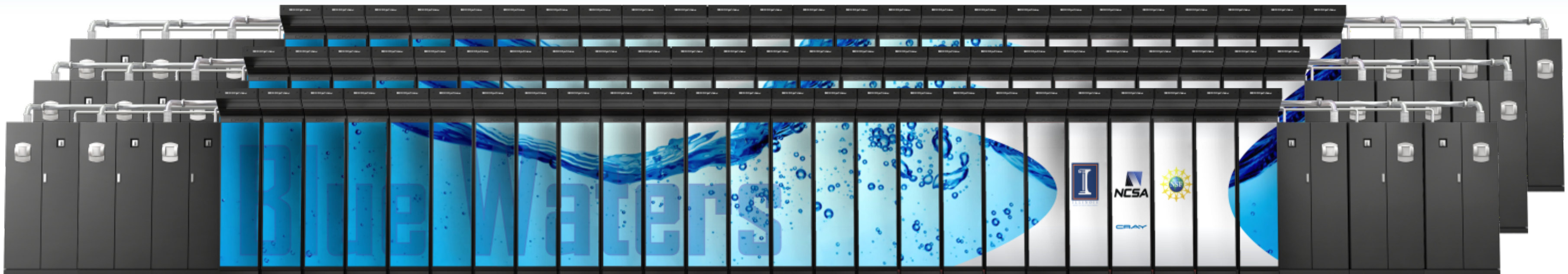
National Center for Supercomputing Applications, University of Illinois

NCSA NSF GREAT LAKES CONSORTIUM FOR PETASCALE COMPUTATION CRAY

Outline

- Blue Waters Update
  - System
  - Applications and Usage
- Observations and Challenges for @Scale Computing
- System Wide Performance Assessment – SSP/SPP
- Comments on the Top500 List and its future

Blue Waters Computing System



Aggregate Memory – 1.66 PB

IB Switch

>1.1 TB/sec

10/40/100 Gb Ethernet Switch

External Servers

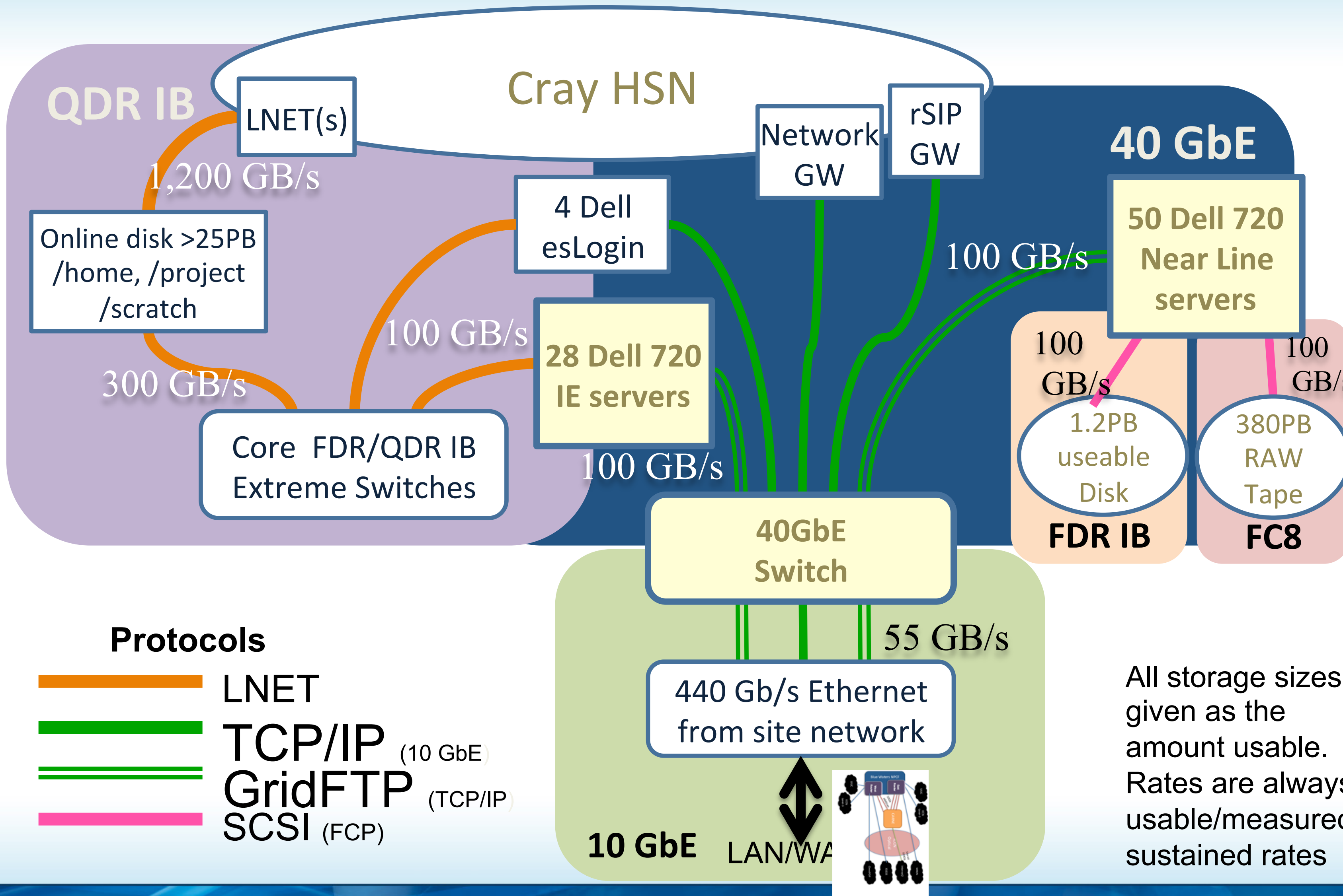
100 GB/sec

120+ - 300 Gb/sec

100-300 Gbps WAN

Spectra Logic: 300 usable PB

Sonexion: 26 usable PB



All storage sizes given as the amount usable. Rates are always usable/measured sustained rates

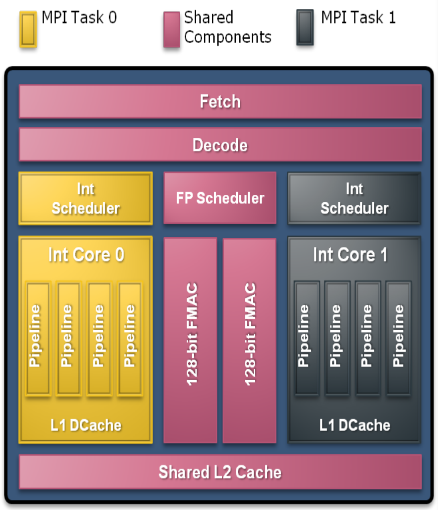
CORE, CPUS, SOCKETS, PROCESSORS, ETC. DO NOT HAVE CLEAR DEFINITIONS

What unit is the smallest schedulable unit?

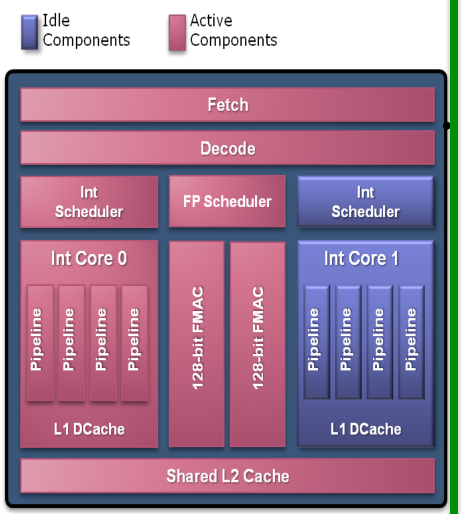
Core - Core Module

- Interlagos is composed of a **Bulldozer** core "modules"
  - Shared and dedicated components
- There are two independent integer units and a *shared*, 256-bit FP resource
- This architecture is very flexible, and can be applied effectively to a variety of workloads and problems
- No one uses the AMD definitions

Shared-core mode
4 FLOPS/Clock per processs



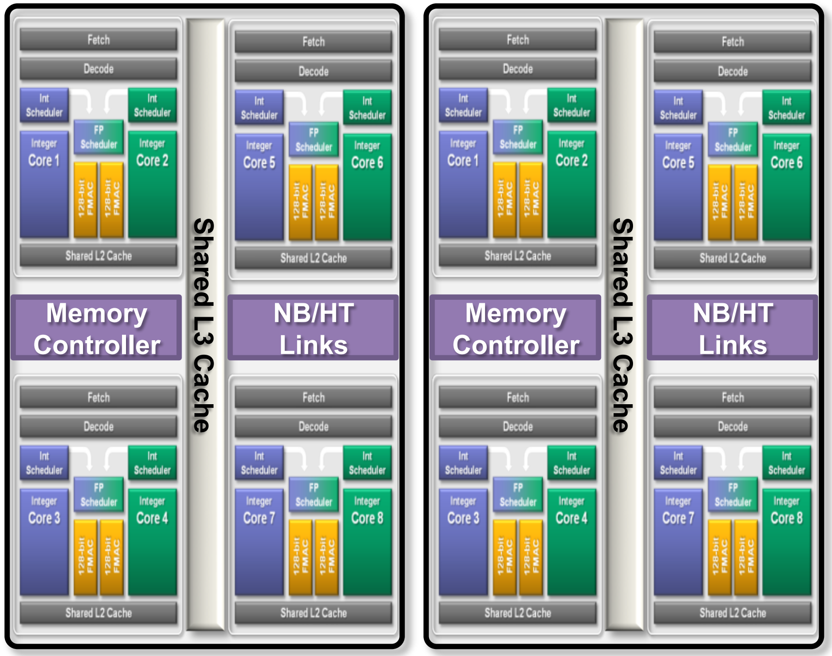
Single-core mode
8 FLOPS/Clock per process



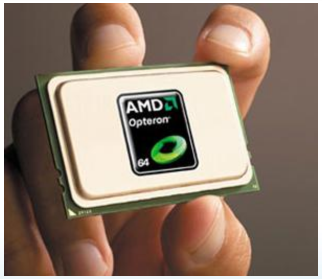
- Single Core Mode, only one integer scheduler unit is used
  - Most common mode for S&E applications
- Implications
  - This core has *exclusive* access to the 256-bit FP unit and is capable of 8 FP results per clock cycle
  - The core has twice the memory capacity
  - The core has twice the memory bandwidth
  - The L2 cache is effectively twice as large
  - The peak performance of the chip is not reduced
- AMD refers to this as a "Core Module"

Interlagos Processor vs Processor Modules

- Each processor die is composed of 4 core modules
- The 4 core modules share a memory controller and 8 MB L3 data cache on one die
- Two die are packaged on a multi-chip module to form a G34-socket Interlagos processor
- Package contains
  - 8 core modules
  - 16 MB L3 Cache
  - 4 DDR3 1600 memory channels



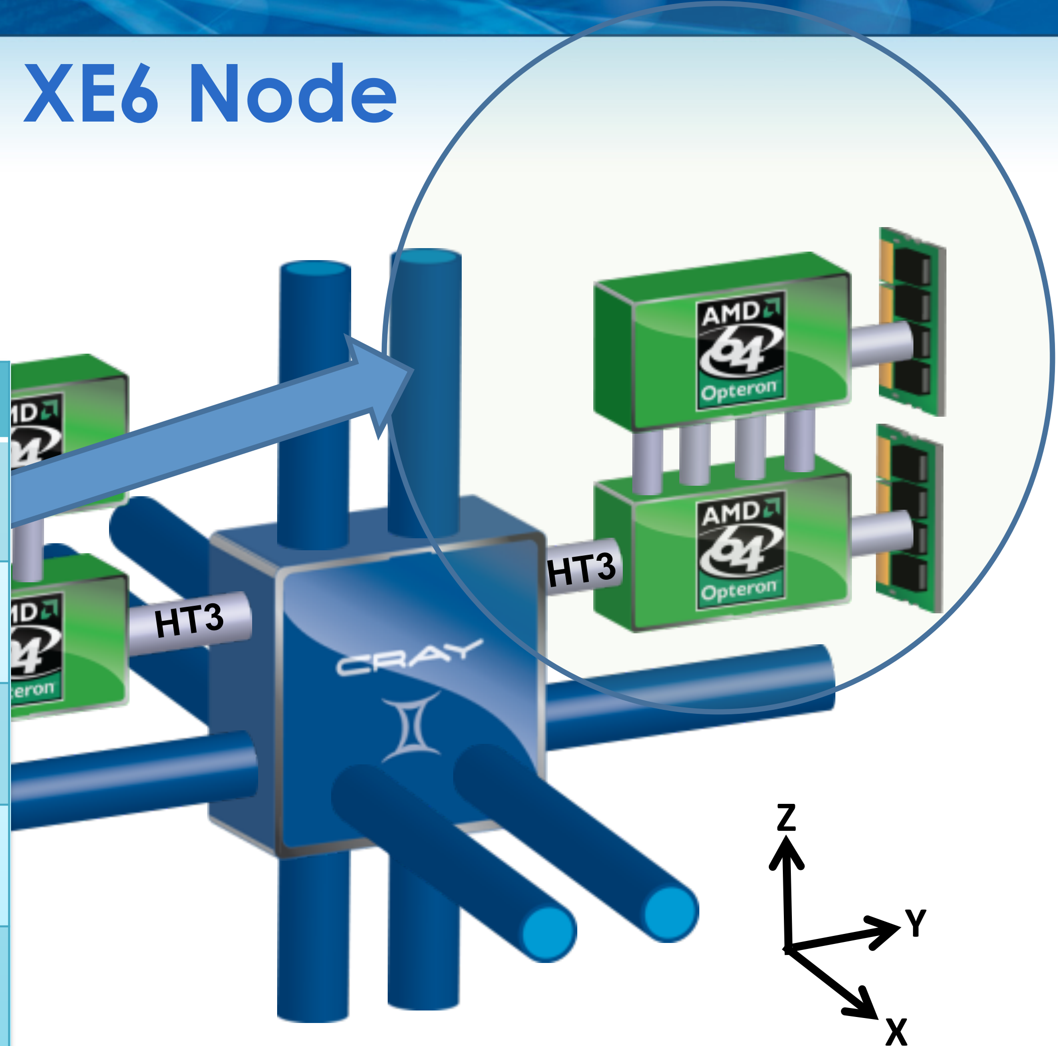
1 Socket



Blue Waters XE6 Node

Blue Waters contains 22,640 XE6 compute nodes

Node Characteristics	
Number of Core Modules*	16
Peak Performance	313 Gflops/sec
Memory Size	64 GB per node
Memory Bandwidth (Peak)	102 GB/sec
Interconnect Injection Bandwidth (Peak)	9.6 GB/sec per direction

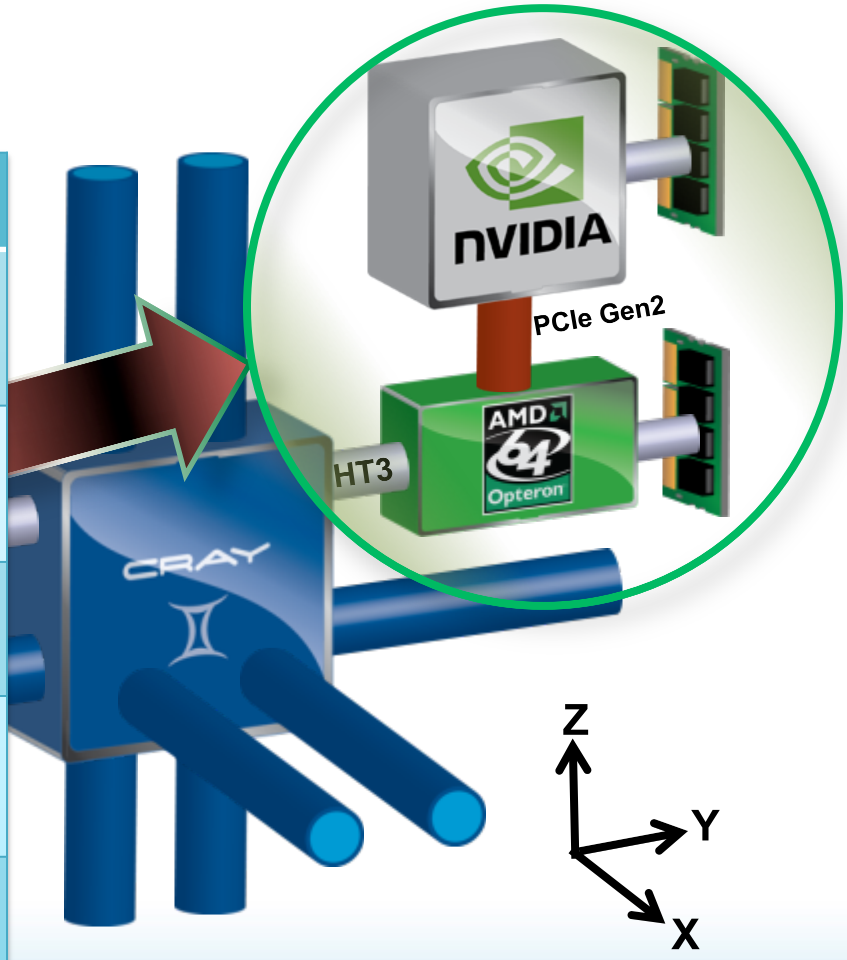


*Each core module (aka Bulldozer) includes 1 256-bit wide FP unit and 2 integer units. This is often advertised as 2 "integer" cores, leading to a 32 core node.

Cray XK7 and a Path to the Future

Blue Waters contains 4,224 NVIDIA Kepler (GK110 K20X) GPUs

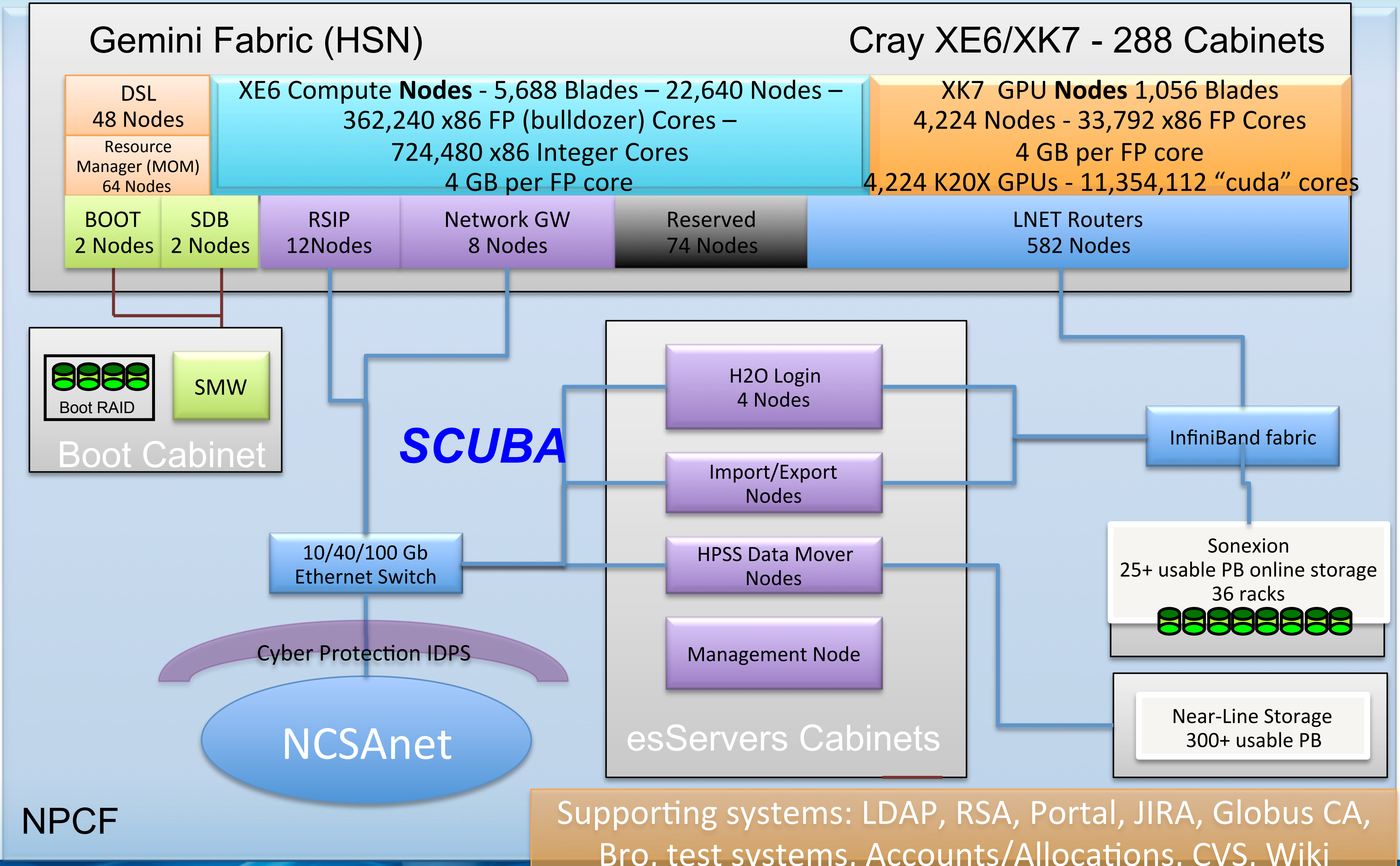
XK7 Compute Node Characteristics	
Host Processor	AMD Series 6200 (Interlagos)
Host Processor Performance	156.8 Gflops
Kepler Peak (DP floating point)	1.311 Tflops (DP)
Host Memory	32GB 51 GB/sec
Kepler Memory	6GB GDDR5 capacity 205 GB/sec



CONCLUSION – WE SCHEDULE NODES SO WE USE NODES FOR COMPARISON

Node = Symetric or NUMA Coherent Unit

BLUE WATERS SUB-SYSTEMS


Gemini Fabric (HSN)
Cray XE6/XK7 - 288 Cabinets
DSL
48 Nodes
Resource Manager (MOM)
64 Nodes
XE6 Compute **Nodes** - 5,688 Blades – 22,640 Nodes – 362,240 x86 FP (bulldozer) Cores – 724,480 x86 Integer Cores
4 GB per FP core
XK7 GPU **Nodes** 1,056 Blades
4,224 Nodes - 33,792 x86 FP Cores
4 GB per FP core
4,224 K20X GPUs - 11,354,112 “cuda” cores
BOOT
2 Nodes
SDB
2 Nodes
RSIP
12Nodes
Network GW
8 Nodes
Reserved
74 Nodes
LNET Routers
582 Nodes
Boot RAID
SMW
Boot Cabinet
SCUBA
H2O Login
4 Nodes
Import/Export
Nodes
HPSS Data Mover
Nodes
Management Node
esServers Cabinets
InfiniBand fabric
Sonexion
25+ usable PB online storage
36 racks
10/40/100 Gb
Ethernet Switch
Cyber Protection IDPS
NCSAnet
Near-Line Storage
300+ usable PB
NPCF
Supporting systems: LDAP, RSA, Portal, JIRA, Globus CA, Bro, test systems, Accounts/Allocations, CVS, Wiki

On-line Storage Subsystem

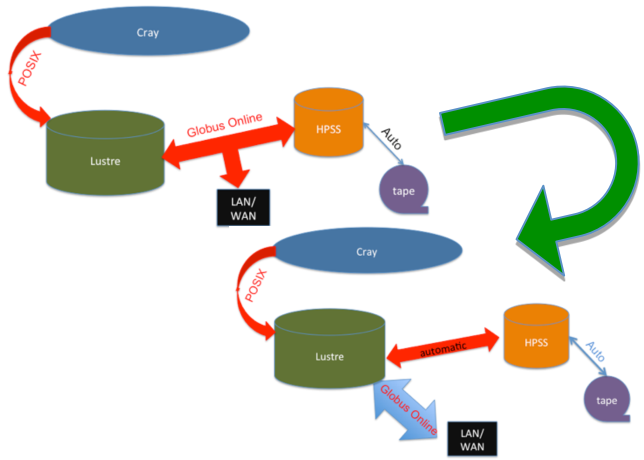
- 36 Sonnexion racks – about 1 PB raw per rack
- 25+ usable PB
- 3 file systems, 3 metadata servers
  - /home – 2.1 usable PB persistent, backup
  - /project – 2.1 usable PB persistent, backup
  - /scratch – 21 usable PB temporary, 30 day residency by file access
- ~ 4,000 Sandybridge cores
- 528 Lnet routers in Cray system
- Tested at 1.1+ TB sustained
  - Re-tested periodically – sustains > 1TB even with 75% full

Near Line Storage Sub-System

- World's Largest HPSS storage system
  - Bandwidth, capacity, file creates, …
  - RAIT implementation

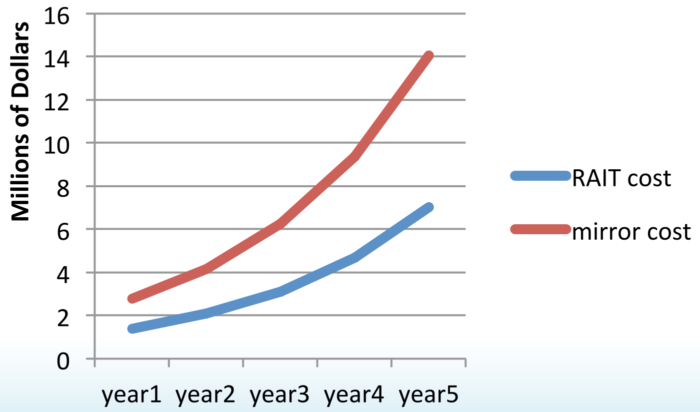
Near-Line Sotrage Subsystem	Phase 1 (test)	Phase 2	Phase 3	Potential
Numbers of HPSS Movers		28	28	
Globus On-line end points	4	50	50	
Number of TS1140 (IBM JAG 4) tape drives	24	244	366	416
Aggregate Bandwidth Performance (GB/s)	5.7	58.5	87.8	100
Number of dual arm SprectraLogic libraries	1	4	6	7
Active Slot count	1,500	63,720	95,580	125,104
Total media capacity	6PB	255PB	382PB	500+PB
HPSS cache		1.3 PB	1.3 PB	

Near-Line Storage



- ***Have the right data at the right place at the right time***
- ***Eliminate Partner Data Pain***
- Cost Efficient
  - RAIT
  - Managing data (limits, transparent movement, consolidation, etc.)
- Import/Export server management and support
- Community Leadership

- **Most balanced and intense storage implementation in open science**
  - Scale and Performance
- Advanced Technologies
  - RAIT, Lustre-HPSS Interface, ILM, etc.
- Maintain storage related software packages
- Maintain and improve BW developed SW
- Performance testing and tuning
- Import/export facility maintenance and service request management

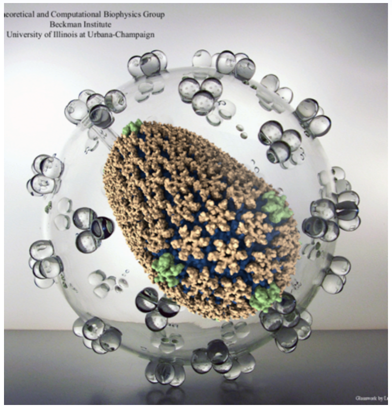


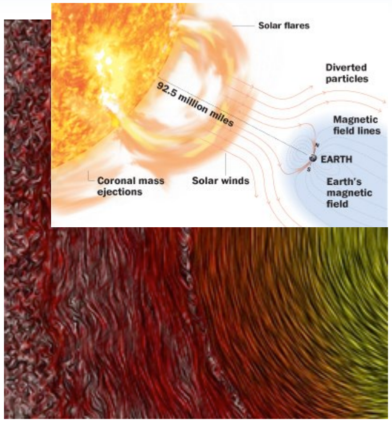
BLUE WATERS IN FULL SERVICE

Interim Full Service Starts April 2, 2013

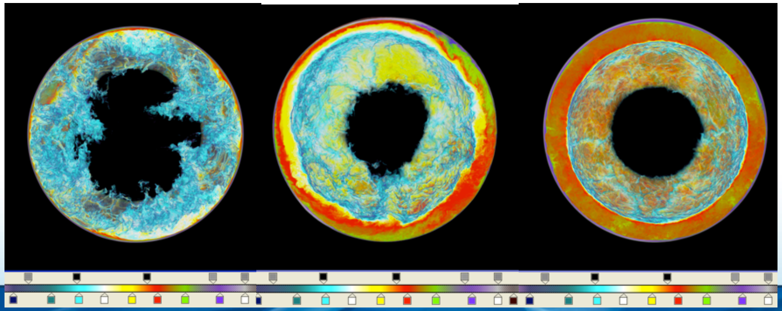
Full Production Service started September 1, 2013

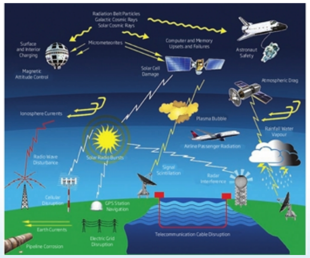
Petascale Usage





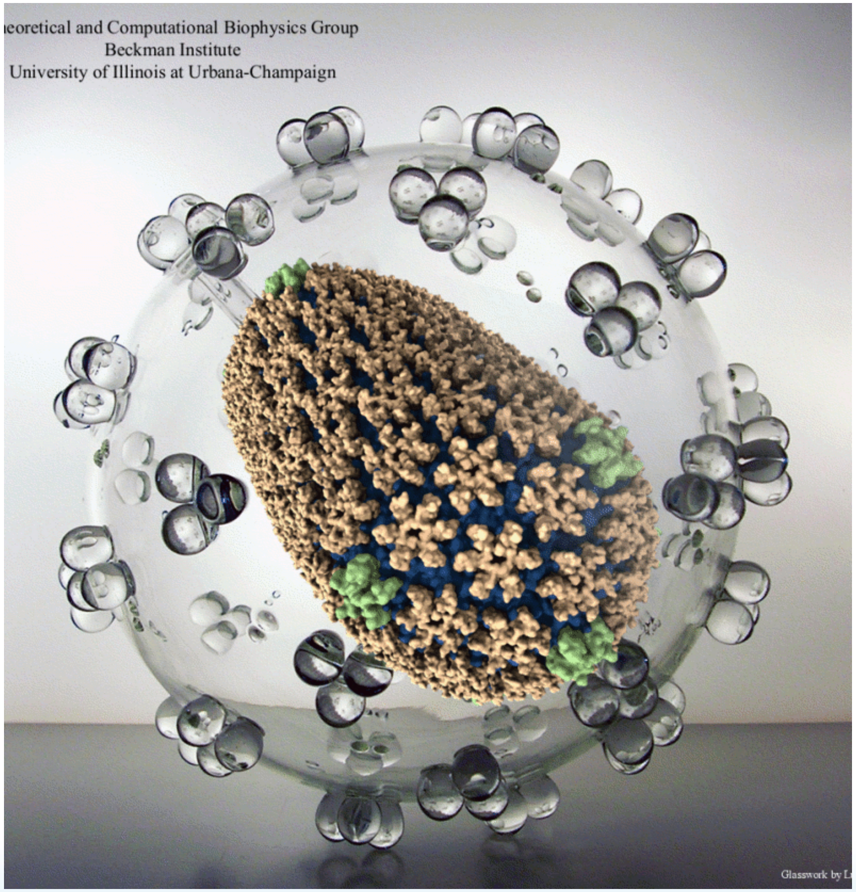
Category	Number of Teams
NSF - PRAC	28 active +6 exploratory 5 have completed
University of Illinois	30 15 General, 15 Exploratory
GLCPC	10
Education	4
Industry	1
Innovation and Exploration	8





Science Area	Number of Teams	Codes	Struct Grids	Unstruct Grids	Dense Matrix	Sparse Matrix	N-Body	Monte Carlo	FFT	PIC	Significant I/O
Climate and Weather	3	CESM, GCRM, CM1/ WRF, HOMME	X	X		X		X			X
Plasmas/Magnetosphere	2	H3D(M),VPIC, OSIRIS, Magtail/ UPIC	X				X		X		X
Stellar Atmospheres and Supernovae	5	PPM, MAESTRO, CASTRO, SEDONA, ChaNGa, MS-FLUKSS	X			X	X	X		X	X
Cosmology	2	Enzo, pGADGET	X			X	X				
Combustion/Turbulence	2	PSDNS, DISTUF	X						X		
General Relativity	2	Cactus, Harm3D, LazEV	X			X					
Molecular Dynamics	4	AMBER, Gromacs, NAMD, LAMMPS			X		X		X		
Quantum Chemistry	2	SIAL, GAMESS, NWChem			X	X	X	X			X
Material Science	3	NEMOS, OMEN, GW, QMCPACK			X	X	X	X			
Earthquakes/Seismology	2	AWP-ODC, HERCULES, PLSQR, SPECFEM3D	X	X			X				X
Quantum Chromo Dynamics	1	Chroma, MILC, USQCD	X		X	X	X		X		
Social Networks	1	EPISIMDEMICS									
Evolution	1	Eve									
Engineering/System of Systems	1	GRIPS,Revisit						X			
Computer Science	1			X	X	X			X		X

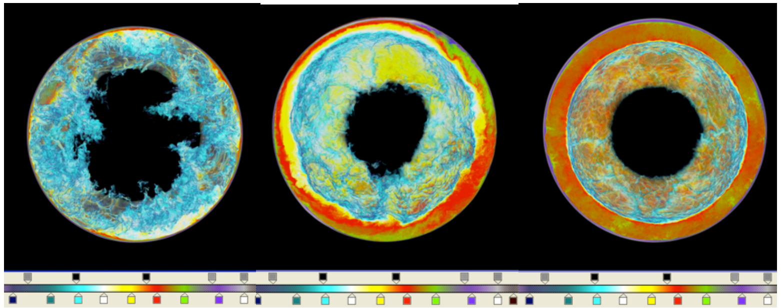
First Unprecedented Result – Computational Microscope

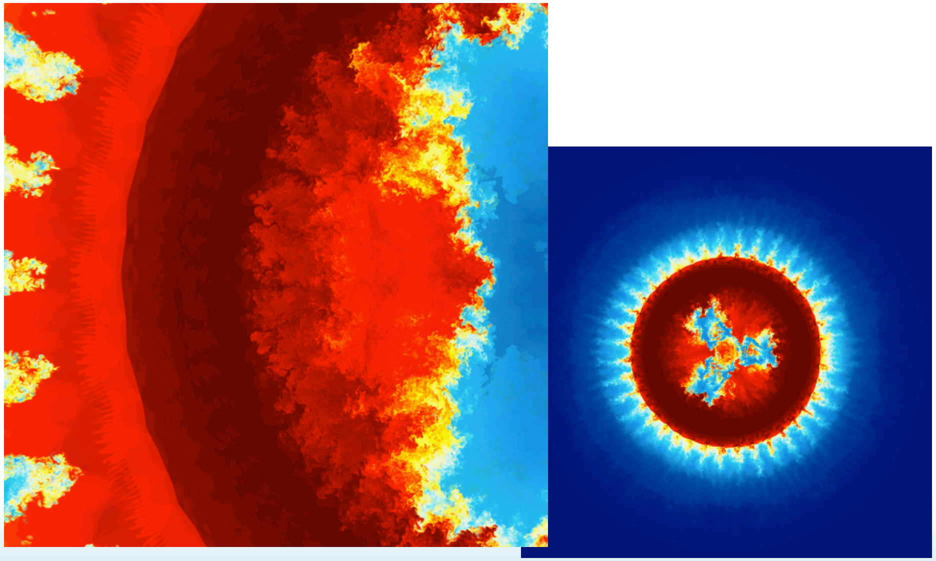


- Klaus Schulten (PI) and the NAMD group - Code NAMD/ Charm++
- Completed the highest resolution study of the mechanism of HIV cellular infection.
- May 30, 2013 Cover of *Nature*
- Orders of magnitude increase in number of atoms – resolution at about 1 angstrom

Petascale Simulation of Turbulent Stellar Hydrodynamics

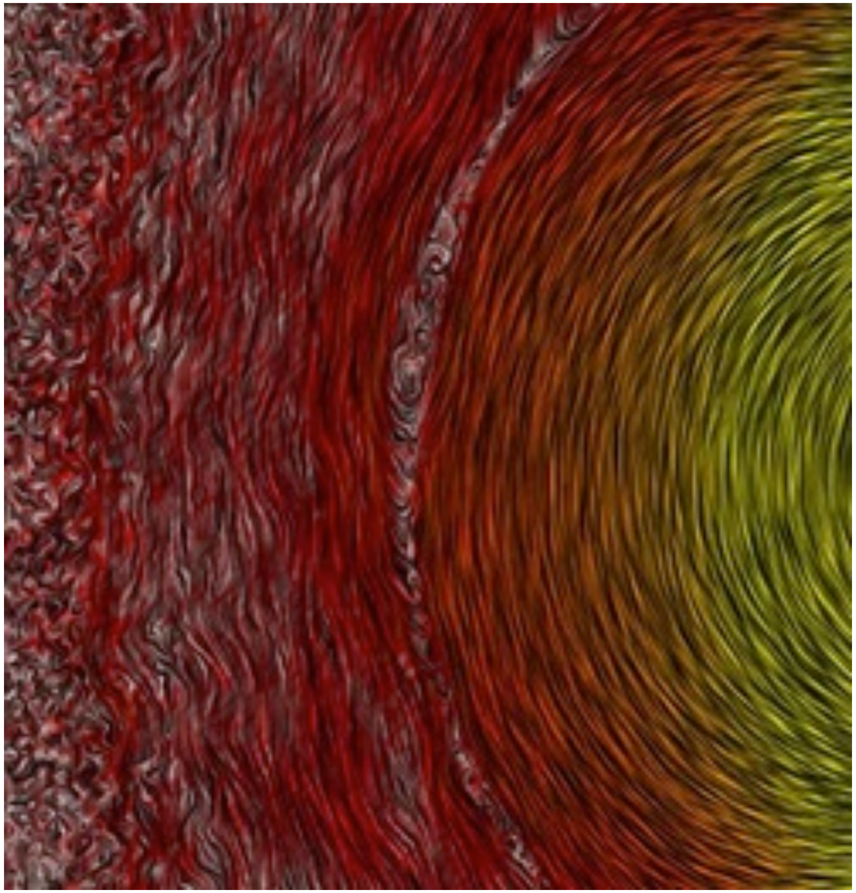
- Paul Woodward PI – Code PPM
  - 1.5 Pflop/s sustained on Blue Waters
  - 10,5603 grid
  - A Trillion Cell, Multifluid CFD Simulation
  - 21,962 XE nodes; 702,784 interger cores; 1331 I/Os; 11 MW
  - All message passing and all I/O overlapped w. comput.
  - 12% theoretical peak performance sustained 41 hrs
  - 1.02 PB data written and archived; 16.5 TB per dump.
  - Ran over 12 days in 6-hour increments



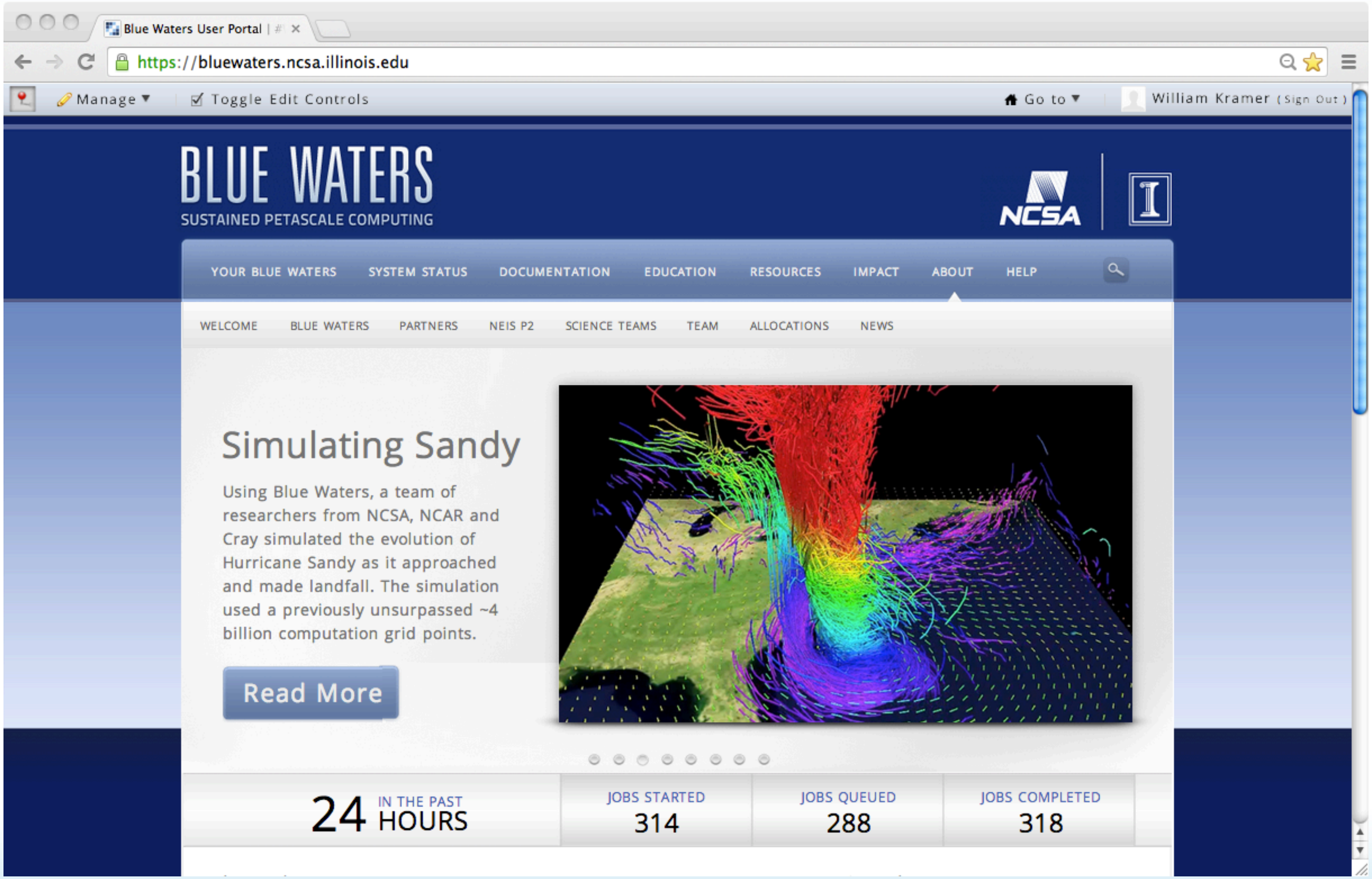


Enabling Breakthrough Kinetic Simulations of the Magnetosphere via Petascale Computing

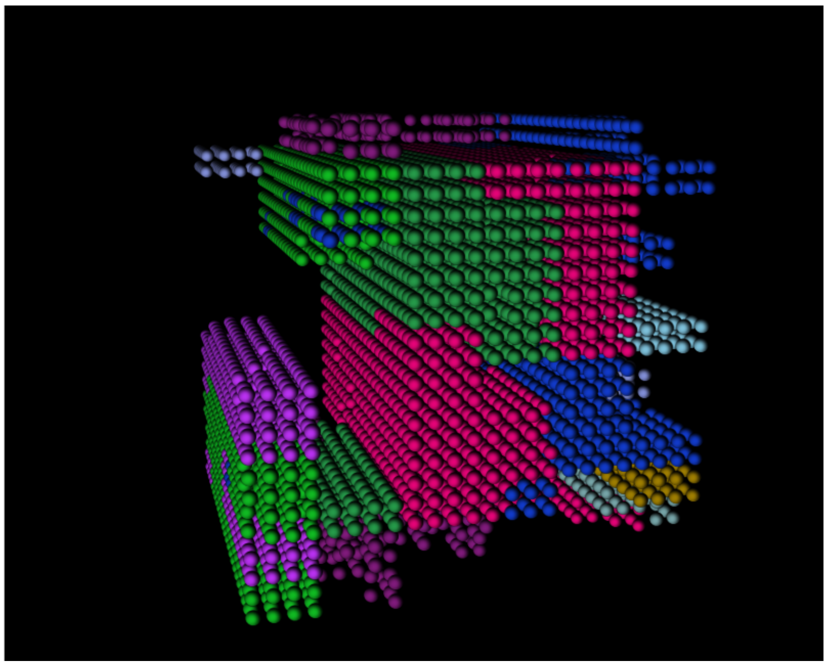
- Homa Karimabadi PI – Code PPM
  - Possible extreme solar storms could significantly disrupt many modern infrastructure systems
  - This project studies the initiation and transmission of the solar wind
  - Produced much higher resolution data sets being shared with 1,000's of other scientists

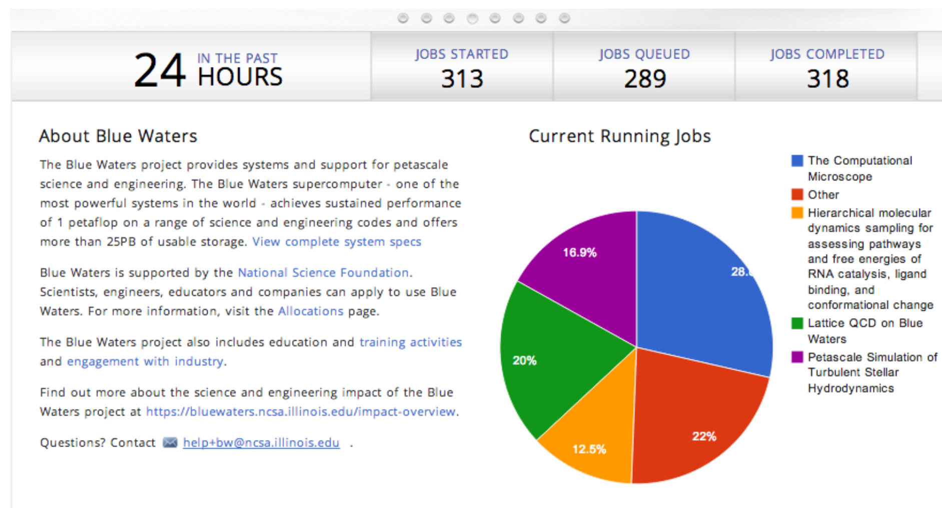


http://bluewaters.ncsa.illinois.edu

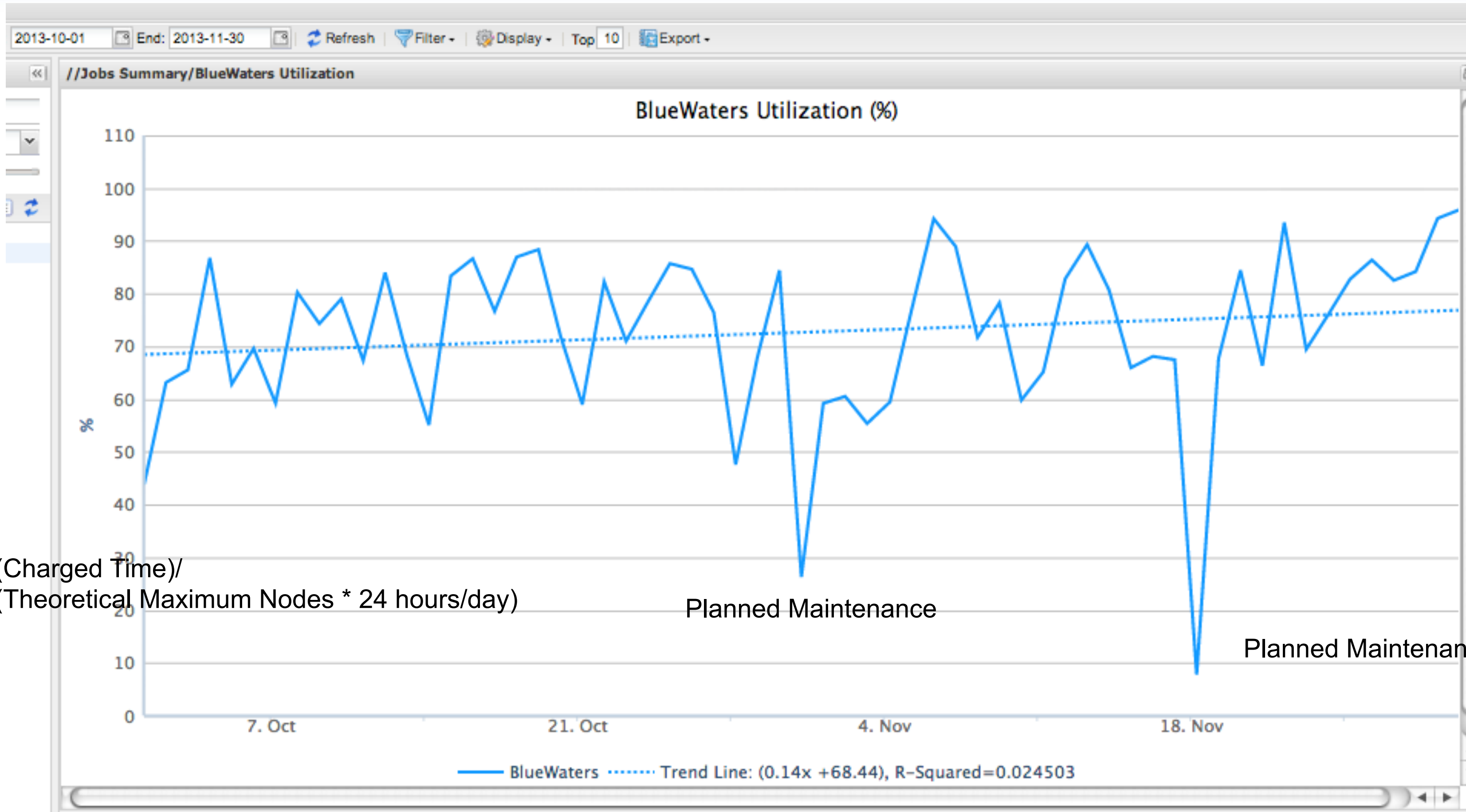


http://bluewaters.ncsa.illinois.edu





Gross Utilization on Blue Waters



(Charged Time)/
(Theoretical Maximum Nodes * 24 hours/day)

Planned Maintenance

Planned Maintenance

Description

- **BlueWaters:** Summarizes jobs reported to the BlueWaters central database (excludes non-BlueWaters usage of the resource).
- **BlueWaters Utilization (%)**: The percentage of resources utilized by BlueWaters jobs.
  BlueWaters Utilization: the ratio of the total CPU hours consumed by BlueWaters jobs over a given time period divided by the total CPU hours that the system could have potentially provided during that period. It does not include non-BlueWaters jobs.
  It is worth noting that this value is a rough estimate in certain cases where the resource providers don't provide accurate records of their system specifications, over time.

Petascale Usage

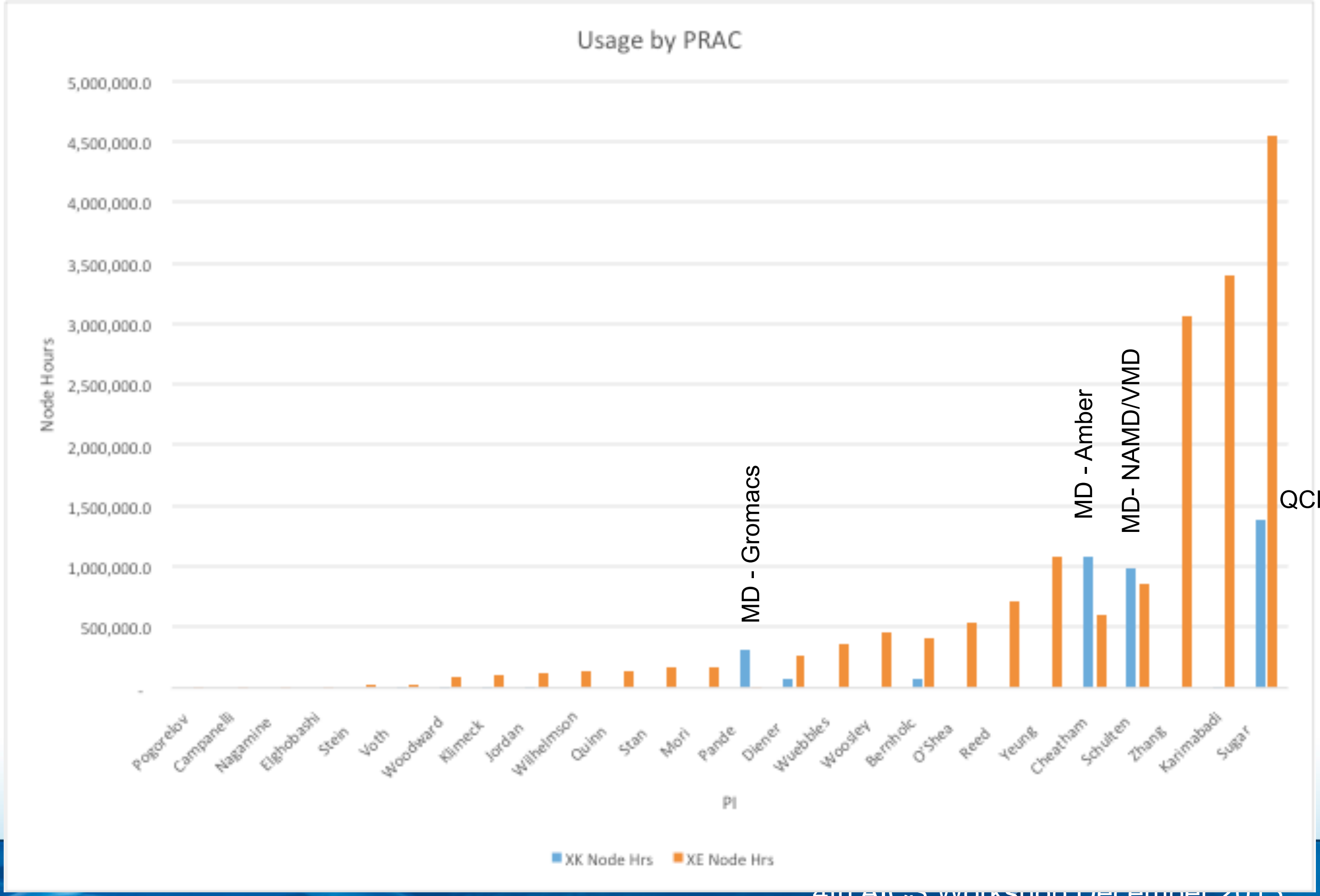
- Petascale Definitions of Scale
  - Not Large ≤ 1,284 nodes
    - ≤ 20,544 FP cores
    - ≤ 41,088 integer cores
  - Large ≥ 1,285 nodes
    - ≥ 20,560 FP cores
    - ≥ 41,120 integer cores
  - Very Large ≥ 4,584 nodes
    - ≥ 123,344 FP cores
    - ≥ 146,688 integer cores

- <u>Year to Date </u>Computational Usage
  - Not Large - 60%
  - Large - 25%
  - Very Large - 15%
    - Does not include any GPU usage

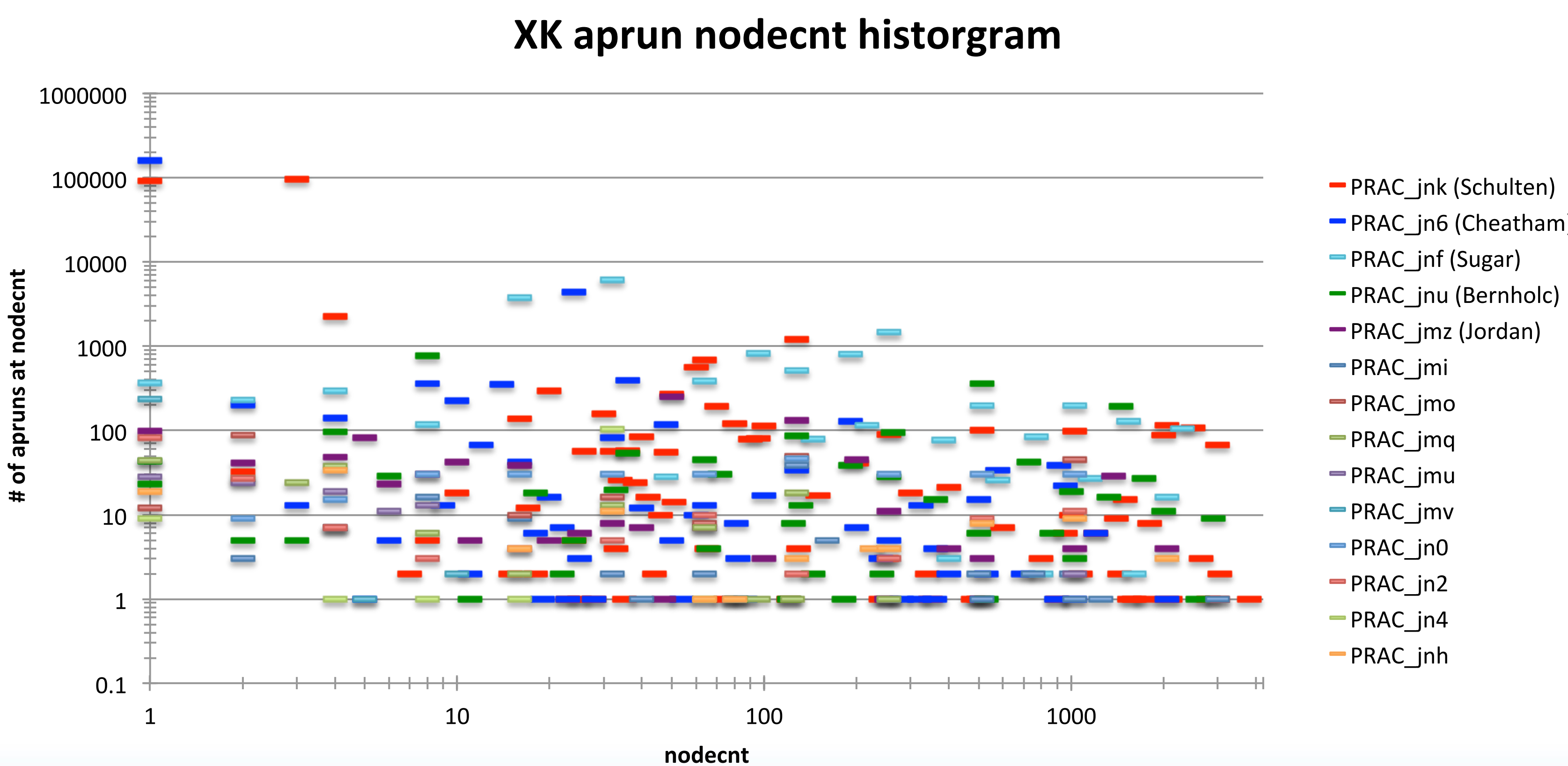
No longer can define core, processor…
- ~380,000 AMD x86 Floating-point Bulldozer cores,
- ~760,000 AMD x86 integer cores,
- 4,224 NVIDIA Kepler K20x GPUs or
- >12 million "cuda-cores"

Usage by NSF PRAC team – A Behavior Experiment

An observed experiment – teams self select what is most useful



XK jobs as of end of September



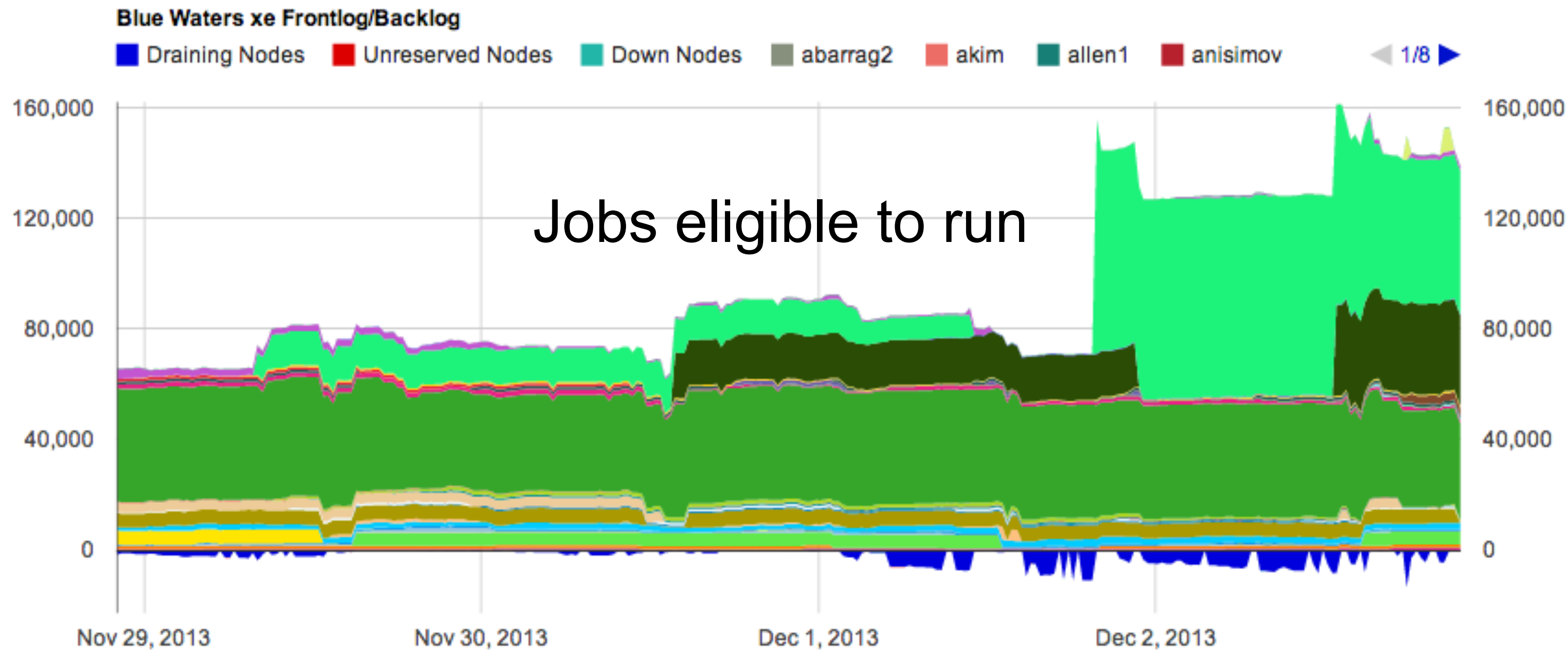
Slide Courtesy Greg Bauer

Frontlog/Backlog

Switch to XK | Previous | Next | Zoom Out | Zoom In

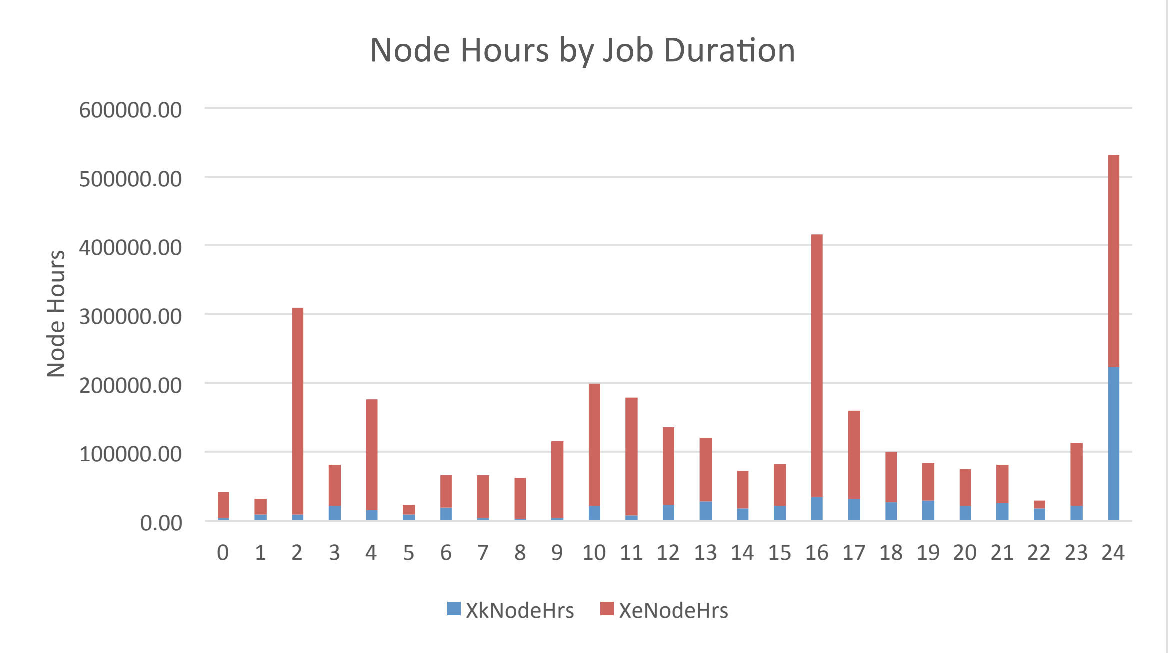
Start: Nov 28,2013 10:26 pm
End: Dec 2,2013 9:39 pm
Overall Utilization Average: 89.98%
Node-Hours lost from Undersubmitted Workload: 589.542 nodehours (0.03% avg)
Node-Hours lost by Down nodes: 1180.411 nodehours (0.05% avg)
Node-Hours spent Draining: 214284.062 nodehours (9.94% avg)
Current Chart Resolution: 30 Minutes

2 people currently viewing this chart



Full Service Job Characteristics

- July-Sept. 2013 Interval
- Large job expansion factor well under target of 10.
  - 1+(time in queue/time requested)



	Not Large Jobs	Large Jobs	Very Large Jobs
XE nodes	1- 1,132 nodes	1,133 - 4,528 nodes	4,529 - 25,712 nodes
XK nodes	1 - 16 nodes	17 - 256 nodes	257 – 4,224 nodes

Expansion factor	Very Large jobs	Large jobs	Not Large jobs
XK nodes	1.56	2.85	2.79
XE nodes	4.75	1.27	1.04

Community Engagement

- Community Outreach and Education activities to enable the general computational science and engineering community to make effective use of petascale systems. This component includes activities to help train the next generation of petascale computational experts through a coordinated **set of courses, workshops, and fellowships.**
- **Key Activities include**
  - **Hands-on workshops**
  - **Virtual School**: semester-long courses on the web to allow participation by students at multiple institutions across the country. The courses will be offered as a traditional college course, including a syllabus with learning outcomes**,**
    - Prototype course taught by Wen-Mei Hwu in Spring 2013.
- **Graduate Fellowships**
  - **Fellowships announced November 11, 2013** http://www.ncsa.illinois.edu/news/story/applications_now_being_accepted_for_blue_waters_graduate_fellowships
    - Candidates must already be enrolled in a PhD program at an accredited US non-profit academic institution at the time of application.
    - They must have completed no more than two years of graduate studies. The fellowship support is for one year, renewable based on performance for up to two additional years.
    - The level of support is up to $50,000 per year encompassing a stipend of $38,000 plus $12K in support of tuition and fees as well as support for travel to augment their learning and present papers in their field.
    - Must be US Citizen or Permanent Resident
  - **Internships**
    - Continuing effort from deployment phase
    - Available to undergraduate and graduate students - $5K, 1 week hands-on workshop at NCSA
    - Interns are paired with researcher(s)
    - Emphasis on engaging women, minorities and people with disabilities
  - **Blue Waters Symposium**
    - Showcases results from the Blue Waters system, and provides a forum for dealing with community issues and solutions for efficient parallel and heterogeneous petascale computing.
    - Planning for this event is underway, but date and location are still under consideration

PETASCALE LESSONS THAT @SCALE SHOULD ADDRESS

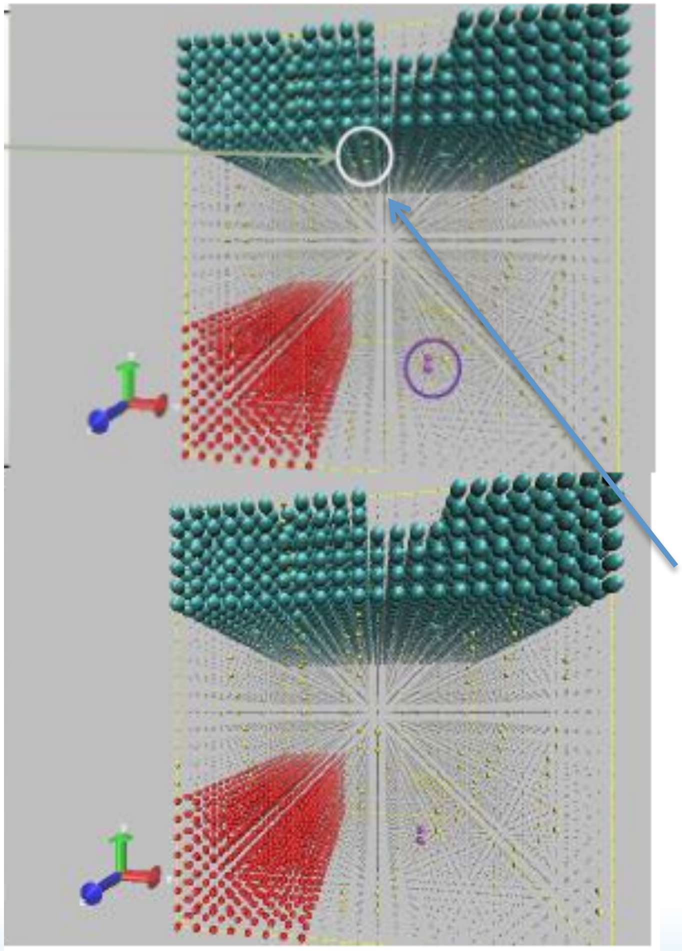
What Science Teams Did to Improve

- Another observed social experiment
- NEIS-P[2] – Direct Support
  - Blue Waters directly funded science teams to make improvements in their codes to enable them to "realize the full potential of the Cray XE6/XK7 system."
  - 20 PRAC teams participated
  - Component was **completed in Summer 2013**
  - Summary of activities and results for each team can be found on the Blue Waters website: https://bluewaters.ncsa.illinois.edu/neis-p2-final-reports
- Reporting by full Science Teams indicated more applications
  - Single node optimization
    - GPU Implementations – prefer a single code base
    - Heterogeneous use of x86 and K20x processors
  - Reduce Communication Pressure
    - Topology Awareness
    - Load Balancing
  - I/O and storage Improvements

Petascale Application Improvement Discovery (PAID) Program

- **Goals**
  - facilitate the creation of new methods and approaches that will dramatically improve the ability to achieve sustained science on petascale systems
  - assist the general computational science community in making effective use of systems at all scales.
- **Major Areas** from Component 1 and Production Experiences
  - Enable application-based topology awareness to more effectively and efficiently use limited bandwidth resources, and to fully exploit the new system functionality for topology aware scheduling that will be available on Blue Waters in 2014.
  - Increasing scalability of full applications, including much work with improving the load balancing within the applications.
  - Improve single node performance for applications, particularly to assist applications in layout, affinity, etc.
  - Increase the number of science applications that can use accelerators and many core technology by lowering the effort to re-engineer applications for these technologies and enabling the teams to maintain a single code base that can be applied to multiple architectures.
  - Enable integrated, at scale applications use of heterogeneous systems that have both general-purpose CPUs and acceleration units.
  - Improve the use of advanced storage and data movement methods to increase the efficiency and time to solution of applications.
  - Assessment and dissemination of science and society impacts resulting from petascale Science

Topology Matters – Good and Bad

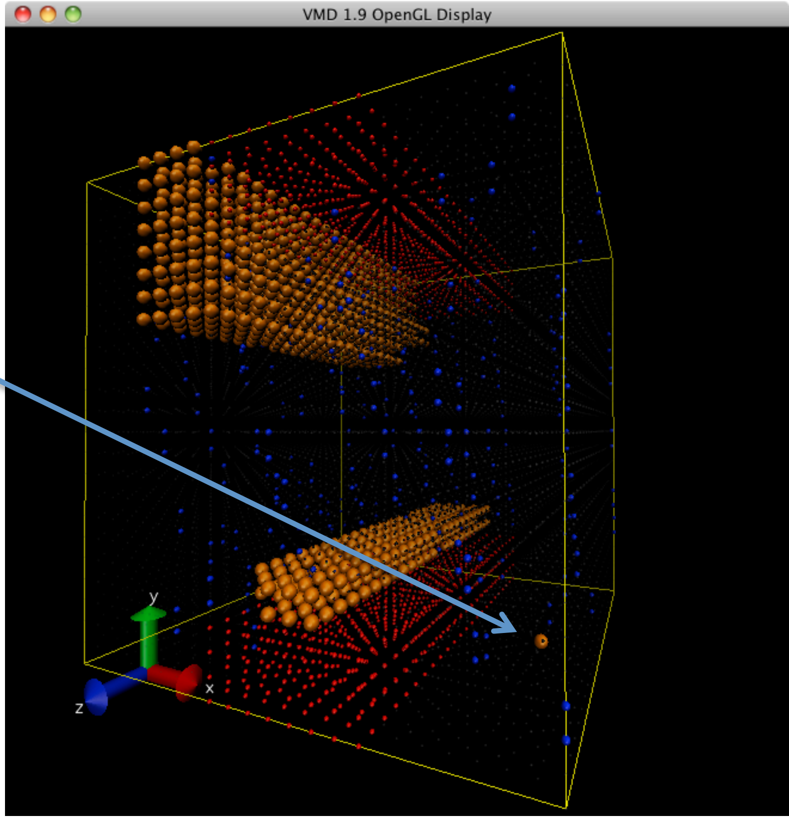


Much of the Benchmark tuning was topology based

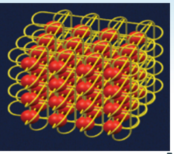
1 poorly placed node out of 4116 (0.02%) can slow an application by >30% (on dedicated system)

Just 1 of 3057 gemini down out in the wrong place of 6114 can slow an application by >20% (P3DNS – 6114 Nodes)

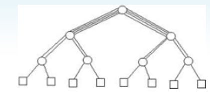
Later Slides for Positive Impacts

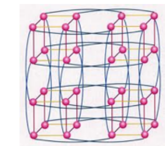


Appears in all system and many applications, but scale makes it clear

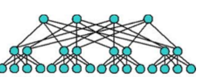


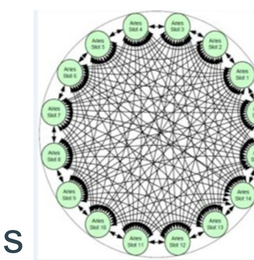
Application Flexibility Performance and Scalability

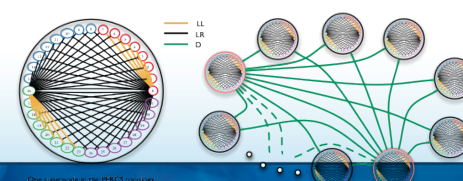




- Applies to all systems and topologies
- Need a system and application partnership to do the best
- Cray developed new management and tuning functions
  - Bandwidth Injection and Congestion Protection features – helps all systems
- BW works with science teams and technology providers to
  - Understand and develop better process-to-node mapping analysis to determine behavior and usage patterns.
  - Better instrumentation of what the network is really doing
  - Topology aware resource and systems management that enable and reward topology aware applications
  - Malleability – for applications and systems
    - Understanding topology given and maximizing effectiveness
    - Being able to express desired topology based on algorithms
    - Mid ware support
- Even if applications scale, consistency becomes an increasing issue for systems applications
- This will only get worse in future systems

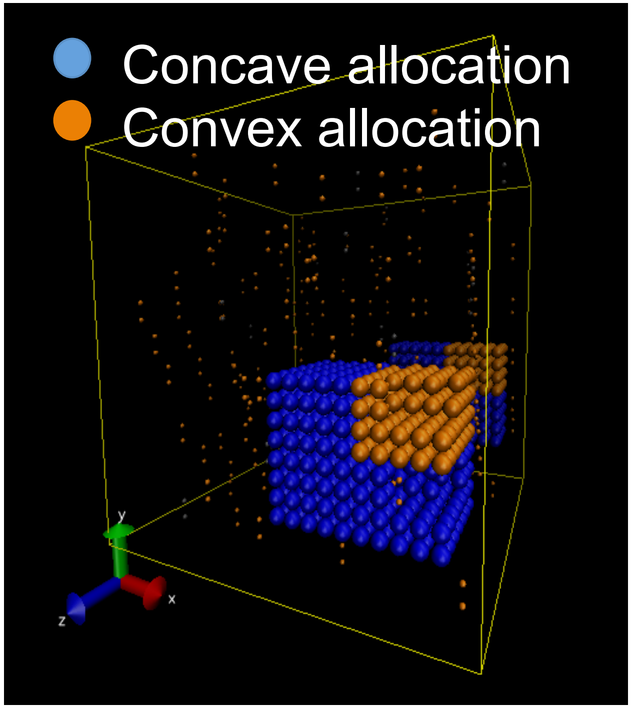






Impact of Node allocation

- Job – Job interaction
  - Analysis of key application communication intensity and sensitivity
  - 20% slowdown typical, 100% or more possible.



Communication	MILC	NAMD	NWCHEM	PSDNS	WRF
Intensive	2	2	3	2	1
Sensitive	2	3	1	2	1

1 – low 3 – high
as viewed by convex app.

Topology Mitigations Today

Shape Targeting

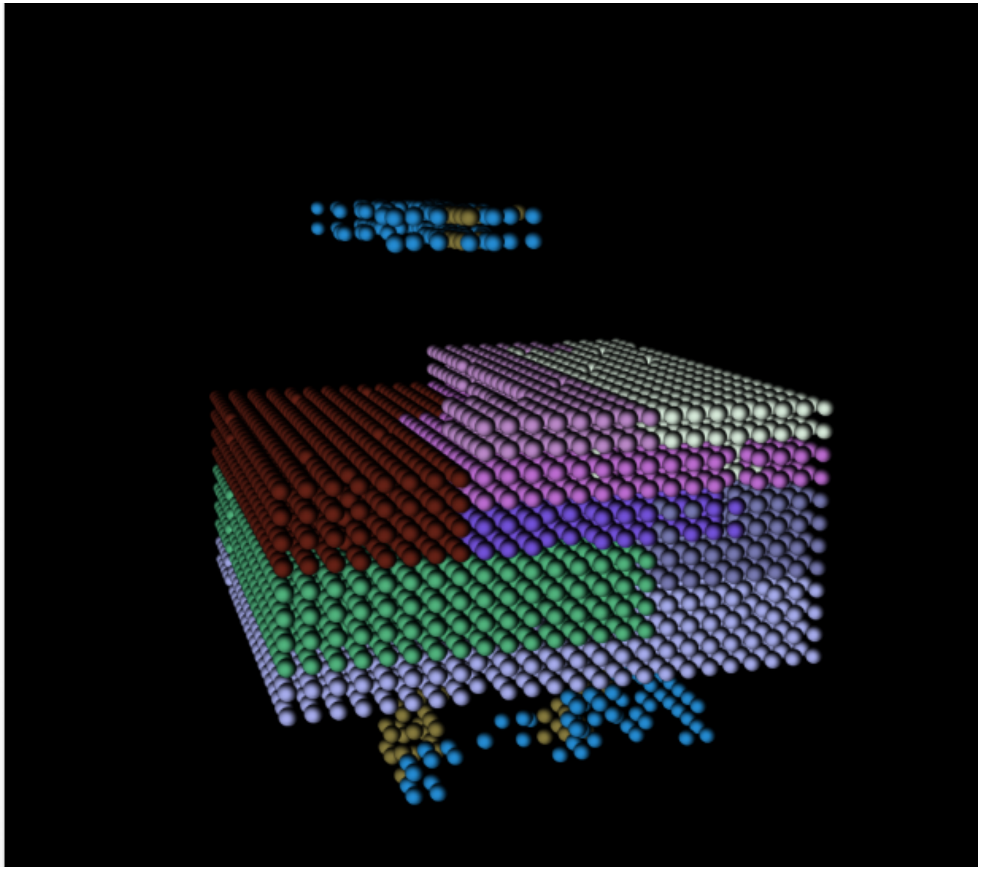
- Slabs/sheets and cubes of popular node counts pre-defined
- Defined in XZ plane to maximize global bandwidth
- Defined to edges to maximize torus wrap-around benefit
  - Also reduces chance of interfering application
- Designated shapes can dodge dateline boundaries
- Use existing scheduler features to target "one of" qualified targets
- Threatens utilization and turnaround time
  - more than 3d topology awareness would
- Improves consistency
- Improves performance

Slide Courtesy Jeremy Enos

Topology Mitigations Today

Alternative ALPS Node ID ordering

- 4x2x8 "bricks" laid out in Z
- Fills XZ plane first
- Return path folding
  - Z-bar returns
  - Plane level
- Ignore XK region remnants



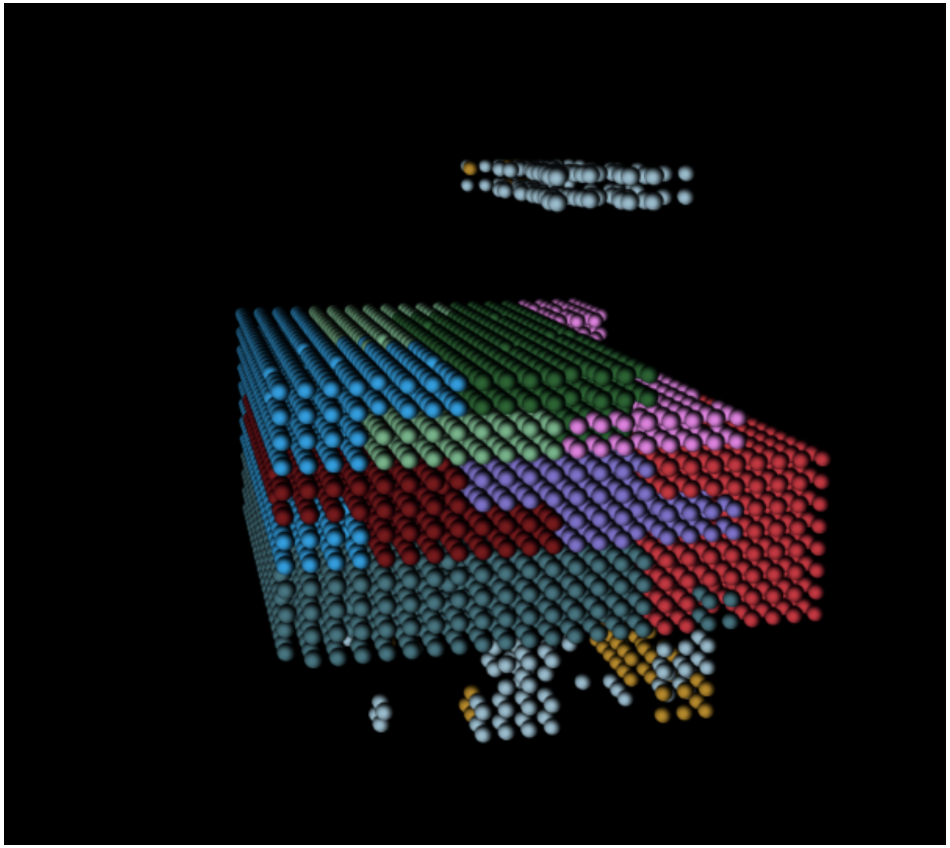
Sheet/slab 2 high in Y

Slide Courtesy Jeremy Enos

Topology Mitigations Today

Alternative ALPS Node ID ordering

- 4x2x8 “bricks” laid out in Z
- Fills XZ plane first
- Return path folding
  - Z-bar returns
  - Plane level
- 4 high in Y (crude, not Hilbert)
- Not optimized, but compare worthy against 4 wide YZ bias
- Ignore XK region remnants



Sheet/slab 4 high in Y

Slide Courtesy Jeremy Enos

Topology Mitigations Today

Alternative Node ID ordering

- Surprise - even crude 4y high did as well or better in most tests
- 10-17% improvement average over baseline for all applications tested
- 2y high probably choice – average skewed by apps that will select for cubic

Application	node count	baseline timing	test 1	test 2
cesm	600	sec	sec	sec
changa	1024	168.9s	169s	177.7s
dns_distuf	512	0.039 sec/step	0.037 sec/step	0.04 sec/step
milc	1372	11.76 sec/step	10.26 sec/step	12.90 sec/step
milc	2744	8.81 sec/step	6.16 sec/step	6.74 sec/step
namd	1368	10.5 ms/step	11.5 ms/step	8.65 ms/step
nwchem	3000	616.0 sec	467.6 sec	442.7 sec
psdns	1024	49.47sec/step	40.68sec/step	28.42sec/step
wrf	1386	0.781025	0.771038	0.771045

Mitigations Today

Alternative Node ID ordering

- Limit XK plane order to half dimension
- Begin XE allocation in full plane area
- Increases the metric for other topology awareness effort
- Fragmentation still a issue but Moab contiguous allocation may be an option to minimize

System wide rank reordering mechanisms in development

Slide Courtesy Jeremy Enos

Congestion Protection

- To avoid data loss, traffic injection is throttled for a period of time, when reaching a point where forward progress is stalling. Throttling is applied and removed until congestion is cleared.
- System monitors percentage of time that traffic trying to enter the network from the nodes and percentage of time network tiles are stalled.
- Fortunately not a common occurrence. It does happen, typically in bursts.
- Can happen with node-node (MPI, PGAS) or node-LNET (IO) traffic.
- Many-to-one and long-path patterns.
- Libraries and user can control node injection as a precaution.
- In CP reports, flit rates represent data arriving at the node from the interconnection network.

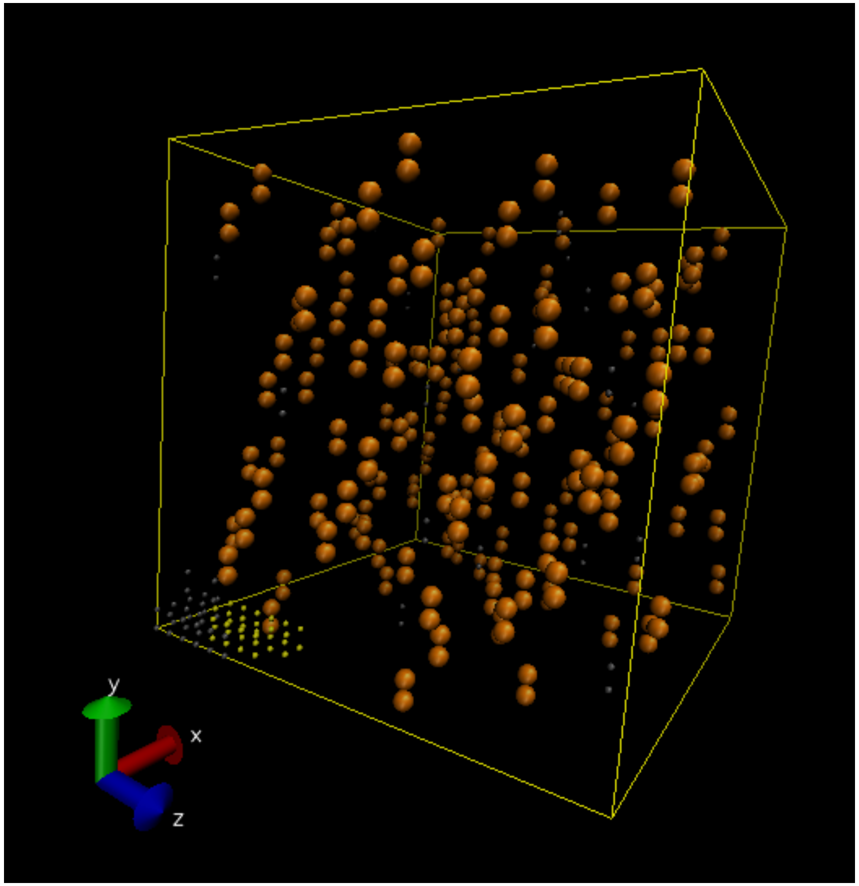
```
                                        Max
APID     Name                   Nodes   Flits/s   UID     Start      End
-----------------------------------------------------------------------------------
2220460 Castro3d.Linux.          2048     31698   46466   16:00:45   19:41:40
2220462 Castro3d.Linux.          2048     81115   46466   16:01:05   19:37:03
2218386 namd2                    2000        --   43448   01:58:31   18:02:09
2220803 psolve                   2000     45732   47252   17:12:34   17:30:30
2218759 su3_rhmd_hisq_q          1536        --   12940   07:29:16
2219859 nwchem                   1000        --   32745   13:58:50   18:02:07
2220668 nwchem                   1000   4128749   32745   17:00:22   18:15:32
2219678 ks_spectrum_his           768        --   12940   11:30:04
2219512 namd2                     700        --   42864   10:35:55
…
…

=====================================================
Top Bandwidth Applications
=====================================================
0: apid 2218386 userid  43448 numnids  2000 apname            namd2 Kflits/sec: Total
3075
1: apid 2219859 userid  32745 numnids  1000 apname           nwchem Kflits/sec: Total
2743
2: apid 2220462 userid  46466 numnids  2048 apname  Castro3d.Linux. Kflits/sec: Total
2715
3: apid 2220460 userid  46466 numnids  2048 apname  Castro3d.Linux. Kflits/sec: Total
2691
4: apid 2219517 userid  42864 numnids   700 apname            namd2 Kflits/sec: Total
2271
5: apid 2219519 userid  42864 numnids   700 apname            namd2 Kflits/sec: Total
2073
6: apid 2218759 userid  12940 numnids  1536 apname  su3_rhmd_hisq_q Kflits/sec: Total
2071
7: apid 2219514 userid  42864 numnids   700 apname            namd2 Kflits/sec: Total
1762
8: apid 2220646 userid  12940 numnids   512 apname  ks_spectrum_his Kflits/sec: Total
1596
9: apid 2217219 userid  47296 numnids   500 apname           python Kflits/sec: Total
1389
…
=====================================================
Congestion Candidate COMPUTE Nodes
=====================================================
1: c17-0c1s0n1 (64051 flits/sec) (nid 18401; apid 2220473 userid 14394 numnids 32 apname
numa_script.sh)
2: c9-0c0s1n0 (61950 flits/sec) (nid 23036; apid 2219894 userid 14394 numnids 32 apname
numa_script.sh)
3: c10-1c0s3n2 (24438 flits/sec) (nid 5798; apid 2219756 userid 14394 numnids 32 apname
numa_script.sh)
4: c3-10c0s5n1 (24238 flits/sec) (nid 25867; apid 2219672 userid 35077 numnids 64 apname
enzo.exe)
5: c12-1c0s2n2 (22544 flits/sec) (nid 8026; apid 2219756 userid 14394 numnids 32 apname
numa_script.sh)
6: c5-10c0s6n3 (20193 flits/sec) (nid 24813; apid 2219672 userid 35077 numnids 64 apname
enzo.exe)
7: c12-1c0s2n0 (20161 flits/sec) (nid 8004; apid 2219756 userid 14394 numnids 32 apname
numa_script.sh)
8: c14-1c0s3n0 (19784 flits/sec) (nid 8120; apid 2219756 userid 14394 numnids 32 apname
```

Slide Courtesy Greg Bauer

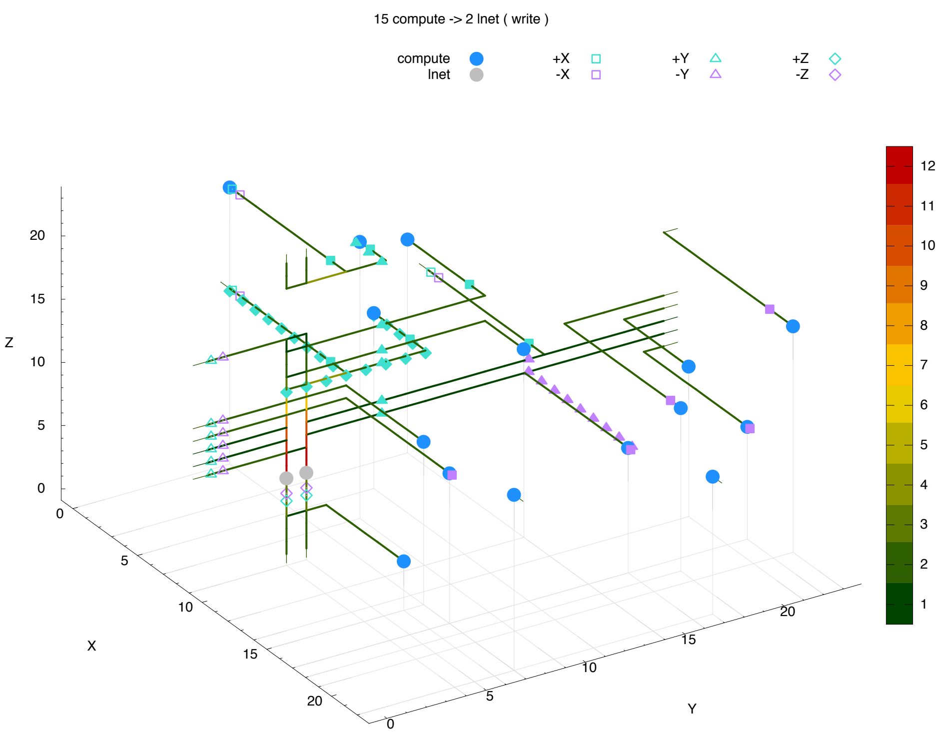
IO and Storage

- LNETs scattered across the torus (orange colored geminis).
- Specific OSTs served by specific LNETs (not a full fat tree for the IB between OSTs and LNETs).
- IO is “topology sensitive”.



Slide Courtesy Greg Bauer

Routing of IO write

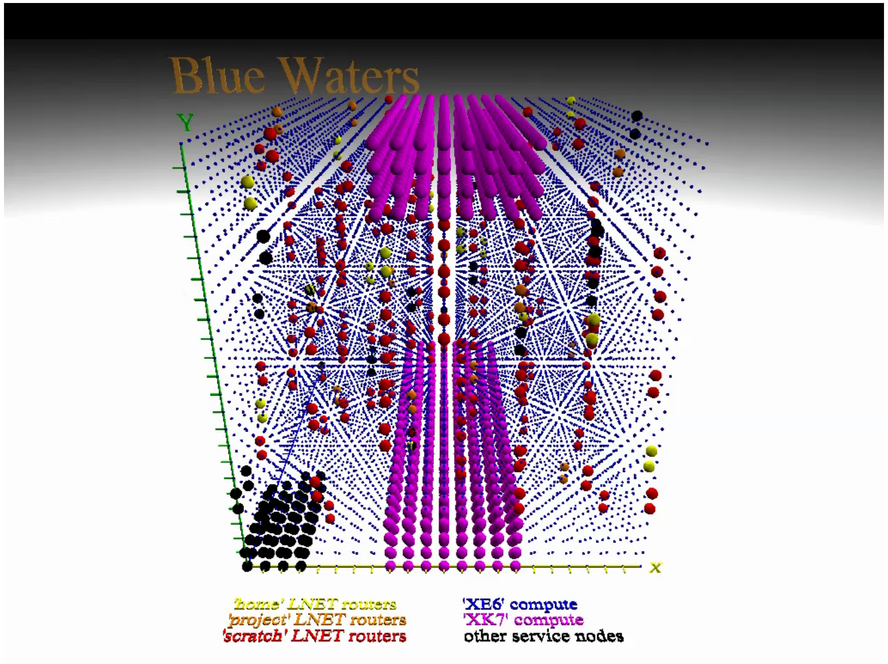


- 15 compute geminis (•) (30 nodes) writing to files served by a LNET pair (•).
- Color scale is the number of convergent routes on the link.

Slide Courtesy Greg Bauer

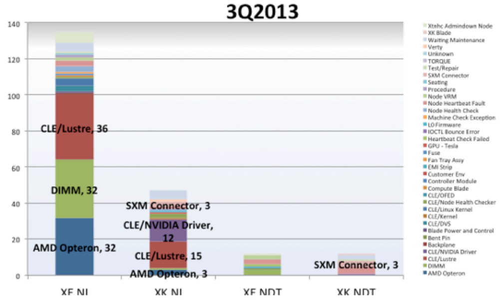
I/O Is Complex

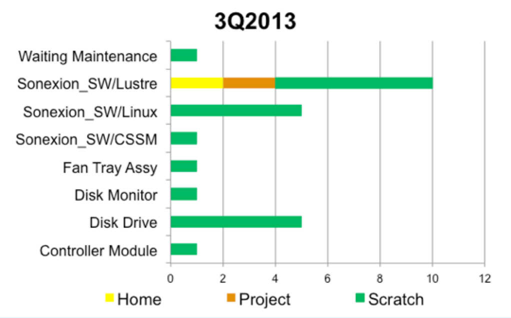
- Many Challenges
  - Scale – timings
  - Filesystem failure over works – just not fast enough all the time
- Data and I/O server placement make this a complicated topology based optimization
- Reads slower than writes at scale – one to many rather than many to one



Resiliency Needs New Approaches

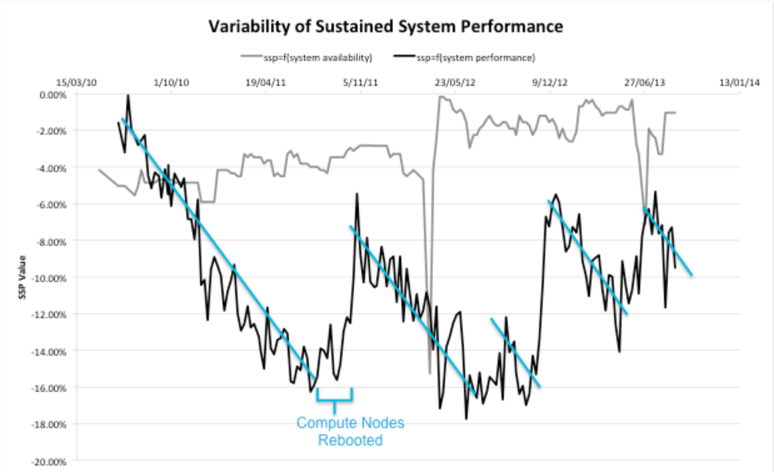
- Current State on Blue Waters
  - 2-2.5 node failures a day – getting less (lower 2's now)
    - Any cause to fail with application running
    - Any cause to fail proactive node health check
  - SWOs – average 5 days system wide MTBF over several months and improving
- Hardware is better than expected
  - E.g. just 7 full HDD failures in 6 months
  - Software error rates not measurable yet
- Software causes the majority of the failures
  - Almost all storage failures
  - More than ½ the node failures
  - More than ½ the system wide outages
  - Possible the cause of congestion events
- Most research is about the hardware not software
  - Most resiliency concerns are about hardware not software
- Defensive I/O (checkpoint) is increasingly intrusive.
  - System-assisted application based flexible resiliency as a path to the future
  - Need multi-vendor implementations or applications will continue to do things they can control





Consistency The Missing Criterial

- Becoming Intolerable
  - Topology helps – sometimes – but not enough
  - Same code – same nodes – different time
- Most is not OS Jitter related
- Congestion Events Intrusive – see Blue Waters' presentations later for more details
  - Prediction should be possible – but not currently available
- Causes overestimation of run time and less efficient system scheduling
- Impacts resource requirements estimation
- Not just at the largest scales – see slide courtesy of Tom Pugh – Australian Met Office



Monitoring at Petascale a Big, Unstructured Data Problem

SOURCE	Average MBs/Day	Max MBs/Day
apstat	0.05	0.06
bwbackup	0.06	0.74
esms	229.79	622.71
hpss	345.29	1391.80
hpss_core	0.08	0.22
ibswitch	0.80	1.59
jcc	0.17	0.93
moab	2539.40	5678.32
sched	0.07	0.08
SEL	0.23	0.42
sonexion	326.67	870.15
syslog	12563.31	102626.18
torque	31.66	103.78
volkseti	11.42	27.38

Average

- **15 GBs/day**
- **>88M events/day**
- **> 10,500 defined events**

Does not include OVIS CPU, Gemini and Darshan data collection

- **Will be significant increases at 1 minute resolution for all nodes**

EVALUATING SUSTAINED PERFORMANCE

Time to Solution is THE Metric

- The consensus of many papers/experts is the only real, meaningful metric that can compare systems or implementations is the time it takes to solve a defined, real problem on systems.
  - Work is a task to carry out or a problem to solve
  - Just like in the real world, work is not a rate, it is not a speed, it is a quantity
- The work is meaningful effort, not overhead work or useless work
- Hence a good evaluation compares how much time it takes to do an amount of meaningful (productive) work
  - Referred to as the System's Potential to do the work
  - Cost effectiveness = system's potential/system's cost
    - Cost can have many components as well

Time to Solution is <u>THE</u> Metric (cont)

- Time to Solution comparisons have their own challenges
  - Defining what the work is in an discrete manner (i.e. data input set)
  - Defining the work process(es) (application/algorithm/code path…)
  - Picking a unit to represent the work
  - Defining work across disciplines for multi use systems
  - Defining useful work vs overhead work (to parallelize, to move data, to set up, key steps)
  - Balancing practical issues
    - Complexity, testable system size, tractable length the test runs, number of tests, quality of implementation, optimizations

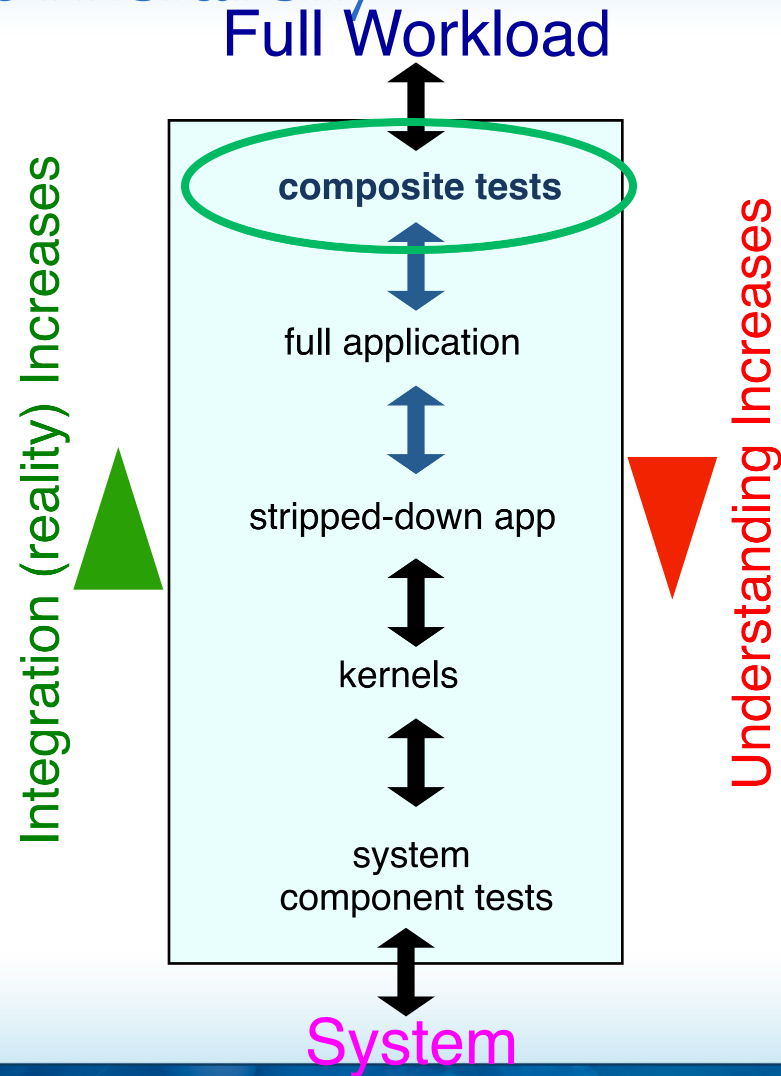
BENCHMARK TEST REFRESHER

What is a Benchmark?

- Benchmark tests are approximations of real, practical work a computer system accomplishes
- Benchmark tests estimate the potential of computer systems to solve a set of problems
- Benchmark tests are made up of computer programs and the input data sets that state a problem for the program to solve
  - Today's real applications are complex and generally solve multiple problems and have different ways to define the methods used
  - Each input data set causes a different code path to execute with possibly different characteristics and performance
  - Many applications have markedly different code paths and characteristics based on input
    - PME steps for chemistry, converging criteria, time step resolution, memory use…
      - E.g MILC for NSF Track1 – same code – two very different problems and characteristics, MADCAP for NERSC-3/4 – same code – multiple different problems and characteristics
  - Hence, one cannot evaluate a benchmark result without defining its input and therefore its code path and execution characteristics
- Sophistication of the approximation represented by the benchmarks depends on the fidelity needed to represent the true workload relative to the goals of getting good measurements

Benchmark and Test Hierarchy

1. Analyze Application Workload
2. Select Representative Applications And Problems
3. Determine Test Cases (e.g. Input, Concurrency)
4. Package and Verify Tests



Slide Courtesy of Katie Antypas and the NERSC-6 Team

The Key Purposes of a Benchmark

1. Evaluation and/or selection of a system from among its competitors.
2. Validating a selected system actually works the way it is expected to operate once a system is built and/or arrives at a site.
   - This purpose may be more important than the first and is particularly key when systems are specified and selected based on performance projections rather than actual runs on the actual hardware.
3. Assuring the system performance stays as expected throughout the system's lifetime (e.g. after upgrades, changes, and regular use)
4. Helping guide future system designs.

Most reports/papers discuss only the first of these four purposes benchmarks play in the life of a system.

The majority of tests claim to do well on the first goal and possibly one other of the goals, but few are effective in all purposes.

From Method To Implementation

- Sustained Petascale Performance Metric is the Blue Waters/NSF implementation of the SSP Method
- To move from the Method to Metric
  1. Select number and instances of applications and problem sets
  2. Select Input sets that determine the code paths
  3. Establish Reference Counts
  4. Optimize (or not)
  5. Run Tests
  6. Composite
  7. Evaluate
  8. Repeat 4 thru 7 or 2 thru 7 or until complete

BW Sustained Performance Measures

- Original NSF Benchmarks
  - Full Size – QCD (MILC), Turbulence (PNSDNS), Molecular Dynamics (NAMD)
  - Modest Size – MILC, Paratec, WRF
- Sustained Petascale Performance (SPP) expands the original requirements as it is a time to solution metric that is using the planned applications on representative parts of the science team problems
  - Represents end to end problem run including I/O, pre and post phases, etc.
  - Coverage for science areas, algorithmic methods, scale
- SPP Application full applications (details and method available)
  - NAMD – Molecular Dynamics; MILC, Chroma – Lattice Quantum Chromodynamics; VPIC, SPECFEM3D – Geophysical Science; WRF – Atmospheric Science; PPM – Astrophysics; NWCHEM, GAMESS – Computational Chemistry; QMCPACK – Materials
- The input, problem sizes, included physics, and I/O performed by each benchmark is comparable to the simulations proposed by the corresponding science team for scientific discovery.
- Well defined reference operation counts used to represent work across disciplines
- Each benchmark sized to use one-fifth to one-half of the number of nodes in the full system.
  - At least three SPP applications run at full system size

Determining Reference Operation Counts

- Determining the total number of reference work operations (e.g. FLOPs) required for each SPP science problem requires specifying the code version and the input problem data set.
- The GigaFLOP value used to calculate $P_{\alpha,i}$ is based on reference FLOP counts obtained using *best practices*. In order of preference, these best practices are:
  - hand-counting the floating-point operations within the code (where feasible),
  - using developer-implemented measures of the number of FLOPs executed, or
  - collecting hardware counter data collected by running the problem on Interlagos processors. When hardware performance counters are collected, the hardware counter data was compared to hand counts or developer-implemented measures (where available) for validation.
  - In order to avoid including extra FLOPs that may result from the extra operations used for scaling such as redundant computations, etc., scaling assessments were collected and compared hardware counter data obtained from multiple runs at different node counts for the same total problem size.
  - Enabled determination of whether the FLOP count for a fixed total problem size increases with the number of nodes, as well as how to eliminate any superfluous FLOPs from FLOP counts obtained at the desired scale.

SPP Method Coverage

Science Area	Struct Grids	Unstruct Grids	Dense Matrix	Sparse Matrix	N-Body/ Agent	Monte Carlo	FFT	PIC	Significant I/O
Climate and Weather	X	X		X		X			X
Plasmas/Magnetosphere	X				X		X		X
Stellar Atmospheres and Supernovae	X			X	X	X		X	X
Cosmology	X			X	X				
Combustion/Turbulence	X						X		
General Relativity	X			X					
Molecular Dynamics			X		X		X		
Quantum Chemistry			X	X	X	X			X
Material Science			X	X	X	X			
Earthquakes/Seismology	X	X			X				X
Quantum Chromo Dynamics	X		X	X	X		X		
Contagion (Social) Networks					X				
Evolution									
Engineering/System of Systems						X			
Computer Science		X	X	X			X		X

BW SPP Test Components

- SPP – is a time to solution metric that is using the planned applications on representative parts of the Science team problems
  - **Represents end to end problem run including I/O, pre and post phases, etc.**
  - Coverage for science areas, algorithmic methods, scale
- SPP Application Mix (details and method available)
  - NAMD – molecular dynamics
  - MILC, Chroma – Lattice Quantum Chromodynamics
  - VPIC, SPECFEM3D – Geophysical Science
  - WRF – Atmospheric Science
  - PPM – Astrophysics
  - NWCHEM, GAMESS – Computational Chemistry
  - QMCPACK – Materials Science
- Minimum SPP for x86 processors plus
- Kepler processors have to add at least 13% more above the x86 SPP
- At least three SPP benchmarks run at full scale

BW SPP Test Components

XE

Area	Code - version	Run Scale (XE Nodes) (Multiply by 16 or 32 to get cores)	Features
Molecular Dynamics	NAMD v2.0	5,000	C++, Charm++
Quantum Monte-Carlo	QMCPACK v52	4,800	C++/Fortran, MPI+OpenMP
Quantum Chromodynamics	MILC 7.6.3	4,116	C/C++, MPI/ pthreads
Quantum Chemistry	NWChem 6.1	5,000	C/Fortran, GA
Climate/ Weather	WRF 3.3.1	4,560	C/Fortran, MPI +OpenMP
Earthquakes/ Seismology	SpecFEM3D 5.13	5,419	F90/C++, MPI
Stellar Atmospheres and Supernovae	VPIC	4,608	Fortran/C, MPI +OpenMP
Plasmas/ Magnetosphere	PPM – 7/2/12	8,256	Fortran, MPI +OpenMP

XK

Area	Code	Run Scale	Method
Molecular Dynamics	NAMD	768	Cuda
Quantum Monte-Carlo	QMCPACK	700	Cuda
Quantum Chromodynamics	CHROMA	768	Cuda
Quantum Chemistry	GAMESS	1,536	OpenACC

- **Composite System SPP – <u>1.31 PF/s</u>**
  - **x86 SPP Contribution – 1.10 PF/s**
  - **Kepler SPP Contribution – 0.21 PF/s**

SPP Metric Definition for BW

- SPP metric is a geometric mean of per node performance rates for a suite of applications, each running in dedicated mode on a 1/5 to a 1/2 of the full number of compute nodes on the Blue Waters system, multiplied by the total number of compute nodes in the system.
- Each set of nodes of a given type is has the SPP contribution calculated independently and those sustained measures are summed to obtain the full system SPP value.
  - More precisely, for a given set of benchmark codes, the performance rate of the i-th code expressed in units of GFLOPS per node of type a, $P_{\alpha,i,}$, is calculated by dividing the reference FLOP count for that benchmark by the number of nodes of that type used to run the problem and by the total wall clock time for that run.
  - For a given number of nodes of a given type α, N_α, the contribution to the SSP from nodes of type a is the geometric mean of $P_{\alpha,i}$ over all applications, multiplied by N_α.
  - The total SSP is the sum of the contributions for each node type. For Blue Waters, α is two for the XE and XK node types. N_{XE} = 22640 and N_{xk}= 4224.
  - The number of GFLOPS per node was computed for the i-th benchmark running on the XE nodes, $P_{XE6,i}$ and the jth benchmark running on the XK nodes, $P_{XK7,j}$.
  - The contribution to the SSP for a given node type is the geometric mean of the $P_{\{XE6,XK7\},i \text{ or } j}$ values times the corresponding numbers of nodes of each type in the full system.
  - Thus, the total SSP of the XE/XK system is:
  - *SSP = Geometric Mean for all i* $(P_{XE6,i}) \times N_{XE6}$ + *Geometric Mean for all j* $(P_{XK7,j}) \times N_{XK7}$

Additional SPP Test Results

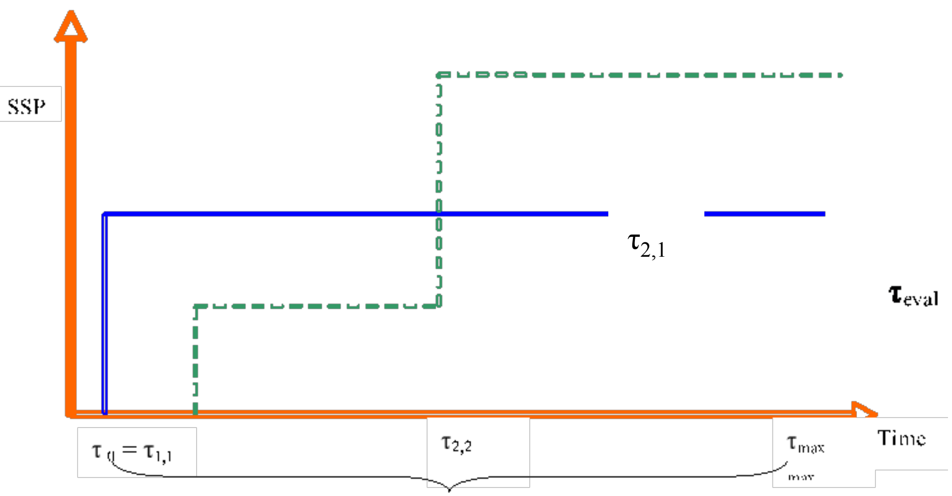
- Three Full Scale NSF applications – defined as problems
  - NAMD, MILC, P3DNS
- Full Scale SPP XE Codes
  - In addition to the NSF Petascale tests, 4 SPP tests ran above 1 PF using the full XE node section of the system
  - Two of the four ran above 1.2 PF
  - Scale ranges from 21,417 to 22,528 nodes
- SPP XK codes x86 to Kepler Speed ups
  - Four XK SPP codes all show a runtime improvement between 3.1-4.9x over x86 version running at same scale.
    - Scale ranges from 700 to 1,536 nodes
  - Three codes were CUDA implementation, 1 code was an OpenACC implementations

Example for SPP - NSF Workload
Blue Waters & Titan Computing Systems

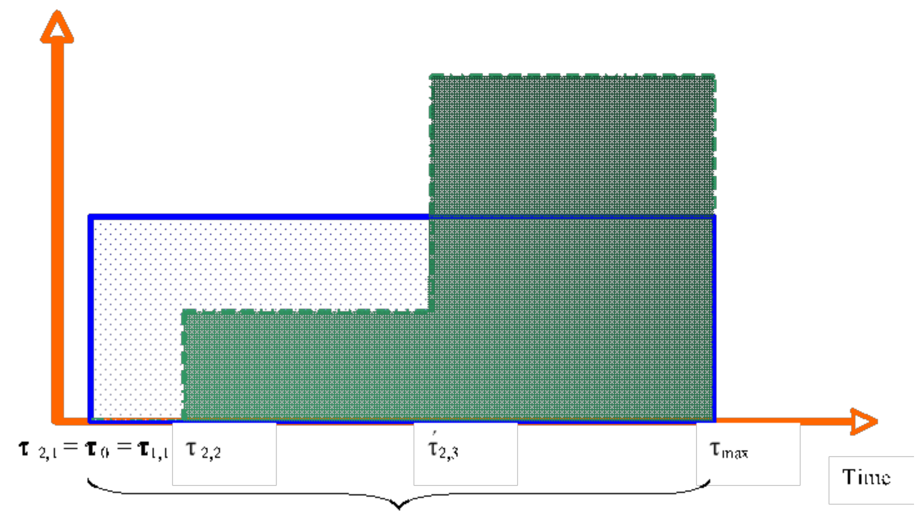
System Attribute (2012)	UIUC/NCSA *Blue Waters*	DOE/ORNL *Titan*
Vendor(s)	Cray/AMD/NVIDIA	Cray/AMD/NVIDIA
Processors	Interlagos 2.3 GHz/Kepler K20X	Interlagos 2.1 GHz /Kepler K20X
Total Peak Performance (PF/s)	*13.1*	*27.11*
Total Peak Performance (CPU/GPU)	7.6/5.5	2.63/24.5
Number of Nodes	27,648	19,200
Number of CPU Modules (8 cores/Module)	49,504	18,688
Number of GPU Chips	4,224	18,688
SPP Sustained Performance (PF/s)	**1.31**	**0.64**
Amount of CPU Memory (TB)	1,660	710
Interconnect	Gemini 3-D Torus	Gemini 3-D Torus
Dimensions	24x24x24	25x16x24
Amount of Usable On-line Disk Storage (PB)	26	>10
2013 upgrade		~40 shared
Sustained Disk Transfer (TB/sec)	1.2	0.245
2013 upgrade		~1 shared
Amount of Near-line/Archival Storage (Usable/Maximum) (PB)	300/400	125/250
2013 upgrade		150/300
Protection from single point of tape failure	Yes	No
Sustained Tape Transfer (GB/sec)	88	18

Information compiled by: Thomas Dunning and Bill Kramer, UIUC/NCSA and Buddy Bland ORNL

Simplified SSP Comparison Across Systems



The proposed deployment time and SSP of two systems



SSP performance chart after periods are aligned. For clarity $\tau'_{2,k}$ replaces $\tau_{2,k}$

τ_{eval}

Value and Price Performance

1. Determine the *Potency* of the system - how well will the system perform the expected work over some time period
   - Potency is the sum, over the specified time, of the product of a system's SSP and the time period of that SSP over some time period
     - Different SSPs for different periods
     - Different SSPs for different types of computation units (heterogeneous)

$$Potency_s = \sum_{k=1}^{K_s} SSP_{s,k} * \left[\min(\tau_{s,k+1}, \tau_{\max}) - \min(\tau_{s,k}, \tau_{\max})\right] \forall \tau_{s,k} \le \tau_{\max}$$

2. Determine the Cost of systems
   - Cost can be any resource units ($, Watts, space…) and with any complexity (Initial, TCO,…)

$$Cost_s = \sum_{k=1}^{K_s} \sum_{l=1}^{L_{s,k}} c_{s,k,l}$$

3. Determine the *Value* of the system
   - Value is the potency divide by a cost function

$$Value_s = \frac{Potency_s}{Cost_s}$$

4. If needed, compare the value of different system alternatives

SSP Method

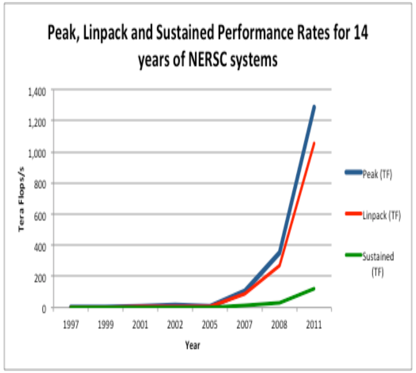
- Used in different forms
  - NSF - Blue Waters SPP 2011-2012
    - Codes and test cases at different scale close to release
  - DOE
    - NERSC – 1998-2017
      - https://www.nersc.gov/research-and-development/performance-and-monitoring-tools/sustained-system-performance-ssp-benchmark/
    - Los Alamos + NERSC + Sandia Trinity – 2015/2016
      - https://www.nersc.gov/systems/trinity-nersc-8-rfp/nersc-8-trinity-benchmarks/ssp/
  - Australian Meteorology Office 2000-present
  - DOD Modernization Office (ERDC, ARL, AFWL, NAVO, MHPCC) 2000-present
  - …..

COMMENTS ON THE TOP500 LIST AND ITS FUTURE REPLACEMENT

Stay True to the Mission
BW Focus on Sustained Performance

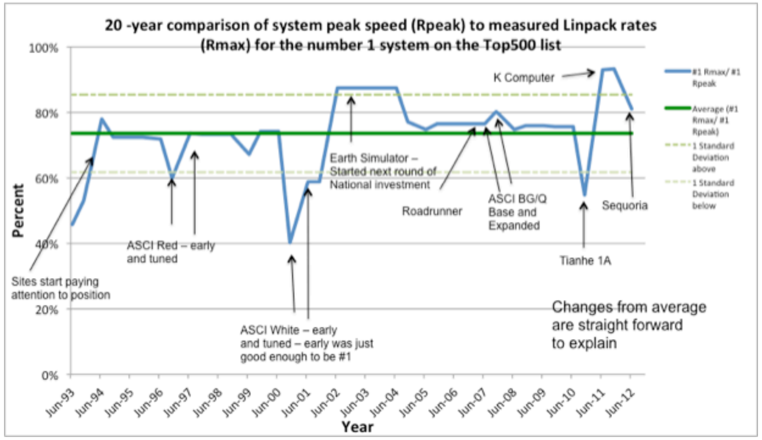
- **Blue Water's and NSF are focusing on *sustained* performance**
- **We intentionally choose not the list Blue Waters on the Top500 List.**
- *Sustained* is the computer's useful, consistent performance on a broad range of applications that scientists and engineers use every day.
  - Time to solution for a given amount of work is the important metric – not hardware Ops/s
    - Work is categorized regardless of scale ofr implementation so tests should approximate this as closely as possible
  - Sustained performance (and therefore tests) include time to read data and write the results
- NSF's call emphasized sustained performance, demonstrated on a collection of application benchmarks (application + problem set)
  - Not just simplistic metrics (e.g. HP Linpack)
  - Applications include both Petascale applications (effectively use the full machine, solving scalability problems for both compute and I/O) and applications that use a large fraction of the system
- Blue Waters project focus is on delivering sustained PetaFLOPS performance to all applications
  - Develop tools, techniques, samples, that exploit all parts of the system
  - Explore new tools, programming models, and libraries to help applications get the most from the system

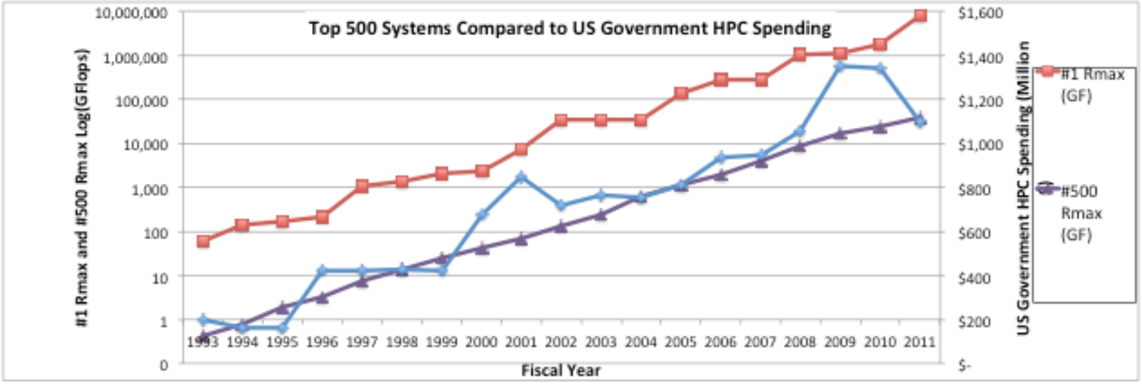
There is Life Beyond the Top500



Top500 values do not correlate with vs measured System Sustained Performance - 13 years of systems at NERSC show this trend

TOP500 is dominated by who has the most money to spend–not what system is the best.





There is Life Beyond the Top500

HPCwire

Since 1986 - Covering the Fastest Computers in the World and the People Who Run Them

Translation Disclaimer

Subscribe | Sign In

Search

Home | News | Topics | Sectors | Resources | Special Features | Market Watch | Events | Job Bank | About

Visit additional Tabor Communication Publications

ENTERPRISETECH

datanami

Digital Manufacturing report

HPCwire JAPAN

November 16, 2012

Blue Waters Opts Out of TOP500

Tiffany Trader

Page: 1 | 2 | 3

The NCSA Blue Waters system is one of the fastest supercomputers in the world, but it won't be appearing on the TOP500 list – nor will it be taking part in the HPC Challenge (HPCC) awards. While it's generally understood that there are an unknown number of classified and commercial systems that don't show up on the list, this is the first time an open science system has opted out in such a fashion.

According to the folks at the National Center for Supercomputing Applications (NCSA), there's a good reason for this. In the days leading up to the 24th annual Supercomputing Conference (SC12) in Salt Lake City, HPCwire spoke with Blue Waters Project Director Bill Kramer to find out what went into this decision.

Off the Wire | Most Read | Blogs

More Off the Wire...

Community is at a tipping point

- Wide spread understanding HPL is not an effective measure of system potential
  - Not representative of many applications – e.g. John McCalpin presented a study of correlation between applications and Linpack.
    - The correlation coefficient was 0.15.
    - The slope of the best-fit line was 0.1;
  - Conclusion – Doubling Linpack performance corresponded to 10% increase in application performance.
  - New HPCG benchmark proposed
  - Jack Dongarra and Mike Heroux
    - SANDIA REPORT SAND2013-4744 June 2013
    - https://software.sandia.gov/hpcg/doc/HPCG-Benchmark.pdf
    - New – Beta – Implementation – Just release
    - Michael A. Heroux, Jack Dongarra and Piotr Luszczek
      - SAND2013- 8752 October 2013
      - https://software.sandia.gov/hpcg/doc/HPCG-Specification.pdf
- Regardless of the measure – a problem remains of a single test combined with a non-peer reviewed list – see future slides
- Workshops may be formed to improve the HPCG approach

Evolutionary Changes that will make the Top500 more meaningful

1. *Require (estimated) cost data be posted for every system listed*
2. *Do not allow a system to be listed until it is fully accepted and performing its mission*
3. *Require a complete description for every system listed to give information about the investment balance*
4. *Move from weak scaling to strong scaling Linpack*
   - *Could use size classes as NPBs do to address large range of system scale*

Revolutionary Improvements - Align Our Community Metric To Best Practices In Benchmarking

- Combining the criteria from (Smith,1988) and (Lilja, 2000) provides the following list of good attributes for benchmarks
- Proportionality – a linear relationship between the metric used to estimate performance and the actual performance. In other words, if the metric increases by 20%, then the real performance of the system should be expected to increase by a similar proportion.
  - A scalar performance measure for a set of benchmarks expressed in units of time should be directly proportional to the total time consumed by the benchmarks.
  - A scalar performance measure for a set of benchmarks expressed as a rate should be inversely proportional to the total time consumed by the benchmarks.
- Reliability means if the metric shows System A is faster than System B, it would be expected that System A outperforms System B in a real workload represented by the metric.
- Consistency so that the definition of the metric is the same across all systems and configurations.
- Independence so the metric is not influenced by outside factors such as a vendor putting in special instructions that just impact the metric and not the workload.
- Ease of use so the metric can be used by more people.
- Repeatability meaning that running the test for the metric multiple times should produce close to the same result.

Align the community metric to best practices in benchmarking (cont)

David Bailey – 12 Ways to Fool the Masses – 1991

1. Quote only 32-bit performance results, not 64-bit results.
2. Present performance figures for an inner kernel, and then represent these figures as the performance of the entire application.
3. Quietly employ assembly code and other low-level language constructs.
4. Scale up the problem size with the number of processors, but omit any mention of this fact.
5. Quote performance results projected to a full system.
6. Compare your results against scalar, unoptimized code on conventional systems.
7. When direct run time comparisons are required, compare with an old code on an obsolete system.
8. If Mflop/s rates must be quoted, base the operation count on the parallel implementation, not on the best sequential implementation.
9. Quote performance in terms of processor utilization, parallel speedups or Mflop/s per dollar.
10. Mutilate the algorithm used in the parallel implementation to match the architecture.
11. Measure parallel run times on a dedicated system, but measure conventional run times in a busy environment.
12. If all else fails, show pretty pictures and animated videos, and don't talk about performance.

David's Update for 2011

A. Cite performance rates for a run with only one processor core active in a shared-memory multi-core node. For example, cite performance on 1024 cores, even though the code was run on 1024 nodes, wasting 15 out of 16 cores on each node.

B. Cite performance rates only for a core algorithms (such as FFT or LU decomposition), even though the paper mentions one or more full-scale applications that were done on the system.

C. List only the best performance figure in the paper, even though the run was made numerous times.

D. Employ special hardware, operating system or compiler settings that are not appropriate for real-world usage.

E. Define "scalability" as successful execution on a large number of CPUs, regardless of performance.

- http://crd.lbl.gov/~dhbailey/dhbtalks/dhb-12ways.pdf

Create A New, Meaningful Suite Of Benchmarks

- Many benchmark suites that were held in high regard (Livermore Loops, NPBs, SPEC) over time are suites of pseudo and/or full applications.
- While the best case for any benchmark is to be a statistically representative sample of real workload, in realty, this is not possible for community tests.
- **SERPOP** (Sample Estimation of Relative Performance of Programs) method is best suited for a generalized test.
  - A sample of a workload is selected to represent a workload. However, the sample is not random and cannot be considered a statistical sample.
  - SERPOP methods occur frequently in performance analysis and reflect very meaning measures that span individual communities.
  - In SERPOP analysis, the workload is related to SERPOP tests, but does not indicate the frequency of usage or other characteristics of any individual workload.
- Many common benchmark suites—including SPEC, TCP and NPB, as well as many acquisition test suites—are SERPOP.

Mashey, John R. "War of the Benchmark Means: Time for a Truce." ACM SIGARCH Computer Architecture News (Association for Computing Machinery) 32, no. 4 (September 2004)

Revolutionary Improvements – Aggregate Multiple Metrics Into A Single Value

- Multiple Benchmarks – not just one
  - Will lose its uniqueness over time
- Compositing Function is necessary
  - SPP
  - Decathlon
  - Flexibly defined sets of criteria – *HPC Sabernetics*

Decathlon Measuring Method

- Proposed by Authors: Satoshi Matsuoka (Tokyo Tech./NII/Riken AICS), William Kramer (NCSA), Daisuke Takahashi (University of Tsukuba)
- 10 tests
  - 10 individual event winners
  - 1 overall winner
- Goal is each test has equal influence in overall best score
- Example - The 2001 IAAF points tables use the following formulae (
  - Points = INT(A(B — P)^C) for tests where faster time produces a better score
  - Points = INT(A(P — B)^C) for tests where greater distance or height produces a better score)
- A, B and C are parameters that vary by discipline and a set according normalized performance aspects of the period
- P is the performance by the test, measured in time or amounts

The HPC Decathlon Assessment Measure Desired Characteristics

- Proportionality
- Scalability
- Reasonable Execution Time
- Reliability
- Consistency
- Independence
- Repeatability
- Verifiabiliy
- Ease of use
- Succinctness of the Rules
- Algorithmic Specification and not Code
- Availability of Efficient, Parallel, and Scalable Reference Implementation
- Single Value Metric Result

- Orthogonality.
- Community agreement and participation
- Maintainability
- Longevity: that allows comparisons of machines of current and past generations and properties of systems to come
- Governance to be able to fairly and responsibly judge the rules applicability
- Composibility
  - should be technically meaningful
  - agreeable by the community,
  - should be changeable in a documented fashion to derive a metric favoring a particular type of workload can be synthesized for respective domains, changes over time, etc.

Reaching Community Consensus

- Determine Key issues – pro and cons, ROI, etc.
  - Full Applications, Mini-applications, kernels
  - Scales and Scaling
    - Sizes
    - Weak vs Strong
    - Explainability
  - Distributions
- Community Ownership
  - Lists must be transparent and community managed
    - Peer Review
    - Professional Society Endorsement
    - Conflict of Interest Avoidance
  - Application Domain Relationships
- Revitialized every 5 years.

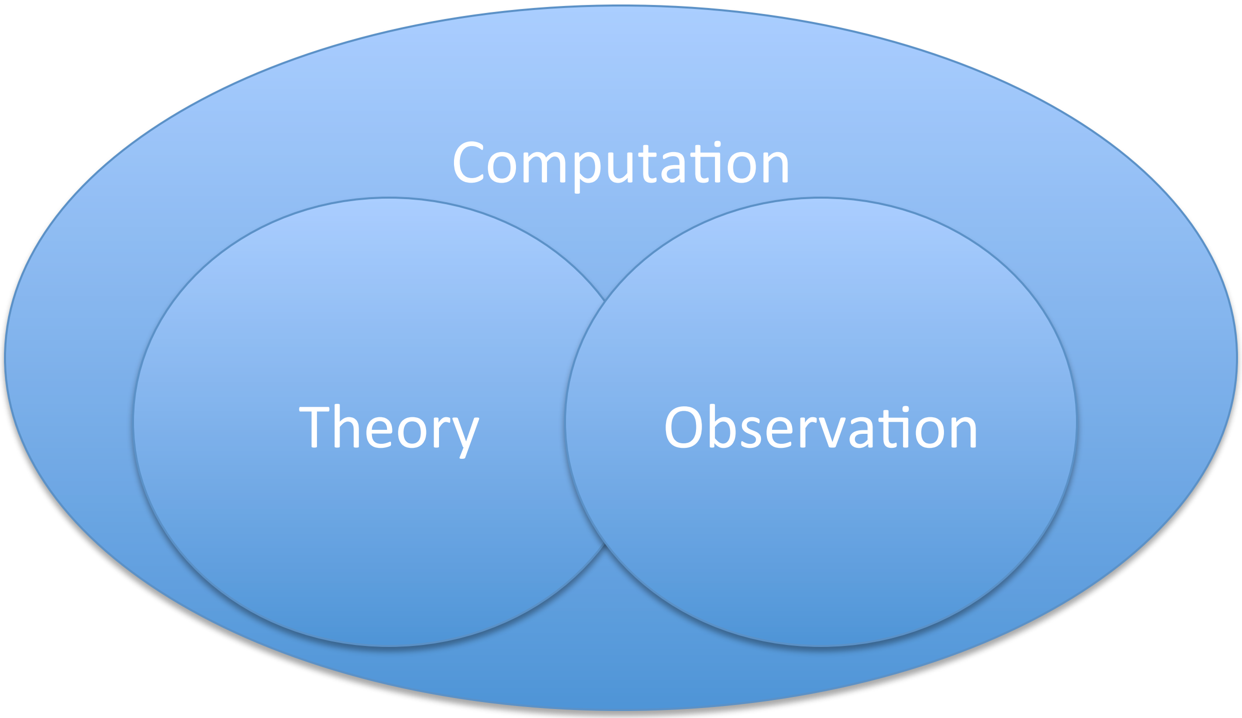
Summary – Let US Be Guided by What Users Want and Need From @Scale Systems

- Performance -
  - How fast will a system process work if everything is working really well
  - Establishes a system's potential to do productive work
- Effectiveness
  - The likelihood users can get the system to do their work when they need it
- Reliability
  - The likelihood the system is available to do the work
- Consistency
  - How often will the system process the same or similar work correctly and in the length of same time
- Usability
  - How easy is it for users to get the system to process their work as fast as possible

We need good PERCU metrics to assess complete systems for the science impact view point

Cost and other "business factors" are also part of a decision making

Questions



Acknowledgements

This work is part of the Blue Waters sustained-petascale computing project, which is supported by the National Science Foundation (award number OCI 07-25070) and the state of Illinois. Blue Waters is a joint effort of the University of Illinois at Urbana-Champaign, its National Center for Supercomputing Applications, Cray, and the Great Lakes Consortium for Petascale Computation.

The work described is achievable through the efforts of the many other on different teams.