BLUE WATERS

SUSTAINED PETASCALE COMPUTING

Blue Waters Redone

Un super système pour résoudre de super défis

William Kramer

National Center for Supercomputing Applications, Department of Chemistry, Department of Computer Science, and Department of Electrical & Computer Engineering

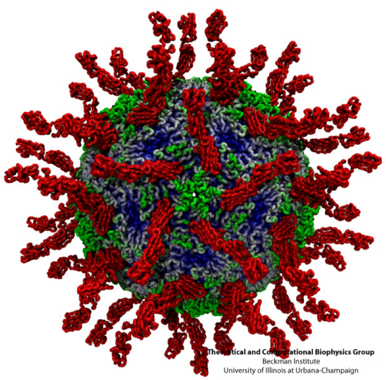
GREAT LAKES CONSORTIUM

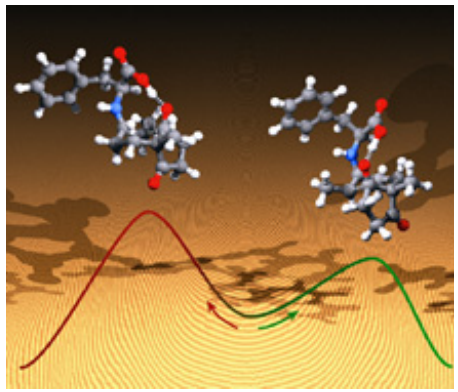
FOR PETASCALE COMPUTATION

Science & Engineering on Blue Waters

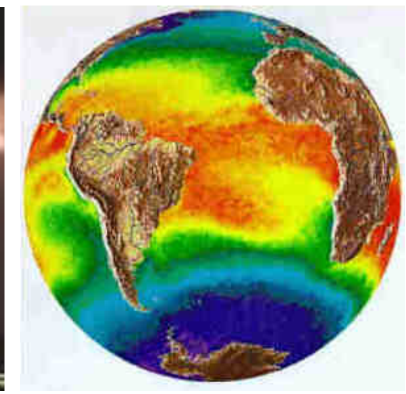
Blue Waters will enable advances in a broad range of science and engineering disciplines. Examples include:

Molecular Science



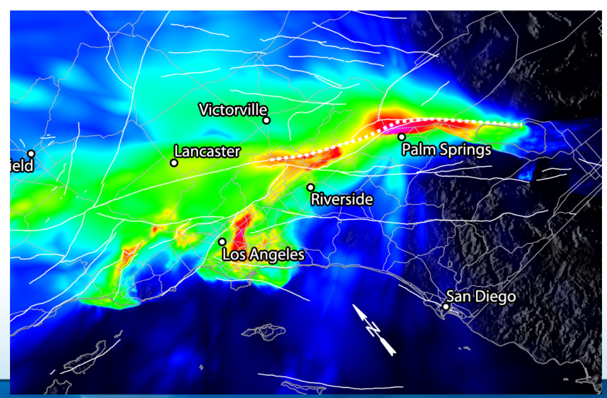


Weather & Climate Forecasting

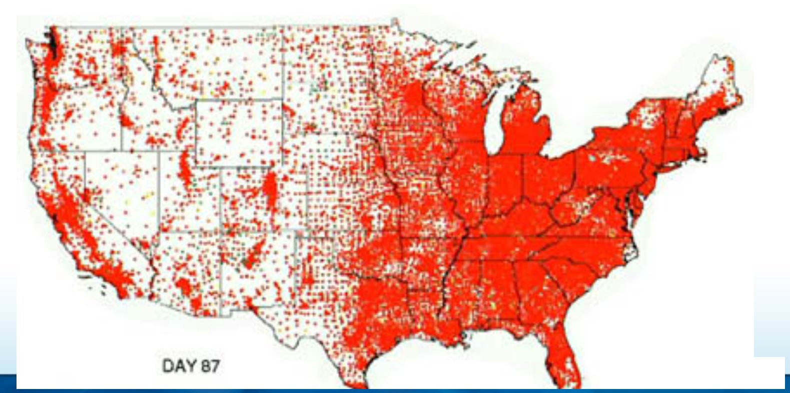


Astronomy

Earth Science



Health



NCSA Has Completed a Grand Challenge

- In August, IBM decided to terminate their contract to deliver the base Blue Waters system
- NSF asked NCSA to propose a change of technology and to adjust the the Project Execution Plan
  - Same expectations and goals
  - Same or better schedule
  - Same or lower budget
  - Less Risk
- In September, NCSA proposed a revised plan to NSF and a Peer Review Panel. - 27 Days!!
  - Complete understanding of applications was key to being able to do this
- NSF approved the plan on November 10, 2011
- I am please to present our new plan to you today
  - All parameters of the project will be met with the new system

Blue Waters Project Components

Petascale Education, Industry and Outreach

Petascale Applications
(Computing Resource Allocations)

Petascale Application Collaboration Team Support

Outstanding User and Production Support
WAN connections, Consulting, System Management, Security, Operations, …

Great Lakes Consortium for Petascale Computing

Value added Software – Collaborations

Value added hardware and software (HPSS, archive HW, LAN/WAN Networking, etc.)

Blue Waters Base System – Processors, Memory, Interconnect, On-line Storage, System Software, Programming Environment

Petascale Computing Facility

Sustained Petascale Performance

Sustained Petascale Performance

Large Memory Integrated Storage at Terabyte/sec

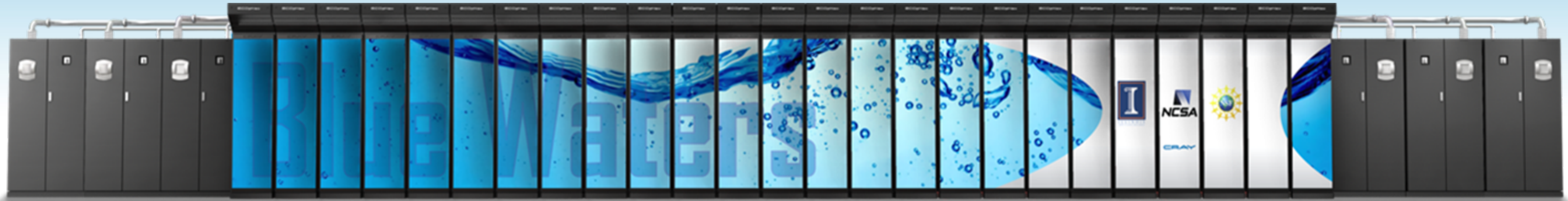
Production Science at Full Scale for All Areas of Science

A Transitional Platform to Exascale

NCSA

ILLINOIS
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

NSF

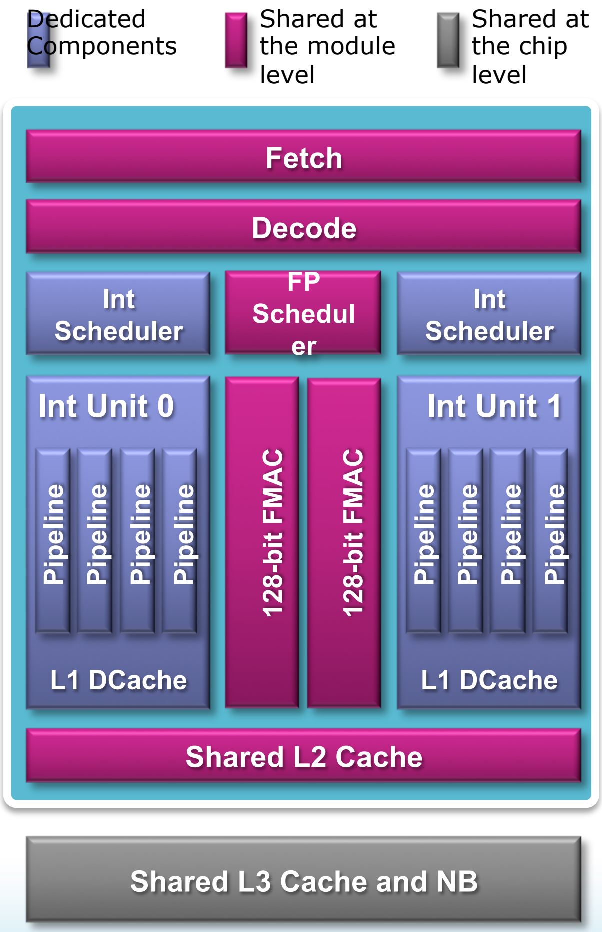


Cray System & Storage cabinets:	• >300
Compute nodes:	• >25,000
Usable Storage Bandwidth:	• >1 TB/s
System Memory:	• >1.5 Petabytes
Memory per core module:	• 4 GB
Gemin Interconnect Topology:	• 3D Torus
Usable Storage:	• >25 Petabytes
Peak performance:	• >11.5 Petaflops
Number of AMD processors:	• >49,000
Number of AMD x86 core module:	• >380,000
Number of NVIDIA GPUs:	• >3,000

NSF NCSA
ILLINOIS
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

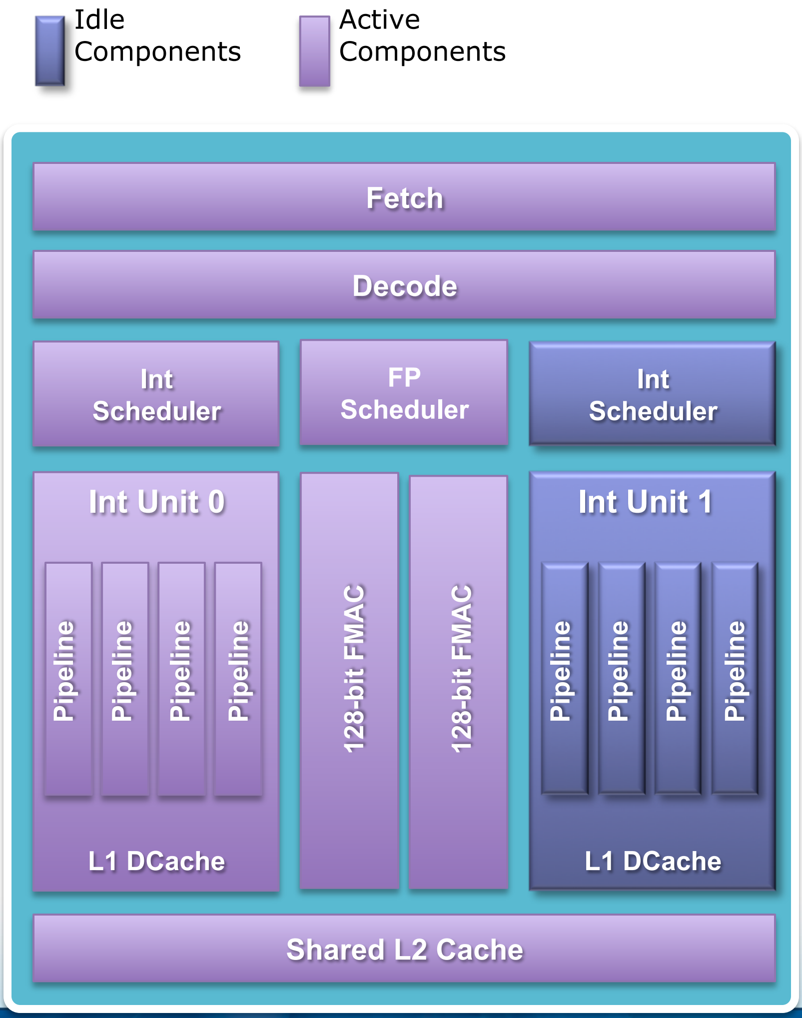
AMD Interlagos Processor Architecture

- Interlagos is composed of a Bulldozer core "modules"
- A core module has shared and dedicated components
- There are two independent integer units and a *shared*, 256-bit FP resource
- A single Integer unit can make use of the entire FP resource with 256-bit AVX instructions
- This architecture is very flexible, and can be applied effectively to a variety of workloads and problems



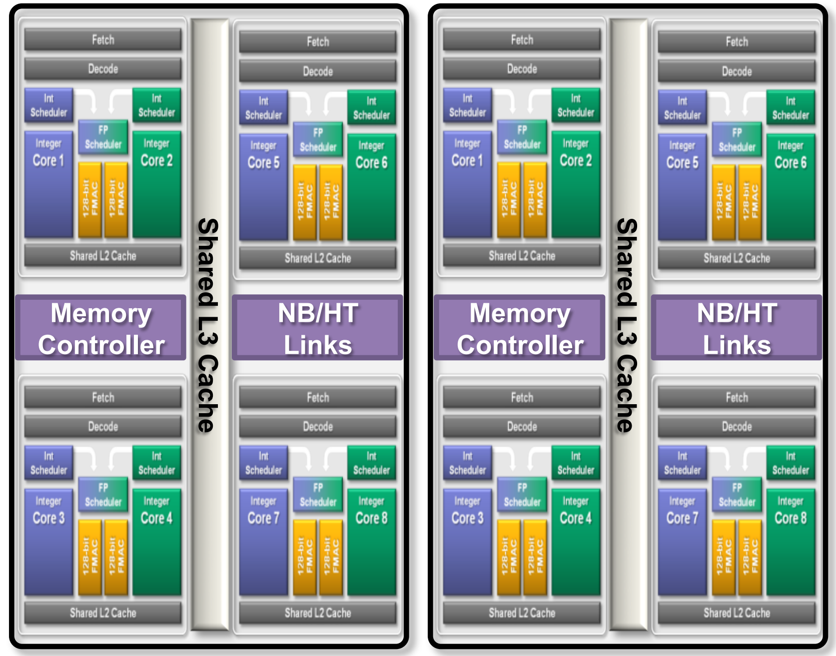
Defining a Core - AMD Wide AVX mode

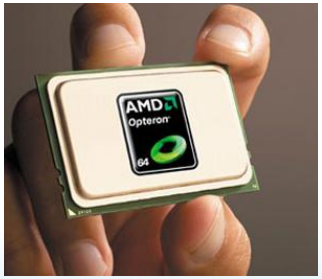
- In this mode, only one integer core is used per core pair
  - Most common mode for PRAC applications
    - Code is Floating Point dominated and makes use of AVX instructions
    - Code needs more memory per MPI rank
- Implications
  - This core has *exclusive* access to the 256-bit FP unit and is capable of 8 FP results per clock cycle
  - The core has twice the memory capacity and memory bandwidth in this mode
  - The L2 cache is effectively twice as large
  - The peak of the chip is not reduced



Interlagos Processor

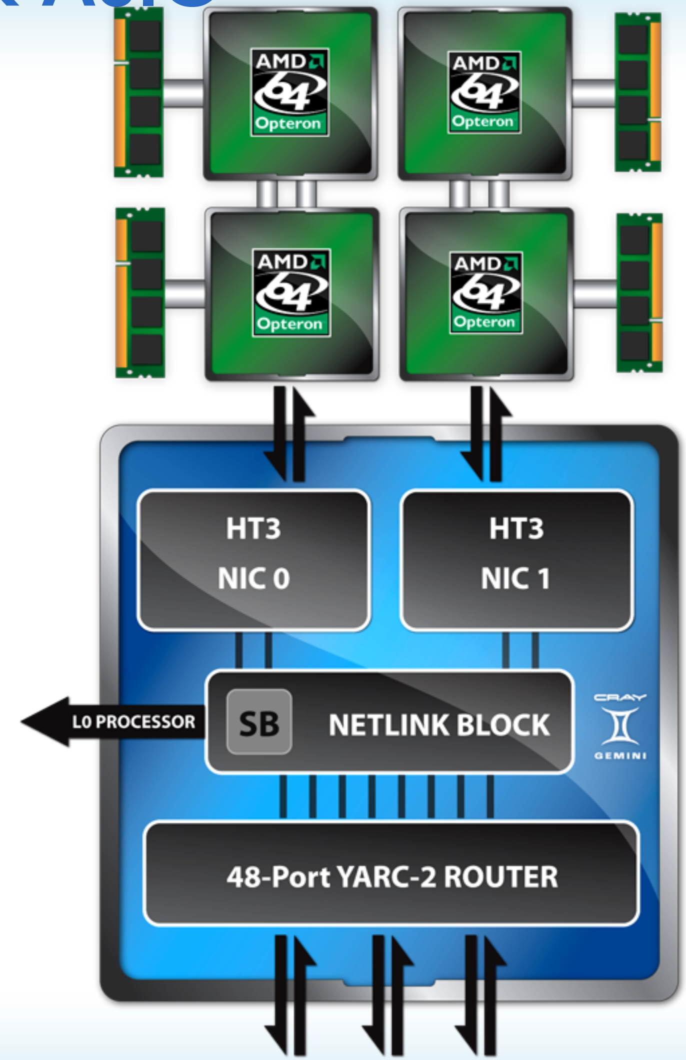
- Each processor die is composed of 4 core modules
- The 4 core modules share a memory controller and 8 MB L3 data cache
- Two die are packaged on a multi-chip module to form a G34-socket Interlagos processor
- Package contains
  - 8 core modules
  - 16 MB L3 Cache
  - 4 DDR3 1600 memory channels



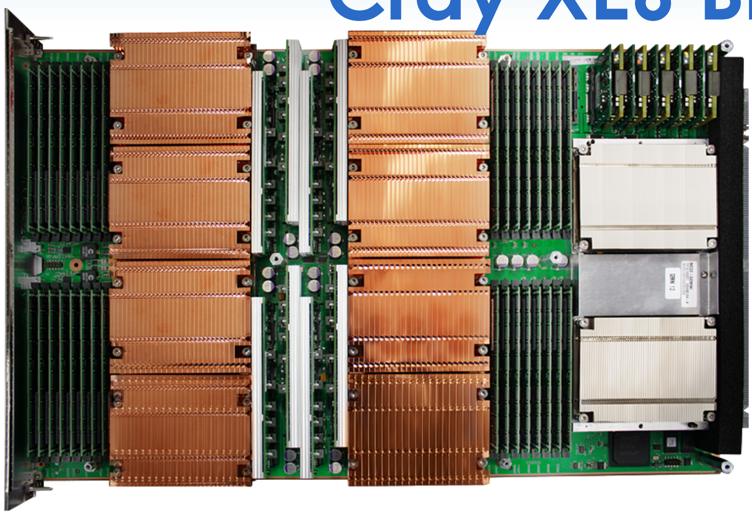


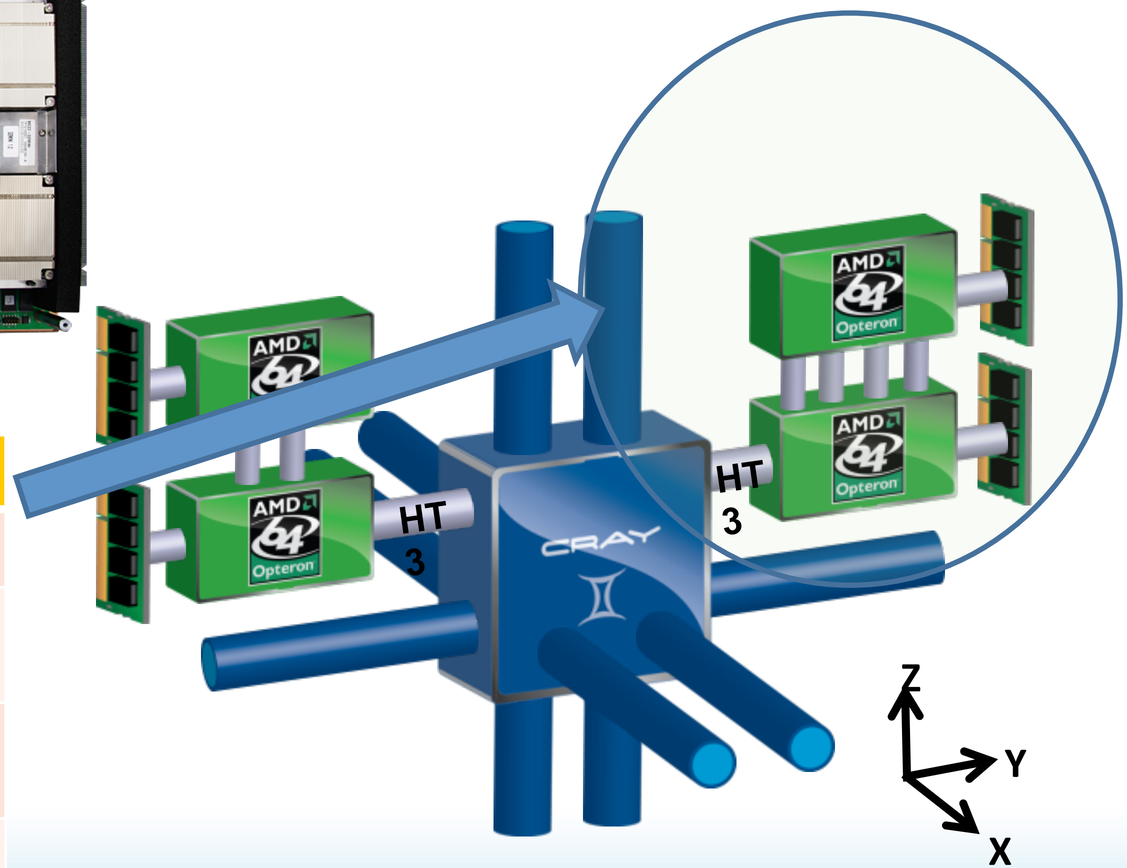
Cray Gemini Network ASIC

- MPI Support
  - 1.2 μs latency
  - >10M independent messages/sec/NIC
  - Fast Memory Access for small messages
  - Block Transfer Engine for large messages
- Advanced Synchronization and Communication Features
  - Efficient support for UPC, CAF, One-sided MPI and Global Arrays
  - Atomic memory operations
  - Pipelined global loads and stores
  - ~25M random Puts/sec/NIC
  - ~65M indexed Puts/sec/NIC
- Resiliency support
  - Extensive error detection and correction
  - Auto link degrade
  - Warm-swap capability
  - Resilient MPI protocols



Cray XE6 Blade and Node

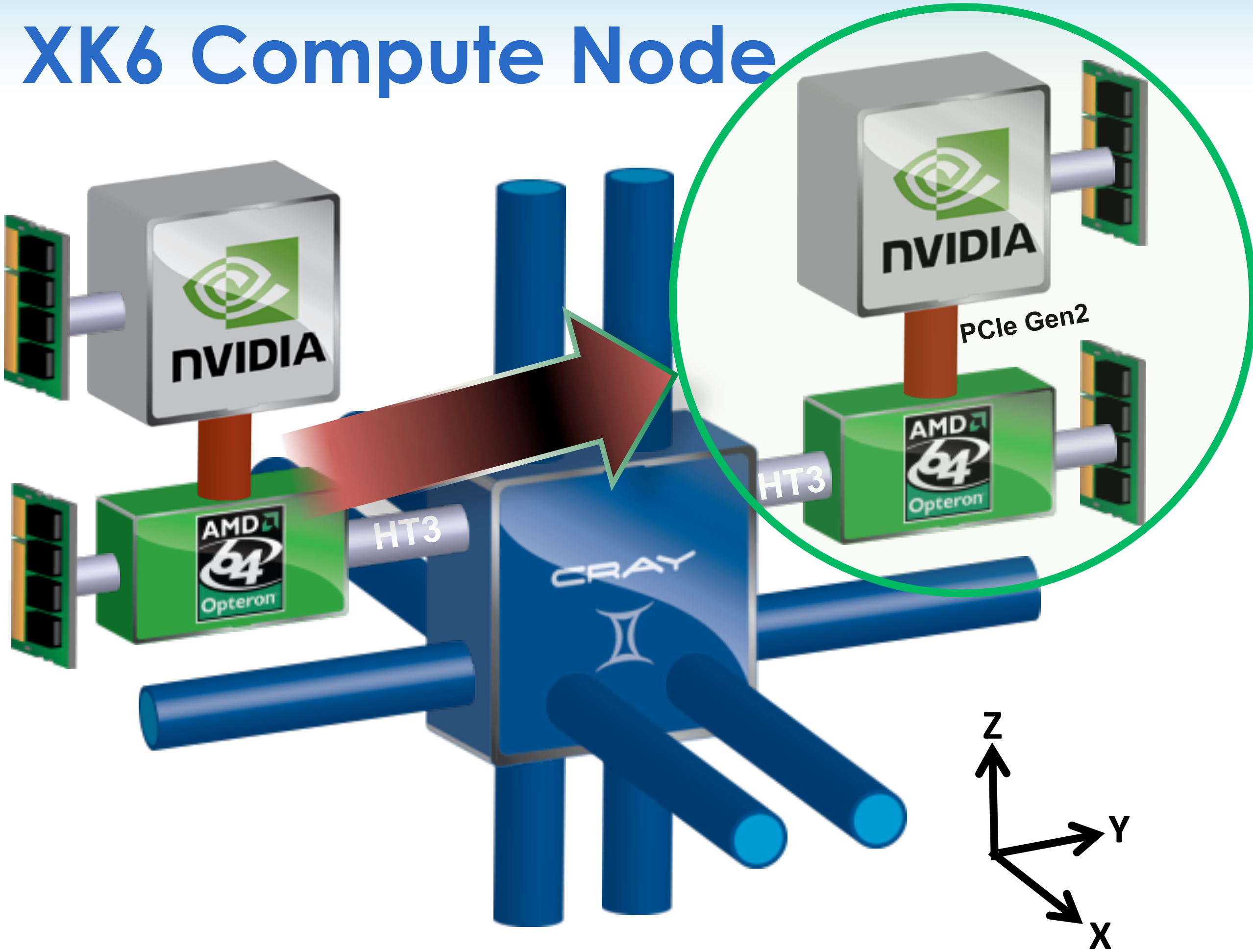




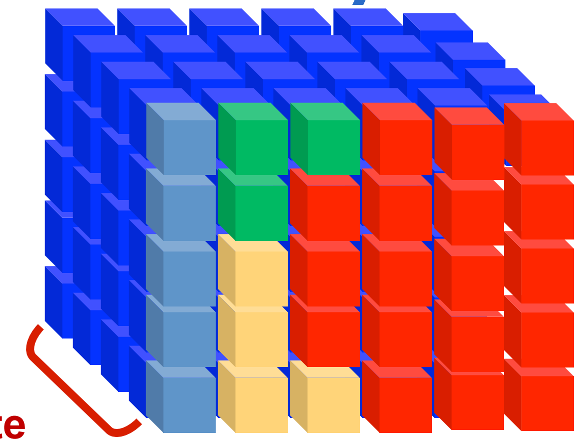
Node Characteristics	
Number of Cores	8 Core modules
Peak Performance	313 Gflops/sec
Memory Size	4 GB per core-m 64 GB per node
Memory Bandwidth (Peak)	102.4 GB/sec

Cray XK6 Compute Node

XK6 Compute Node Characteristics
AMD Series 6200 (Interlagos)
NVIDIA Kepler
Host Memory 32GB 1600 MHz DDR3
NVIDIA Tesla X2090 Memory 6GB GDDR5 capacity
Gemini High Speed Interconnect
Upgradeable to future GPUs



Cray Linux Environment



Compute Nodes

Service Nodes

Specialized Linux nodes

Compute PE

Login PE

Network PE

System PE

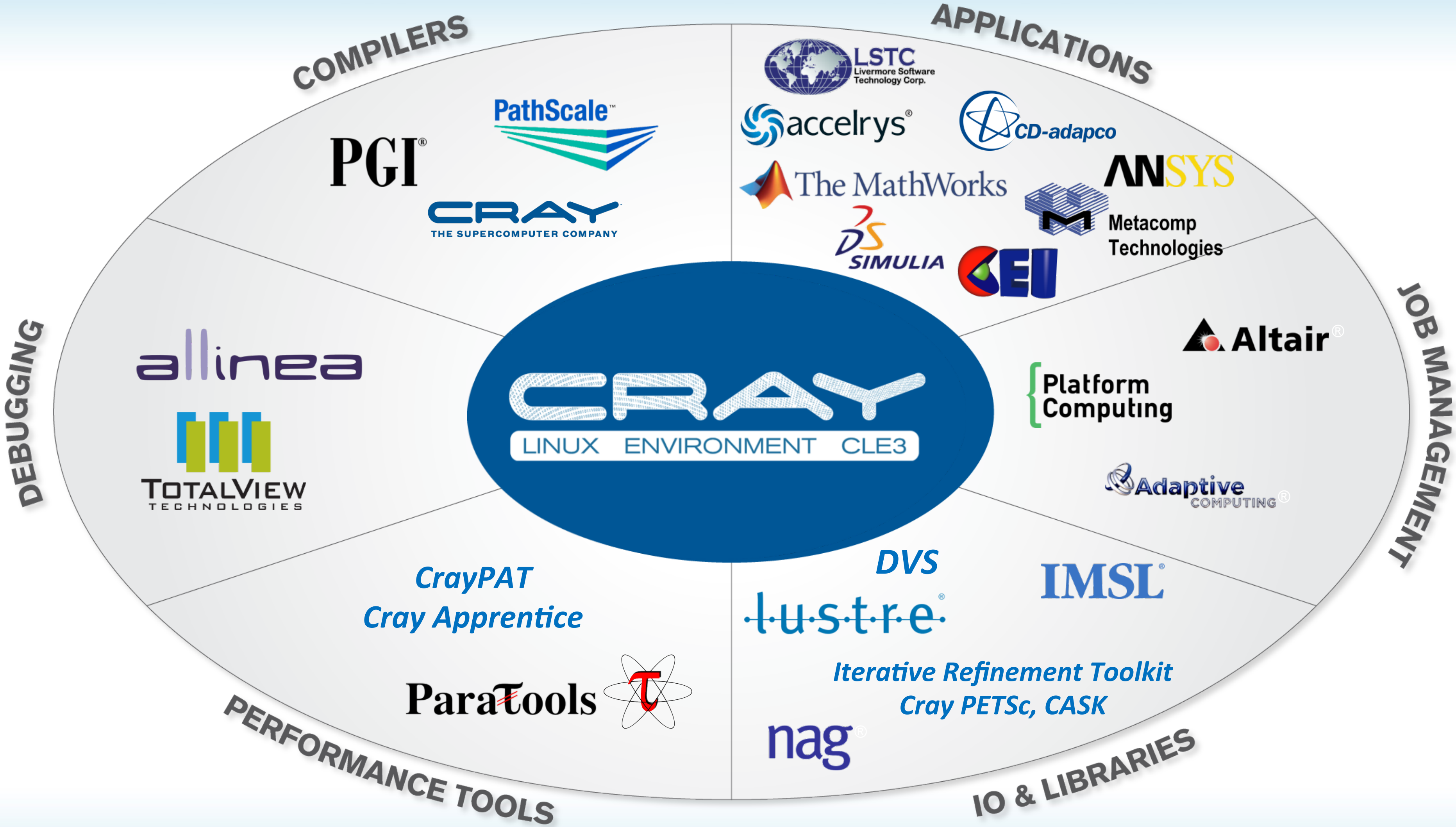
I/O PE

- Streamlined Linux distribution on Compute PEs, full distribution on Service PEs
- Ability to dynamically configure nodes to trade off services and scalability
- Software Architecture
  - Reduces OS "Jitter"
  - Enables reproducible runtimes
- Large machines boot in under 30 minutes, including filesystem
- Job Launch time is a couple seconds on 1000s of PEs

Cray's focus is on all aspects of scalability (noise reduction, bottleneck removal, resilience, flexibility, etc.)

Significant R&D continues to be spent

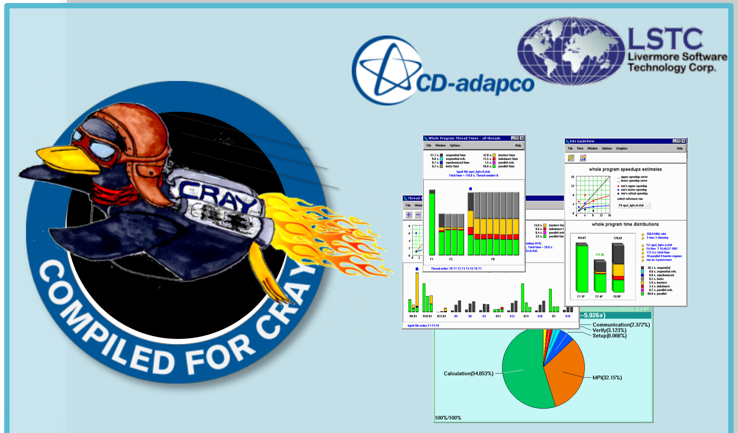
Cray Software Ecosystem



CLE3, An Adaptive Linux OS designed specifically for HPC

ESM – *Extreme Scalability Mode*

- No compromise *scalability*
- Low-Noise Kernel for scalability
- Native Comm. & Optimized MPI
- Application-specific performance tuning and scaling



CCM –*Cluster Compatibility Mode*

- No compromise *compatibility*
- Fully standard x86/Linux
- Standardized Communication Layer
- Out-of-the-box ISV Installation
- ISV applications simply install and run

CLE3 run mode is set by the user on a job-by-job basis to provide full flexibility

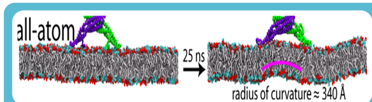
Approach to Accelerator Programming

- Most important hurdle for widespread adoption of accelerated computing is programming difficulty
- Need a single programming model that is portable across machine types, and also forward scalable in time
  - Portable expression of heterogeneity and multi-level parallelism
  - Programming model and optimization should not be significantly difference for “accelerated” nodes and multi-core x86 processors
  - *Allow users to maintain a single code base*
- Approach:
  - Support 3rd party GPU/Accelerator tools and languages for compatibility
    - CUDA and OpenCL
    - PGI Fortran compiler
    - Allinea, TotalView, etc.
  - Optimized scientific libraries for Accelerator
  - Cray compiler with native support for Accelerator
    - C, C++ and Fortran; MPI and OpenMP
    - Directives based on OpenMP for identifying parallel work
  - Whole program scoping tools

Focus on Sustained Performance

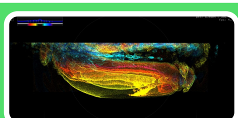
- **Blue Water's and NSF are focusing on *sustained* performance in a way few have been before.**
- *Sustained* is the computer's performance on a broad range of applications that scientists and engineers use every day.
  - Time to solution is the metric – not Ops/s
  - Tests include time to read data 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 fraction of the system
    - Metric is the time to solution
- 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

More than 25 PRAC science teams
12 distinct research fields
selected to run on the new Blue Waters
Expect ~10 more major teams

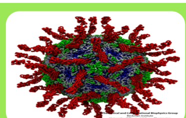


Nanotechnology

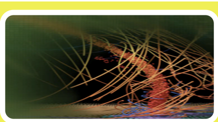
Astronomy



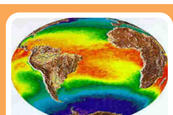
Earthquakes and the damage they cause



Viruses entering cells



Severe storms



Climate change

Science Area	Number of Teams	Codes	Structured Grids	Unstructured Grids	Dense Matrix	Sparse Matrix	N-Body	Monte Carlo	FFT	Significan t I/O
Climate and Weather	3	CESM, GCRM, CM1, HOMME	X	X		X		X		
Plasmas/ Magnetosphere	2	H3D(M), OSIRIS, Magtail/UPIC	X				X		X	X
Stellar Atmospheres and Supernovae	2	PPM, MAESTRO, CASTRO, SEDONA	X			X		X		X
Cosmology	2	Enzo, pGADGET	X			X	X			
Combustion/ Turbulence	1	PSDNS	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, MILD, USQCD	X		X	X	X		X	
Social Networks	1	EPISIMDEMICS								
Evolution	1	Eve								
Computer Science	1			X	X	X			X	X

Sustained Petascale Performance Applications

- In addition to all of the NSF RPF Petascale benchmarks, NCSA is using the SPP to assess sustained performance
  - NAMD – molecular dynamics
  - MILC – lattice QCD
  - PPM – turbulent stellar atmospheres
  - QMCPACK – materials science
  - H3D(M) – Earth's magnetosphere and plasma physics
  - WRF – weather and climate
  - SPECFEM3D– geodynamics
  - NWChem– chemistry
- The input, problem sizes, included physics, and I/O performed by each benchmark will be comparable to the simulations proposed by the corresponding science team for scientific discovery.
- Each benchmark will be sized to use one-fifth to one-half of the number of nodes in the full system.
  - Multiple of the applications will be >1 PF sustained a full size
- GPUs will quantitatively increase the SPP

INTELLECTUAL CHALLENGES

Extreme Scale Intellectual Challenges

- Challenges that will be faced by the Science Teams include:
  - Scaling applications to large processor counts.
  - Effective using of many core and accelerator components.
  - Using of both general purpose and accelerated nodes in single application.
  - Application based resiliency
- NCSA establishing a focused effort in **Extreme-scale Scientific Computing Applications (ESCA)** to work directly with the Science Teams to enable them to take full advantage of the extraordinary capabilities of extreme scale systems.
- This plan includes participation from the broader scientific computing community.
- Other challenges (productivity, cost of ownership, etc.) also factors

Extreme Scalability

- Developing better process-to-node mapping using for graph analysis to determine MPI behavior and usage patterns.
- Topology Awareness in Applications and in Resource Management
- Improve use of the available bandwidth (MPI implementations, lower level communication, etc.).
  - For example, the DNS analysis assumes that only a relatively low fraction of available bandwidth will be achieved – can this be improved? Most likely.
- Considering alternative programming models that improve efficiency of calculations (e.g., CAF one-sided access can reduce memory bandwidth requirements).
- UI Staff and other NCSA collaborators and partners, working closely with the Science Teams, will explore the above approaches.
  - Most of the above approaches will provide an increase of a factor of 2-6 in effective bandwidth.

Many Core and Accelerated Units

- Help the science teams to make more effective use of GPUs consists of two major components.
  - Introduce compiler and library capabilities into the science team workflow to significantly reduce the programming effort and impact on code maintainability. Examples:
    - Compiler based directives
    - GMAC - a library that provides global shared memory and automates data transfer/coherence between the CPUs and the GPUs in a node
    - DL is a compiler-based memory layout transformation tool that uses a combination of compiler and runtime support to easy the task of adjusting memory layout to satisfy conflicting needs between the CPU and the GPU
    - TC is a compiler based tool for thread coarsening and data tiling.
  - Provide expert support to the science teams through hand-on workshops, courses, and individualized collaboration programs.

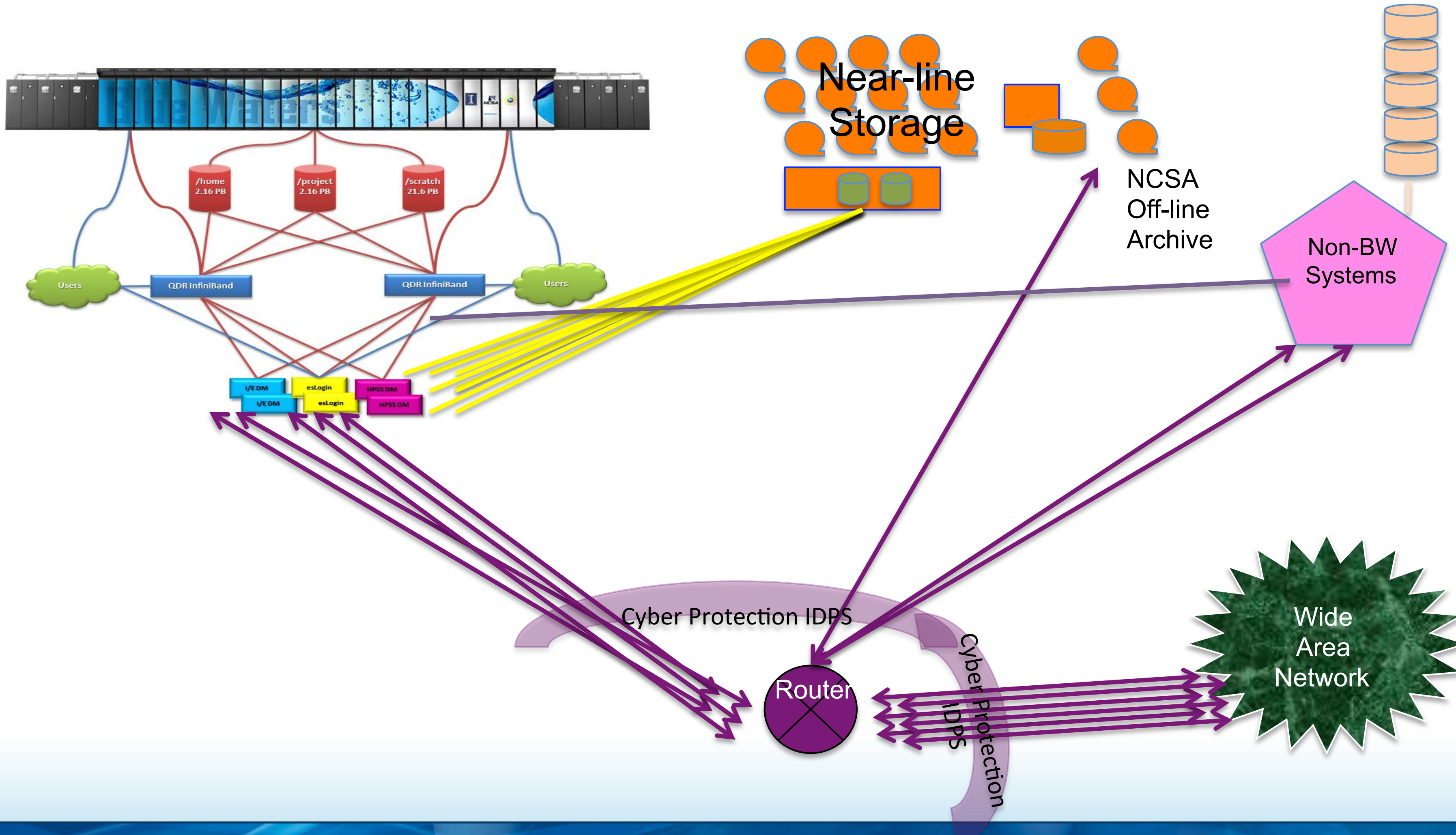
Using Of Both General Purpose And Accelerated Nodes In Single Application.

- For multi-physics applications that provide a natural decomposition into modules is to deploy the most appropriate module(s) different computational units.
  - NCSA will assist in identifying appropriate modules, and in the mechanics of heterogeneous partitioning.
- For applications, such as NAMD, Episimdemics, and possibly ENZO, that use the Charm++ adaptive runtime system, heterogeneity can be handled without significant changes to the application itself.
- MPI applications may be able to leverage the Charm++ runtime system by converting them to adaptive MPI (AMPI) first - EVE and CM1.
- Some applications naturally involve assigning multiple blocks to individual processors include multiblock codes (typically in fluid dynamics), and the codes based on structured adaptive mesh refinement.
  - The application-level load balancing algorithms can be modified to deal with the performance heterogeneity created by the mix of nodes. The NCSA/Illinois staff will assist in such modification.
- Some applications use frameworks for accomplishing their load-balancing (Zoltan, UNITAH, Paramesh and Chombo, etc.) that already address the issue of differential performance of different processors.

Application Based Resiliency

- Multiple layers of Software and Hardware have to coordinate information and reaction
- Analysis and understanding is needed before action
- Correct and actionable messages need to flow up and down the stack to the applications so they can take the proper action with correct information
- Applications need to understand circumstances and take action
- Flexible resource provisioning needed in real time
- Interaction with other constraints so sub-optimization does not adversely impact overall system optimization

Blue Waters's System Architecture



Summary/Questions

- Blue Waters technology is now determined and will be in early use 2012
- It will be the most significant general purpose computational capability in the US for the diverse science
- BW will be a total capability rivaling any others
- BW will be a exceptional transitional platform to help the NSF computational community to move to Exascale like architectures
- BW will probably have the largest amount of memory of any system in it generation
- BW will have one of the most robust I/O sub-systems of its generation
- *Co-design* works <u>but</u> may take a project on to very unexpected paths

Acknowledgements

This research 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, IBM, and the Great Lakes Consortium for Petascale Computation.

The work described is only achievable through the efforts of the Blue Waters Project.

References

- David H. Bailey, "Twelve Ways to Fool the Masses When Giving Performance Results on Parallel Computers," *Supercomputing Review, Aug 1991, pg. 54-55. - http://crd.lbl.gov/~dhbailey/dhbpapers/twelve-ways.pdf*
- David H. Bailey, et. al, "The NAS Parallel Benchmarks," *International Journal of Supercomputer Applications, vol. 5, no. 3 (Fall 1991), pg. 66-73.*
- http://crd.lbl.gov/~dhbailey/dhbtalks/dhb-12ways.pdfMashey, 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).
- McMahon, F. *The Livermore Fortran Kernels: A computer test of numerical performance range.* Technical Report , Lawrence Livermore National Laboratory, Livermore, CA: University of California,, 1986.
- Lilja, David. *Measuring Computer Performance: A Practitioner's Guide.* Cambridge University Press, 2000.
- Kramer, William and Clint Ryan. "Performance Variability on Highly Parallel Architectures." *International Conference on Computational Science 2003.* Melbourne Australia and St. Petersburg Russia, 2003.
- John, Lizy Kurian. "More on finding a Single Number to Indicate Overall Performance of a Benchmark Suite,." *ACM SIGARCH Computer Architecture News* (Association for Computing Machinery) 31, no. 1 (March 2004).
- John, Lizy Kurian, and Lieven Kurian, . *Performance Evaluation and Benchmarking.* 6000 Broken Sound Parkway, NW, Suite 300, Boca Raton,, FL, 33487-2742: CRC Press Taylor and Francis Group, 2006.
- Hockney, Roger W. "The Science of Computer Benchmarking." *SIAM.* Society for Industrial and Applied Mathematics, 1996.
- "High Performance Technology Insertion 2006 (TI-06) ." *DOD Modernization Program.* 2005. http://www.fbodaily.com/archive/2005/05-May/08-May-2005/FBO-00802613.htm (accessed 2005).
- Flemming, Philip J., and John J. Wallace. "How not to lie with statistics: the correct way to summarize benchmark results." *Communications of the ACM* (Association for Computing Machinary) 29, no. 3 (March 1986).
- Dongarra, Jack. *Performance of Various Computers Using Standard Linear Equations Software.* Computer Science , University of Tennessee, Knoxville TN, 37996: University of Tennessee, 1985.

References

- Culler, David E., and Jaswinder Pal Singh. *Parallel Computer Architecture: A Hardware/Software Approach.* First. Edited by Denise P.M. Penrose. San Francisco, CA: Morgan Kaufmann Publishers, Inc., 1999.
- Patterson, David, and John Hennessey. *Computer Architecture – A Quantitative Approach.* Second. Burlington, MA: Morgan Kaufmann Publishers, 1996.
- Carrington, Laura, M. Laurenzano, Allan Snavely, Roy Campbell, and Larry Davis. "How Well Can Simple Metrics Represent the Performance of HPC Applications?" *SC 05 - The High Performance Computing, Storage, Networking and Analysis Conference 2005.* Seattle, WA: Association of Computing Machinary (ACM), 2005.
- Bucher, Ingrid, and Joanne Martin. *Methodology for Characterizing a Scientific Workload.* Technical Report, Los Alamos, NM 87545: Los Alamos National Laboratory, 1982.
- "SSP Project Page." *NERSC.* 2008. http://www.nersc.gov/projects/ssp.php.
- *Streams Benchmark.* http://www.cs.virginia.edu/stream/
- Simon, Horst, and Erich Strohmaier. *Statistical Analysis of NAS Parallel Benchmarks and LINPACK Results.* Vol. 919, in *Lecture Notes In Computer Science*, edited by Bob Hertzberger and Guiseppe Serazzi, 626 - 633. London: Springer-Verlag, 1995.
- Alex Kaiser, Samuel Williams, Kamesh Madduri, Khaled Ibrahim, David H. Bailey, James W. Demmel, Erich Strohmaier, "TORCH Computational Reference Kernels: A Testbed for Computer Science Research," manuscript, Dec 2010.