CRAY
THE SUPERCOMPUTER COMPANY

XT# Hardware Architecture and System Software

Workshop at UDE
June 9-10
Stefan Andersson

Agenda

- Wednesday : Basic Overview
  - XT Architecture
  - XT Programming Environment
  - XT MPT : CRAY MPI
  - Cray Scientific Libraries
  - CRAYPAT : Basic HOWTO
  - Handons
- Thursday : Optimization
  - Where and How to Optimize on the XT
  - More CRAYPAT
  - More Handons (bring your application day)

University of Duisburg Essen

- Two Cabinet Cray XT6m
- Peak Performance of 31 TFLOPS
- AMD 12-Core Magny Cours Processors, 1.9 GHz
- 4128 processor cores
- *This is the first Cray XT6m order in Europe*

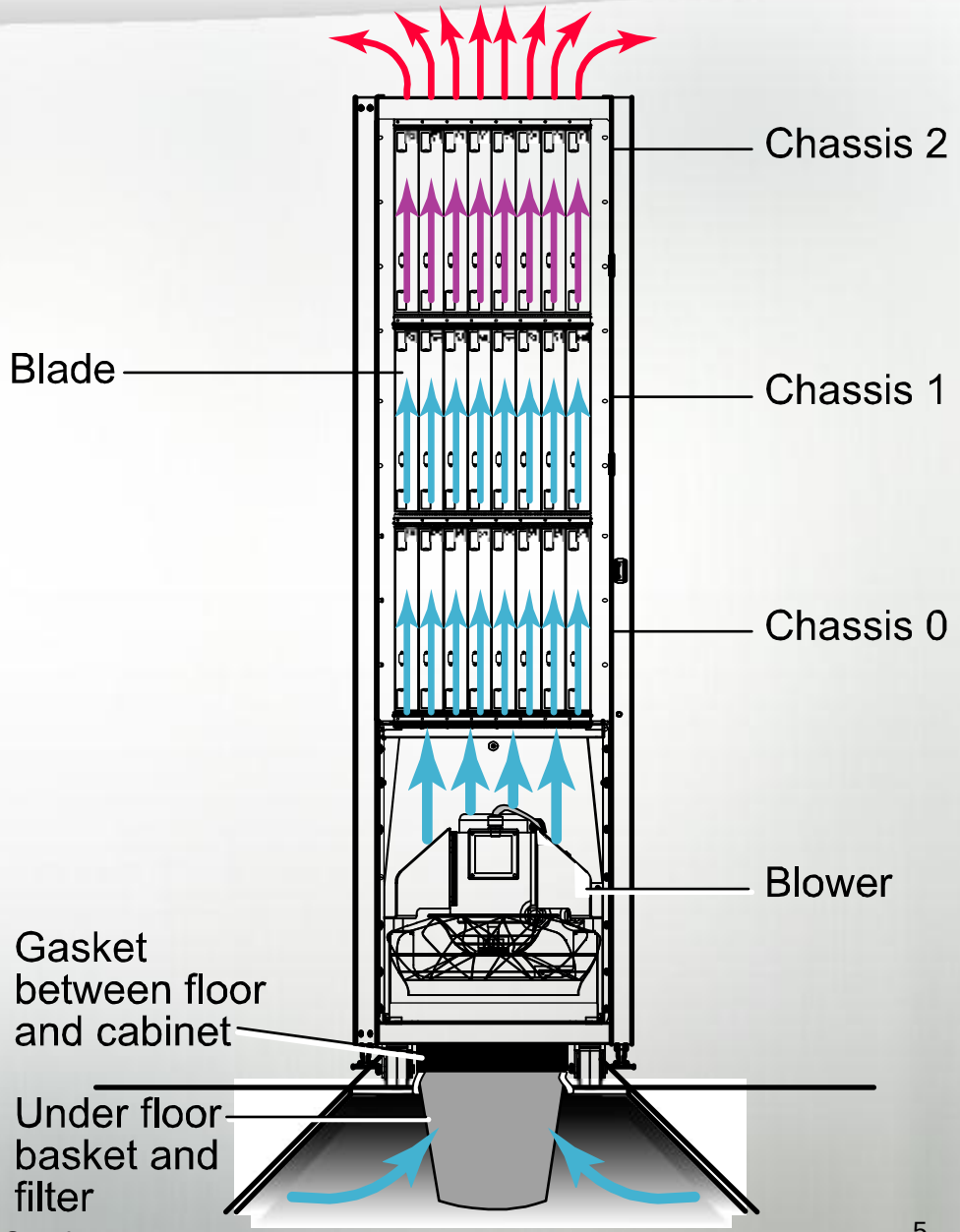
Introducing the Cray XT6m Supercomputer

Cray MPP product for the mid-range HPC market using proven Cray XT6 technologies

- Leading Price/Performance
- Divisional/Supercomputing HPC configurations
  - 1-6 Cabinets
- "Right-sized" Interconnect
  - SeaStar2+ Interconnect
  - 2D Torus Topology
- Proven "petascale" hardware and software technologies
- New "Customer Assist" Service Plan

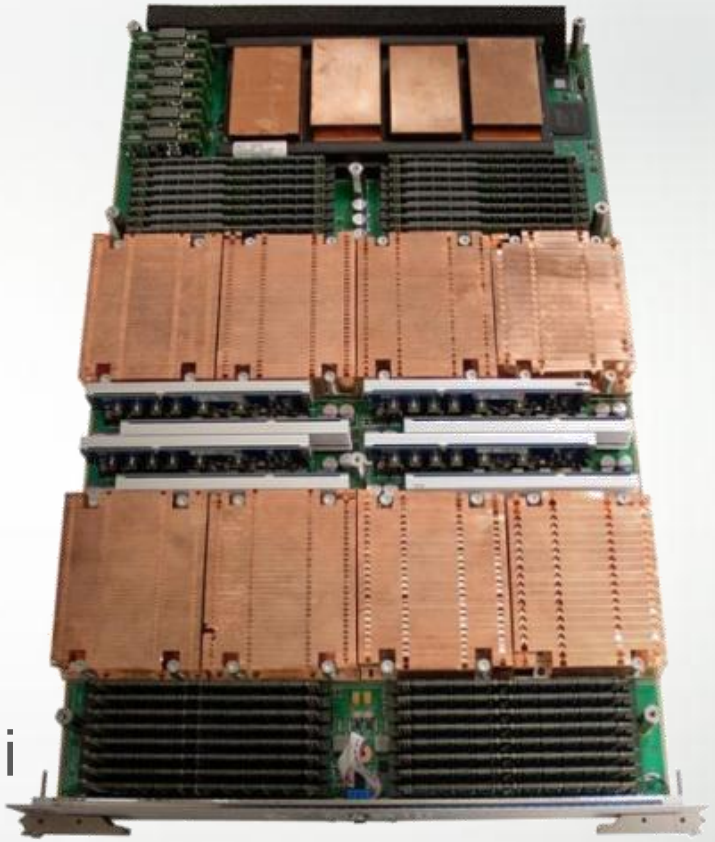
Cray XT Cabinet

- The cabinet contains three chassis, a blower for cooling, a power distribution unit (PDU), a control system (CRMS), and the compute and service blades (modules)
- All components of the system are air cooled
  - A blower in the bottom of the cabinet cools the blades within the cabinet
    - Other rack-mounted devices within the cabinet have their own internal fans for cooling
- The PDU is located behind the blower in the back of the cabinet



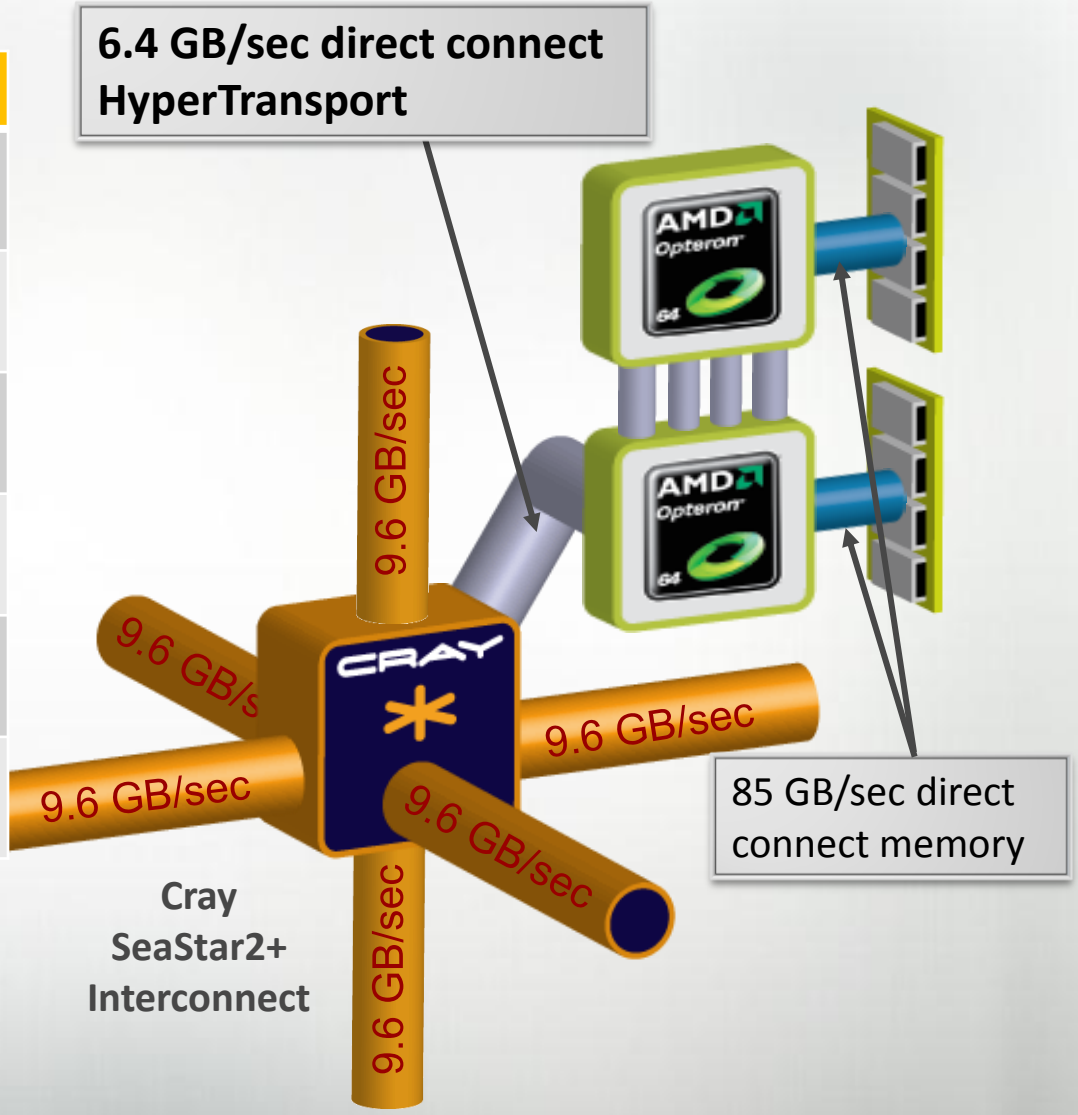
Cray XT6 Compute Blade

- 8 AMD Magny-Cours processors (Opteron 6100 series)
- 8 channels of DDR3 memory per dual-socket node
- Plug-compatible with XT5 cabinets and backplanes
- Now shipping with SeaStar interconnect as the Cray XT6
- Upgradeable shortly to Gemini Interconnect
- Upgradeable in 2011 to AMD's "Interlagos" series

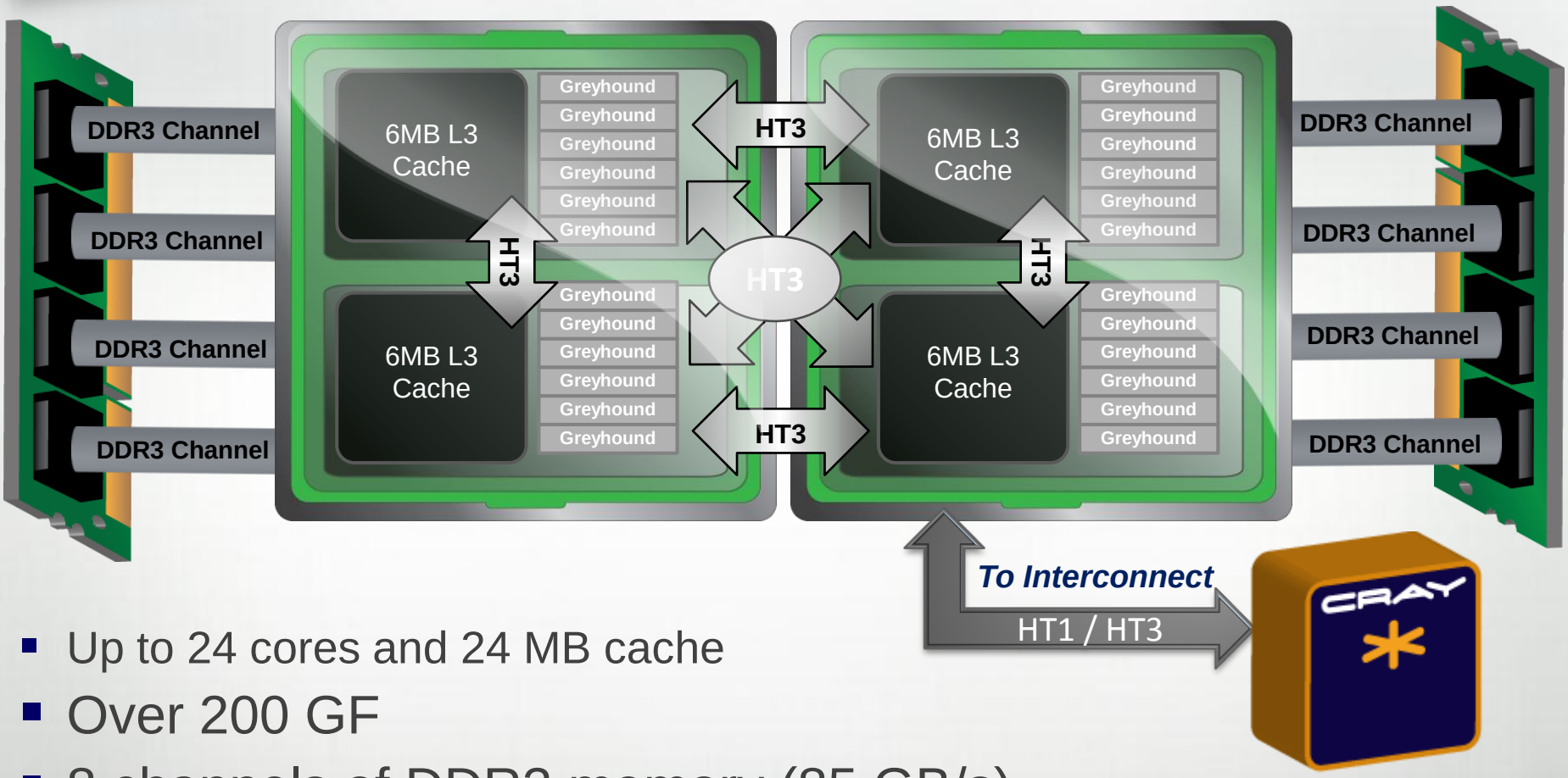


Cray XT6 (or XT6m) Node

Characteristics	
Number of Cores	24 (MC) 32 (IL)
Peak Performance MC-8 (2.4)	153 Gflops/sec
Peak Performance MC-12 (2.2)	211 Gflops/sec
Peak Performance IL-16 (2.2)	282 Gflops/sec
Memory Size	32 or 64 GB per node
Memory Bandwidth	85 GB/s (1333 MHz) 102 GB/s (1600 MHz)

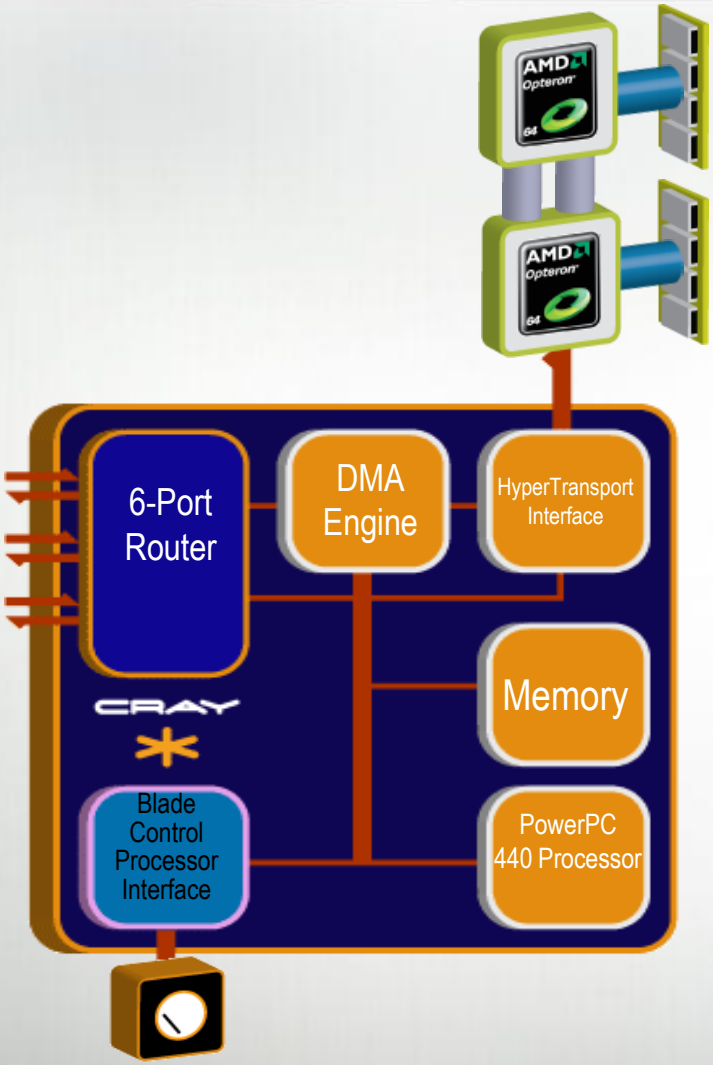


AMD Opteron 6100 Series Processor



- Up to 24 cores and 24 MB cache
- Over 200 GF
- 8 channels of DDR3 memory (85 GB/s)
- Dies are fully connected with HT3
- Snoop Filter for high-performance SMP coherence scaling

Cray SeaStar2+ Interconnect



Now Scaled to 225,000 cores

- **Cray XT5/XT6 systems ship with the SeaStar2+ interconnect**
- Custom ASIC
- **Integrated NIC / Router**
- MPI offload engine
- Connectionless Protocol
- Link Level Reliability
- Proven scalability to 225,000 cores

Cray Programming Environment

Every XT6 Cray System Includes

Cray Integrated Tools

- Cray Compilation Environmen
  - Fortran/C/UPC/CAF/C++
- Optimized OpenMP/MPI Libraries
- CrayPat, Cray Apprentice2
- Optimized Math Libraries
  - Iterative Refinement Toolkit
  - Cray PETSc, CASK

Customer-selected Options

Compilers

- PGI, PathScale

Debuggers

- TotalView, Allinea DDT

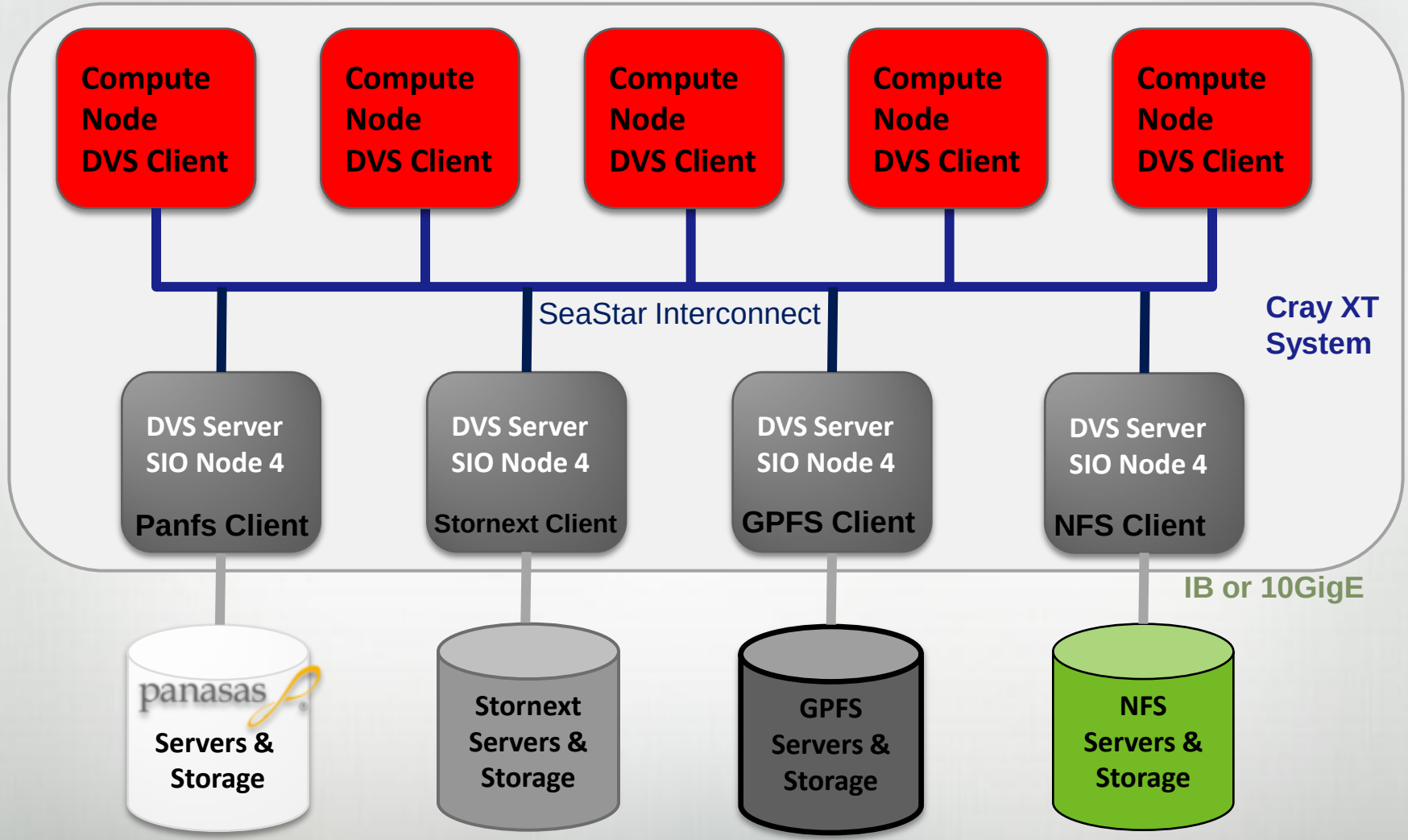
Schedulers

- Moab, PBS Professional, LSF

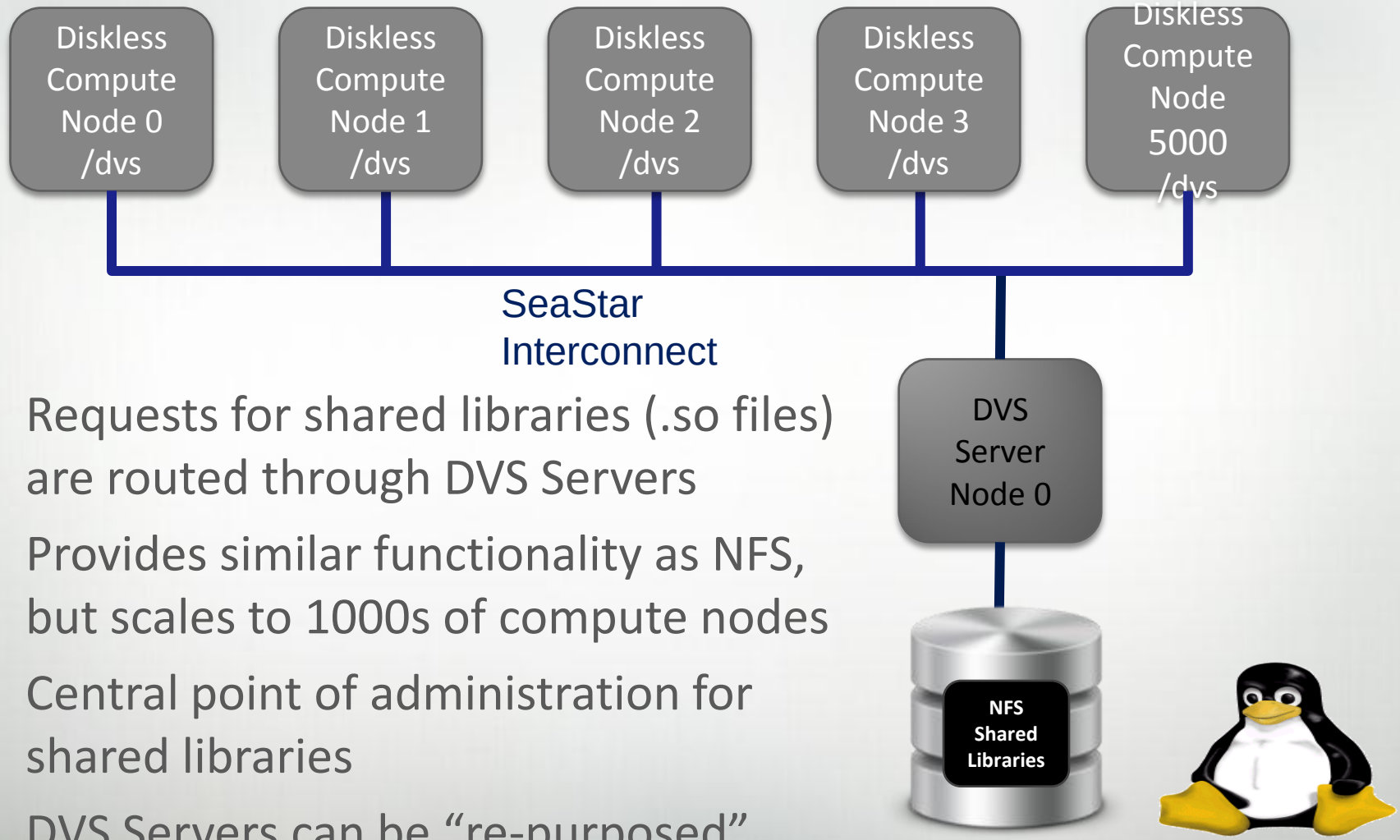
New CLE Features for 2009 / 2010

- Parallel Data Virtualization Service support
- Scalable Dynamic Libraries
- Cluster Compatibility Mode
- Noise Isolation
- NodeKARE (Node Knowledge and Reconfiguration) resiliency features
- Checkpoint / Restart

Mounting Other Filesystems with DVS



Scaling Shared Libraries with DVS



- Requests for shared libraries (.so files) are routed through DVS Servers
- Provides similar functionality as NFS, but scales to 1000s of compute nodes
- Central point of administration for shared libraries
- DVS Servers can be “re-purposed” compute nodes

DSL : Dynamic Shared Libraries

- Benefit: root file system environment available to applications
- Shared root from SIO nodes will be available on compute nodes
- Standard libraries / tools will be in the standard places
- Able to deliver customer-provided root file system to compute nodes
- Programming environment supports static and dynamic linking
- Performance impact negligible, due to scalable implementation

An Adaptive Linux OS designed specifically for HPC

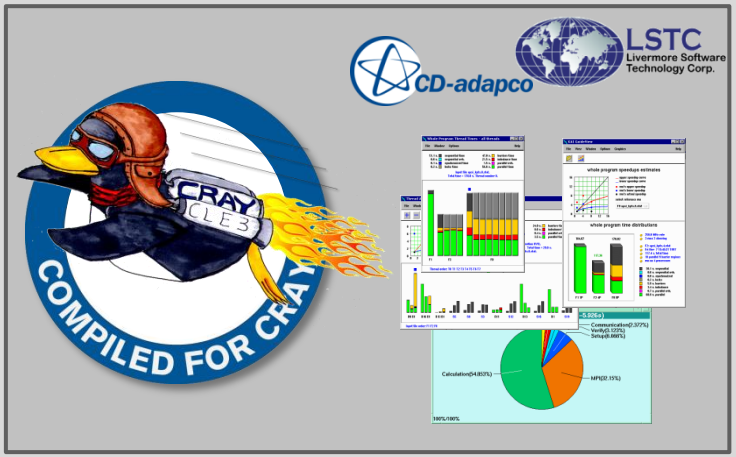
CRAY LINUX ENVIRONMENT CLE3

ESM – *Extreme Scalability Mode*

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

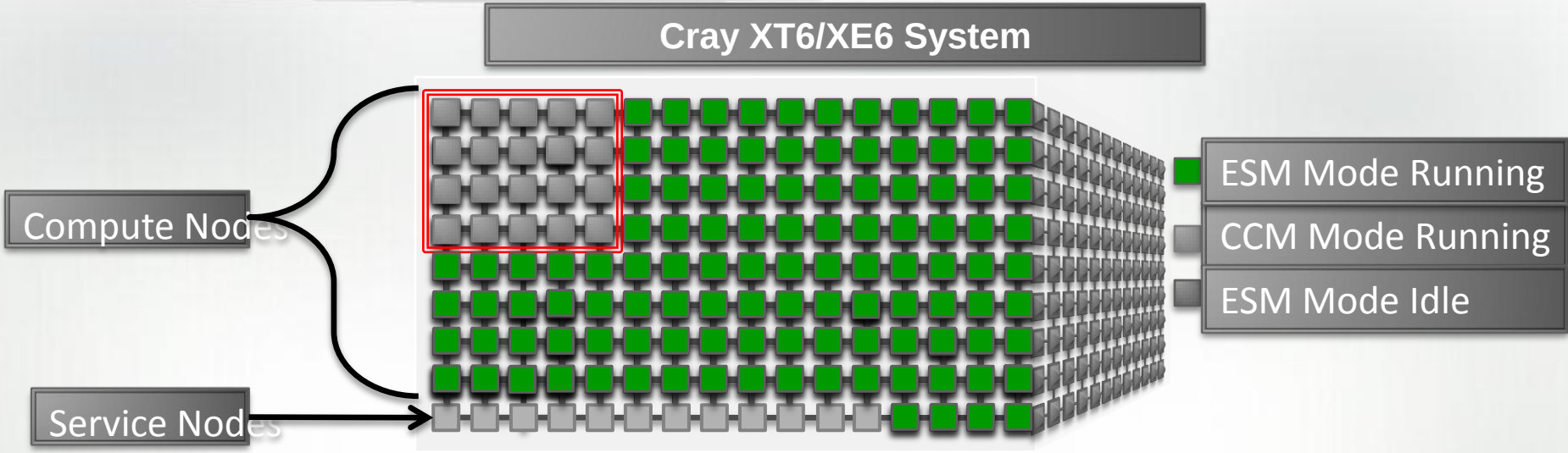
CCM –*Cluster Compatibility Mode*

- 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

CLE3 : Allows simultaneous CCM and ESM Modes



- Many Applications running in Extreme Scalability Mode (ESM)
- Submit CCM application through Batch Scheduler, nodes reserved
  qsub -q ccm Qname AppScript
- Previous jobs finish, nodes configured for CCM
- Executes the batch script and Application
- Other nodes scheduled for ESM or CCM applications as available
- After CCM job completes, CCM nodes cleared
- CCM nodes available for ESM or CCM mode Applications

CRAY
THE SUPERCOMPUTER COMPANY

Cray XE6

Cray XE6 System

- System announced two weeks ago in Edinburgh, Scotland
- Over $200M in booked orders
- 3 Additional Petaflop Machines
- Deliveries start in July
- Key New Technologies
  - Gemini interconnect
  - Series 6 Processor blade to support AMD's new 6100 series Opteron
  - New XIO blade for I/O
  - CLE 3 Operating System

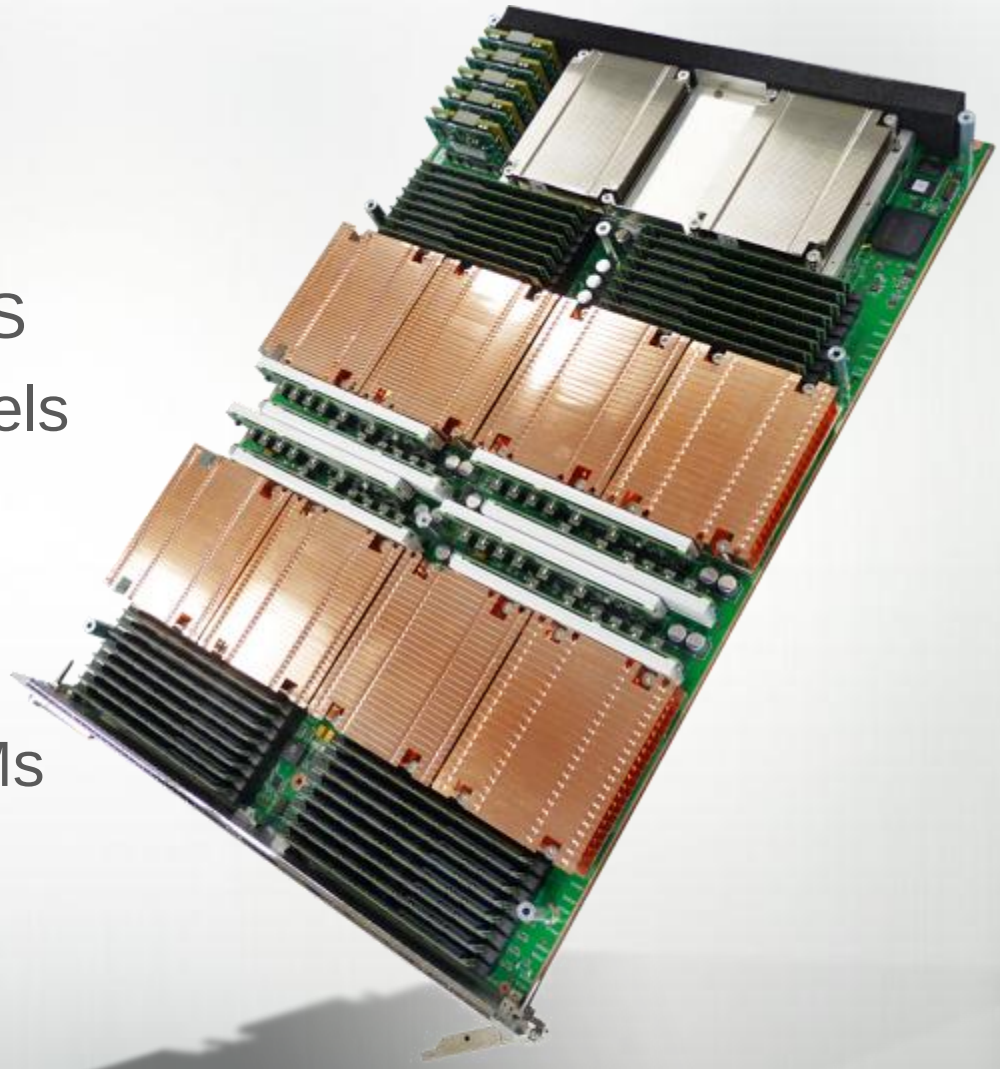
Gemini Network in the Cray XE6

- Shipping 3Q 2010
- XT systems upgraded by swapping mezzanine card
- Topology remains a 3D torus

- Dramatic increase in network performance and capabilities:
  - 50x higher message throughput per node
  - 4x reduction in message latency
  - Hardware support for one-sided MPI and PGAS languages
  - Suite of global synchronization primitives
  - Advanced resiliency features – fully resilient MPI communication

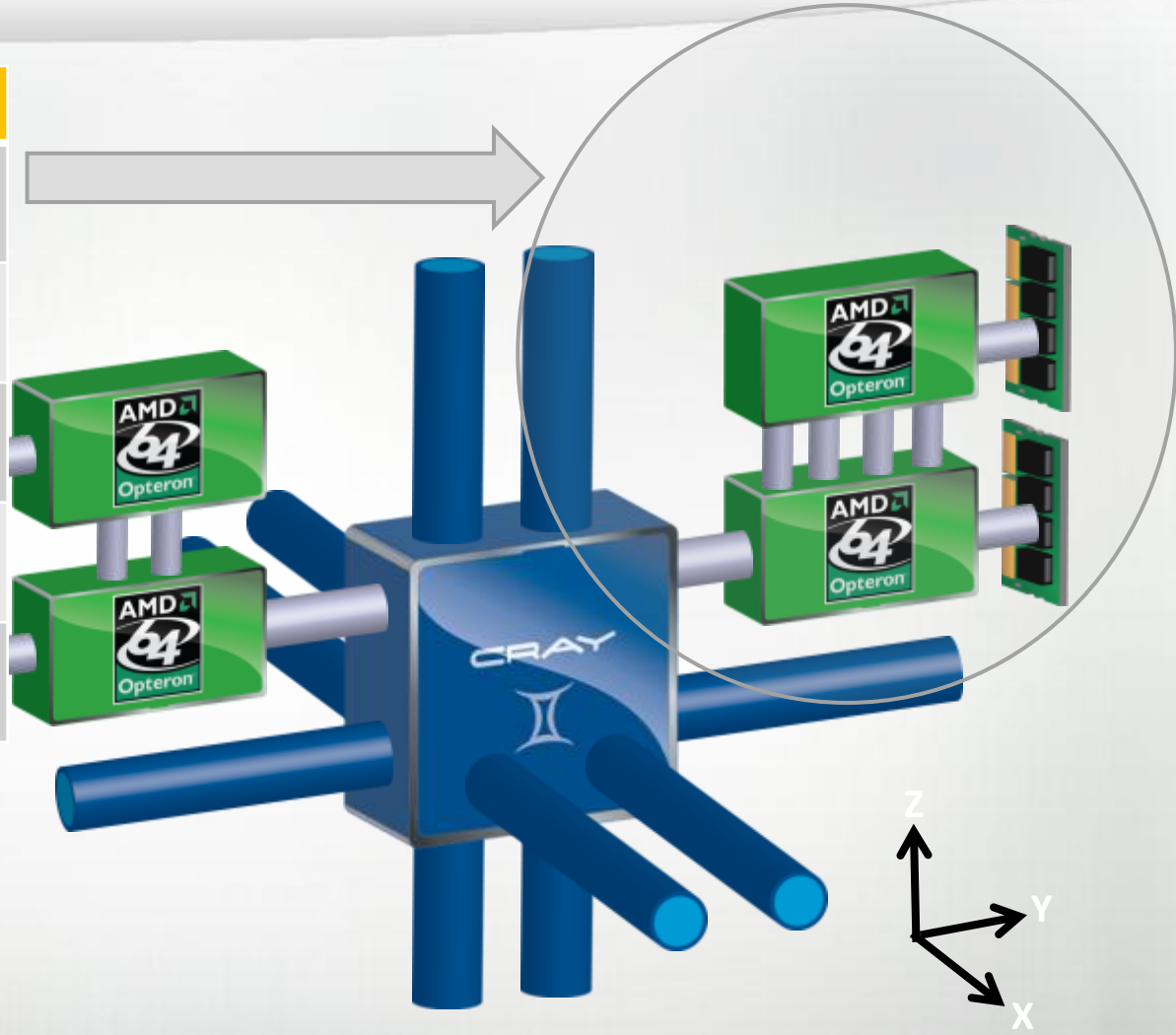
Cray XE6 Compute Blade

- 8 Magny Cours Sockets
- 96 Compute Cores
- 32 DDR3 Memory DIMMS
- 32 DDR3 Memory channels
- 2 Gemini ASICs
- L0 Blade management processor
- Redundant Vertys & VRMs



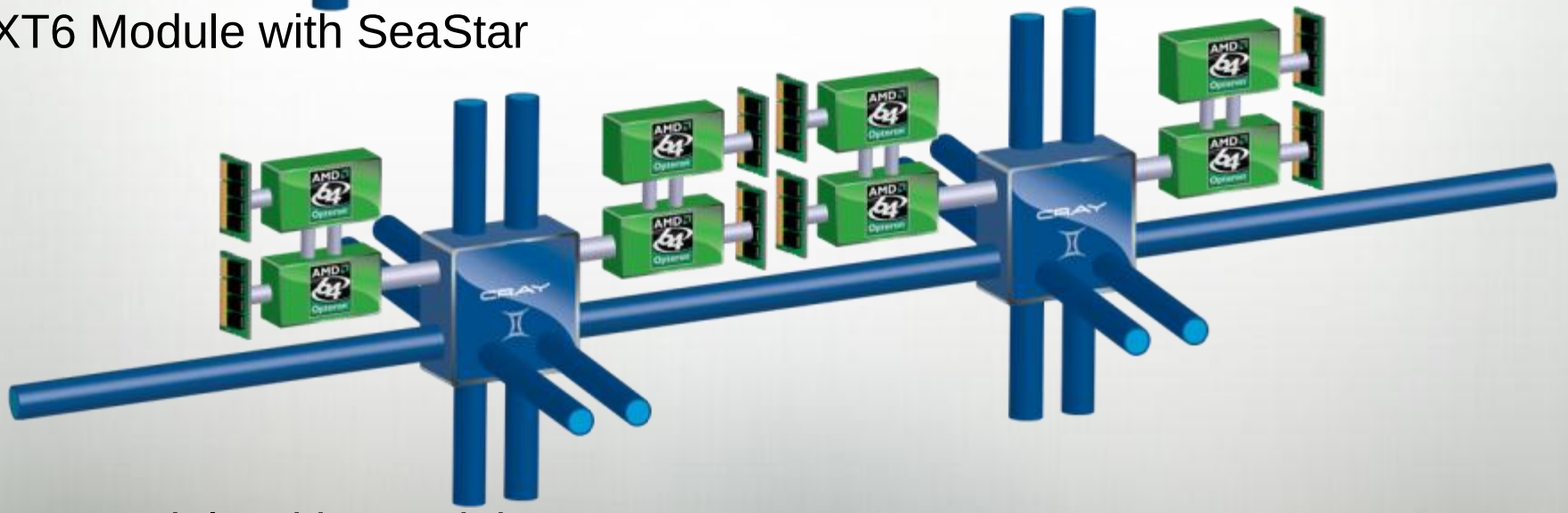
Cray XE6 Node

Node Characteristics	
Number of Cores	24 (Magny Cours)
Peak Performance MC-12 (2.2)	211 Gflops/sec
Peak Performance MC-8 (2.4)	153 Gflops/sec
Memory Size	32 GB per node 64 GB per node
Memory Bandwidth (Peak)	83.5 GB/sec



Gemini vs SeaStar – Topology

XT6 Module with SeaStar



XE6 Module with Gemini

CRAY
THE SUPERCOMPUTER COMPANY

Gemini Interconnect

Cray Network Evolution

SeaStar

- Built for scalability to 250K+ cores
- Very effective routing and low contention switch

Gemini

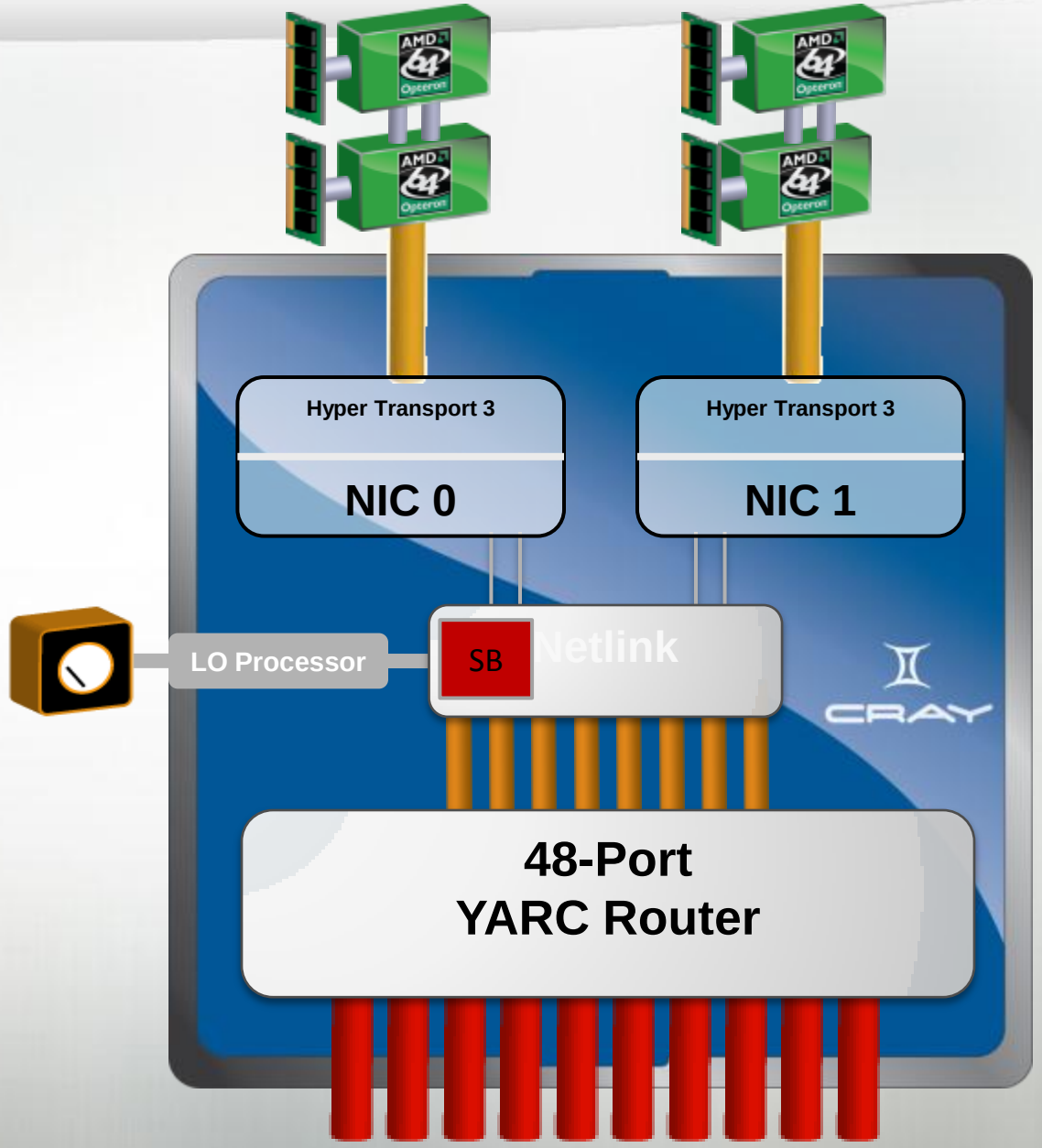
- 100x improvement in message throughput
- 3x improvement in latency
- PGAS Support, Global Address Space
- Scalability to 1M+ cores

Aries

- Up to 10x improvement with Low Radius, High Bandwidth Network
- Very effective routing and low contention switch
- Electro-Optical Signaling

Cray Gemini

- Supports 2 Nodes per ASIC
- 168 GB/sec routing capacity
- Scales to over 100,000 network endpoints
- Link Level Reliability and Adaptive Routing
- Advanced Resiliency Features
- Provides global address space
- Advanced NIC designed to efficiently support
  - MPI
  - One-sided MPI
  - Shmem
  - UPC, Coarray FORTRAN



Gemini MPI Features

- Like SeaStar, Gemini has a DMA offload engine allowing large transfers to proceed asynchronously
- Gemini provides low-overhead OS-bypass features for short transfers
  - MPI latency targeted at ~ 1 us
  - NIC provides for many *millions* of MPI messages per second (measured 8M on a single core)
  - “Hybrid” programming not a *requirement* for performance
- RDMA provides a much improved one-sided communication mechanism
- AMOs provide a faster synchronization method for barriers
- Gemini supports adaptive routing, which
  - Reduces problems with network hot spots
  - Allows MPI to survive link failures

Gemini Advanced Features

- Globally addressable memory provides efficient support for UPC, Co-array FORTRAN, Shmem and Global Arrays
  - Cray Programming Environment will target this capability directly
- Pipelined global loads and stores
  - Allows for fast irregular communication patterns
- Atomic memory operations
  - Provides fast synchronization needed for one-sided communication models

Example: Random Gather (Shared memory code)

```
parameter (n=2**30)
real table(n)
buffer(nelts)  ! nelts << n
…
do i=1,nelts
   buffer(i) = table(index(i))
enddo
```

- Yes, this is purely synthetic, but simulates “irregular” communication access patterns.
- We do have customers that do this stuff.

The Same Code, Written in MPI

```
 if(my_rank.eq.0)then
! first gather indices to send out to individual PEs
        do i=1,nelts
          indpe = ceiling(real(index(i))/real(myelts)) - 1
          isum(indpe)=isum(indpe)+1
          who(isum(indpe),indpe) = index(i)
        enddo
! send out count and indices to PEs
        do i = 1, npes-1
          call MPI_SEND(isum(i),8,MPI_BYTE,i,10,
     &          MPI_COMM_WORLD,ier)
          if(isum(i).gt.0)then
           call MPI_SEND(who(1,i),8*isum(i),MPI_BYTE,i,11,
     &          MPI_COMM_WORLD,ier)
          endif
        enddo
! now wait to receive values and scatter them.
        do i = 1,isum(0)
          offset = mod(who(1,0)-1,myelts)+1
          buff(i,0) = table(offset)
        enddo
        do i = 1,npes-1
          if(isum(i).gt.0)then
           call MPI_RECV(buff(1,i),8*isum(i),MPI_BYTE,i,12,
     &          MPI_COMM_WORLD,status,ier)
          endif
        enddo
        do i=nelts,1,-1
          indpe = ceiling(real(index(i))/real(myelts)) - 1
          offset = isum(indpe)
          isum(indpe) = isum(indpe) - 1
          buffer(i) = buff(offset,indpe)
        enddo
        else  !if my_rank.ne.0
          call MPI_RECV(my_sum,8,MPI_BYTE,0,10,
     &          MPI_COMM_WORLD,status,ier)
          if(my_sum.gt.0)then
           call MPI_RECV(index,8*my_sum,MPI_BYTE,0,11,
     &          MPI_COMM_WORLD,status,ier)
           do i = 1, my_sum
             offset = mod(index(i)-1,myelts)+1
           do i = 1, my_sum
             offset = mod(index(i)-1,myelts)+1
             buffer(i) = table(offset)
           enddo
           call MPI_SEND(buffer,8*my_sum,MPI_BYTE,0,12,
     &          MPI_COMM_WORLD,ier)
          endif
        endif
```

The Same Code, Written in Co-Array FORTRAN

```
      parameter (n=2**30)              ! 1 Gigaword
      parameter (NPES=2**7)            ! 128 Pes
      parameter (eltspe = 2**23)       ! Elements per PE
      real table (eltspe)[NPES]
…
      do i=1,nelts
        PE =( index(i) + eltspe-1)/eltspe
        offset = mod(index(i)-1,eltspe)+1
        buffer(i) = table(offset)[PE]
      enddo
```

- Remote references *will* pipeline with this loop (library calls do not!)
- Resulting performance is orders of magnitude faster than MPI

Summery : An overview of Cray XT systems

	XT3	XT4	XT5	XT6	XE6
Number of cores/socket	2	4	4-6	12	12
Number of cores/node	2	4	8-12	24	24
Clock Cycle (CC)	2.6	2.3	2.6	1.8-2.4	1.8-2.4
Number of 64 bit Results/CC	2	4	4	4	4
GFLOPS/Node	10.4	36.8	83.6-124.8	~200	~200
Interconnect	Seastar 1	Seastar 2+	Seastar 2+	Seastar 2+	Gemini
Link Bandwidth GB/sec	6x2.4	6x4.8	6x4.8	6x4.8	10x~14
MPI Latency microseconds	6	6	6	6	1.2
Messages/sec	400K	400K	400K	400K	10M
Global Addressing	No	No	No	Yes	Yes

Cray XE6 Compute Blade