The Cray Gemini Network: Basic Architecture and Failure Analysis

Forest Godfrey
Principal Engineer, Cray Inc.
March 1, 2012

Outline

- About the Speaker
- Cray XE/XK System Overview
- Cray Gemini NIC Hardware Architecture Overview
- System Resiliency Analysis
- Application Resiliency Analysis

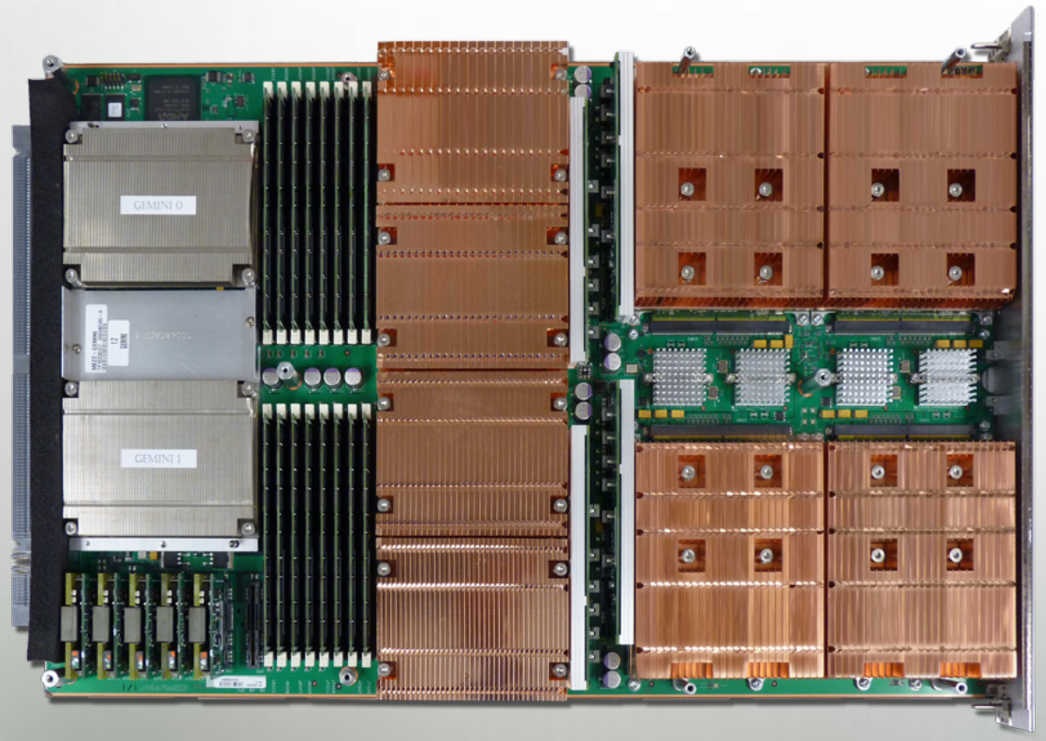
- Note: This is not specifically a Blue Waters talk!

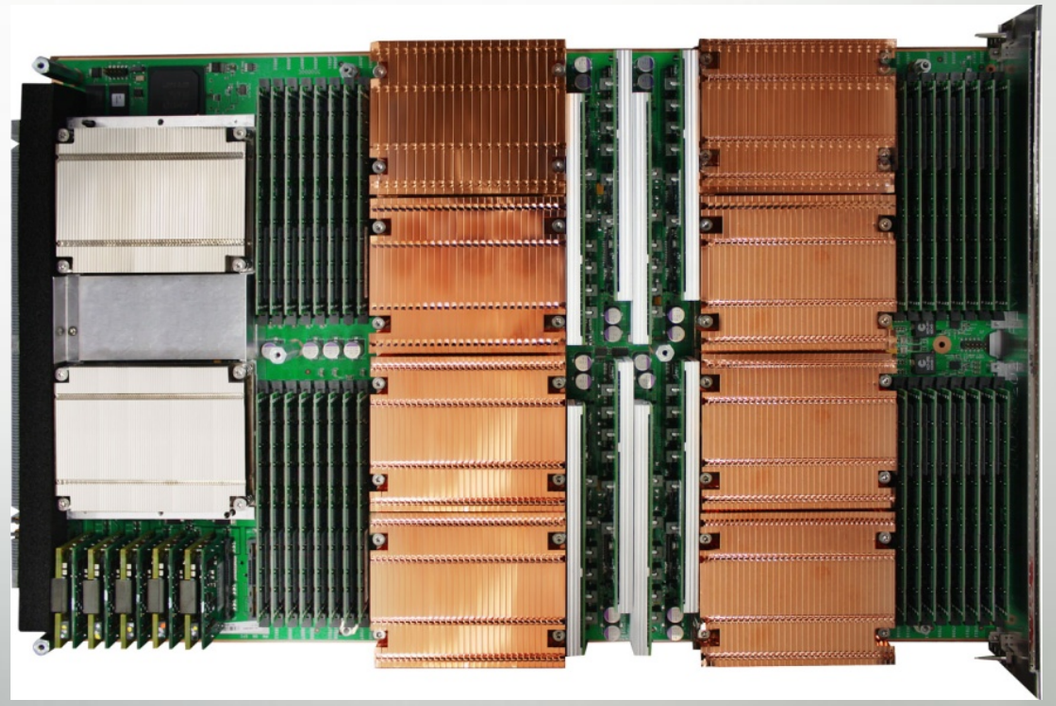
About Me

- Graduated from Carnegie Mellon in 1999 with Bachelor of Science in Computer Science
- Have been with Cray (or SGI when it owned Cray) ever since
- Started as a kernel programmer (Irix and Linux)
- Worked on SGI Origin and Altix systems as well as Cray X1, X1E, X2, XT series, XE series, and XK series (and two upcoming products). Served on architecture team for X2, XE and XK.
- Lead software architect for GPUs and future system control networks
- My brother, Brighten, is a professor in the CS Department at UIUC

Outline

- About the Speaker
- Cray XE/XK System Overview
- Cray Gemini NIC Hardware Architecture Overview
- System Resiliency Analysis
- Application Resiliency Analysis



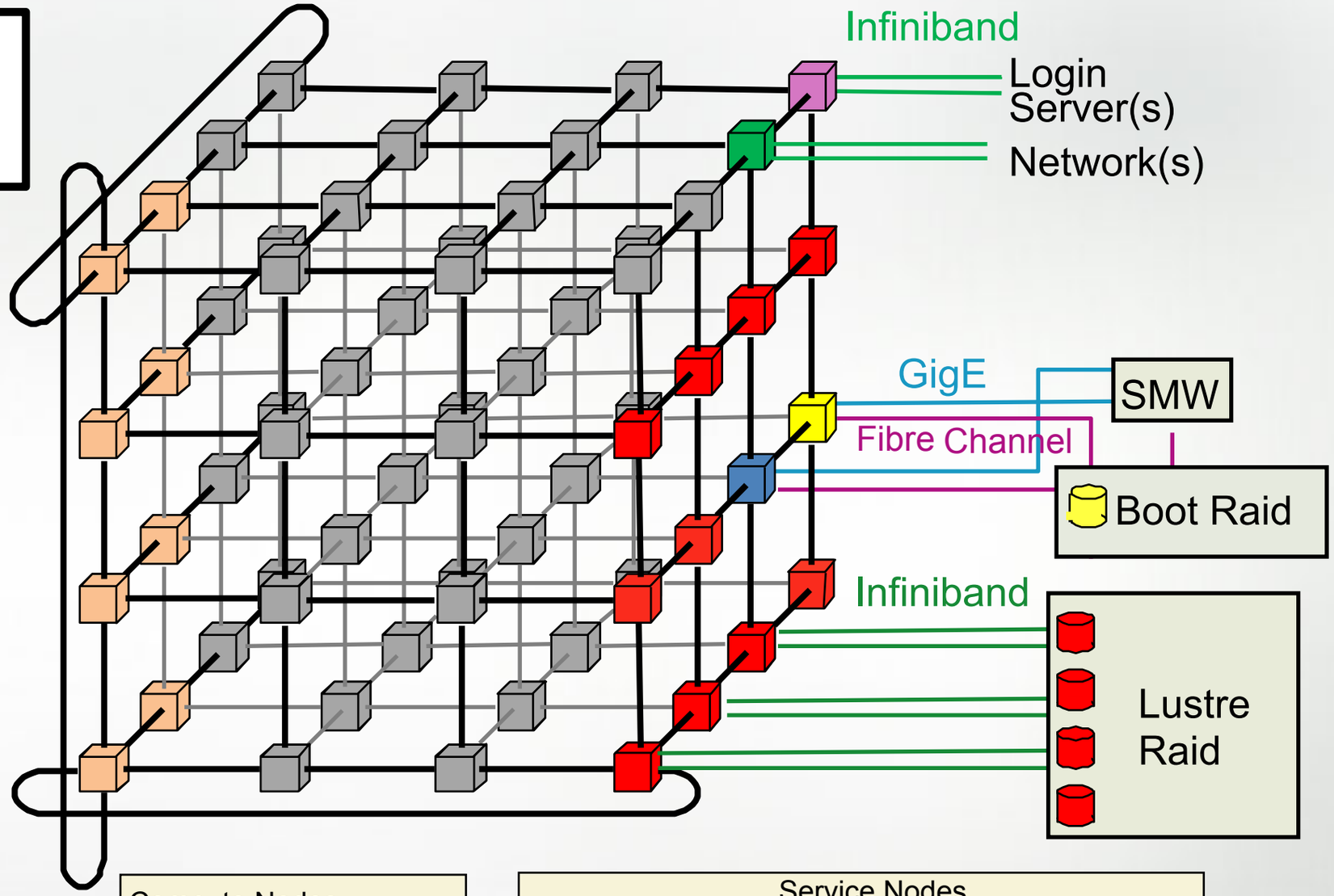


Gemini Overview: Router Architecture

- Two NICs per chip (hence "Gemini")
  - Called "NIC" even though it's not a "card" or "chip"
- Integral 48 port router
  - 8 ports are internal only
  - 40 external ports are arranged to form 3 dimensional torus
    - 6 links: X+, X-, Y+, Y-, Z+, Z-

Gemini Overview: Router Architecture

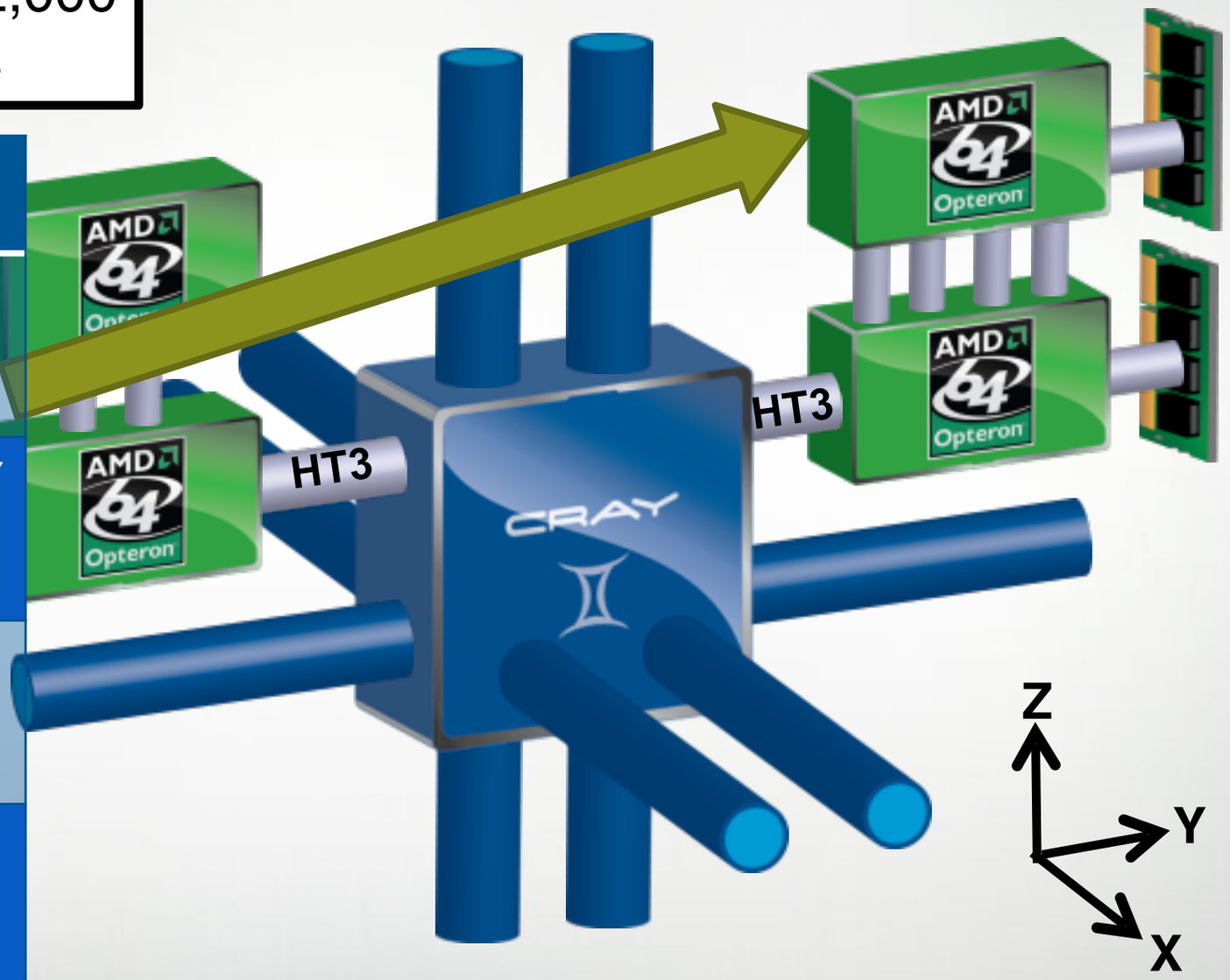
Blue Waters 3D Torus
23 x 24 x 24 Gemini



Cray XE6 Node Block Diagram

Blue Waters is currently scheduled to contain ~22,000 XE6 compute nodes

Node Characteristics	
Number of Cores*	***16***
Approximate* Performance	***300 Gflops/ sec***
Memory Size	***64 GB per node***
Memory Bandwidth (Peak)	***102 GB/sec***
Interconnect Injection Bandwidth (Peak)	***9.6 GB/sec per direction***

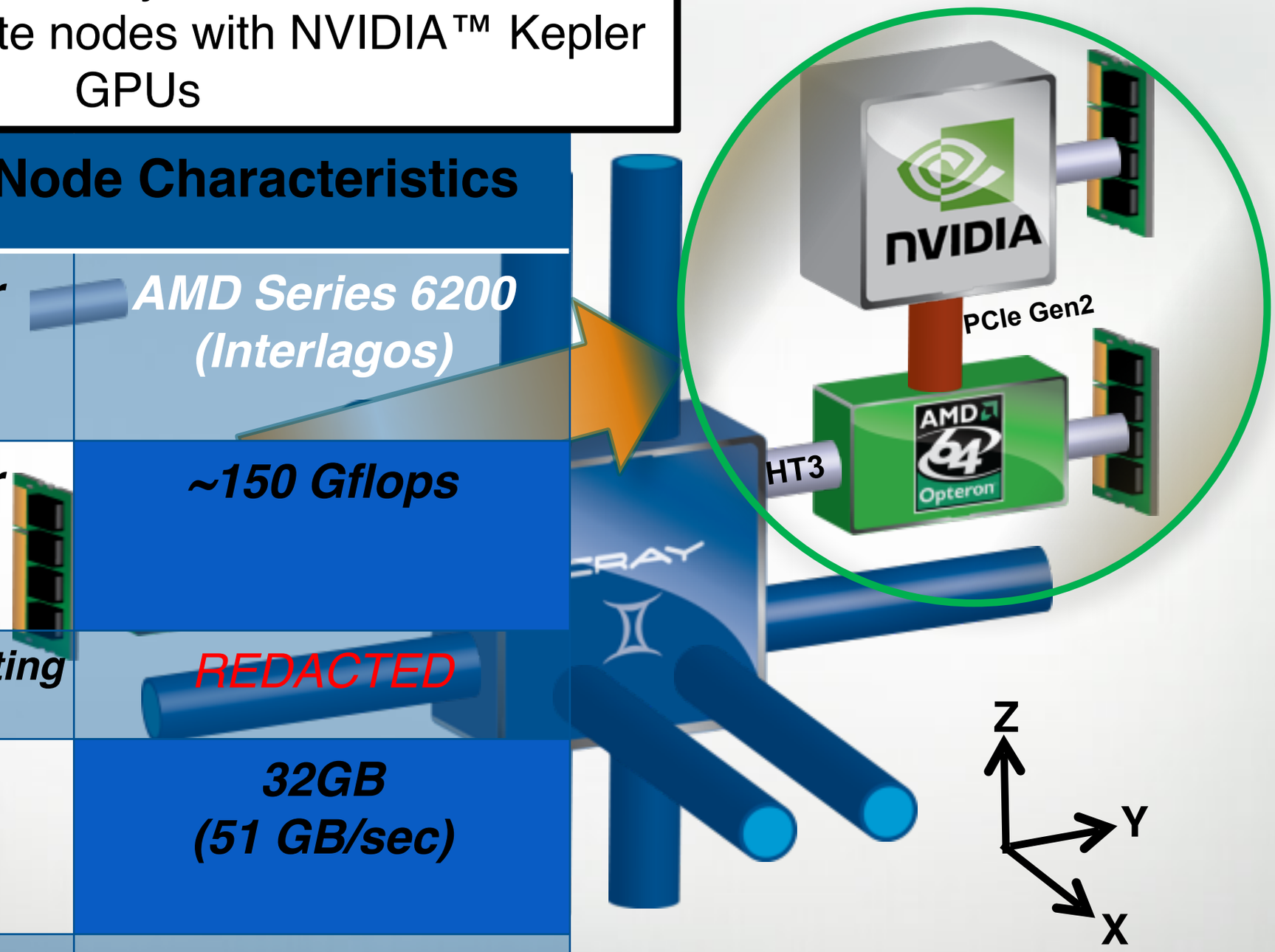


**Exact calculation of these numbers is beyond the scope of this talk*

Cray XK6 Block Diagram

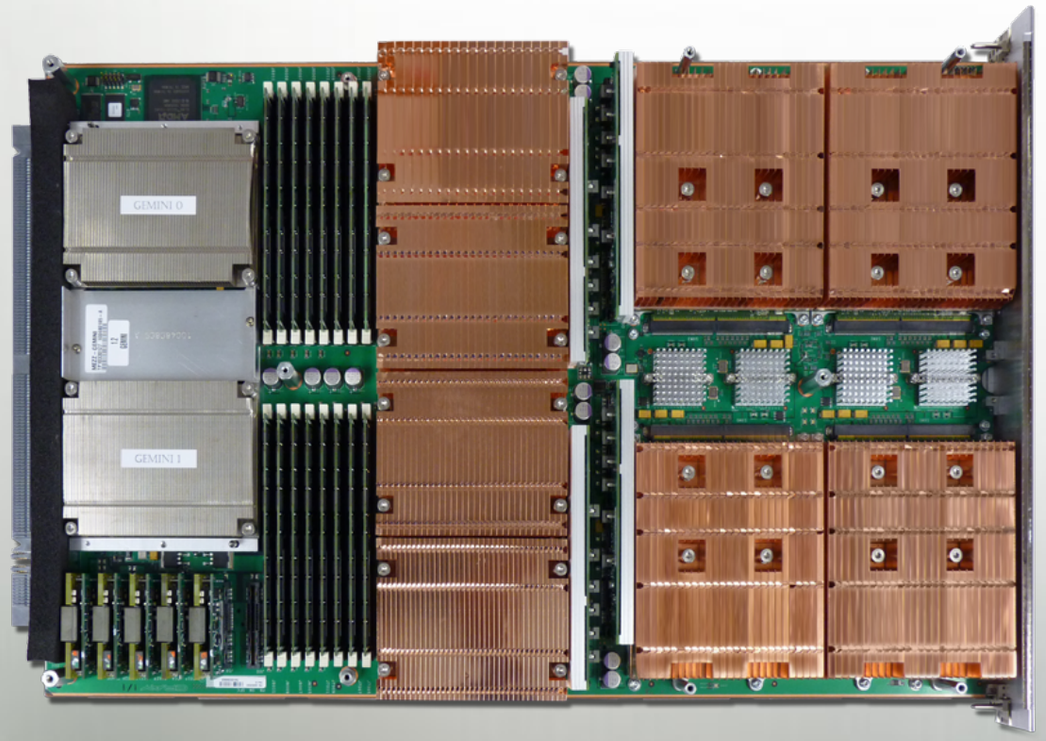
Blue Waters is currently scheduled to contain ~3000 XK compute nodes with NVIDIA™ Kepler GPUs

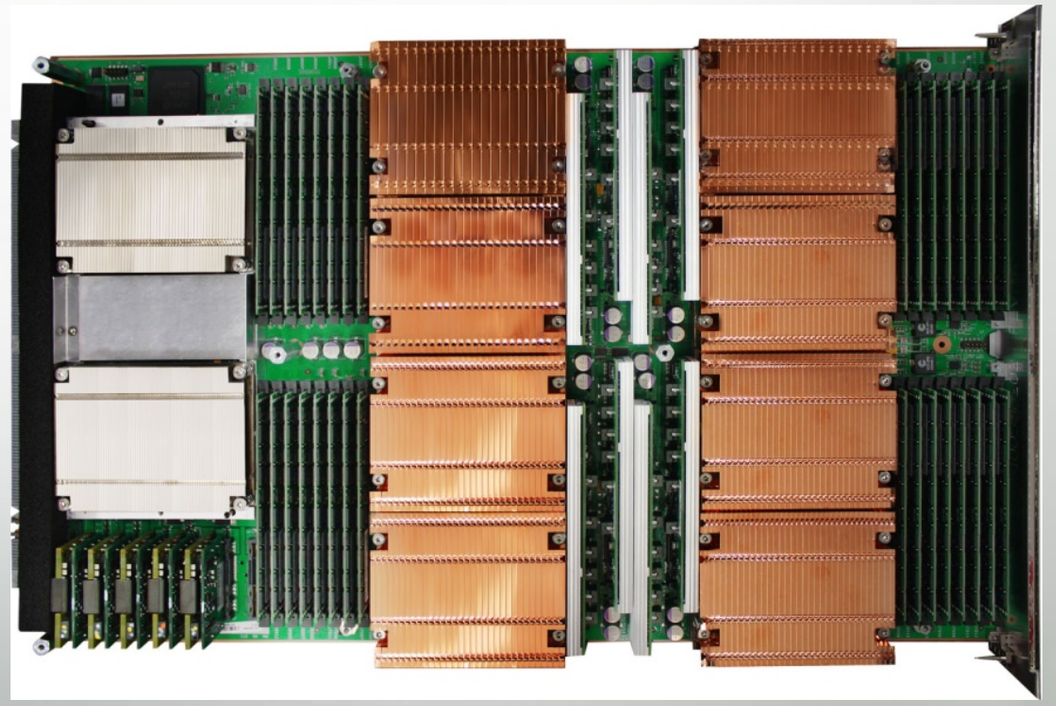
XK7 Compute Node Characteristics	
Host Processor	***AMD Series 6200 (Interlagos)***
Host Processor Approximate Performance	***~150 Gflops***
Kepler Peak (DP floating point)	*REDACTED*
Host Memory *(peak)*	***32GB (51 GB/sec)***
Kepler Memory	***6GB GDDR5 capacity***



Outline

- About the Speaker
- Cray XE/XK System Overview
- Cray Gemini NIC Hardware Architecture Overview
- System Resiliency Analysis
- Application Resiliency Analysis





Gemini Hardware Overview: The problem

- High Level Goals
  - Machine must be excellent at MPI
  - Machine must be excellent at GAS languages (UPC, Co-Array FORTRAN)
  - Machine must be reliable
- Combination of MPI and GAS languages means that the machine must have low latency access to global memory

Gemini Overview: Low Latency The Old Way

- Have the processor be able to load directly from remote memory regions
  - Advantages
    - Easy to program
    - NIC hardware architecture is very simple
    - Fit well with the Cray instruction set
  - Disadvantages
    - Requires lots of physical address bits
      - At least 15 bits for Node ID, 35 bits for physical address = 49 bits! Opteron just recently got to 48 bits!
    - If network stalls, so does the processor
    - Must cover latency/bandwidth product of network (see next slide)

Gemini Overview: Latency & Bandwidth Problem

- Latency Bandwidth Product is literally latency x bandwidth
- Oversimplified Example:
  - Gemini network on Blue Waters is 23x24x24 for a diameter of 36 so we need minimum of 72 hops for the round trip
  - Assume per-hop latency is approximately 100ns (real number depends on a variety of parameters)
  - Injection bandwidth is approximately 5.5GB/sec
  - 72 * 100ns * 5.5GB/sec = 42521 bytes
  - At max HyperTransport packet size (256 bytes) that's 167 packets!
  - Opteron supports at most 32 outstanding non-posted packets (packets that require a response)!

Gemini Overview: Solutions

- Need to decouple network operations from the processor
  - Could just use a block transfer engine (BTE)
    - That only works at large enough transfer sizes
    - GAS languages need to deal with small transfers
- In addition to BTE, use a “Fast Memory Access (FMA) Window”
  - Goal: Allow processor directed network reads/ writes directly from user space without coupling processor instructions to network operations

Gemini Overview: Solutions: FMA Windows

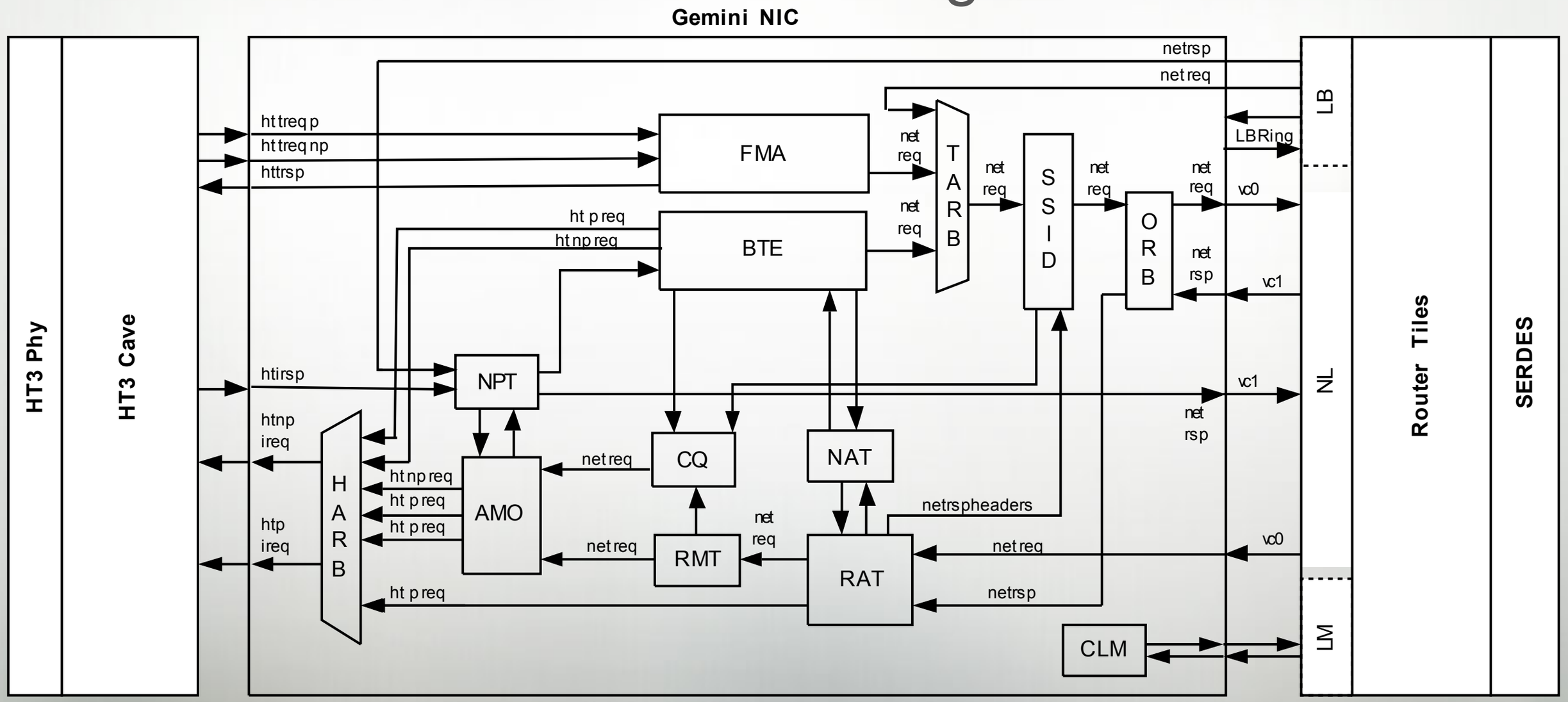
- Need to disconnect loads and stores that occur in the processor pipeline from the actual network PUT and GET operations.
- Allows the processor to deal with errors by not stalling indefinitely.
- Allows the network to have many more outstanding operations than the processor would otherwise support.

Gemini Overview: Solutions: FMA Windows

- Split into two pieces – large (512MB) access window and small (4KB) control window
- Control window "aims" the access window at remote memory (sets target node, which memory registration, protection information, type of operation, etc.)
- Processor then writes directly into access window
- To do a remote read, command is set to read and the write into the local window causes remote memory to be written back to local memory
- In addition to reads/writes, various atomic memory operations are supported

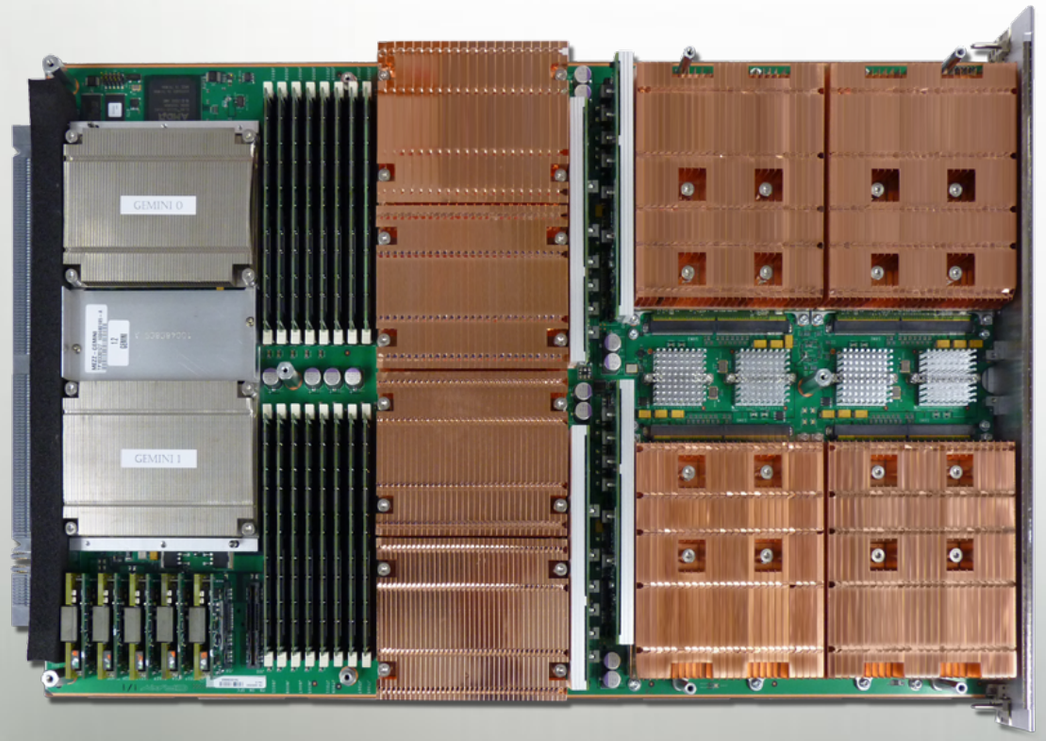
Gemini Overview: NIC Architecture

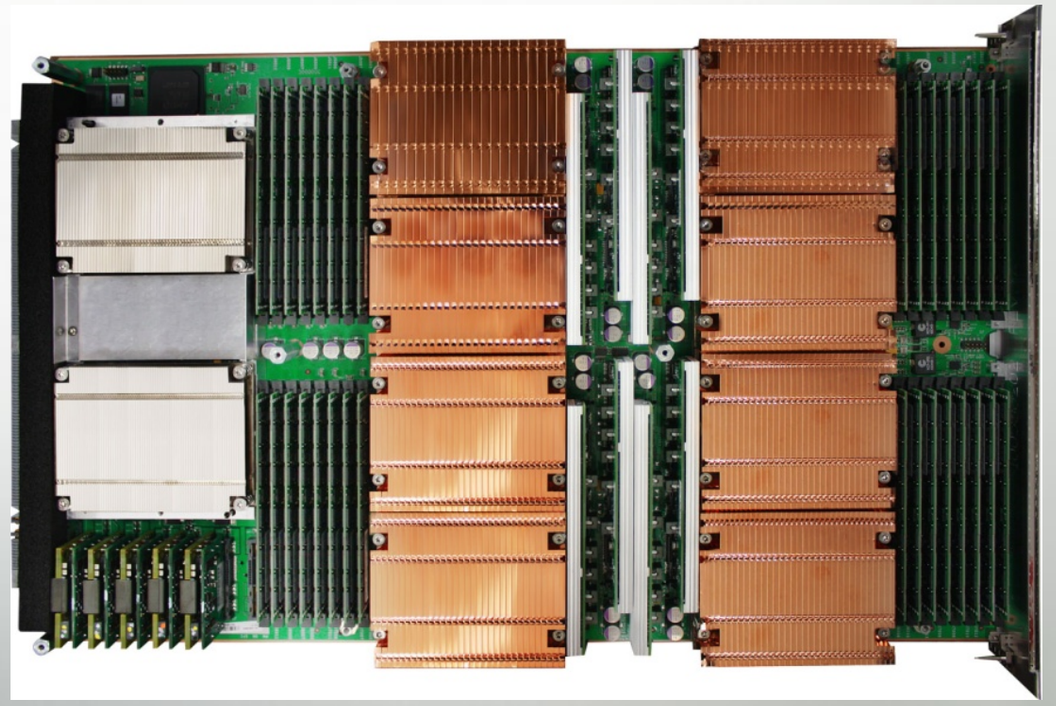
- Need to add a variety of blocks to handle network operations/timeouts and completion of operations
- Final NIC Hardware block diagram



Outline

- About the Speaker
- Cray XE/XK System Overview
- Cray Gemini NIC Hardware Architecture Overview
- System Resiliency Analysis
- Application Resiliency Analysis





Resiliency Analysis: The Basics

- Unfortunately, vendors, including Cray, don't like giving out hardware reliability numbers.
- For the rest of this section, we will talk about a particular size system. It would be considered "leadership class". It is not any currently shipping system.
- People with Intel NDA's may be able to get actual numbers from Intel on processor reliability.

Resiliency Analysis: The Basics

- Hardware failure rates are measured using “Failures In Time” (FITS). 1 FIT = 1 failure during 1 billion hours of operation
- The system in question has a FIT rate of 215,505,538
- Convert FIT to Mean Time Between Failures:

$$\frac{1}{215{,}505{,}538\ \frac{Failures}{1\ billion\ Hours}} = \frac{1\ billion\ Hours}{215{,}505{,}538\ Failures}$$

$$= 4.64\ \frac{Hours}{Failure}$$

- System hardware MTBF of 4.64 hours!!! That is not good!
- Not a useful number because software can recover some failures & some failures don’t crash the system

Resiliency Analysis: Software Recovery

- Need to decompose failures into individual types
  - Processor Uncorrectable Memory Error
  - Processor Silent Data Corruption
    - Yes, it happens. Fortunately not often. No, I can't give this number either.
  - Voltage Regulator Failure
  - High Speed Network Link Failure
  - Etc.
- Used a total of 33 categories of failure

Resiliency Analysis: Software Recovery

- Next analyze how software will handle the failure and its impact on the system as a whole
  - Example: Uncorrectable Error in RAM
  - Should never cause a total system failure, but maybe we'll get unlucky. May cause failure 0.1% of the time.
  - When doing analysis such as this, should justify each statement.
  - In this case, a DRAM error will likely crash the node. The software is designed such that no individual node failure can take down the system. We can fail over boot nodes, filesystem server nodes (and other I/O nodes), and login nodes. Failures on compute nodes do not affect system services. A variety of software and hardware timeouts are in place and verified to work to assure this.

Resiliency Analysis: Software Recovery

- Another example: High Speed Network Link Failure
- In the event of a failed HSN link, we can recover all adaptively routed packets. System services only use adaptive packets.
- However, must reroute the system after the failure. This is a complex procedure. At the time of this analysis, it was in testing and showed a 75% success rate (it's well above 90% now).
- Thus, 25% of the time, an HSN link failure will crash the system

Resiliency Analysis: After Recovery

- Now we can compute new “software adjusted” failure rates for each category

$$Software\ Adjusted\ FIT\ rate = Hardware\ Fit\ Rate * (1 - Software\ Recovery\ Success\ Rate)$$

- Then sum the individual categories to get the new failure rate

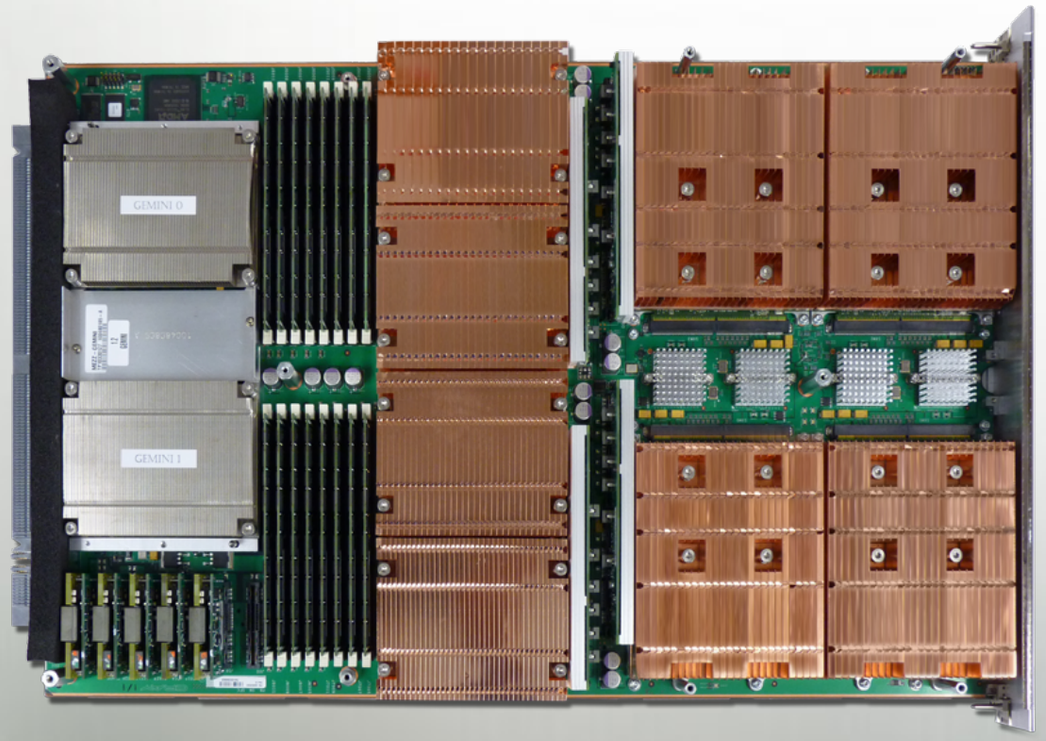
Resiliency Analysis: After Recovery

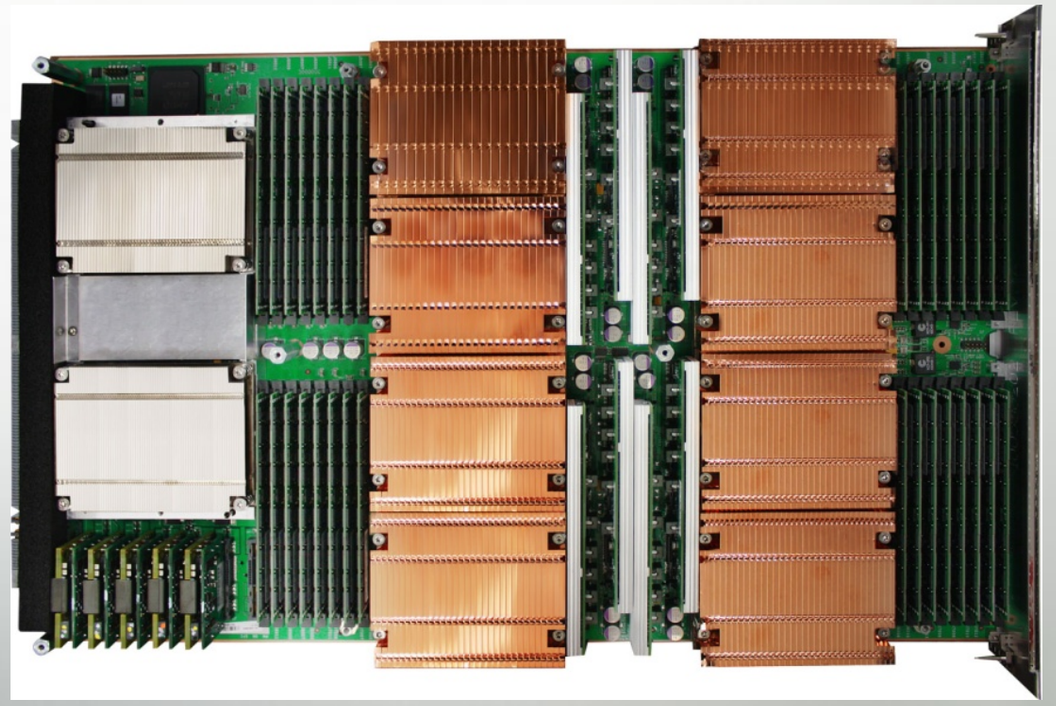
- 99.2713% of hardware failures are not fatal to the system after software recovery
- In our example system, we now have a FIT rate of 1,570,560

$$\frac{1}{1{,}570{,}560\ \frac{Failures}{1\ billion\ Hours}} = \frac{1\ billion\ Hours}{1{,}570{,}560\ Failures} = 637\ \frac{Hours}{Failure}$$

Outline

- About the Speaker
- Cray XE/XK System Overview
- Cray Gemini NIC Hardware Architecture Overview
- System Resiliency Analysis
- Application Resiliency Analysis





Resiliency Analysis: Application Failures

- System level resilience is acceptable
- Run the same calculation for standard MPI application running on our example system
- Only 15.2% of the failures are recoverable
- **No failure that loses even a single byte of memory of an MPI 2 application is recoverable!**

Resiliency Analysis: Application Failures

- After software recovery, application FIT rate is 182,719,264.

$$\frac{1}{182{,}719{,}264\ \frac{Failures}{1\ billion\ Hours}} = \frac{1\ billion\ Hours}{182{,}719{,}264\ Failures}$$

$$= 5.47\ \frac{Hours}{Failure}$$

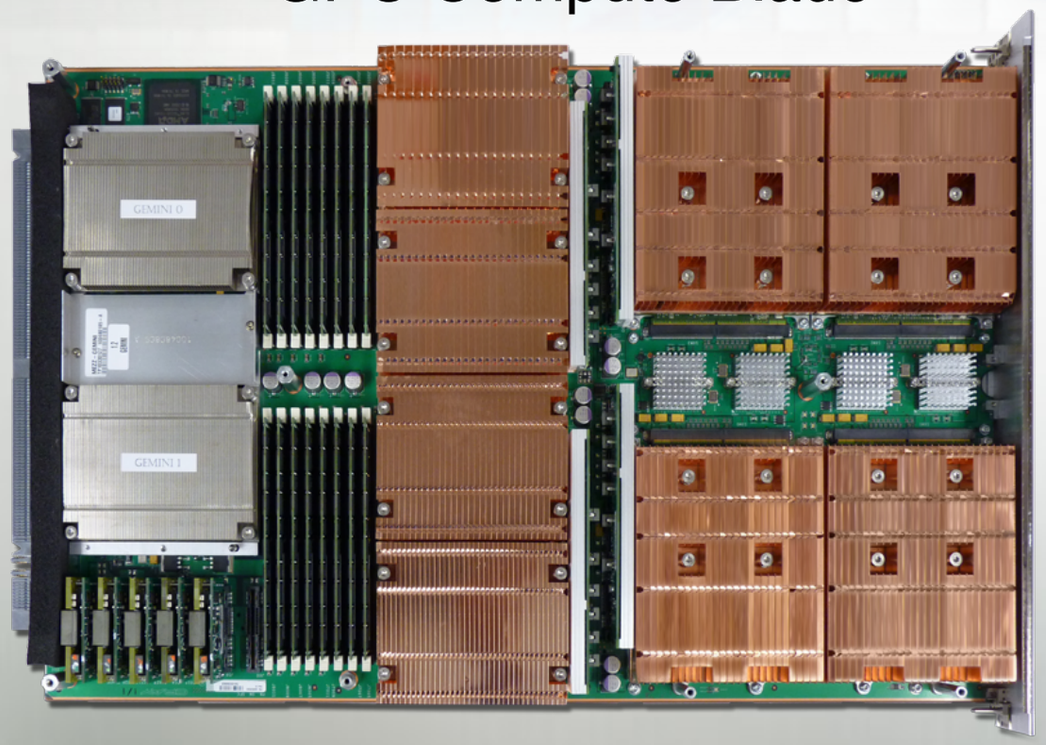
Resiliency Analysis: Application Failures

- As time goes on, processor voltages drop and transistor counts. That will increase, not decrease failure rates.
- With memory sizes increasing, it is not possible to checkpoint/restart out of that failure rate!
- Must speed adoption of programming models capable of dealing with hardware failures (including loss of data).
- Many universities working on this including the Parallel Programming Laboratory at UIUC!

Questions?

- Forest Godfrey can be reached at fgodfrey@cray.com.

GPU Compute Blade



Opteron Compute Blade