THOR'S HAMMER

RED STORM

Bill Camp

& Jim Tomkins

National Nuclear Security Administration

Sandia National Laboratories

The Design Specification and Initial Implementation of the Red Storm Architecture --in partnership with Cray, Inc.

William J. Camp & James L. Tomkins

CCIM, Sandia National Laboratories

Albuquerque, NM

bill@sandia.gov

Our rubric

- Mission critical engineering & science applications
- Large systems with a few processors per node
- Message passing paradigm
- Balanced architecture
- Use commodity wherever possible
- Efficient systems software
- Emphasis on scalability & reliability in all aspects
- Critical advances in parallel algorithms
- Vertical integration of technologies

Computing domains at Sandia

Domain	Volume		Mid-Range		Peak
# Procs	1	10^1	10^2	10^3	10^4
Red Storm			X	X	X
Cplant		X	X	X	
Beowulf	X	X	X		
Desktop	X				

- Red Storm is targeting the highest-end market but has real advantages for the mid-range market (from 1 cabinet on up)

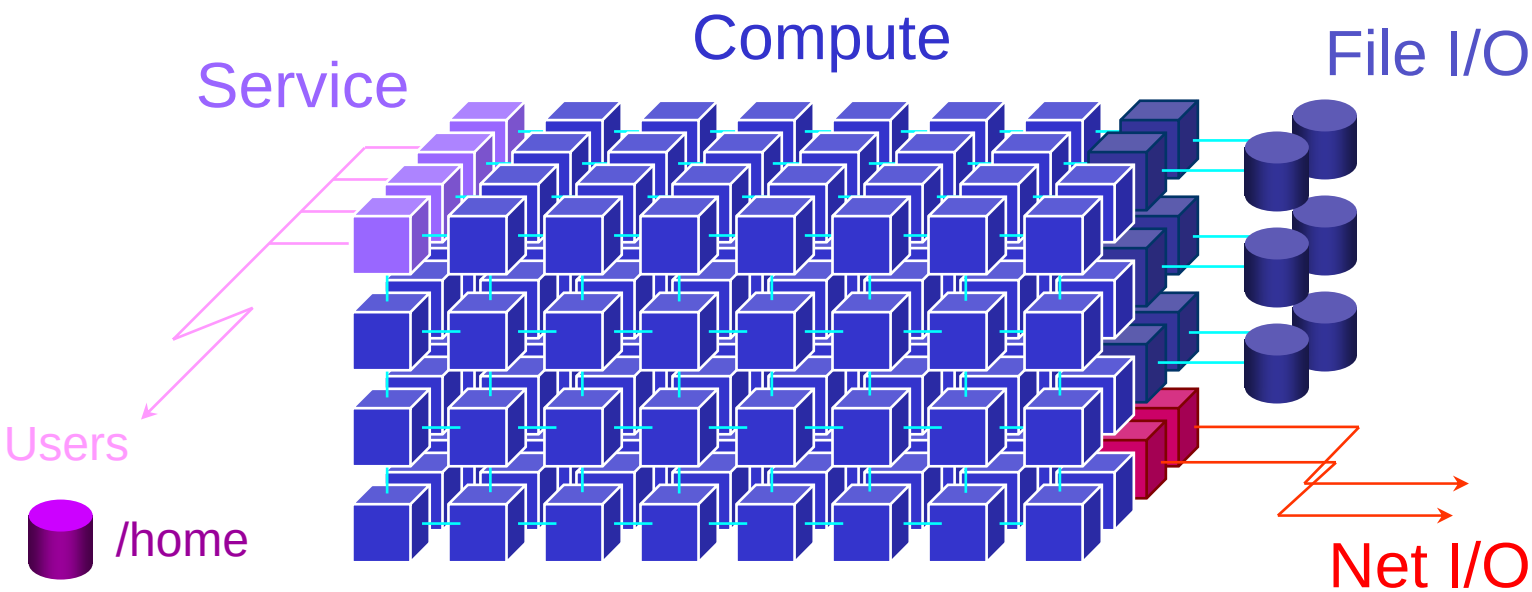
Red Storm Architecture

- True MPP, designed to be a single system
- Distributed memory MIMD parallel supercomputer
- Fully connected 3D mesh interconnect. Each compute node processor has a bi-directional connection to the primary communication network
- 108 compute node cabinets and 10,368 compute node processors (AMD Sledgehammer @ 2.0 GHz)
- ~10 TB of DDR memory @ 333MHz
- Red/Black switching: ~1/4, ~1/2, ~1/4
- 8 Service and I/O cabinets on each end (256 processors for each color)-- may add on-system viz nodes to SIO partition
- 240 TB of disk storage (120 TB per color)

Red Storm Architecture

- Functional hardware partitioning: service and I/O nodes, compute nodes, and RAS nodes
- Partitioned Operating System (OS): LINUX on service and I/O nodes, LWK (Catamount) on compute nodes, stripped down LINUX on RAS nodes
- Separate RAS and system management network (Ethernet)
- Router table-based routing in the interconnect
- Less than 2 MW total power and cooling
- Less than 3,000 ft^2 of floor space

A partitioned, scalable computing architecture



THOR'S HAMMER
RED STORM

Designing for scalable supercomputing

Challenges in:

- Design
- Integration
- Management
- Use

National Nuclear Security Administration

Sandia National Laboratories

A design approach for high-end systems:

SURE:

- Scalability
- Usability
- Reliability
- Expense minimization

SURE Architectural tradeoffs:

- Processor and memory sub-system balance
- Compute vs interconnect balance
- Topology choices
- Software choices
- RAS
- Commodity vs. Custom technology
- Geometry and mechanical design

Sandia Strategies:

-build on commodity
-leverage Open Source (eg Linux)
-Add to commodity selectively (in RS there is basically *one* truly custom part!)
-leverage experience with previous scalable supercomputers

THOR'S HAMMER
RED STORM

Context - Very Large Parallel Computer Systems

Scalability - Full System Hardware and System Software

Usability - Required Functionality Only

Reliability - Hardware and System Software

Expense minimization- use commodity, high-volume parts

SURE poses Computer System Requirements:

NNSA National Nuclear Security Administration

Sandia National Laboratories

Scalability

Hardware:

System Hardware Performance increases linearly with the number of processors to the full computer system size - Scaled Speedup.

- Avoidance of Hardware bottlenecks
  - Communication Network performance
  - I/O System

Machine must be able to support ~30,000 processors operating as a single system.

Usability

Application Code Support:
Software that supports scalability of the Computer System

- Math Libraries
- MPI Support for Full System Size
- Parallel I/O Library
- Compilers

Tools that Scale to the Full Size of the Computer System

- Debuggers
- Performance Monitors

Full-featured OS support at the user interface

Scalability

System Software;

System Software Performance scales nearly perfectly with the number of processors to the full size of the computer (~30,000 processors). This means that System Software time (overhead) remains nearly constant with the size of the system or scales at most logarithmically with the system size.

- Full re-boot time scales logarithmically with the system size.
- Job loading is logarithmic with the number of processors.
- Parallel I/O performance doesn't depend on how many PEs are doing I/O
- Communication Network software must be scalable.
    - No connection-based protocols among compute nodes.
    - Message buffer space independent of # of processors.
    - Compute node OS gets out of the way of the application.

Scaling Analysis for design tradeoffs

Consider three application parallel efficiencies on 1000 processors. What is the most productive way to increase overall application performance?

Case 1: 90% Parallel Efficiency

10X faster processor yields ~5X application code speedup

Cut parallel inefficiency by 10X makes 5% increase in speed

Case 2: 50% Parallel Efficiency

10X faster processor yields <2X application code speedup

Cut parallel inefficiency by 10X makes ~2X increase in speed

Case 3: 10% Parallel Efficiency

10X faster processor yields ~10% application code speedup

Cut parallel inefficiency by 10X makes ~9X increase in speed

System Scalability Driven Requirements

Overall System Scalability - Complex scientific applications such as radiation transport should achieve scaled parallel efficiencies greater than 70% on the full system (~20,000 processors).

- This implies the need for excellent interconnect performance, hardware and software.
- Overall System Reliability - The usefulness of the system is strongly dependent on the time between interrupts.
- Ratio of calculation time to time spent checkpointing should be ~ 20 to 1 to make good progress.
- 100 hour MTBI is desirable

What makes a computer scalable

- Balance in the node hardware:
  - Memory BW must match CPU speed

    Ideally 24 Bytes/flop (never yet done)
  - Folk Theorem:

    Real Speed < Min[(CPU Speed, Mem.BW)/4]
  - Communications speed must match CPU speed
  - I/O must match CPU speeds
  - Scalable System SW(OS and Libraries)
  - Scalable Applications

What doesn't help scalability

- Large Coherent Shared Memory Spaces:
  - Cache Coherency can actually hurt scalability for large #'s of CPUs
  - Shared memory programming methods (eg threads) do not currently scale to large #'s of CPUs
  - Virtual Memory in App's space-- "Paging to where?"

Let's Compare Balance In Parallel Systems

Machine	**Node Speed Rating(MFlops)**	**Link BW (Mbytes/s)**	**Ratio (Bytes/flop)**
ASCI RED	400	800(533)	2(1.33)
T3E	1200	1200	1
ASCI RED**	666	800(533)	(1.2)0.67
Cplant	1000	140	0.14
Blue Mtn*	500	800	1.6
BlueMtn**	64000	**1200 (9600*)**	0.02 (0.16*)
Blue Pacific	2650	300 (132)	0.11 (0.05)
White	24000	2000	0.083
Q*	2500	650	0.2
Q**	10000	400	0.04

Why is Comm's the Killer Concern?

Amdahl's Law limits the scalability of parallel computation

-- but not due to serial work in the application

Why?

Amdahl's Law

$$S_{Amdahl}(N) = [1 + f_s]/[1/N + f_s]$$

where S is the speedup on N processors and f_s is the serial (non-parallelizable) fraction of the work to be done.

Amdahl says that in the limit of an infinite number of processors, S cannot exceed $[1+ f_s]/ f_s$. So, for example if $f_s = 0.01$, S cannot be greater than 101 no matter how many processors are used.

Amdahl's Law

Example:

How big can f_s be if we want to achieve a speedup pf 8,000 on 10,000 processors (80% parallel efficiency)?

Answer:

f_s must be less than 0.000025 !

Amdahl's Law

Contrary to Amdahl & most folks' early expectations, well designed codes on balanced systems can routinely do this well or better !

However in applying Amdahl's Law, we neglected the overhead due to **communications.**

A REAListic Use of Amdahl's Law

The actual scaled speedup is more like

$$S(N) \sim S_{Amdahl}(N)/[1 + f_{comm} \times R_{p/c}],$$

where f_{comm} is the fraction of work devoted to communications and $R_{p/c}$ is the ratio of processor speed to communications speed.

REAL Law Implications

for $S_{real}(N) / S_{Amdahl}(N)$

Let's consider three cases on two computers:

the two computers are identical except that one has an $R_{p/c}$ of 1 and the second an $R_{p/c}$ of 0.05

The three cases are f_{comm} = 0.01, 0.05 and 0.10

REAL Law Implications S(N) / $S_{Amdahl}(N)$

$R_{p/c}$ \ f_{comm}	0.01	0.05	0.10
1.0	0.99	0.95	0.9
0.05	0.83	0.50	0.33

Bottom line:

A "well-balanced" architecture is nearly insensitive to communications overhead

By contrast a system with weak communications can lose over half its power for applications in which communications is important

THOR'S HAMMER
RED STORM

Applications Scalability Driven Requirements

High Performance Machine Interconnect

- Bandwidth - at least 1 B/F
- MPI Latency (ping-pong divided by 2) - ~3000 CPU clocks

System Software Scalability

- -No large SMPs-- N^2 cost and overhead scaling
- -No connection based networks - N^2 scaling
- -Source based routing preferred; tables acceptable
- -Compute Node OS - No time sharing of nodes, No compute node paging, No sockets, No spurious demons,
- -Minimize number of OS initiated interrupts.
- -Keep it simple

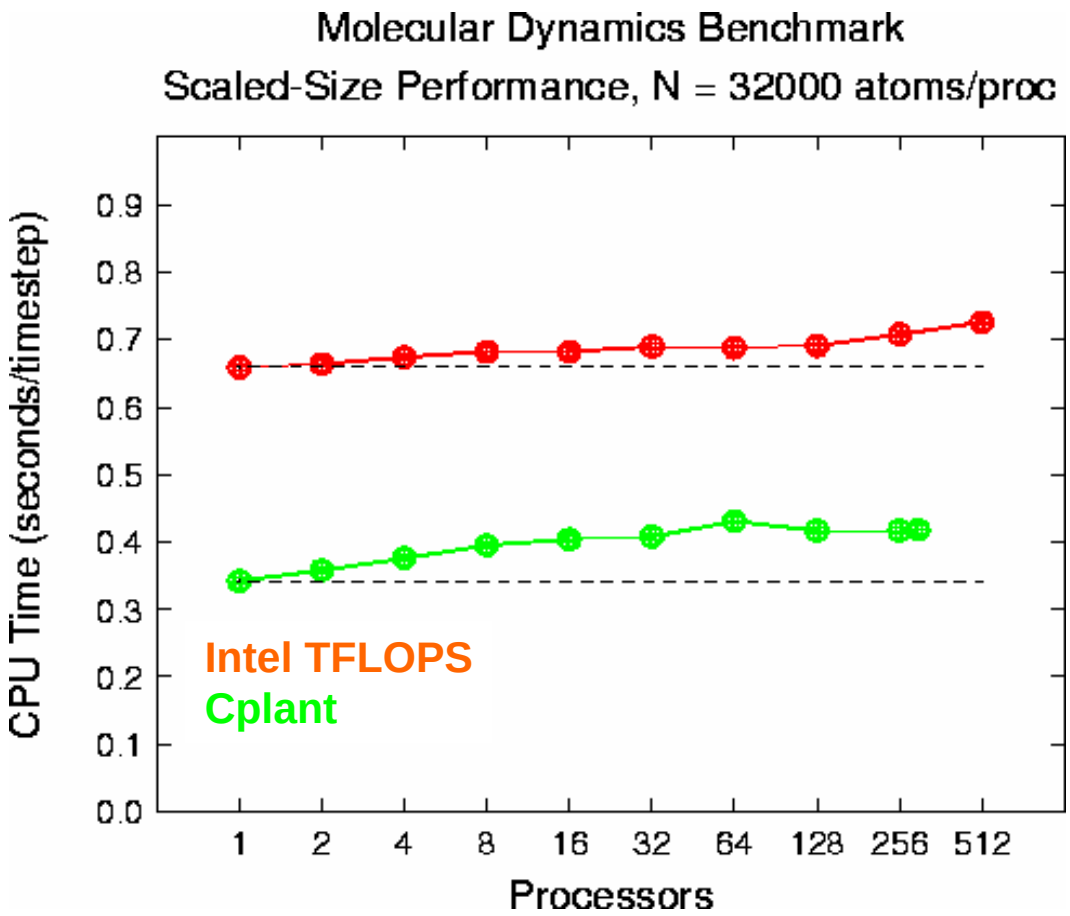
Overall System Reliability

- -System MTBI of 50 hrs or more to get useful work done

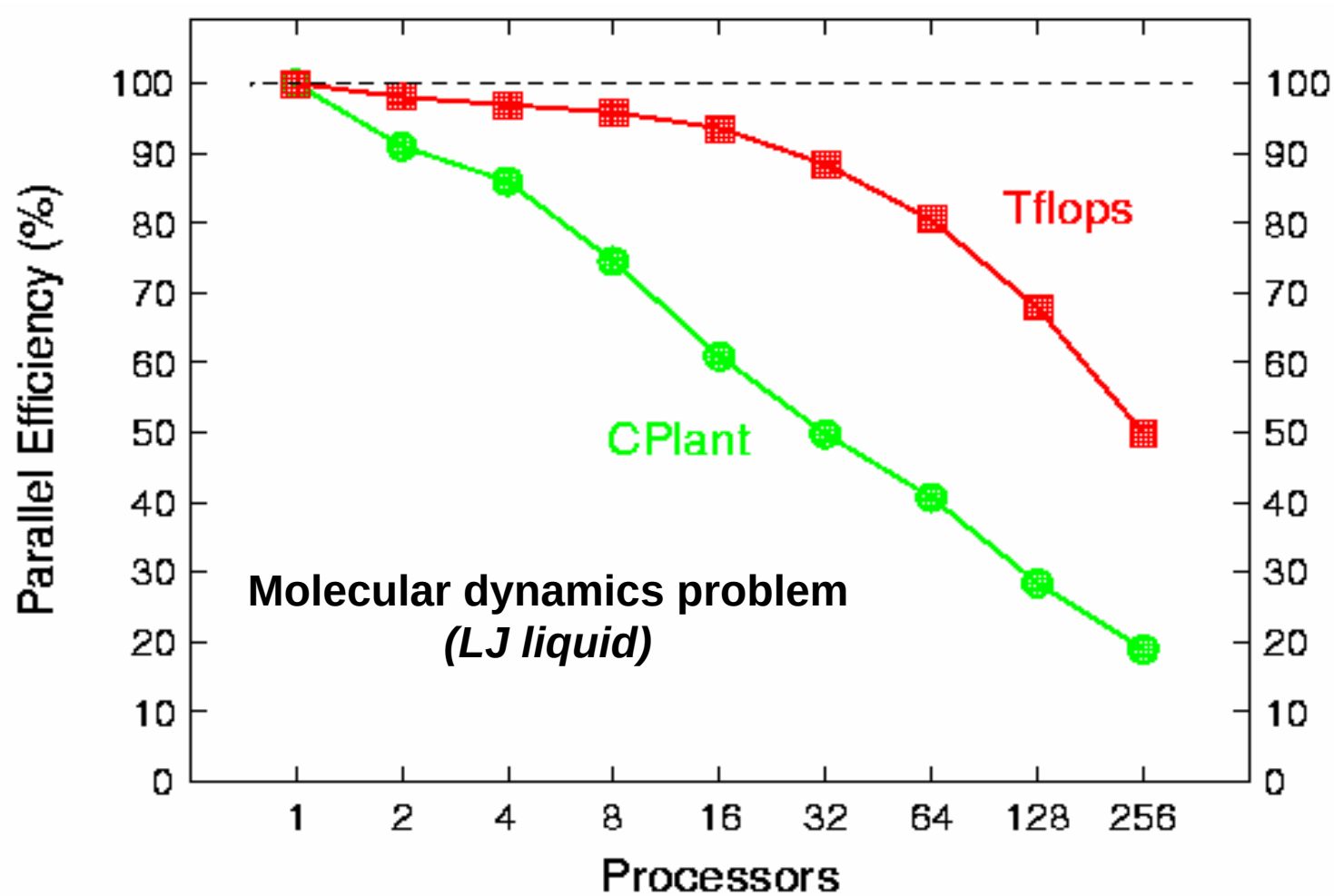
National Nuclear Security Administration

Sandia National Laboratories

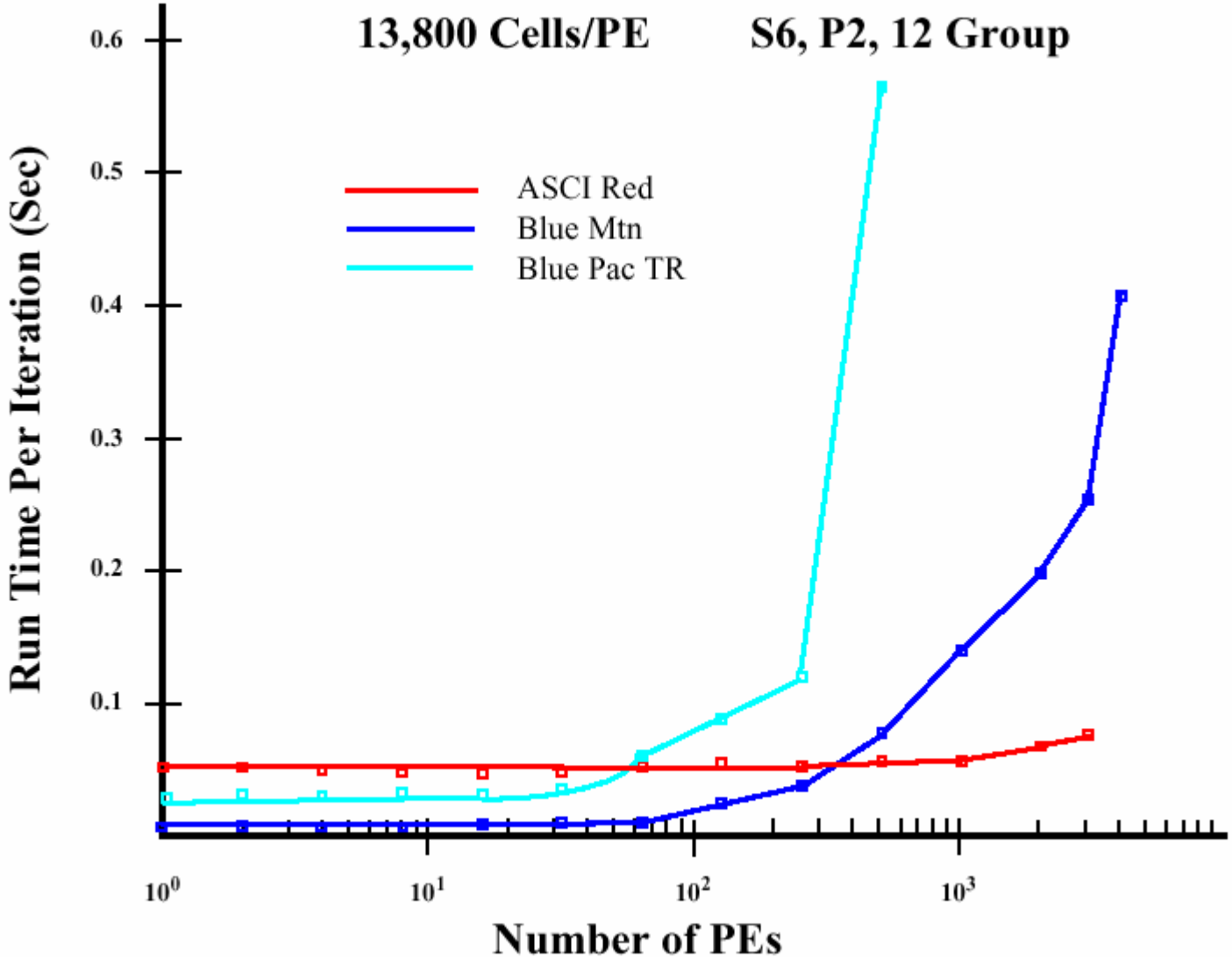
Scaled problem performance



Fixed problem performance



Parallel S_n Neutronics (provided by LANL)

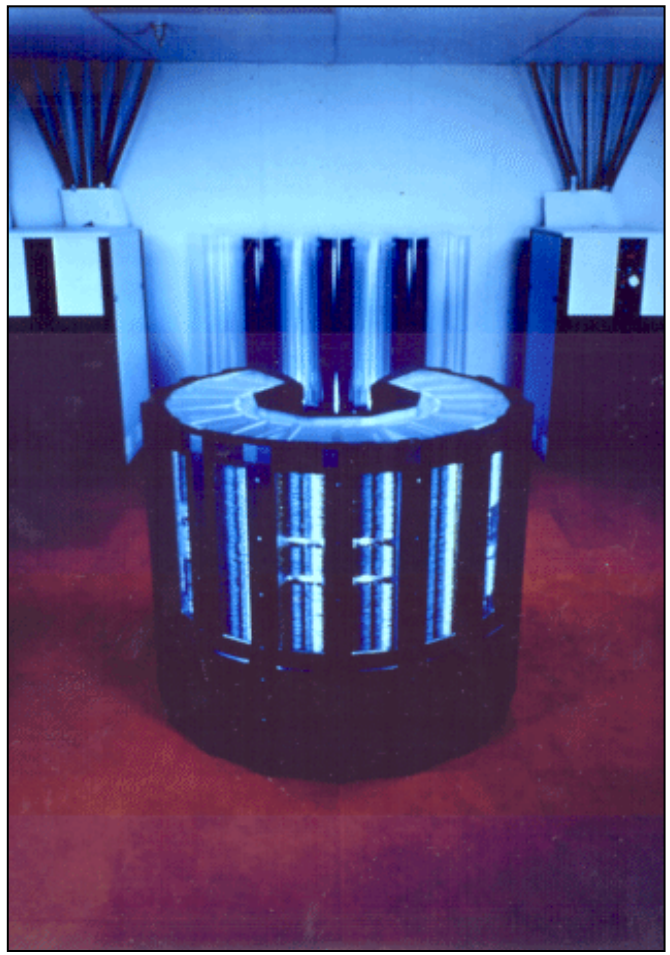


Conclusion:

For most large scientific and engineering applications the performance is more determined by parallel scalability and less by the speed of individual CPUs. There must be balance between processor, interconnect, and I/O performance to achieve overall performance.

To date, only a few tightly-coupled, parallel computer systems have been able to demonstrate a high level of scalability on a broad set of scientific and engineering applications.

The balance bible
In the beginning …



… and then there was:

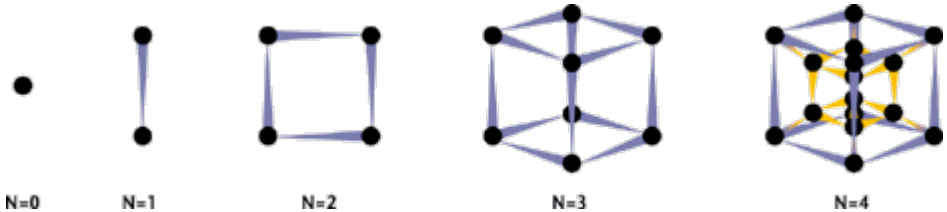
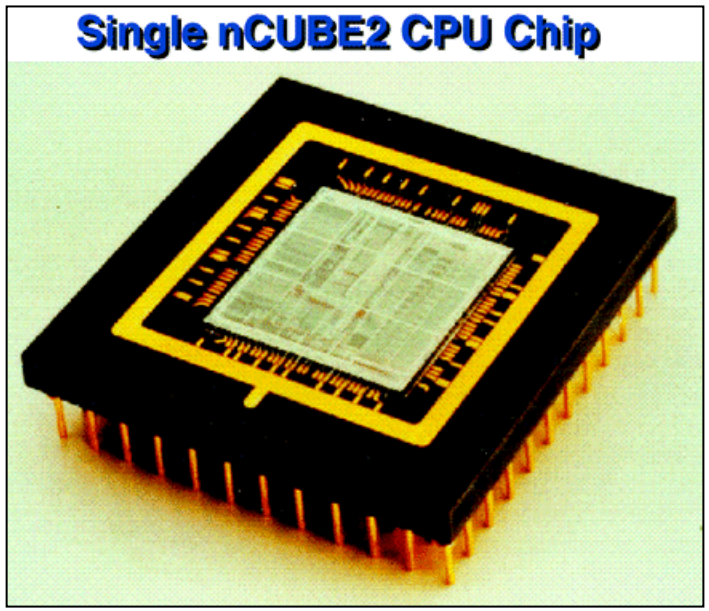
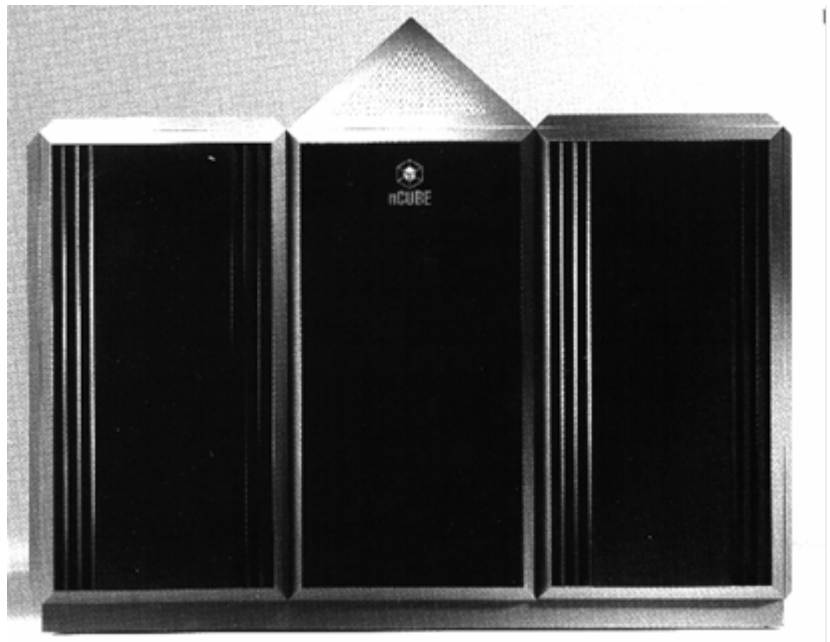


Figure 1: Two hypercubes of the same dimension, joined together, form a hypercube of the next dimension. N is the dimension of the hypercube.





Massively Parallel Processors

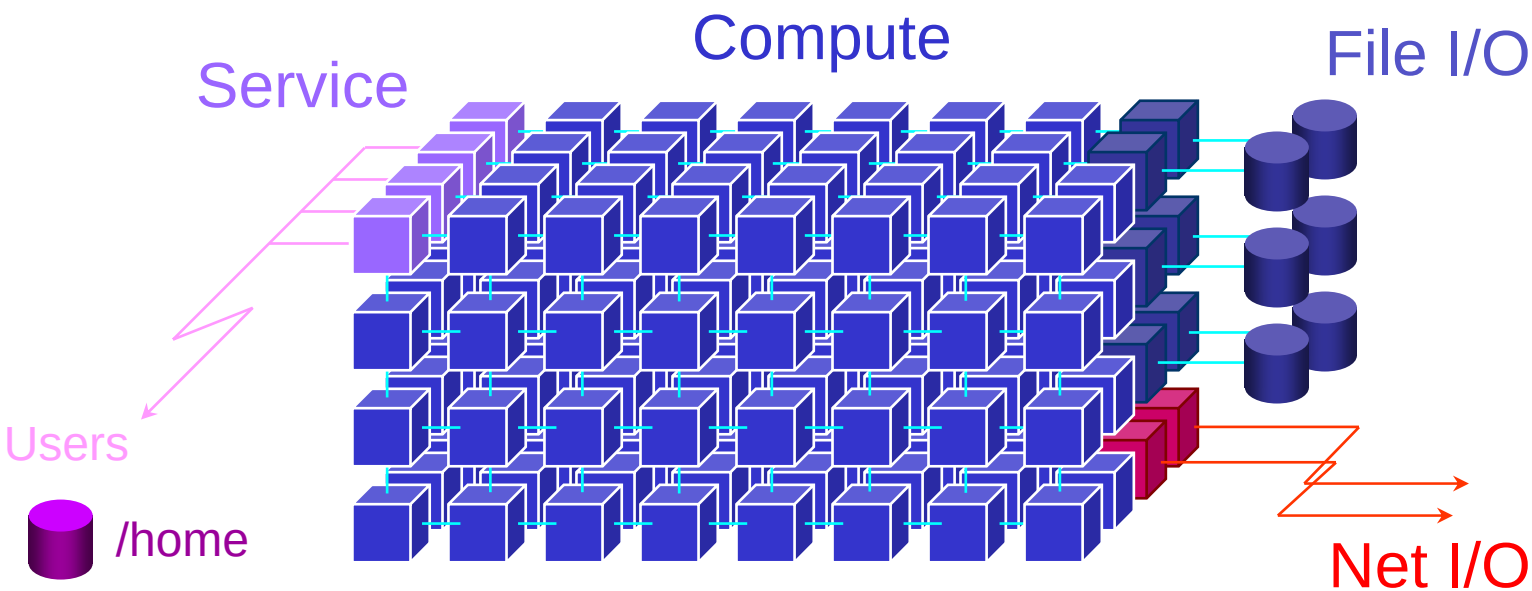
- Intel Paragon
- 1,890 compute nodes
- 3,680 i860 processors
- 143/184 GFLOPS
- 175 MB/sec network
- SUNMOS LWK

Intel Tflops

4,576 compute nodes

- 9,472 Pentium II processors
- 2.38/3.21 TFLOPS
- 400 MB/sec network
- Puma/Cougar LWK

A scalable computing architecture



Our new MPP architecture

- Scalability
- Reliability
- Usability
- Simplicity
- Cost effectiveness

Building blocks

Architecture-- cost and scalability

Processor

Memory Subsystem

Interconnect-- topology and scalability

I/O System-- scalability

RAS-- scalability

Look at Processor/Memory issues

What about vectors?

We looked at them very hard

We decided that for our applications commodity processors were a better choice.

What about vectors?

Let p be the fraction of work that is "vectorizable."

Let N be the average speed advantage on long vectors relative to a competing superscalar processor

Let M be the average speed advantage of the superscalar processor over the vector processor on work that doesn't vectorize

THOR'S HAMMER
RED STORM

If W is the work to be done and s is the (superscalar) speed, then

$$S = T_S / T_V$$

$$S = [\ W / s] /[pW / (s\ N)\ \ + (1-p)W / (s/M)\ ]$$

$$S = [\ p/N + M(1-p)\ ]^{-1}.$$

National Nuclear Security Administration

Sandia National Laboratories

In comparing a Pentium-4 @2GHz with the SX-6 processor we may guess that N=M=4

So,

$$S = [\ p/4 + 4(1-p)\ ]^{-1}.$$

For p < 0.8, we actually get slowdown!

In comparing a Pentium-4 @2GHz with the SX-6 processor what is our experience?

Our two major hydrocodes,Alegra and CTH, run about the same speed on an SX-6 as they do on the Pentium-4.

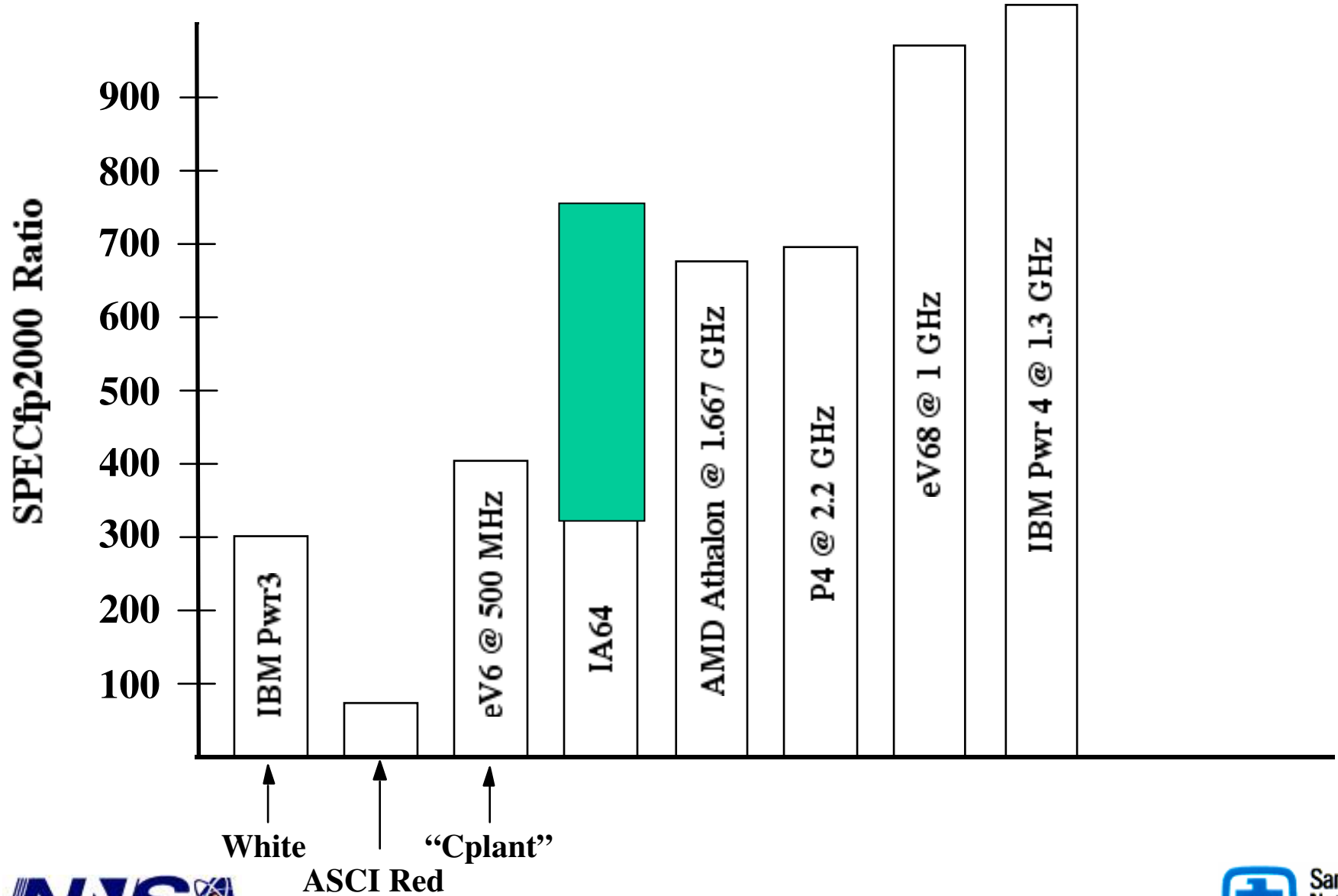
Our principal Monte Carlo Rad Transport code, ITS, runs about 1/10th as fast on the ES as on a P-4.

What about benchmarks?

SpecFP-2000 turns out to have a benchmark that is representative of DOE finite-element app's: the Sam Key benchmark.

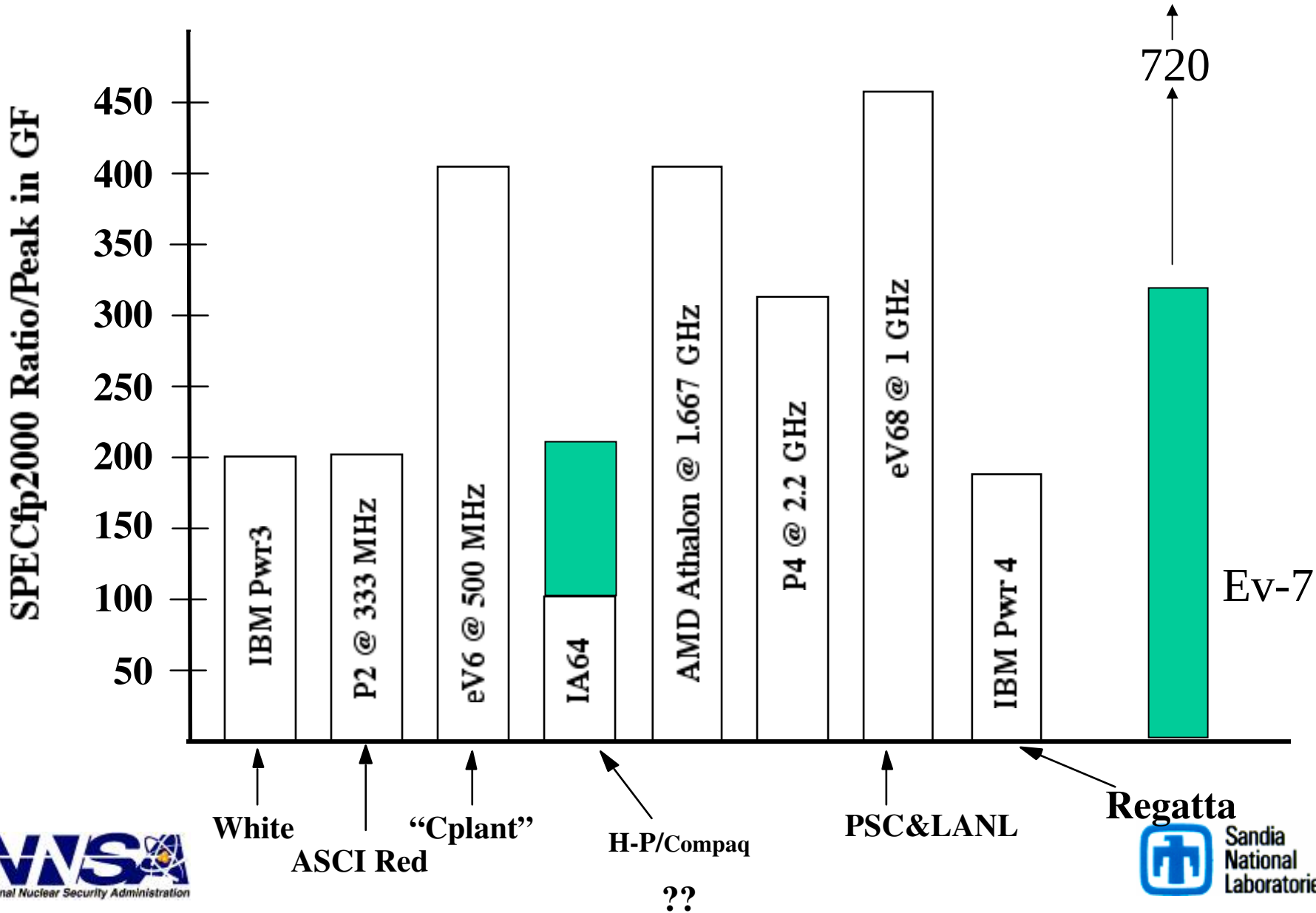
Let's look at processor performance on Sam's benchmark.

Processor Performance
SPECfp2000 - Sam Key Benchmark (FMA-3D)



THOR'S HAMMER
RED STORM
NNSA National Nuclear Security Administration
Sandia National Laboratories

Normalized Processor Performance
SPECfp2000 - Sam Key Benchmark (FMA-3D)



THOR'S HAMMER
RED STORM

Q

Cplant

RED

White

Purple-?

H.-P.

processor	Peak in GF	SpecFP-2000 ratio	Normalized SpecFP-2000 -ratio	Peak needed to match EV7 based 20 TF*
EV-7 @1.25Ghz*	2.5	~1800*	720*	20
AMD Hammer @2.5 Ghz*	2.5	~1300*	520*	27.7
EV-68 @1.0 Ghz	2.0	917	458.5	31.4
Athalon@1.67Ghz	1.67	680	407.9	35.3
EV-6 @500Mhz	1.0	406	406	35.5
P4@ 2.2 Ghz	2.2	699	317	45.4
P2@333Mhz	0.333	~70**	210**	68.6
Pwr 3@ 375 Mhz	1.5	302	201.3	71.5
Pwr 4@1.3 Ghz	5.2	978	188	76.6
Itanium-2 @1.0 Ghz	4.0	776	194	73.2

SpecFP -2000(FMA-3D) based estimates of Required Peak MPP Speeds

NNSA National Nuclear Security Administration

*** On this benchmark**

Sandia National Laboratories

A couple of real National Security App’s

Alegra Grind times

EV-7 @ 1.2 Ghz (2.4 GF) 163 microsecs
Power-4 @ 1.3 Ghz (5.2 GF) (176) 213 microsecs
ASCI RED cpu @ 0.333 GHz 2216 microsecs
IA32 P4 @2.0 GHz 261 microsecs
IA64 Itanium-1@733MHz (2.9 GF) 2995 microsecs
IA64 Itanium-2 @1GHz-- about the same as P4@2Ghz
Earth Simulator ~ same speed as a 2 GHz IA32

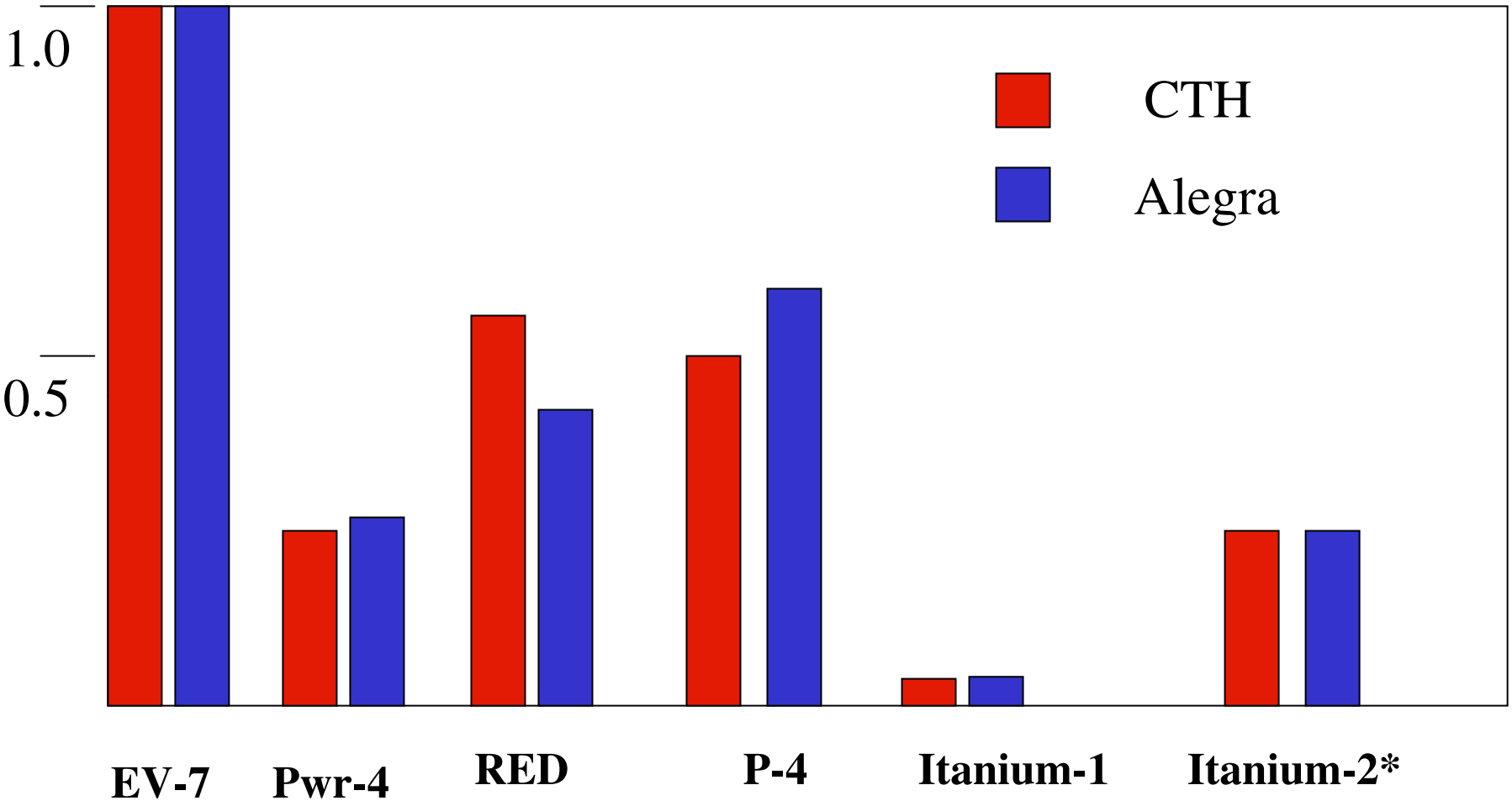
CTH Grind times

EV-7 @ 1.2 Ghz (2.4 GF)	8.9 microsecs
Power-4 @1.3 Ghz (5.2 GF)	12.5 microsecs
ASCI RED cpu @ 0.333 GF	90.5 microsecs
IA32 P4 @2.0 GHz	17.4 microsecs
SX-6 (ES)	~15 microsecs
IA64 Itanium-1@733MHz (2.9 GF)	143 microsecs

IA64 Itanium-2 @1GHz about the same as P4@2Ghz

Earth Simulator ~ 1.2 speed of IA32 @2.0 GHz

Normalized Relative Speed



THOR'S HAMMER
RED STORM

Take away message:

The Alpha family are the clear performance winners.

The IA32 is also pretty good; as is AMD Athlon

Our tests show tht the AMD "Hammer" is between IA32 and the Alpha with a price point more like the IA32

Interconnect

Connection Choices:

1. PCI/PCIX based processor connections-- adequate
2. Memory sub-system based connections --much better (e.g. Marvel interconnects and AMD Hypertransport Layer)

Interconnect

Switch Fabric:

1. Commercial networks:

   Myrinet (cheaper, fairly fast)

   Quadrix (more costly; currently faster)

   Gigabit Ethernet (Cheap, Not A good idea for scaling to 10^4 nodes)

2. Custom interconnects:

   e.g., IBM; ASCI Red; T3E;SGI; Cray, …

Interconnect Tradeoffs

Switch Fabric:

1. Commercial networks:

   Quadrix can get within a factor 2--4 of the latency requirements and within a factor of 4 of the bandwidth targets for Red Storm. Cabling costs may be higher than for custom interconnects.

2. Custom interconnects:

   Easily meet the BW and latency requirements for Red Storm. Need to pay the NRE costs somehow; takes 24--30 months to bring it to production

Interconnect Choice

Custom interconnects if possible:

- If cost & schedule can be controlled, this is the best solution
  - should permit rolling upgrades
  - meets all scaling targets

Quadrics (with mods) might be an acceptable alternative

Interconnect Topology

1. Large Switches

   Full Xbar (Some folks' Holy Grail)

   IBM Colony & follow-on

   Quadrics Fat Tree

   Myricom Clos Switch

2. Mesh or Mesh-like

   e.g., Cplant, ASCI Red; T3E; Cray SV-2*, …•

Topology Choice

Switch Topology (modulo photonic switches):

1. Large Switches

   These are excellent for modest-size clusters. Their cost grows faster than linearly and the cabling issues grow enormously difficult for large systems

2. 3D meshes

   Cost is linear in both switches and in cables. For our applications on large system, this is by far the best choice.

Reliability for Scientific and Engineering Applications

What is Reliability:

High Mean Time Between Interrupts for hardware and system software

High Mean Time Between errors/failures that affect users

What it is not:

High availability

Our take on how to get Reliability: System Software

Partitioned Operating System (OS)

- Service Partition - Full function OS
- I/O Partition - Full function OS
- Compute Partition - Light Weight Kernel OS
- System Partition - System control functions
- Provide only needed functionality for each partition.

System Software Adaptation

- Automatic OS re-boots on OS failures
- Automatic system reconfiguration for hardware failures

Keep it Simple

How to Get Reliability-- Hardware

A full system approach - Machine must be looked at as a whole and not a collection of separate parts or sub-systems.

Hardware

- Partitioning based on function
- Redundant Components
- Error Correction
- Hot Spares
- Integrated Full System Monitoring and Scalable Diagnostics
- Preventive Maintenance
- High volume parts as appropriate

Is a 50 Hour MTBI Possible?

ASCI Red Experience in 1999
Hardware MTBI - > 900 hours
System Software MTBI - > 40 hours
ASCI Red has over 9000 processors
~4 hours Preventive Maintenance is performed per week
Integrated full system monitoring capability
Almost all unscheduled interrupts occur as a result of OSF/1 (TOS) failures
(We believe that the software MTBI would be much better if Intel had remained in the supercomputer business.)

Expense minimization

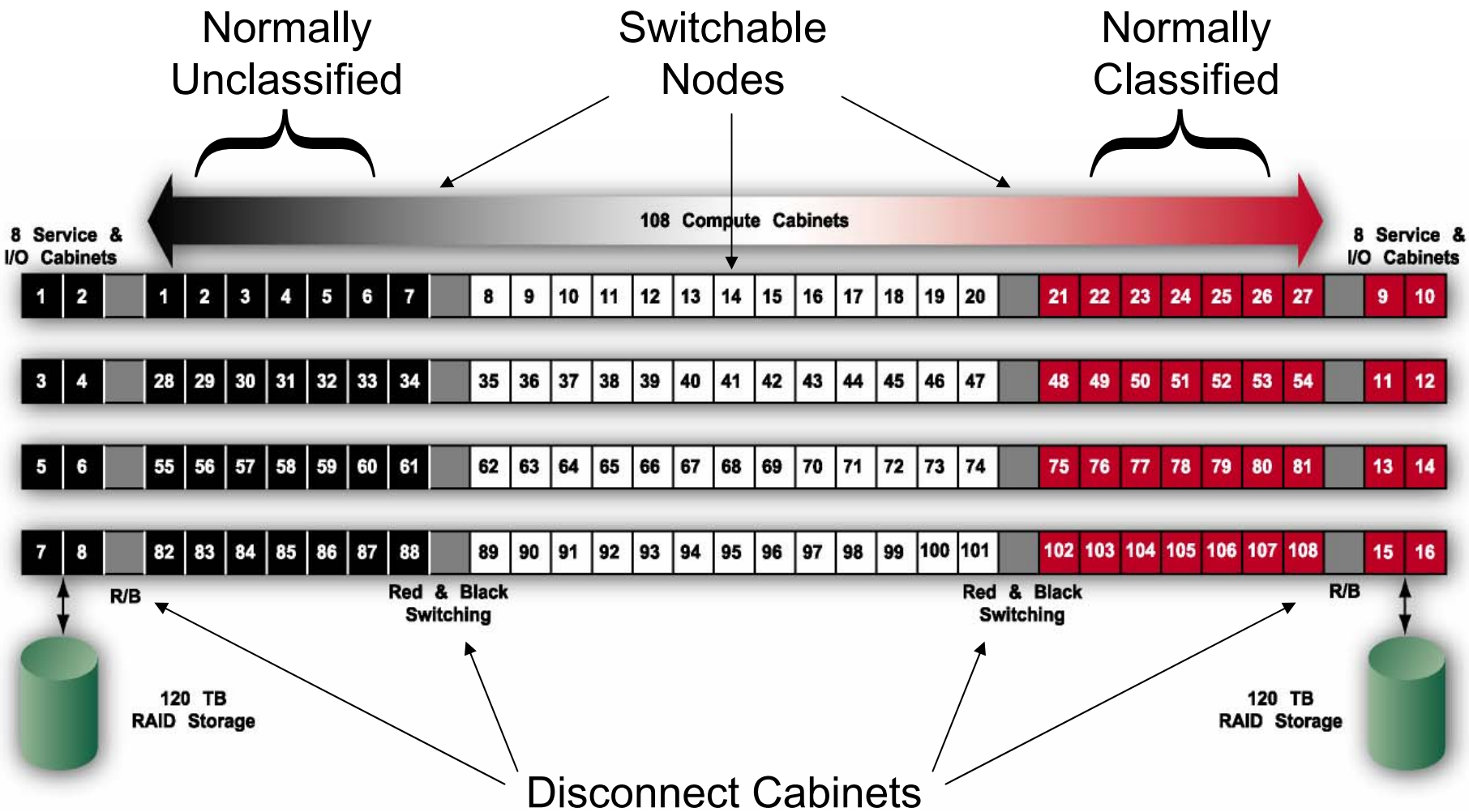
1. Use high-volume parts where possible
2. Minimize power requirements
   - Cuts operating costs
   - Reduces need for new capital investment
3. Minimize system volume
   - Reduces need for large new capital facilities
4. Use standard manufacturing processes where possible-- minimize customization
5. Maximize reliability and availability/dollar
6. Maximize scalability/dollar
7. Design for integrability

New Building for Thor's Hammer

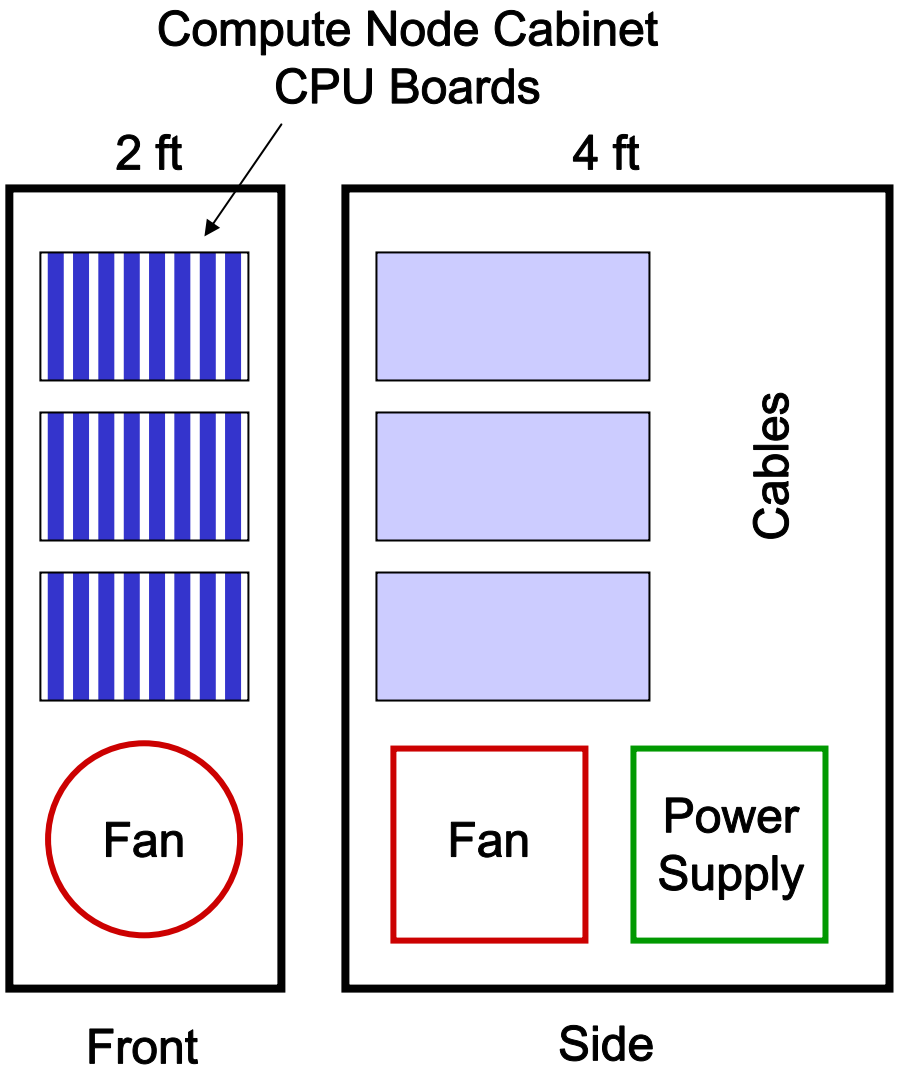
Thor's Hammer Topology

- 3D-mesh Compute node topology:
  - 27 x 16 x 24 (x, y, z) – Red/Black split: 2,688 – 4,992 – 2,688
- Service and I/O node topology
  - 2 x 8 x 16 (x, y, z) on each end
  - 128 full bandwidth links to Compute Node Mesh (384 available)
- Visualization Partition:
  - Subset of Service and I/O partition
  - May have about 256 processors on each color

System Layout
(27 x 16 x 24 mesh)



Thor's Hammer Cabinet Layout



- Compute Node Cabinet
  - 3 Card Cages per Cabinet
  - 8 Boards per Card Cage
  - 4 Processors per Board
  - 4 NIC/Router Chips per Board
  - N + 1 Power Supplies
  - Passive Backplane
- Service and I/O Node Cabinet
  - 2 Card Cages per Cabinet
  - 8 Boards per Card Cage
  - 2 Processors per Board
  - 4 NIC/Router Chips per Board
  - Dual PCI-X for each processor
  - N + 1 Power Supplies
  - Passive Backplane

Thor's Hammer Performance

- Peak of ~40 TF based on 2 floating point instruction issues per clock.
- We required 7-fold speedup but based on our benchmarks expect performance will be 8-9 time faster than ASCI Red.
- Expected MP-Linpack performance: >>20 TF (14 is required)
- Aggregate system memory bandwidth: ~55 TB/s
- Interconnect Performance:
  - Latency <2 μs (neighbor), <5 μs (full machine)
  - Link bandwidth ~ 6.0 GB/s bi-directional (sustained 4.1 GB/s)
  - Minimum bi-section bandwidth ~2.3 TB/s (peak) 1.6 TB/s (sustained)

Thor's Hammer Performance

- I/O System Performance
  - Sustained file system bandwidth of 50 GB/s for each color
  - Sustained external network bandwidth of 25 GB/s for each color
- Node memory system
  - Page miss latency to local memory is ~80 ns
  - Peak bandwidth of ~5.4 GB/s for each processor

Red Storm System Software

- Operating Systems
  - LINUX on service and I/O nodes
  - Sandia's LWK (Catamount) on compute nodes
  - LINUX on RAS nodes
- Run-Time System
  - Logarithmic loader
  - Fast, efficient Node allocator
  - Batch system – PBS
  - Libraries – MPI, I/O, Math
- File Systems being considered include
  - PVFS – interim file system
  - *Lustre* – Pathforward support,
  - *Panassas*
  - …

Red Storm System Software

- Tools
  - All IA32 Compilers, all AMD 64-bit Compilers – Fortran, C, C++
  - Debugger – *Totalview (also examining alternatives)*
  - Performance Monitor
- System Management and Administration
  - Accounting
  - RAS GUI Interface

Comparison of ASCI Red and Red Storm

	ASCI Red	Red Storm
Full System Operational Time Frame	June 1997 (processor and memory upgrade in 1999)	August 2004
Theoretical Peak (TF)	3.15	41.47
MP-Linpack Performance (TF)	2.379	>20 (estimated)
Architecture	Distributed Memory MIMD	Distributed Memory MIMD
Number of Compute Node Processors	9,460	10,368
Processor	Intel P II @ 333 MHz	AMD Opteron @ 2 GHz
Total Memory	1.2 TB	10.4 TB (up to 80 TB)
System Memory Bandwidth	2.5 TB/s	55 TB/s
Disk Storage	12.5 TB	240 TB
Parallel File System Bandwidth	1.0 GB/s each color	50.0 GB/s each color
External Network Bandwidth	0.2 GB/s each color	25 GB/s each color

Comparison of ASCI Red and Red Storm

	ASCI Red	RED STORM
Interconnect Topology	3D Mesh (x, y, z) 38 x 32 x 2	3D Mesh (x, y, z) 27 x 16 x 24
Interconnect Performance MPI Latency Bi-Directional Bandwidth Minimum Bi-section Bandwidth	15 µs 1 hop, 20 µs max 800 MB/s 51.2 GB/s	2.0 µs 1 hop, 5 µs s max 6.0 GB/s 2.3 TB/s
Full System RAS RAS Network RAS Processors	10 Mbit Ethernet 1 for each 32 CPUs	100 Mbit Ethernet 1 for each 4 CPUs
Operating System Compute Nodes Service and I/O Nodes RAS Nodes	Cougar TOS (OSF1) VX-Works	Catamount LINUX LINUX
Red/Black Switching	2260 – 4940 – 2260	2688 – 4992 - 2688
System Foot Print	~2500 ft^2	~3000 ft^2
Power Requirement	850 KW	1.7 MW

Red Storm Project

- 23 months, design to FPS!
- System software is a joint project between Cray and Sandia
  - Sandia is supplying Catamount LWK and the service node run-time system
  - Cray is responsible for Linux, NIC software interface, RAS software, file system software, and *Totalview* port
  - Initial software development is being done on a cluster of workstations with a commodity interconnect. ASCI Red support machines will be used for system software development and testing
- System design is going on now
  - Cabinets-- exist
  - NIC/Router-- initial design and independent architectural review done
- Full system installed and turned over to Sandia in stages culminating in August 2004

THOR'S HAMMER

RED STORM

NNSA
National Nuclear Security Administration

Sandia National Laboratories