Sustained Petascale: The Next MPI Challenge

Al Geist
Chief Technology Officer
Oak Ridge National Laboratory

EuroPVM-MPI 2007

Paris France
September 30-October 3, 2007

Managed by UT-Battelle
for the Department of Energy

Research Sponsored by DOE Office of Science

OAK RIDGE National Laboratory

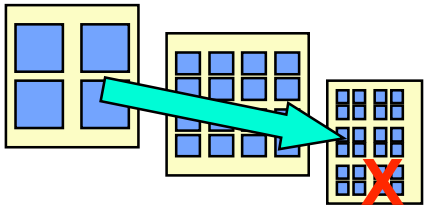
Outline

Sustained petascale systems will soon be here!

- 10-20 PF peak systems in NSF and DOE around 2011
- Time for us to consider the impact on MPI, OpenMP, others…

Disruptive shift in system architectures, a similar shift from vector computers 15 years ago drove the creation of PVM and MPI

- Heterogeneous nodes
- Multi-core chips
- Million or more cores



What is the impact on MPI ?

- New features for performance and application fault recovery?
- Hybrid models using a mix of MPI and SMP programming?

Productivity - how hard does sustained petascale have to be?

- Debugging and performance tuning tools
- Validation and knowledge discovery tools

Sustained Petascale Systems by 2011

DOE and NSF plan to deploy computational resources needed to tackle global challenges	Vision: Maximize scientific productivity and progress on the largest scale computational problems

· Energy, ecology and security
· Climate change
· Clean and efficient combustion
· Sustainable nuclear energy
· Bio-fuels and alternate energy
	· DOE Leadership Computing Facilities
	  · 1 PF ORNL
	  · ½ PF ANL
	· NSF Cyberinfrascructure
	  · Track-1 NCSA 10+ PF
	  · Track-2 TACC 550 TF
	  · Track-2 UT/ORNL 1 PF

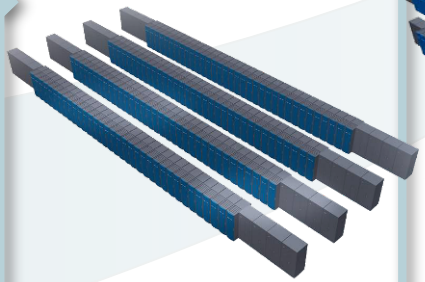
Eg. ORNL Leadership Computing Facility Hardware roadmap

Cray XT4: 119 TF
11,706 nodes
23,412 cores
46 TB

FY2007

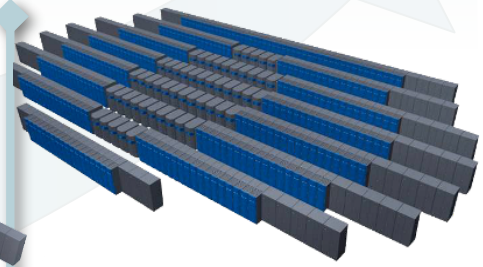
Cray XT4: 250+ TF
11,706 nodes
36,004 cores
71 TB

FY2008



Cray XT5: 1 PF
24,576 nodes
98,304 cores
175 TB

FY2009



Cray Cascade: 20 PF
6,224 nodes
800,000 cores
1.5 PB

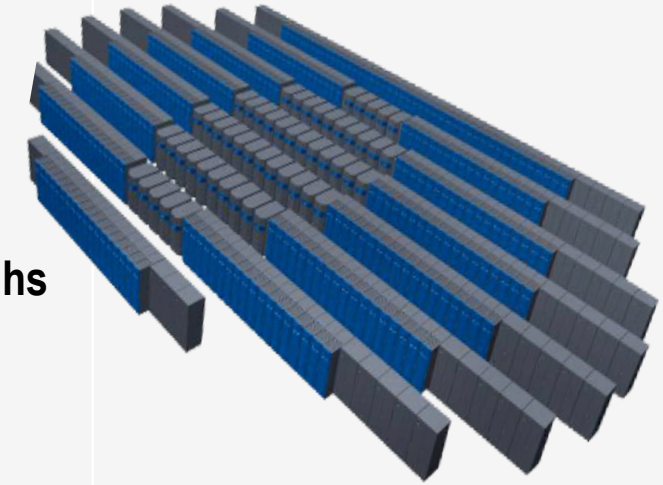
FY2011

Maximizing usability by designing based on large scale science needs

ornl

Let application needs drive the system configuration

- 22 application walkthroughs were done for codes in:
  - Physics
  - CFD
  - Biology
  - Geosciences
  - Materials, nanosciences
  - Chemistry
  - Astrophysics
  - Fusion
  - Engineering



Walkthrough analysis showed:

- Injection bandwidth and interconnect bandwidth are key bottlenecks to sustained petascale science

- 6,224 SMP nodes, each with 8 Opterons
- 1.5 PB, globally addressable across system (256 GB per node)
- Global bandwidth: 234 TB/s (fat tree + hypercube)
- Disk: 46 PB; archival: 0.5 EB
- Physical size
  - 264 cabinets
  - 8,000 ft^2 of floor space
  - 15 MW of power

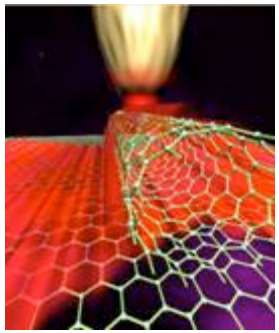
MPI performance has important role in avoiding these bottlenecks

Scientists are making amazing discoveries on the ORNL Leadership Computers

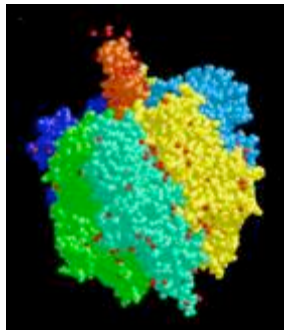
Focus on computationally intensive projects of large scale and high scientific impact

Provide the capability computing resources (flops, memory, dedicated time) needed to solve problems of strategic importance to the world.

ORNL 250 TF Cray XT4
December 2007

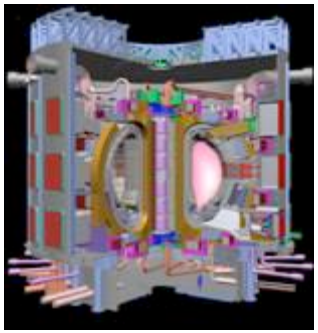


Design of innovative nano-materials



Understanding of microbial molecular and cellular systems

100 yr Global climate to support policy decisions



Predictive simulations of fusion devices

Science Drivers for Sustained PF
New problems from Established Teams

ornl

Science Domains	Science Driver
Nanoscience	Designing high temperature superconductors, magnetic nanoparticles for ultra high density storage
Biology	Can efficient ethanol production offset the current oil and gasoline crisis?
Chemistry	Catalytic transformation of hydrocarbons; clean energy and hydrogen production and storage
Climate	Predict future climates based on scenarios of anthropogenic emissions
Combustion	Developing cleaner-burning, more efficient devices for combustion.
Fusion	Plasma turbulent fluctuations in ITER must be understood and controlled
Nuclear Energy	Can all aspects of the nuclear fuel cycle be designed virtually? Reactor core, radio-chemical separations reprocessing, fuel rod performance, repository
Nuclear Physics	How are we going to describe nuclei whose fundamental properties we cannot measure?

MPI Dominates the Largest HPC Applications

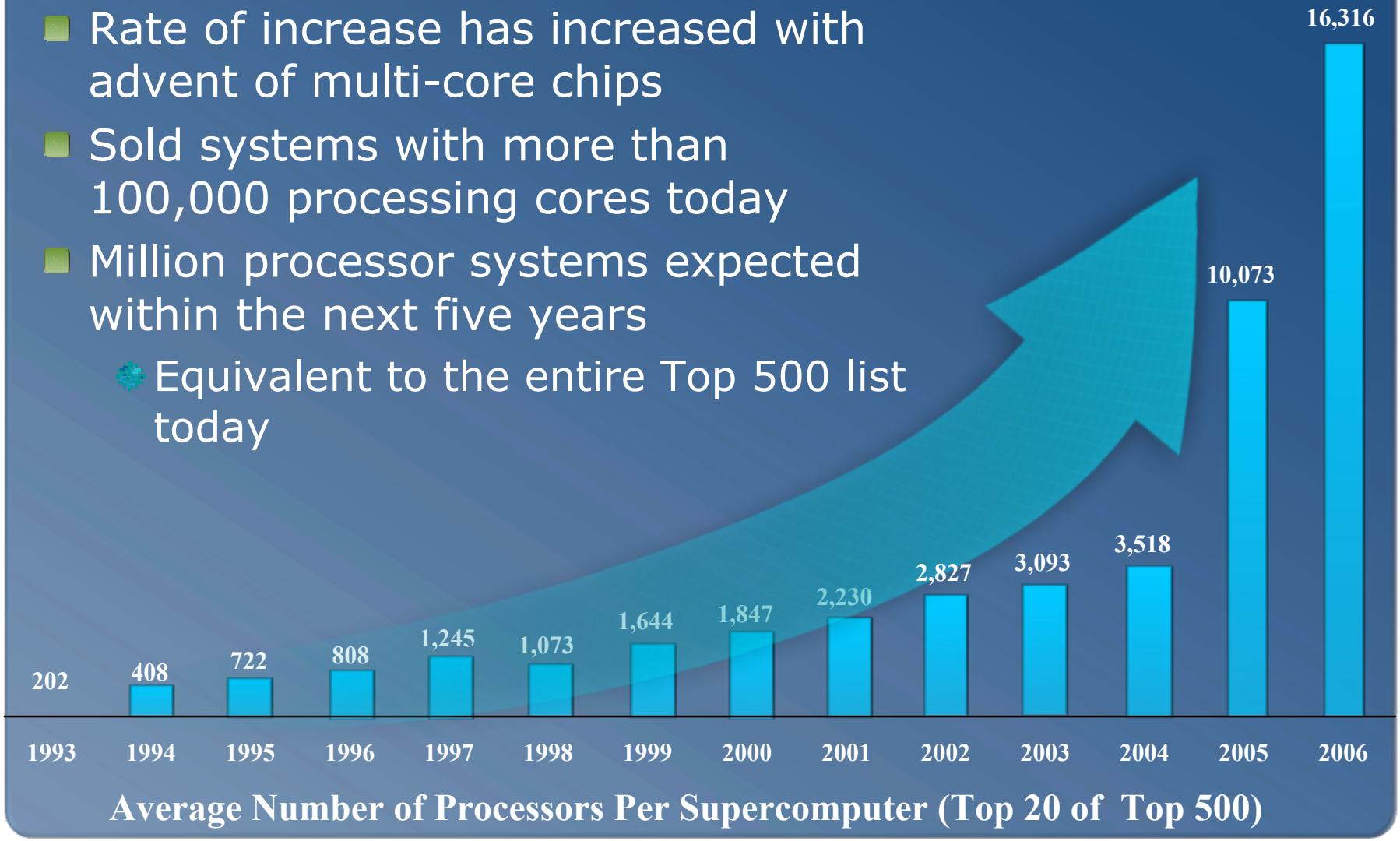
	Language/Prog Models										RT		Dev		I/O						Math												Sys					
	Fortran77	Fortran90	Fortran95	C/C++	CAF	OpenMP	MPI	MPI2	Global Arrays	Python	ARMCI	BLACS	GNU Make	CCA-Tools	HDF5	PHDF5	MPI-IO	netCDF	pnetCDF	XML	BLAS	Cray Scilib	FFTPACK	FFTW	LAPACK	MUMPS	PBLAS	PEIGS	Parmetis	PARPACK	PetSC	Scalapack	SGI SCSL	SPRNG	ZOLTAN	ctime	tar	CP	Mkdir	Rm	Other
LSMS	1	1		1				1					1		1					1	1				1										1					
QMC		1					1														1				1								1							
CHIMERA		1					1								1				1						1															
POP/CICE		1			x	x	1											1																						
SD3		1					1																													1	1	1	x	
GTC		1		1		x	1								1	1	1	1		1										1										1
MADNESS		1					1														1																			
AORSA	1	1										1						1					1				1				1									
LAMMPS				1			1																	1																
FLASH		1		1			1			1			1		1				1																					
Milc/Chroma				1			1																																	1
PFLOTRAN		1					1														1									1										
QBOX				1			1					1								1	1			1	1						1									
CAM		1		1	x													1				1										1							x
CCSD		1					1										1				1																		
T3P				1			1											1								1			1						1					1
VASP		1					1														1										1								x
NEWTRNX		1		1						1			1	1	1										1				1										
NWChem	1			1					1		1		1								1		1					1			1								
OReTran			1				1														1				1														
CASINO		1					1														1																		1

Must have Can use

Multi-core is driving scaling needs

ornl

- Rate of increase has increased with advent of multi-core chips
- Sold systems with more than 100,000 processing cores today
- Million processor systems expected within the next five years
  - Equivalent to the entire Top 500 list today



Multi-core – How it affects MPI

ornl

The core count rises but the number of pins on a socket is fixed. This accelerates the decrease in the bytes/flops ratio per socket.

The bandwidth to memory (per core) decreases

- Utilize the shared memory on socket
- Keep computation on same socket
- MPI take advantage of core-core communication

The bandwidth to interconnect (per core) decreases

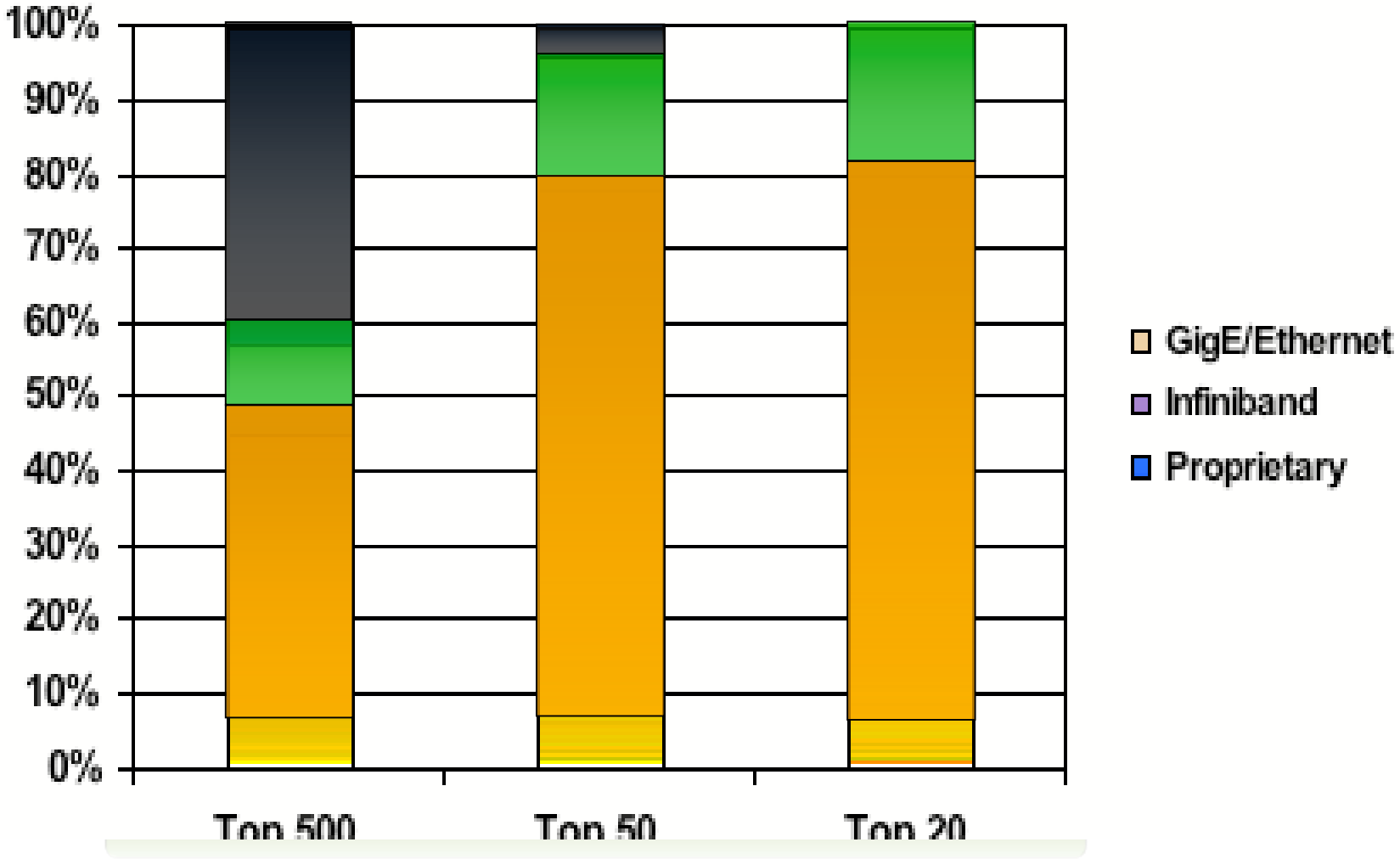
- Better MPI collective implementations
- Stagger message IO to reduce congestion
- Aggregate messages from multiple cores

The bandwidth to disk (per core) decreases

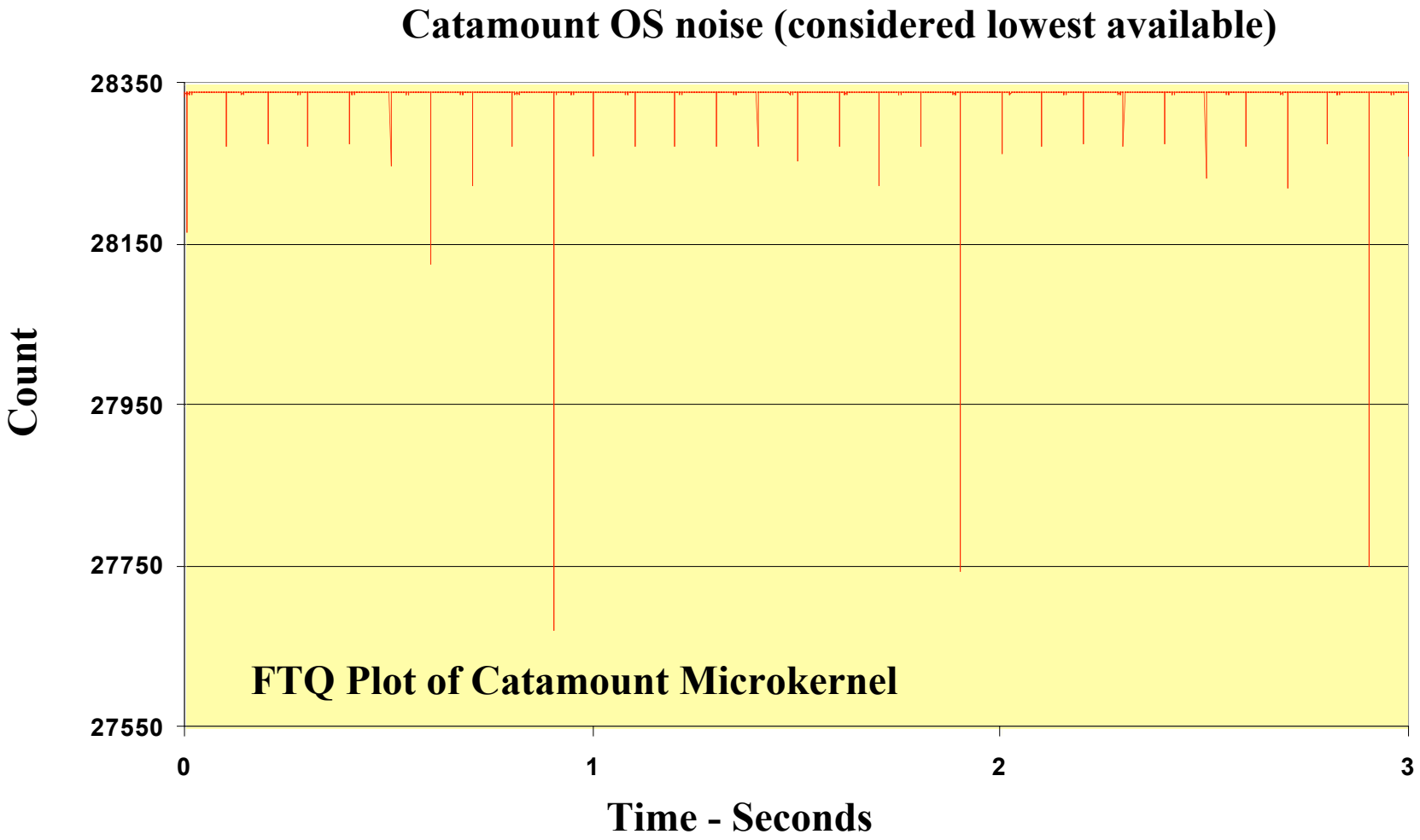
- Improved MPI-IO
- Coordinate IO to reduce contention

MPI Must Support Custom Interconnects

Interconnects in the Top 500

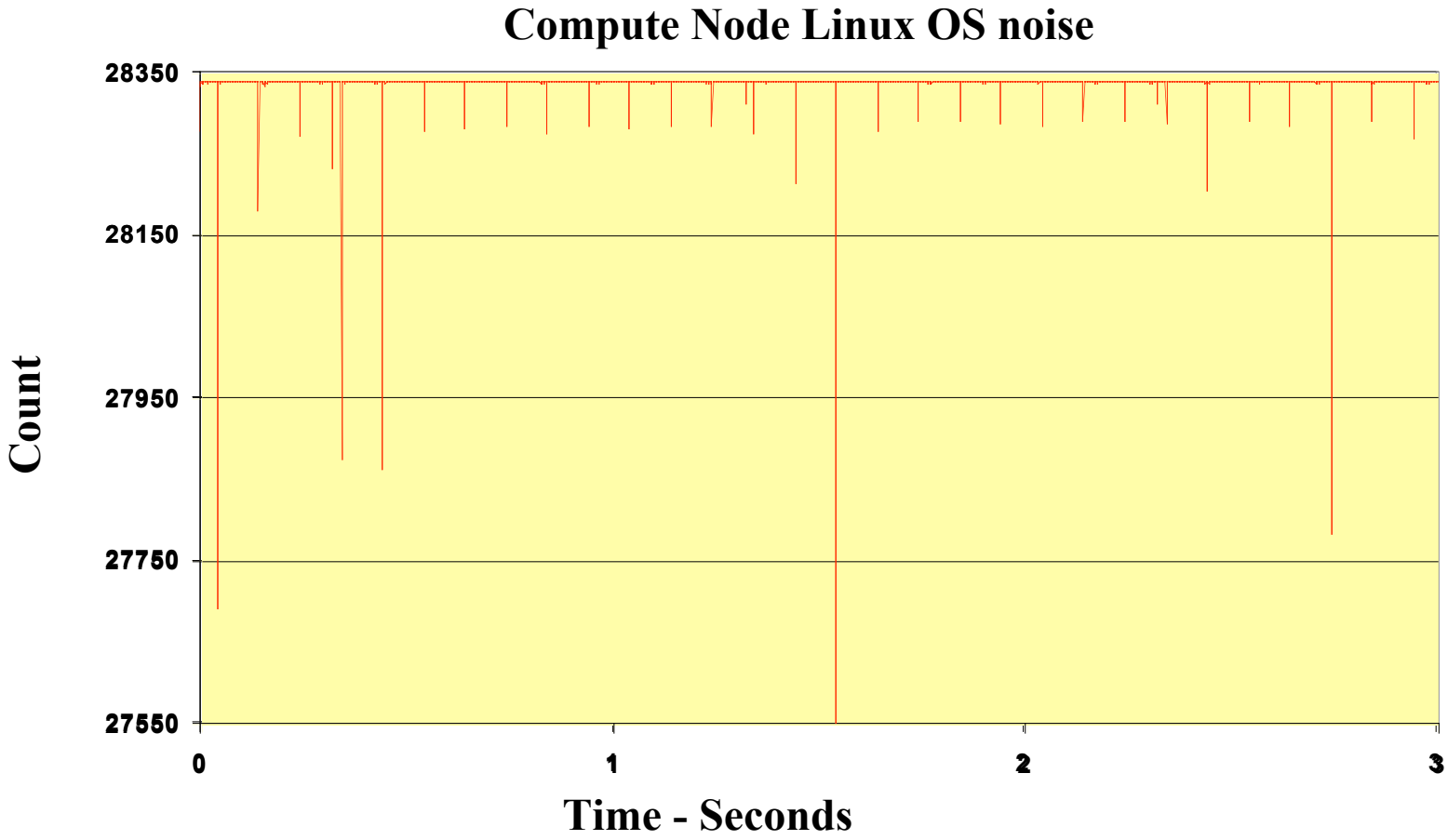


Trend is away from Custom Microkernels



Cray Compute Node Linux

Issue of Linux “jitter” killing scalability solved in 2007 through a series of tests on ORNL 11,000 node XT4.



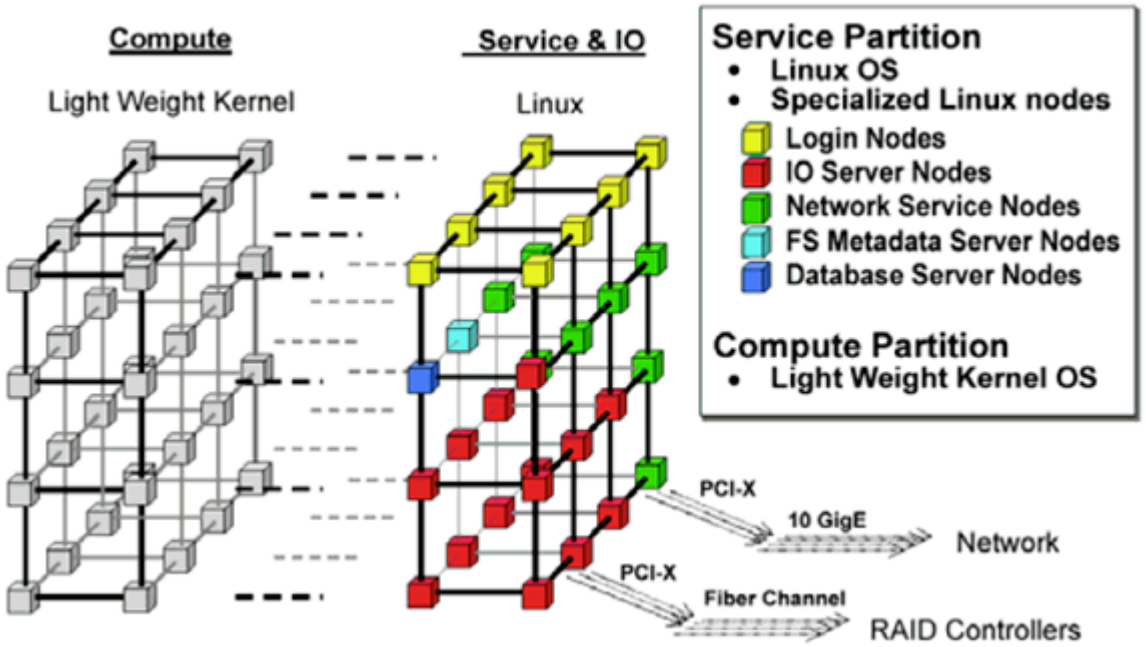
Heterogeneous Systems

ornl

How do we keep MPI viable as the heterogeneity of the systems increases?

Hybrid systems, for example:
Clearspeed accelerators (Japan TSUBAME)
IBM Cell boards (LANL Roadrunner)

Systems with heterogeneous node types:
IBM Blue Gene and Cray XT systems
(6 node types)

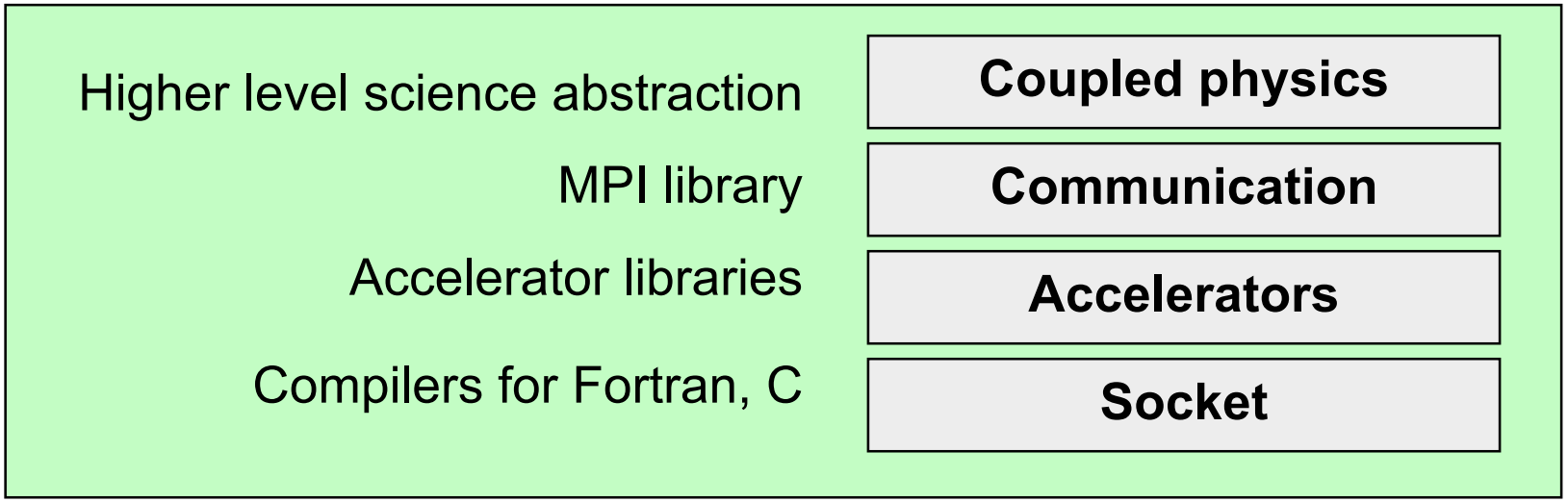


Heterogeneous Systems MPI Impact

ornl

How do we keep MPI viable as the heterogeneity of the systems increases?

One possible solution: Software layering
MPI becomes just one layer and doesn't have to solve everything



Big Computers and Big Applications

ornl

Can a computer ever be too big for MPI?

Not in the metric of number of nodes – has run on 100,000 node BG

but what about a million nodes of sustained petascale systems???

MPI-1 and MPI-2 standards suffer from a lack of fault tolerance

In fact the most common behavior is to abort the entire job if one node fails. (and restart from checkpoint if available)

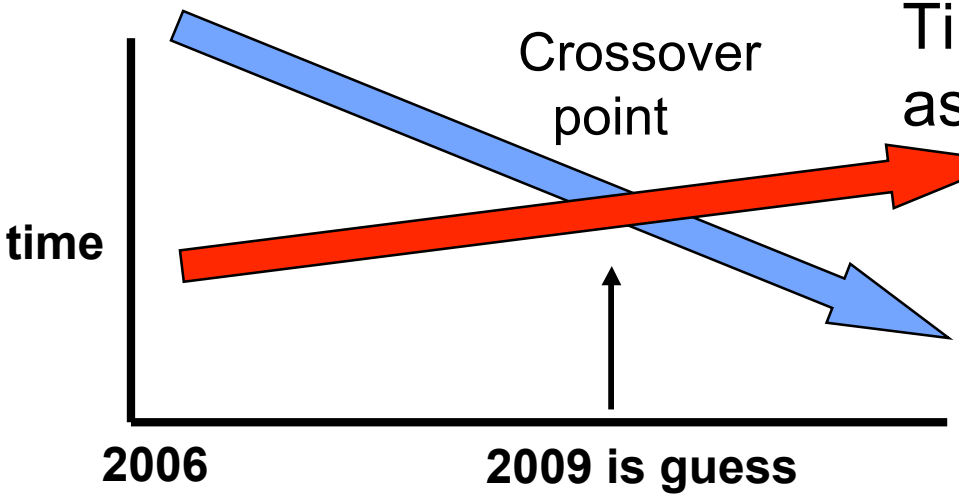
As number of nodes grows it becomes less and less efficient or practical to kill all the remaining nodes because one has failed.

Example: 99,999 nodes running nodes are restarted because 1 node fails. That is a lot of wasted cycles.

Checkpointing can actually increase failure rate by stressing IO system

The End of Fault Tolerance as We Know It
Point where checkpoint ceases to be viable

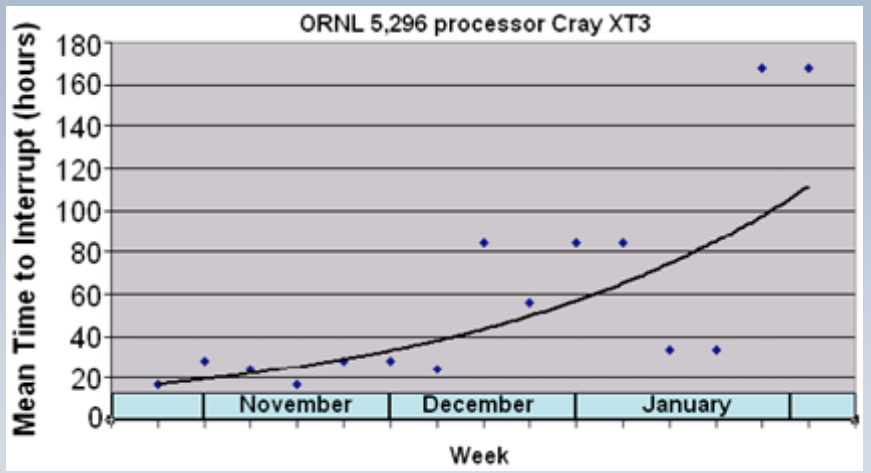
MPI apps will no longer be able to rely on checkpoint on big systems



Time to checkpoint grows larger as problem size increases

MTTI grows smaller as number of parts increases

Good news is the MTTI is better than expected for LLNL BG/L and ORNL XT4 a/b 6-7 days not minutes

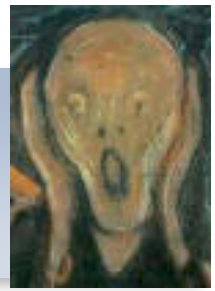


Applications need recovery modes not in standard MPI

Harness project (follow-on to PVM) explored 5 modes of MPI recovery in FT-MPI. The recoveries effect the size (extent) and ordering of the communicators

- ABORT: just do as vendor implementations
- BLANK: leave holes
  - But make sure collectives do the right thing afterwards
- SHRINK: re-order processes to make a contiguous communicator
  - Some ranks change
- REBUILD: re-spawn lost processes and add them to MPI_COMM_WORLD
- REBUILD_ALL: same as REBUILD except rebuilds all communicators, groups and resets all key values etc.

May be time to consider an MPI-3 standard that allows applications to recover from faults

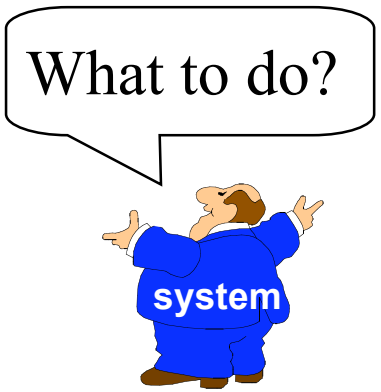


What other features are needed?

Need a mechanism for each application (or component) to specify to system what to do if fault occurs

System Options include:

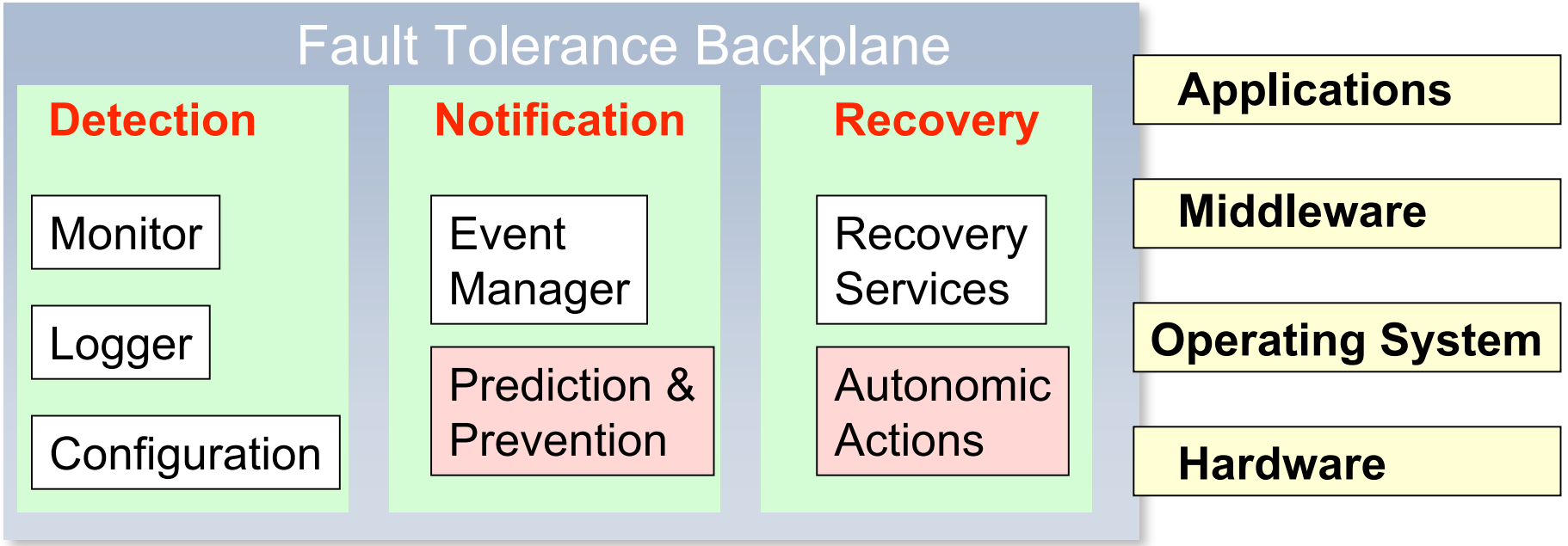
- Restart – from checkpoint or from beginning
- Ignore the fault altogether – not going to affect app
- Migrate task to other hardware before failure
- Reassignment of work to spare processor(s)
- Replication of tasks across machine
- Notify application and let it handle the problem



Holistic Solution

We need coordinated fault awareness, prediction and recovery across the entire HPC system from the application to the hardware.

"Prediction and prevention are critical because the best fault is the one that never happens"



CIFTS project underway at ANL, ORNL, LBL, UTK, IU, OSU

Productivity - Validation

Validation of answer on such large systems when the problem size and more realistic physics has never been run before. There is a lack of tools and rigor today.

Fault may not be detected

Algorithms may introduce rounding errors

Eg. Linpack on ORNL 119 TF

Cosmic rays may introduce perturbations

Eg. VaTech Big Mac

Result looks reasonable but is actually wrong

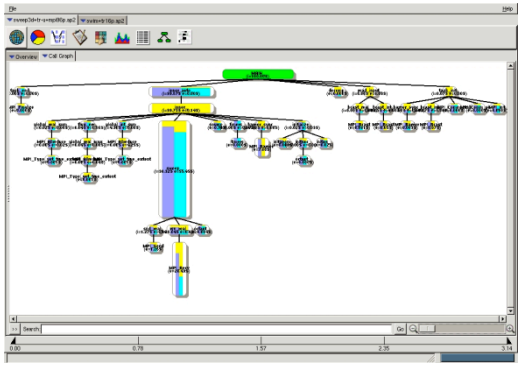
Can't afford to run every job three (or more) times
Yearly Allocations are like $5M-$10M grants

Performance Tools for Petascale

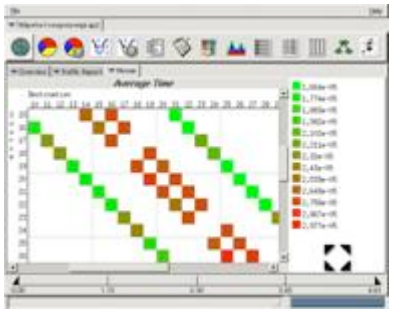
Example Cray's Apprentice^2 tool for large scale performance analysis. Routinely used on 11,000 node XT4 at ORNL

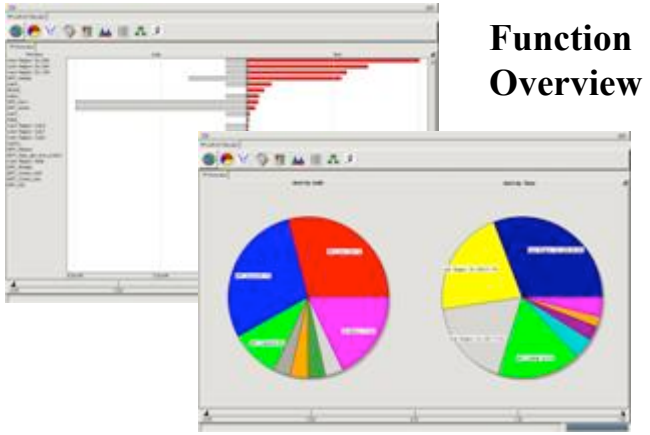
But what happens at 100,000? At million?

Call Graph Profile



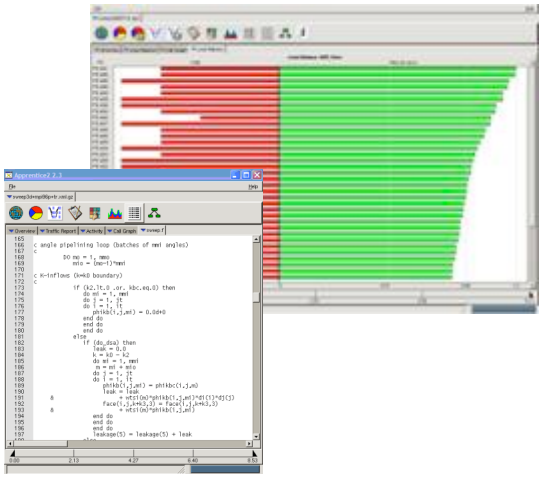
Pair-wise Communication View

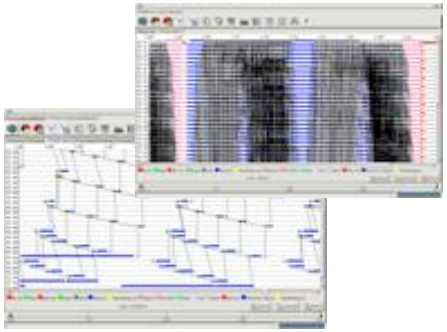




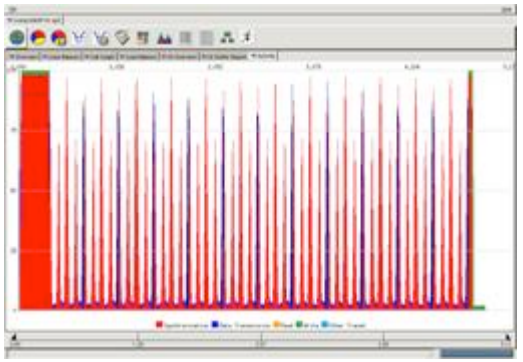
Function Overview

Load balance views





Time Line & I/O Views



Communication & I/O Activity View

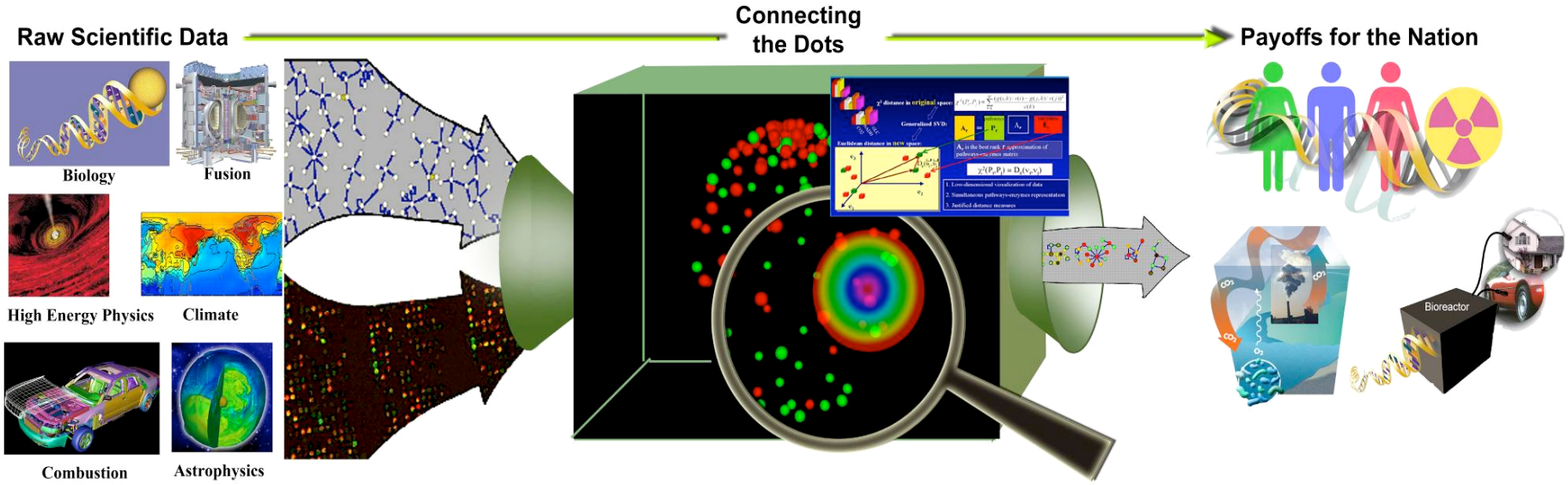
Petascale Debugger is viewed as major missing component of productivity suite

Both Petascale and Exascale workshops held in 2007 pointed this out.

- **Comparative Debugging is just one solution being explored**
  - Simultaneous run of two MPI applications
  - Ability to compare data from different applications
  - Ability to assert the match of data at given points in execution
- **Scenarios**
  - Porting between architectures
  - Serial converted to parallel
  - One optimization level versus another
  - Small scaling versus large scaling
  - One programming language converted to another
  - COTS only (a la cluster) versus MPP
  - threaded versus vector

Productivity – what to do with the data

The increase in data output at sustained petascale drives the need for scalable knowledge discovery tools



Sheer Volume of Data

Climate

5 years: 5-10 Petabytes/year

Fusion

5 years: 1000 Megabytes/2 min

Advanced Mathematics and Algorithms

- **Huge dimensional space**
- **Combinatorial challenge**
- **Complicated by noisy data**

Providing Predictive Understanding

- **Biology**
- **Nanotechnology**
- **Alternate Energy**

90% of stored data
is never read and costs $10,000/PB to archive on tape

Final Thoughts

ornl

- Sustained petascale systems will have disruptive architectures, but applications have inertia against change
- MPI programming model dominates the HPC applications
  - But MPI will need to evolve to be effective on sustained petascale systems.
  - Multi-core chips, heterogeneous architectures, and fault tolerance will drive the evolution of MPI
- There is a critical need for tools to increase productivity on the largest scale systems, especially in validation and knowledge discovery.

Questions?