

# High Availability for Ultra-Scale Scientific High-End Computing

**Christian Engelmann**

Network and Cluster Computing Group

Computer Science and Mathematics Division

Oak Ridge National Laboratory, Oak Ridge, USA

---

# Overview

- Research at Oak Ridge National Laboratory.
- Ultra-scale scientific high-end computing.
- Fault-tolerant heterogeneous metacomputing.
- High availability system software framework.
- Super-scalable algorithms for computing on 100,000 processors.

# Research at Oak Ridge National Laboratory

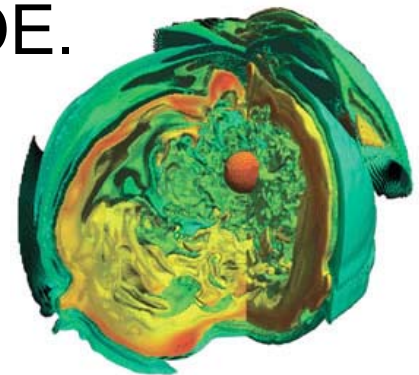
## **Christian Engelmann**

Network and Cluster Computing Group  
Computer Science and Mathematics Division  
Oak Ridge National Laboratory, Oak Ridge, USA

# OAK RIDGE NATIONAL LABORATORY

MANAGED BY UT-BATTELLE FOR THE DEPARTMENT OF ENERGY

- Multiprogram science and technology laboratory.
- Privately managed for the U.S. Department of Energy.
- Basic and applied research and development.
- In biological, chemical, computational, engineering, environmental, physical, and social sciences.
- Staff: 3800 total, 1500 scientists and engineers
- Budget: \$1.06 billion, 75% from US DOE.
- Total land area: 58mi<sup>2</sup> (150km<sup>2</sup>).
- ~3000 guest researchers each year.
- ~30,000 visitors each year.



# East Campus of Oak Ridge National Laboratory

Computational Sciences Building



Research Office Building



Engineering Technology Facility

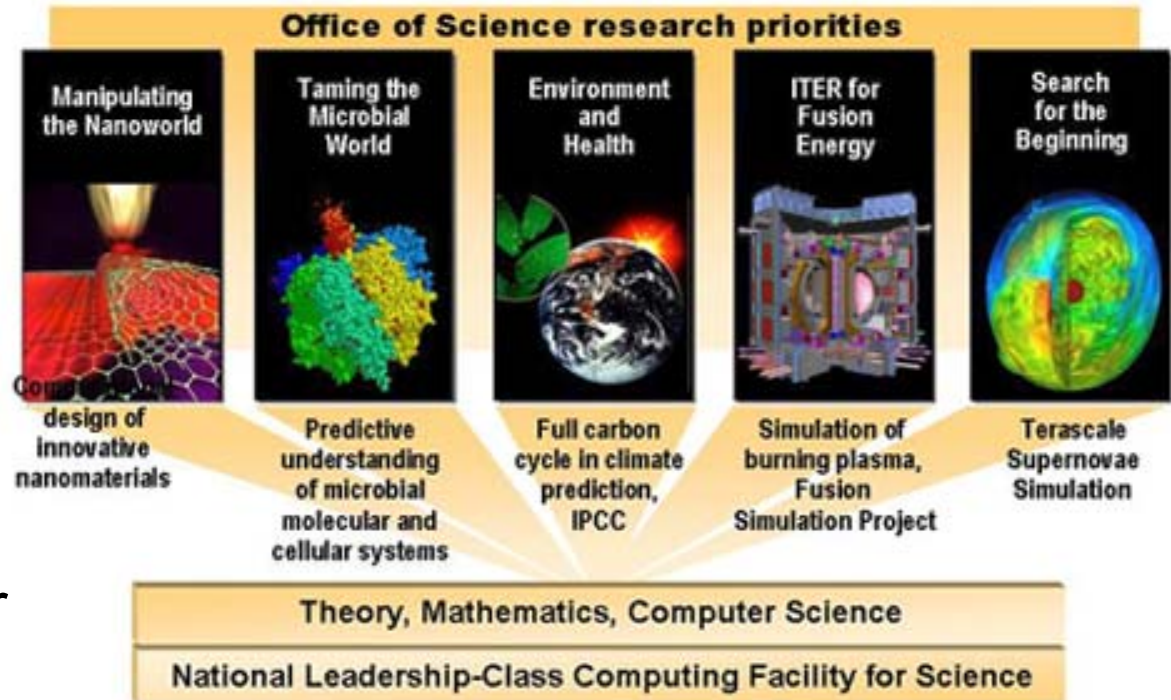
Joint Institute for Computational Sciences

Research Support Center (Cafeteria, Conference, Visitor)

# National Leadership Computing Facility

- Established in 2004.
- \$25M from US DOE.
- Lead by Oak Ridge National Laboratory.
- Collaboration with other laboratories and universities.
- Using capability over capacity computing.
- Advancing the race for scientific discovery.

## Leadership Computing for Science Critical for success in key national priorities



# Center for Computational Sciences

- Computer center with 40,000 ft<sup>2</sup> (3700m<sup>2</sup>) floor space.
- 3 systems in the Top 500 List of Supercomputer Sites:
  - 29. Cray X1 Vector with 504P/2TB ⇒ 5.9 TFLOPS.
  - 80. IBM Power 4 Cluster with 864P/1TB ⇒ 2.3 TFLOPS.
  - 245. SGI Altix IA64 SSI with 256P/2TB ⇒ 1.2 TFLOPS.



# Leadership Computing Roadmap

- New system installations and upgrades this year:
  - Cray XT3 Opteron Cluster with 5212P/10TB  $\Rightarrow$  25 TFLOPS.
  - Cray X1 to X1e with 1024P/ 4TB  $\Rightarrow$  18 TFLOPS.
- Upgrades next year:
  - Cray XT3 to 20000P/40TB  $\Rightarrow$  100 TFLOPS.
- Future roadmap:
  - Upgrade Cray X1e to X2 ~ 2007.
  - Upgrade Cray XT3 to 250 TFLOPS ~ 2007.
  - Installation of a 1 PFLOP system ~ 2009.



Cray XT3

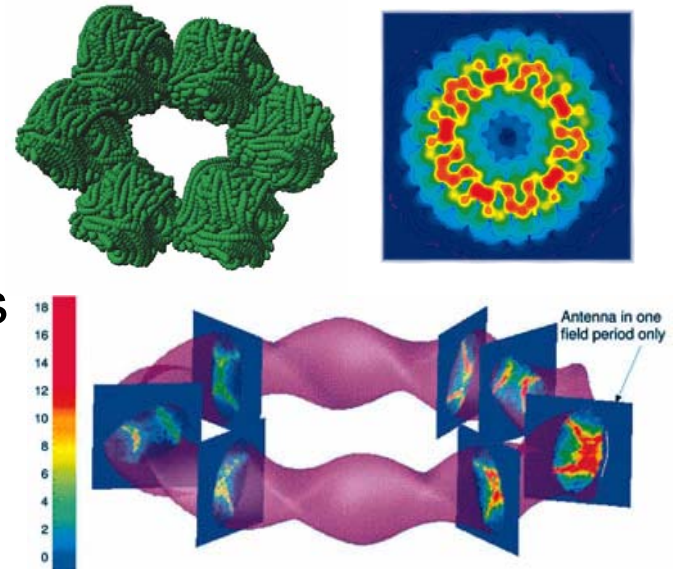


Cray X1



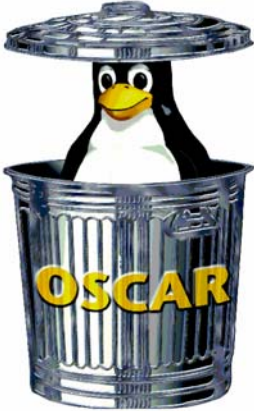
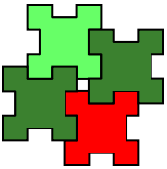



# Computer Science Research Groups

- Computer Science and Mathematics Division.
  - Applied research focused on computational sciences, intelligent systems, and information technologies.
- CSM Research Groups:
  - Climate Dynamics
  - *Computational Biology*
  - Computational Chemical Sciences
  - Computational Materials Science
  - Computational Mathematics
  - ...
  - *Network and Cluster Computing (~23 researchers)*



# Computer Science Research Projects

- Parallel Virtual Machine (PVM). 
- MPI Specification, FT-MPI and Open MPI. 
- Common Component Architecture (CCA).
- Open Source Cluster Application Resources (  ).
- Scalable Systems Software (SSS).
- ...
- Fault-tolerant metacomputing (HARNESS). 
- High availability for high-end computing (RAS-MOLAR).
- Super-scalable algorithms research. 

# Ultra-scale Scientific High-End Computing

**Christian Engelmann**

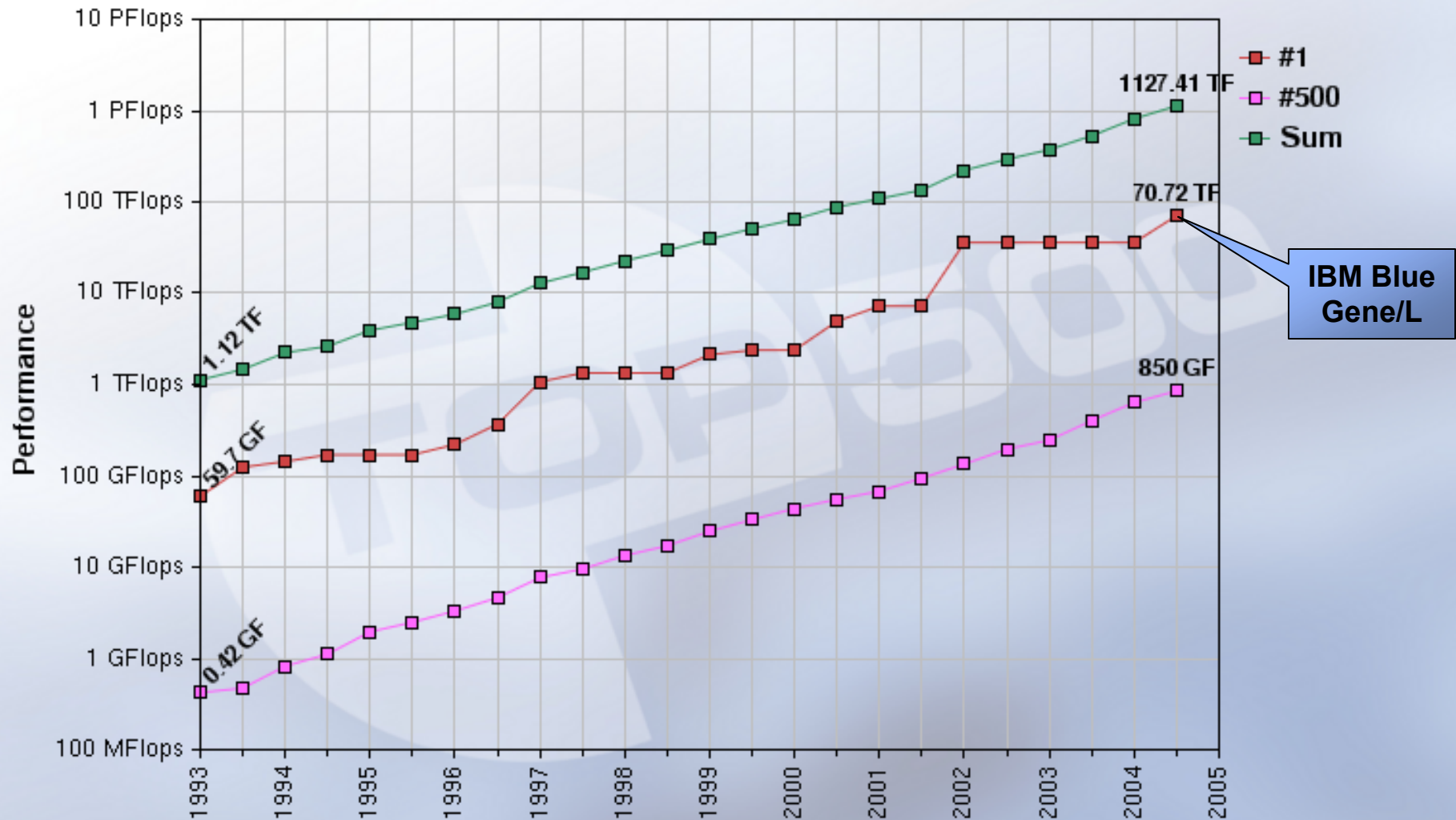
Network and Cluster Computing Group

Computer Science and Mathematics Division

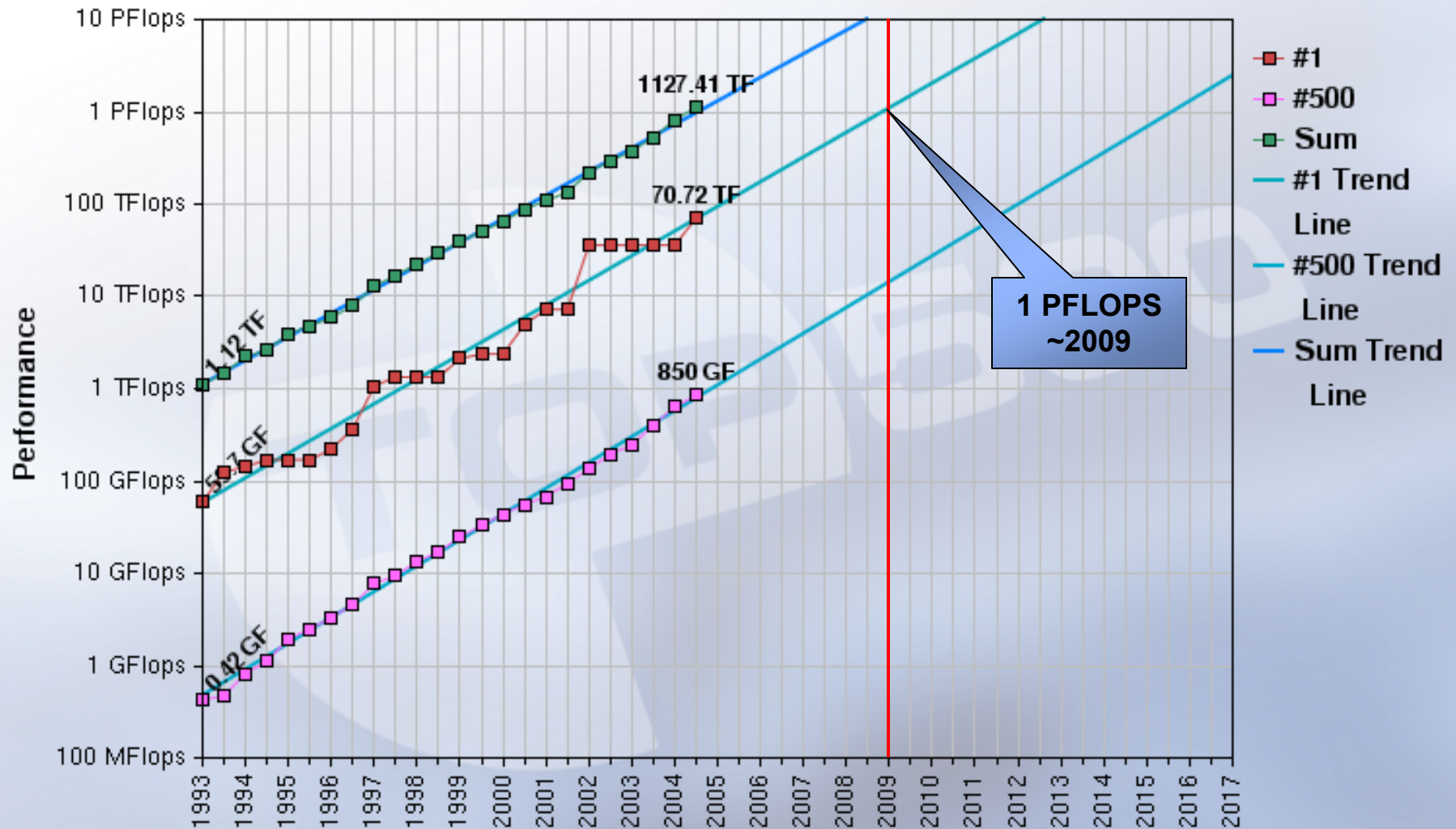
Oak Ridge National Laboratory, Oak Ridge, USA

# Scientific High-End Computing

- Next generation supercomputing.
  - Large-scale cluster, parallel, distributed and vector systems.
  - 131,072 processors for computation in IBM Blue Gene/L.
- Computationally and data intensive applications.
  - Many research areas: (multi-)physics, chemistry, biology...
  - Climate, supernovae (stellar explosions), nuclear fusion, material science and nanotechnology simulations.
- Ultra-scale = upper end of processor count (+5,000).
  - 25+ TeraFLOPS (25,000,000,000,000 FLOPS and more).



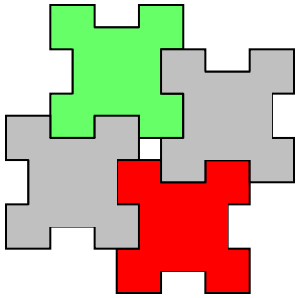
# Projected Performance Development



# Ultra-scale Software Research Issues

- Capability computing applications require ultra-scale systems and long runtimes (weeks or even months).
  - However, larger and more complex systems result in an increase of failure rates and system downtimes.
  - Furthermore, application efficiency drops off with increased system scale due to Amdahl's Law.
- 
- ➔ Application software fault-tolerance.
  - ➔ High availability system software.
  - ➔ Super-scalable algorithms for 100,000 processors.

# Fault-tolerant Heterogeneous Metacomputing with Harness



**Christian Engelmann**

Network and Cluster Computing Group  
Computer Science and Mathematics Division  
Oak Ridge National Laboratory, Oak Ridge, USA





# What is Harness

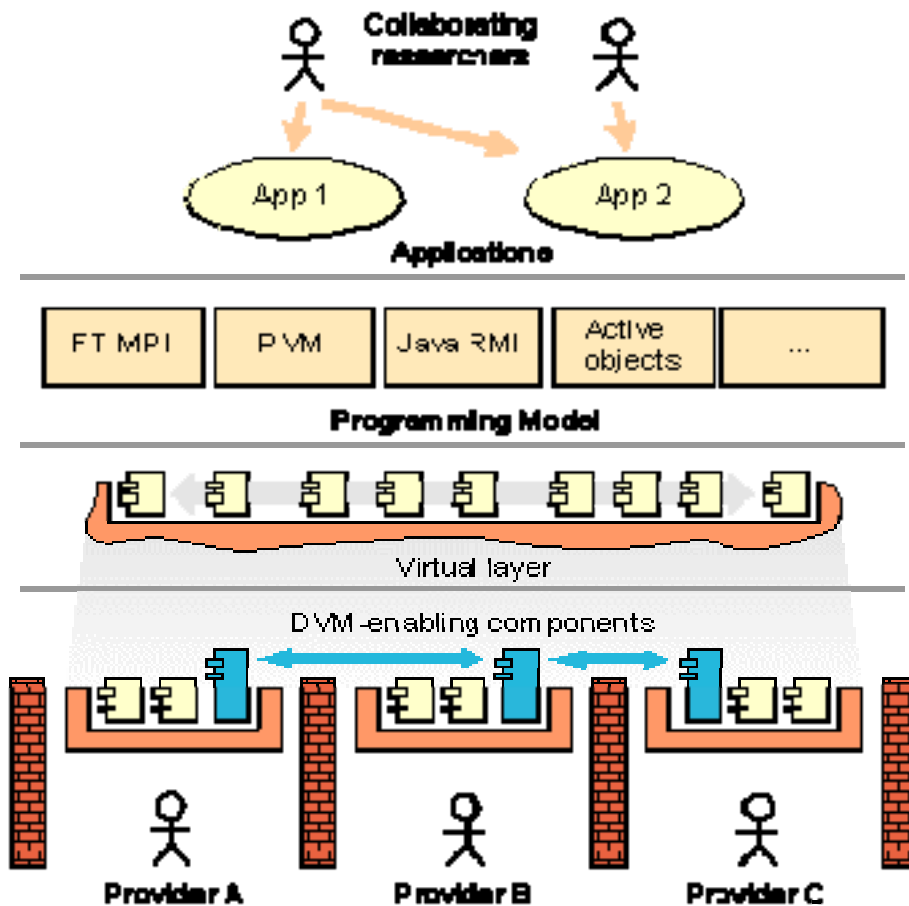
- A pluggable, reconfigurable, adaptive framework for heterogeneous distributed computing.
- Allows aggregation of resources into high-capacity distributed virtual machines.
- Provides runtime customization of computing environment to suit applications needs.
- Enables dynamic assembly of scientific applications from (third party) plug-ins.
- Offers highly available distributed virtual machines through distributed control.

# Harness Research Areas



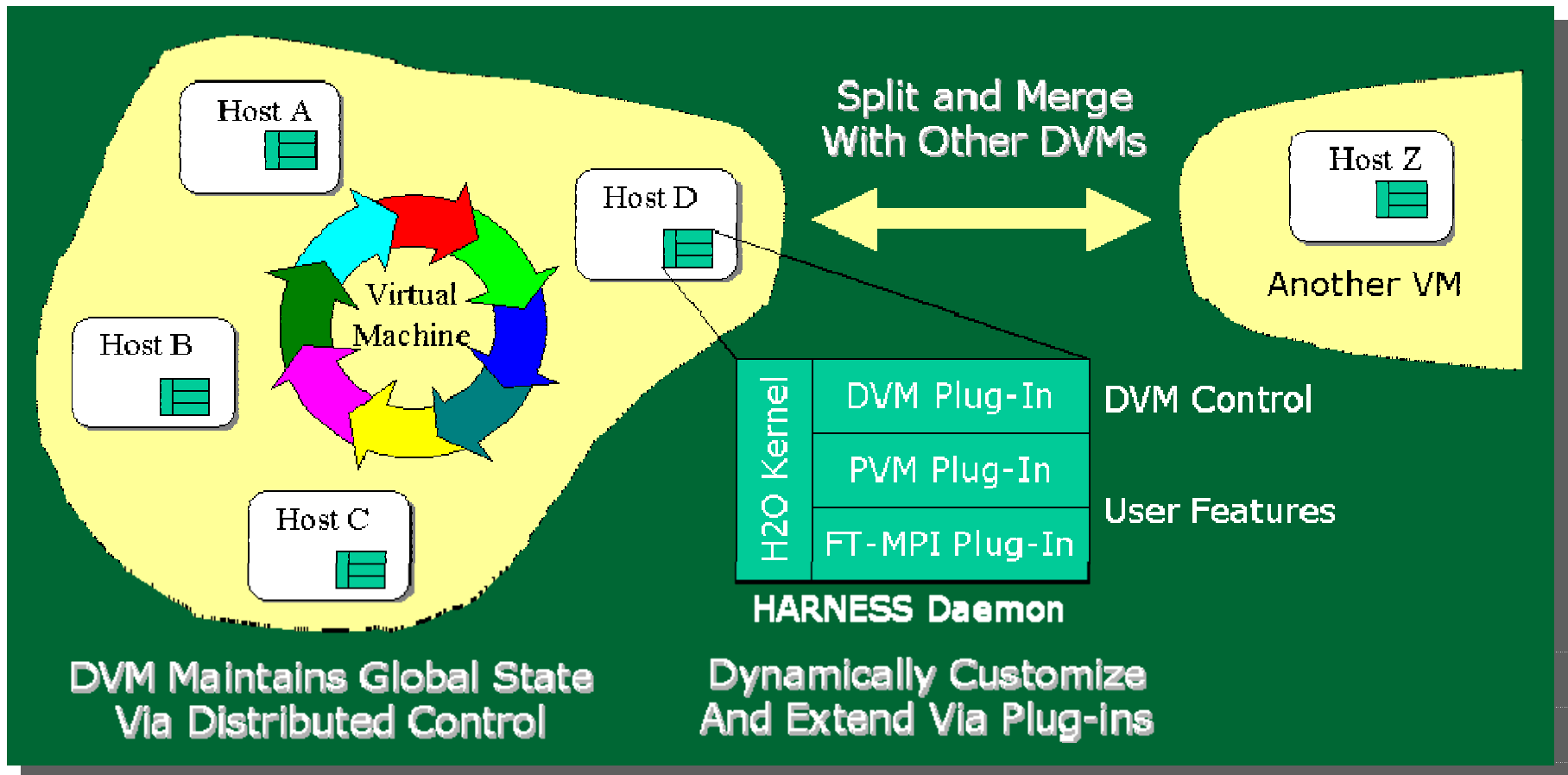
- Lightweight, pluggable software frameworks.
- Adaptive, reconfigurable runtime environments.
- Parallel plug-ins and diverse programming paradigms.
- Highly available distributed virtual machines (DVMs).
- Advanced ultra-scale approaches for fault tolerance.
- Fault-tolerant message passing (FT-MPI).
- Mechanisms for configurable security levels.
- Dynamic, heterogeneous, reconfigurable communication frameworks (RMIX).

# Harness Architecture



- Light-weight kernels share their resources.
- Plug-ins offer services.
- Support for diverse programming models.
- Distributed Virtual Machine (DVM) layer.
- Highly available DVM.
- Highly available plug-in services via DVM.

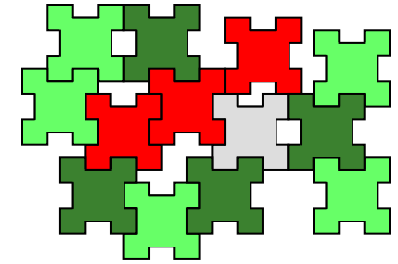
# Harness DVM Architecture



# Harness Plug-ins



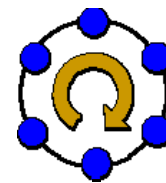
**FT-MPI**



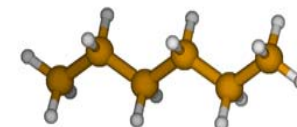
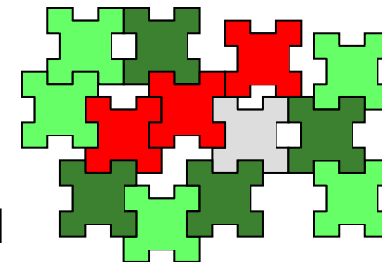
- PVM emulation plug-in:
  - Replaces the PVM daemon.
  - Allows users a seamless transition to Harness.
  - Plug-ins and applications just link libpvm.
  - PVM is controlled with the Harness console.
- Fault-tolerant MPI (FT-MPI) plug-in:
  - Combines several FT-MPI services in one plug-in.
  - Plug-ins and applications just use ftmpiCC.
  - FT-MPI is controlled with the Harness console.

# Harness Plug-ins

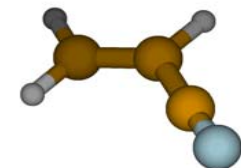
- DVM plug-in:
  - Allows to aggregate multiple Harness kernels.
- Distributed control plug-in:
  - Provides high availability through virtual synchrony.
- RMIX plug-in:
  - Offers multi-protocol RMI (JRMPX, SOAP and RPC).
- Several application plug-ins:
  - Molecular dynamics.
  - Quantum chemistry.



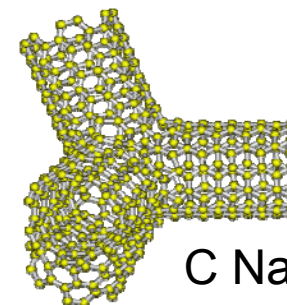
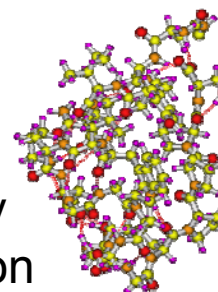
Distributed Control



SCC-DFTB



Geometry Optimization



C Nanotubes

# The Future of the Harness Project

## The Harness Workbench:

- *Simplify HPC application development & deployment.*
- Virtualized Command Toolkit (VCT) providing a common view across diverse HPC environments.
- Next generation runtime environment (Harness-RTE) for automatic adaptation to target architectures.
- Plug-ins for adaptive compilation, linking and staging.
- Configurable application programming environments.
- Back-end for parallel software toolkits and integrated development environments, such as Eclipse.

# High Availability System Software Framework

**Christian Engelmann**

---

Network and Cluster Computing Group  
Computer Science and Mathematics Division  
Oak Ridge National Laboratory, Oak Ridge, USA



# Availability of HEC Systems

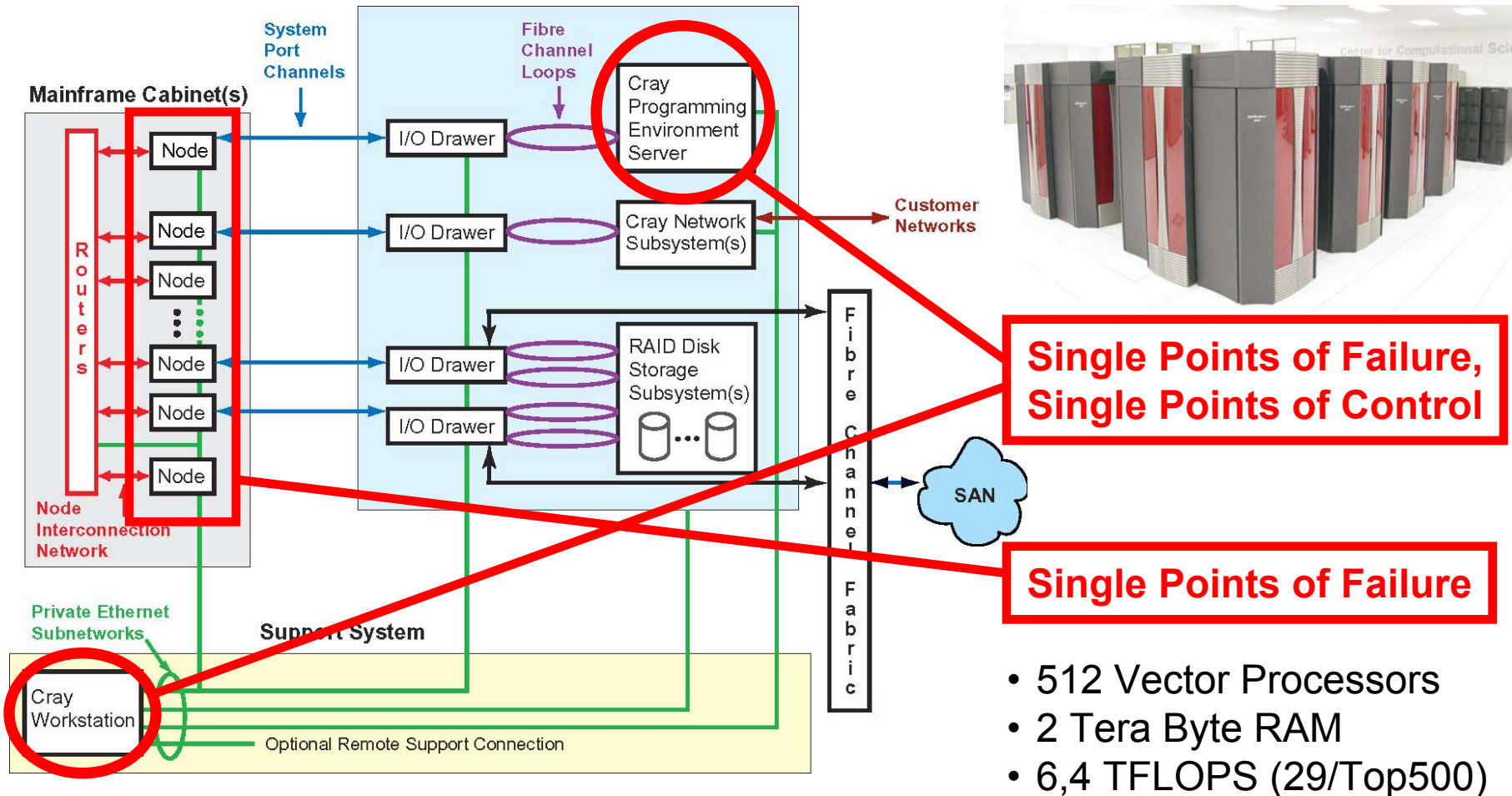
- Today's supercomputers typically need to reboot to recover from a single failure.
- Entire systems go down (regularly and unscheduled) for any maintenance or repair.
- Compute nodes sit idle while their head node or one of their service nodes is down.
- Availability will get worse in the future as the MTBI decreases with growing system size.
- *Why do we accept such significant system outages due to failures, maintenance or repair?*

# Availability Measured by the Nines

9's	Availability	Downtime/Year	Examples
1	90.0%	36 days, 12 hours	Personal Computers
2	99.0%	87 hours, 36 min	Entry Level Business
3	99.9%	8 hours, 45.6 min	ISPs, Mainstream Business
4	99.99%	52 min, 33.6 sec	Data Centers
5	99.999%	5 min, 15.4 sec	Banking, Medical
6	99.9999%	31.5 seconds	Military Defense

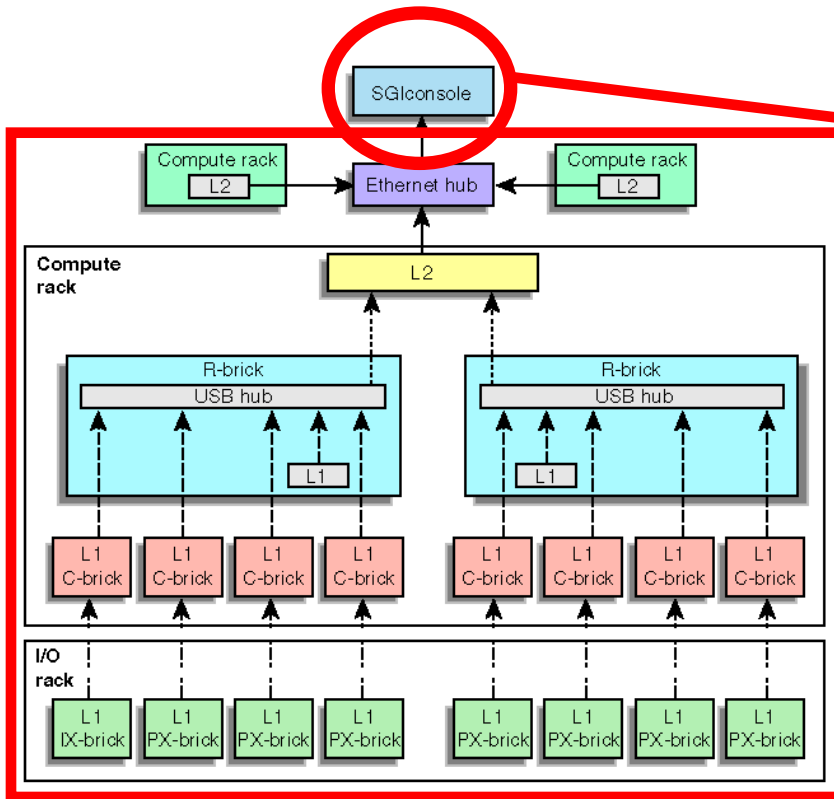
- Enterprise-class hardware + Stable Linux kernel = 5+
- Substandard hardware + Good high availability package = 2-3
- Today's supercomputers = 1-2
- My desktop = 1-2

# Vector Machines: Cray X1 (Phoenix)



# SSI Clusters: SGI Altix (Ram)

- 256 Itanium 2 Processors
- 2 Tera Byte RAM
- 1,5 TFLOPS (245/Top500)



**Single Point of Failure,  
Single Point of Control**

**Single Points of Failure**

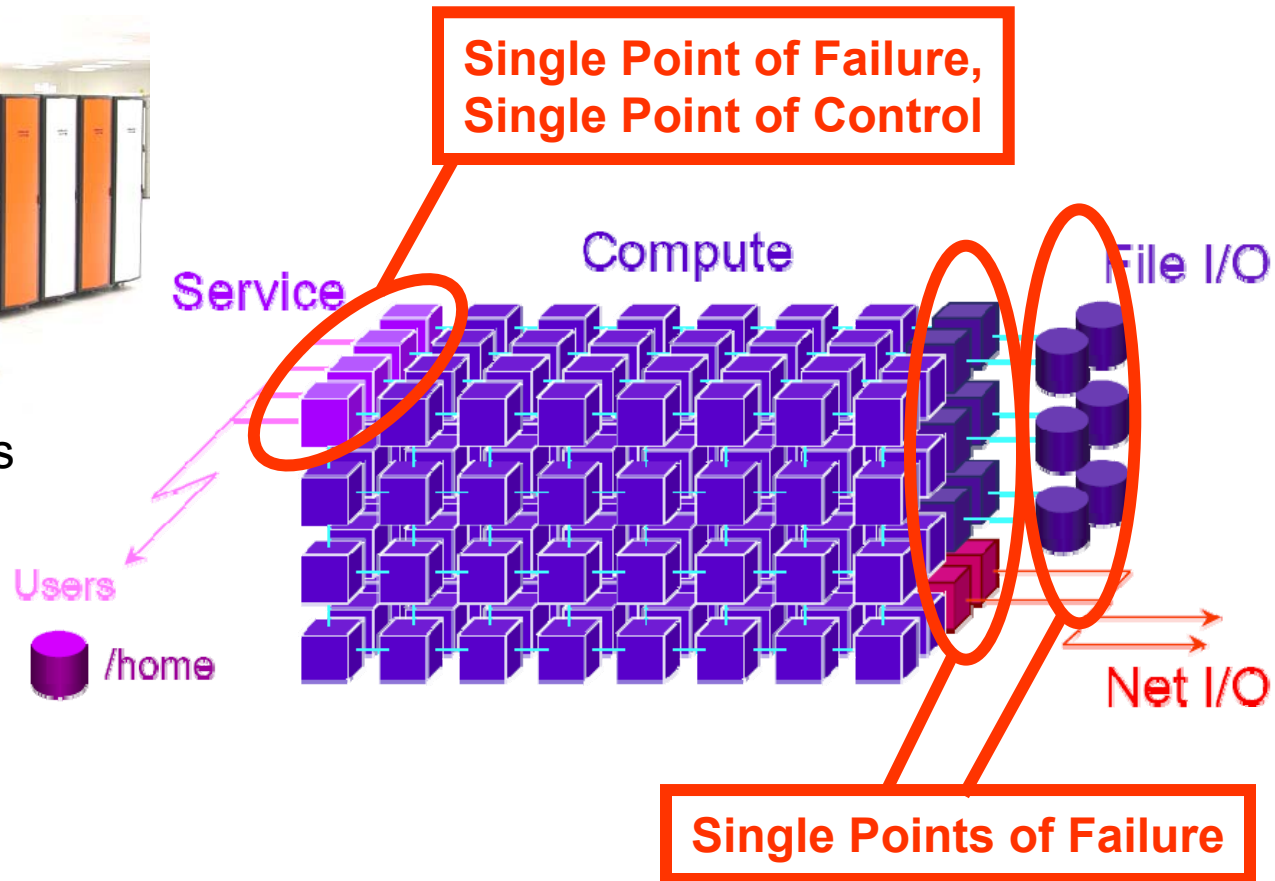


— Ethernet  
- - - - - USB signals in NUMalink3 cable (L1 of C-brick to USB hub in R-brick)  
· · · · · USB cable  
- · - · - RS-422 signals in Crosstown2 cable

# Clusters/MPPs: Cray XT3 (Jaguar)



- AMD Opteron Processors
- Installation in progress
- 25 TFLOPS in 2005
- 100 TFLOPS in 2006
- 250 TFLOPS in 2007



# High Availability Methods

## Active/Hot-Standby:

- Single active head node.
- Backup to shared storage.
- Simple checkpoint/restart.
- Rollback to backup.
- Idle standby head node(s).
- Service interruption for the time of the fail-over.
- Service interruption for the time of restore-over.
- Possible loss of state.

## Active/Active:

- Many active head nodes.
- Work load distribution.
- Symmetric replication between participating nodes.
- Continuous service.
- Always up-to-date.
- No restore-over necessary.
- Virtual synchrony model.
- Complex algorithms.

# High Availability Technology

## Active/Hot-Standby:

- HA-OSCAR with active/hot-standby head node.
- Similar projects: HA Linux
- Cluster system software.
- No support for multiple active/active head nodes.
- No application support.

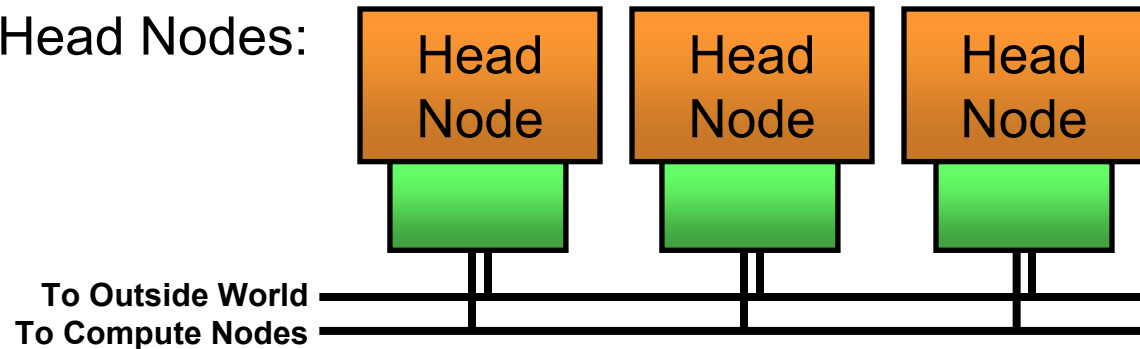
## Active/Active:

- HARNESSESS with symmetric distributed virtual machine.
- Similar projects: Cactus ...
- Heterogeneous adaptable distributed middleware.
- No system level support.
- Solutions not flexible enough.

- *System-level data replication and distributed control service needed for active/active head node solution.*
- *Reconfigurable framework similar to HARNESSESS needed to adapt to system properties and application needs.*

# Modular HA Framework on Active/ Active Head Nodes

Highly Available Head Nodes:



Reliable Services:

Scheduler	MPI Runtime	...
-----------	-------------	-----

Virtual Synchrony:

Distributed Control Service
-----------------------------

Symmetric Replication:

Data Replication Service
--------------------------

Reliable Server Groups:

Group Communication Service
-----------------------------

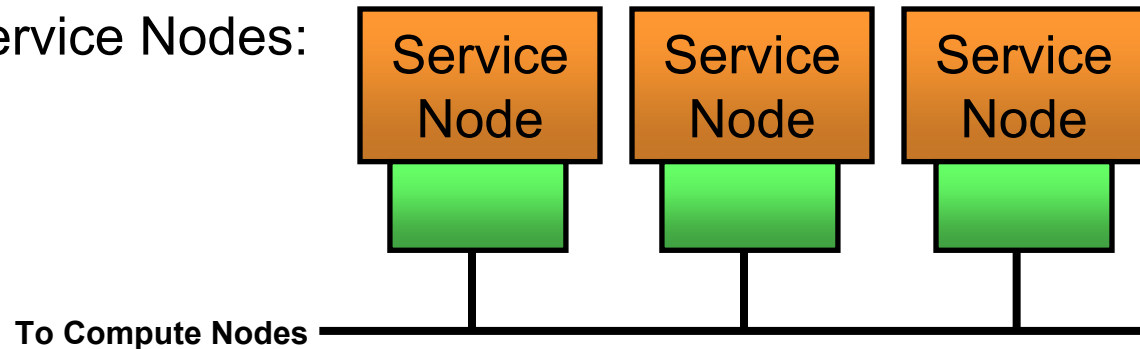
Communication Methods:

TCP/IP	Shared Memory	Etc.
--------	---------------	------



# Modular HA Framework on Active/ Active Service Nodes

Highly Available Service Nodes:



Reliable Services:

File System	MPI Runtime	...
-------------	-------------	-----

Virtual Synchrony:

Distributed Control Service
-----------------------------

Symmetric Replication:

Data Replication Service
--------------------------

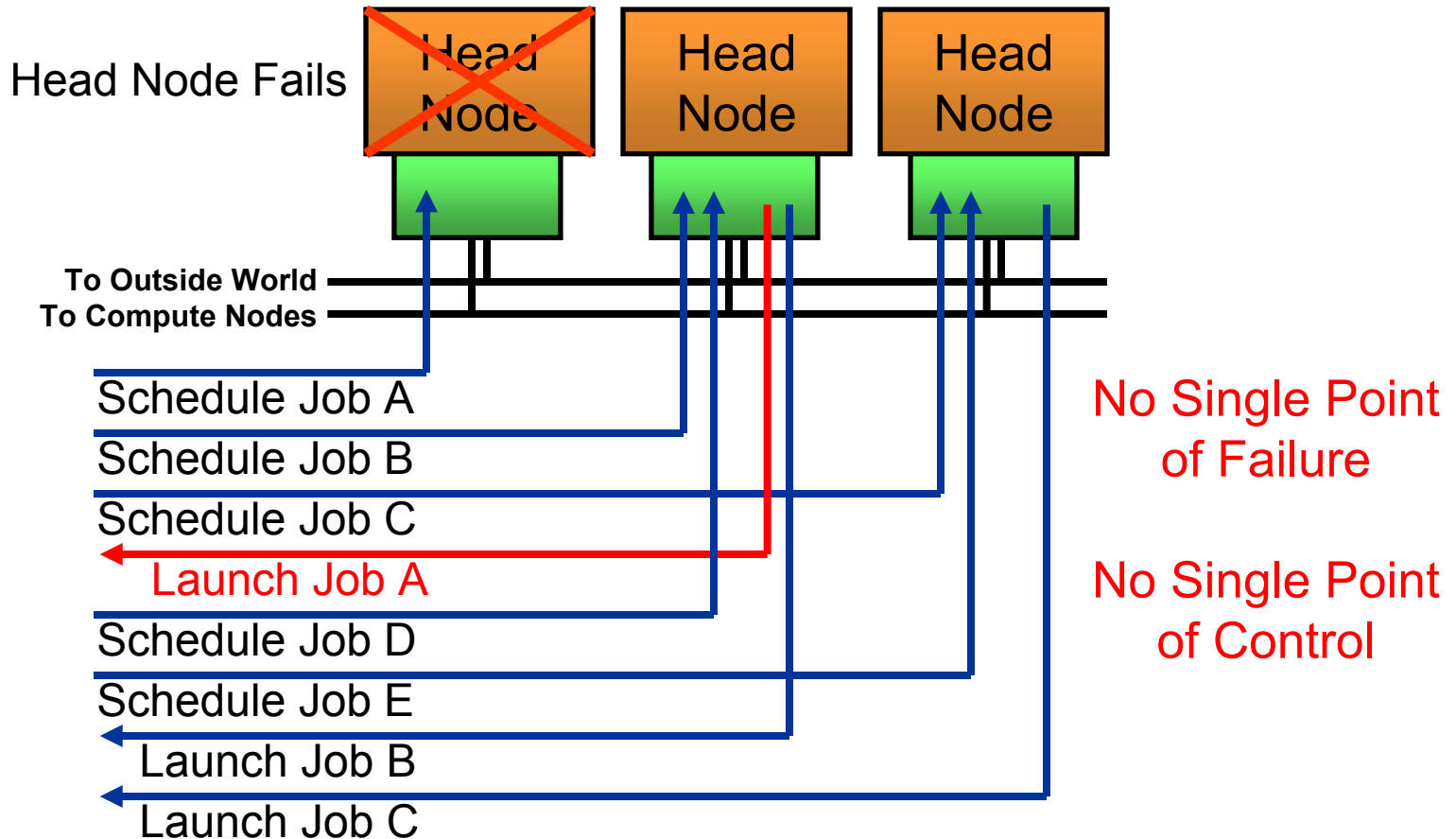
Reliable Server Groups:

Group Communication Service
-----------------------------

Communication Methods:

TCP/IP	Shared Memory	Etc.
--------	---------------	------

# Modular HA Framework on Active/ Active Head Nodes: Scheduler Example



# Many HA Framework Use Cases

- Active/Active and Active/Hot-standby process state replication for multiple head or service nodes.
  - Reliable system services, such as scheduler, MPI-runtime and system configuration/management/monitoring.
- Memory page replication for SSI and DSM.
- Meta data replication for parallel/distributed FS.
- Super-scalable peer-to-peer diskless checkpointing.
- Super-scalable localized FT-MPI recovery.
- !!! No protection from Byzantine failures !!!

# MOLAR: Modular Linux and Adaptive Runtime Support for High-end Computing Operating and Runtime Systems

- The HA Framework is part of the MOLAR project.
- MOLAR addresses the challenges for operating and runtime systems to run large applications efficiently on future ultra-scale high-end computers.
- MOLAR is a collaborative effort:



OAK RIDGE NATIONAL LABORATORY

MANAGED BY UT-BATTELLE FOR THE DEPARTMENT OF ENERGY

NC STATE UNIVERSITY



The University of Reading



LOUISIANA TECH  
UNIVERSITY

CRAY

# MOLAR: HEC OS/R Research Map

**MOLAR: Modular Linux and Adaptive Runtime Support**

**HEC Linux OS: Modular, Custom, Light-weight**

Kernel Design

**Performance  
Observation**

Communications, IO

**Monitoring**

Extend/Adapt  
Runtime/OS

Root Cause  
Analysis

**RAS**

High  
Availability

**Testbeds**

Provided

# Super-Scalable Algorithms for Computing on 100,000 Processors

**Christian Engelmann**

Network and Cluster Computing Group

Computer Science and Mathematics Division

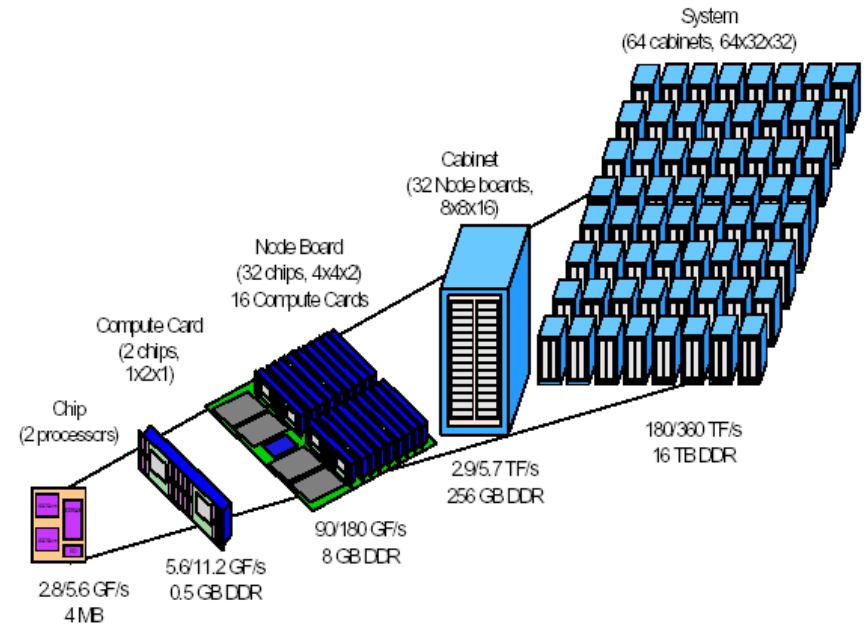
Oak Ridge National Laboratory, Oak Ridge, USA

# Super-Scale Architectures

- Current tera-scale supercomputers have up to 10,000 processors.
- Next generation peta-scale systems will have 100,000 processors and more.
- Such machines may easily scale up to 1,000,000 processors in the next decade.
- IBM is currently deploying the Blue Gene/L system at research institutions world-wide.

# IBM Blue Gene/L

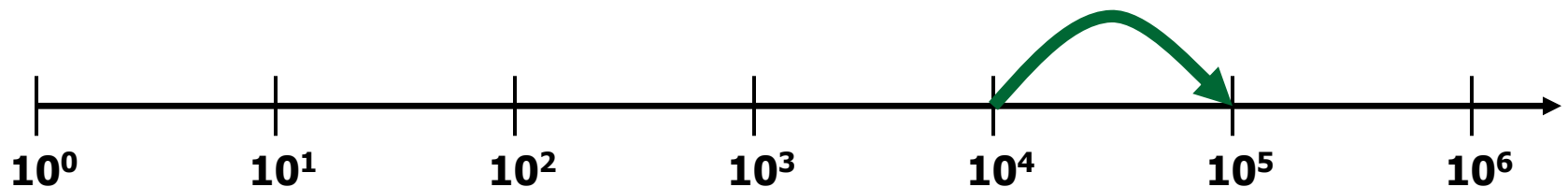
- 64K diskless nodes with 2 processors per node.
- 512MB RAM per node.
- Additional service nodes.
- 360 Tera FLOPS.
- Over 150k processors.
- Various networks.
- Operational in 2005.
- Partition (512 nodes) outages on single failure.
- MTBF = hours, minutes?





# Scalability Issues

- How to make use of 100,000 processors?
- System scale jumps by a magnitude.
- Current algorithms do not scale well on existing 10,000-processor systems.
- Next generation super-scale systems are useless if efficiency drops by a magnitude.



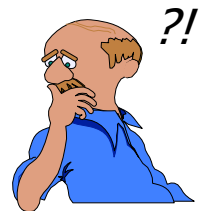
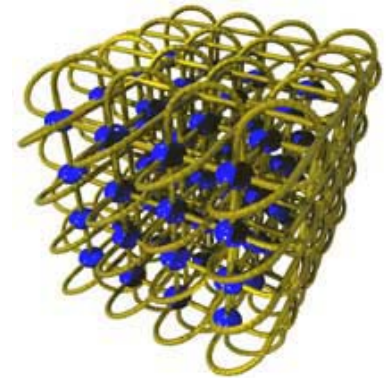
---

# Fault-tolerance Issues

- How to survive on 100,000 processors?
- Failure rate grows with the system size.
- Mean time between failures (MTBF) may be a few hours or just a few minutes.
- Current solutions for fault-tolerance rely on checkpoint/restart mechanisms.
- Checkpointing 100,000 processors to central stable storage is not feasible anymore.

# ORNL/IBM Collaboration

- Development of biology and material science applications for super-scale systems.
- Exploration of super-scalable algorithms.
  - Natural fault-tolerance.
  - Scale invariance.
- Focus on test and demonstration tool.
- Get scientists to think about scalability and fault-tolerance in super-scale systems!



# Cellular Algorithms Theory

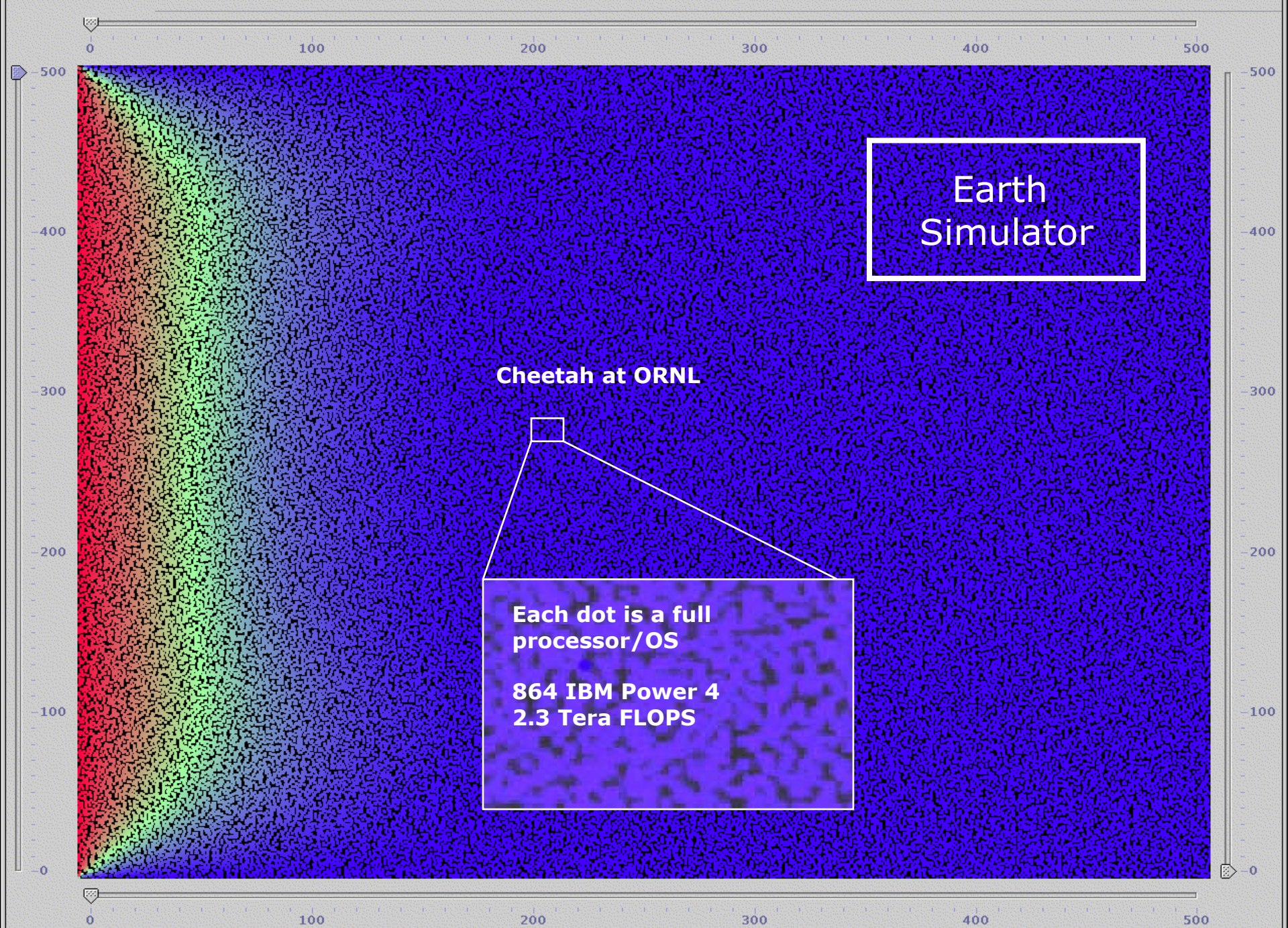
- Processes have only limited knowledge mostly about other processes in their neighborhood.
- Application is composed of local algorithms.
- Less inter-process dependencies, e.g not everyone needs to know when a process dies.
- Peer-to-peer communication with overlapping neighborhoods promotes scalability.



- MIT Media Lab. Research: Paintable Computing.

# Cellular Architecture Simulator

- Developed at ORNL in Java with native C and Fortran application support using JNI.
- Runs as standalone or distributed application.
- Lightweight framework simulates up to 1,000,000 lightweight processes on 9 real processors.
- Standard and experimental networks:
  - Multi-dimensional mesh/torus.
  - Nearest/Random neighbors.
- Message driven simulation is not in real-time.
- Primitive fault-tolerant MPI support.

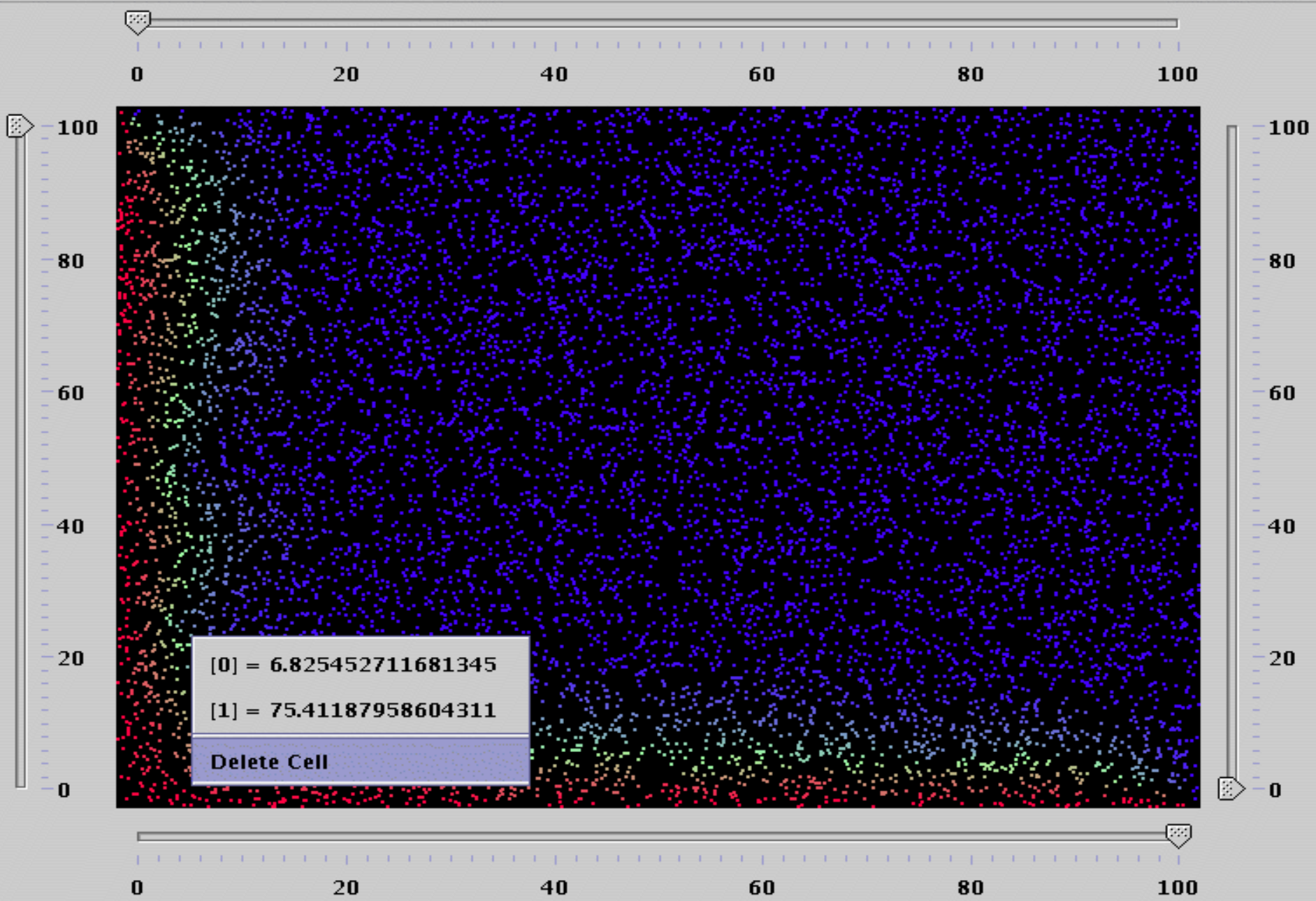


Earth  
Simulator

Cheetah at ORNL

Each dot is a full  
processor/OS

864 IBM Power 4  
2.3 Tera FLOPS



# Super-scalable Algorithms Research

- Extending the cellular algorithms theory to real world scientific applications.
- Exploring super-scale properties:
  - Scale invariance – fixed scaling factor that is independent from system and application size.
  - Natural fault-tolerance – algorithms get the correct answer despite failures without checkpointing.
- Gaining experience in programming models for computing on 100,000 processors.



# Explored Super-scalable Algorithms

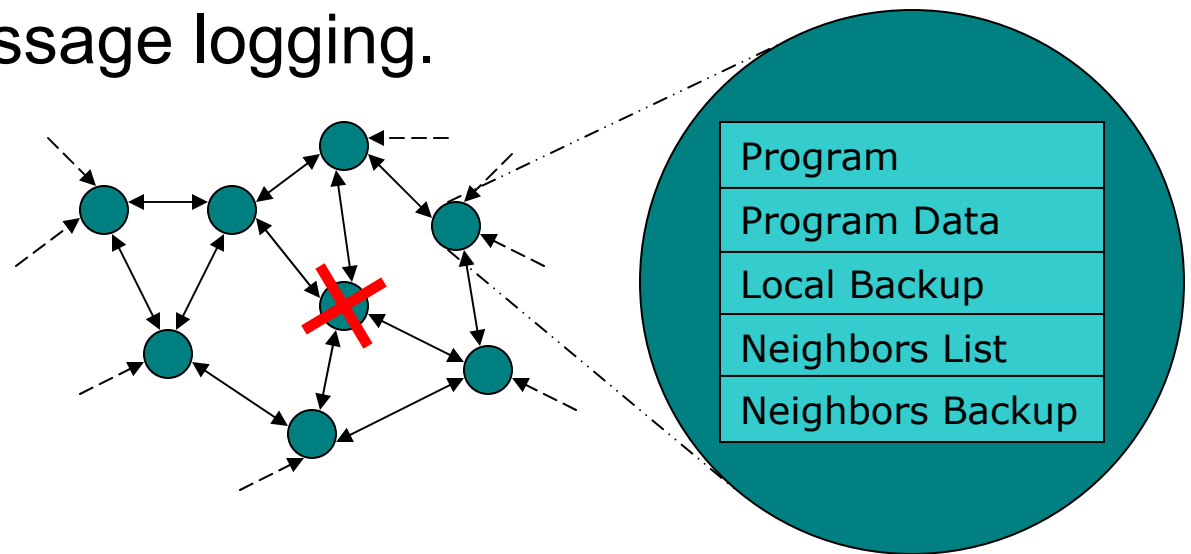
- Local information exchange:
  - Local peer-to-peer updates of values.
  - Mesh-free chaotic relaxation (Laplace/Poisson).
  - Finite difference/element methods.
  - Dynamic adaptive refinement at runtime.
  - Asynchronous multi-grid with controlled or independent updates between different layers.
- Global information exchange:
  - Global peer-to-peer broadcasts of values.
  - Global maximum/optimum search.

# Super-scalable Fault Tolerance

- For non-naturally fault tolerant algorithms.
  - Does it makes sense to restart all 100,000 processes because of one failure?
  - The mean time between failures (MTBF) is likely to be a few hours or just a few minutes.
  - Traditional centralized checkpointing and message logging are limited by bandwidth (bottleneck).
- Frequent checkpointing decreases app. efficiency.
- The failure rate is going to outrun the recovery rate.

# Super-scalable Diskless Checkpointing

- Decentralized peer-to-peer checkpointing.
- Processors hold backups of neighbors.
- Local checkpoint and restart algorithm.
- Coordination of local checkpoints.
- Localized message logging.



# Super-scalable Algorithms Research

- Super-scale systems with 100,000 and more processors become reality very soon.
- Super-scalable algorithms that are scale invariant and naturally fault-tolerant do exist.
- Diskless peer-to-peer checkpointing provides an alternative to natural fault-tolerance.
- A lot of research still needs to be done.



# Conclusions

- Oak Ridge National Laboratory performs basic and applied research in various areas.
- Capability computing is our path to world-class leadership computing.
- Next generation ultra-scale scientific high-end computing is a research challenge for:
  - ❑ Application software fault-tolerance.
  - ❑ High availability system software.
  - ❑ Super-scalable algorithms.

# High Availability for Ultra-Scale High-End Scientific Computing

**Christian Engelmann**

Network and Cluster Computing Group

Computer Science and Mathematics Division

Oak Ridge National Laboratory, Oak Ridge, USA

---

# Laboratories and Research Centers

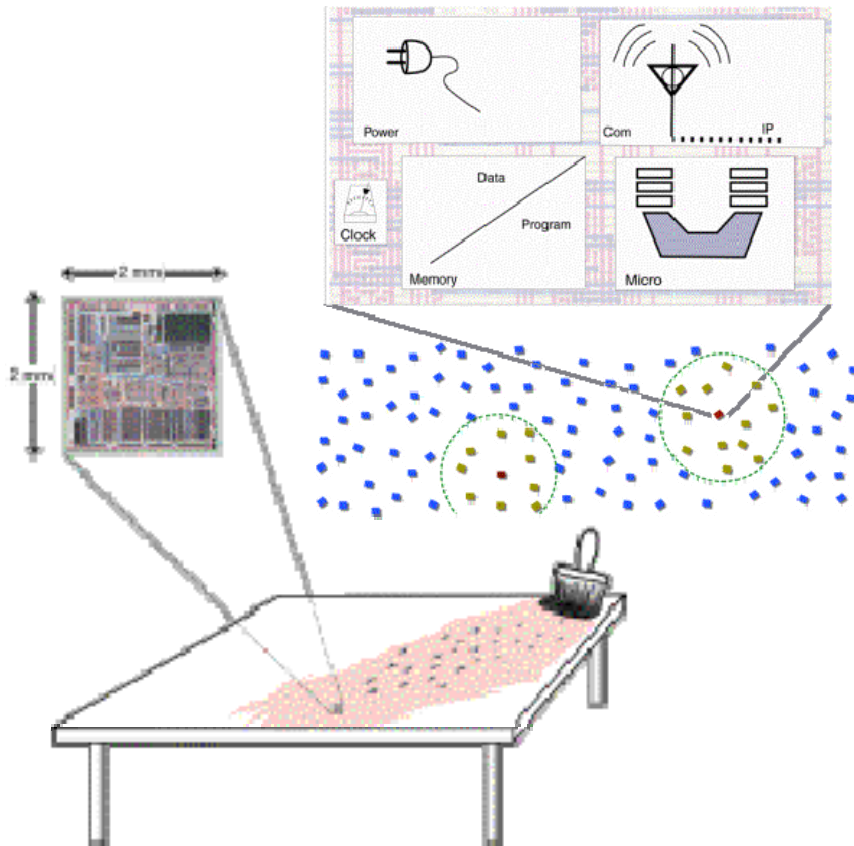
- Oak Ridge Electron Linear Accelerator (ORELA).
- Holifield Radioactive Ion Beam Facility (HRIBF).
- High Flux Isotope Reactor (HFIR).
- Spallation Neutron Source (SNS), *see next slide*.
- High Temperature Materials Laboratory (HTML).
- National Transportation Research Center (NTRC).
- ...
- Joint Institute for Computational Science (JICS).
- National Leadership Computing Facility (NLCF).

# Spallation Neutron Source at Oak Ridge National Laboratory





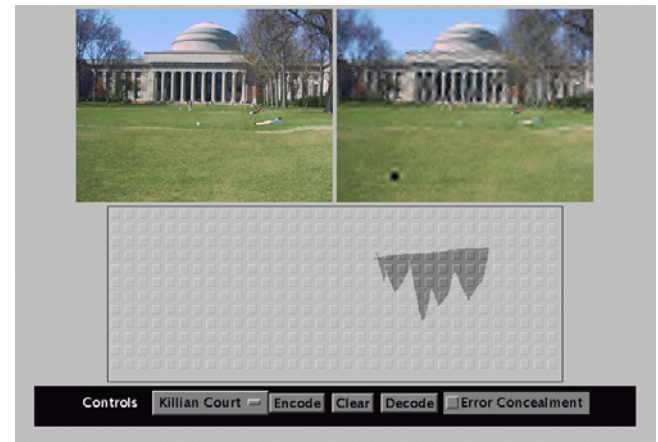
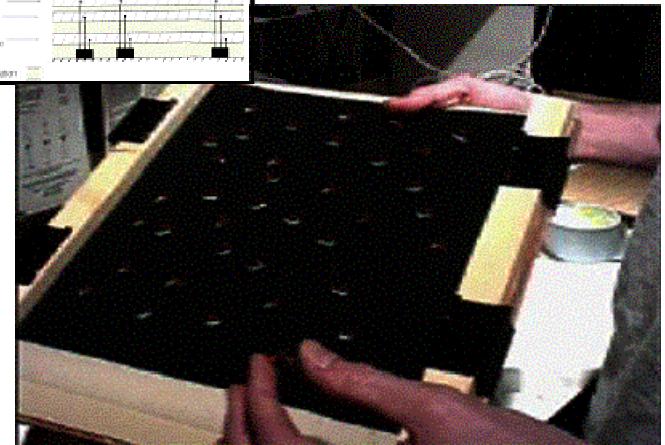
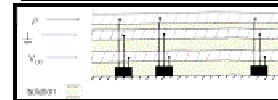
# MIT Research: Paintable Computing



- In the future, embedded computers with a radio device will get as small as a paint pigment.
- Supercomputers can be easily assembled by just painting a wall of embedded computers.
- Applications are driven by cellular algorithms.

# MIT Research: Pushpin Computing

- 100 embedded nodes.
- 1.25m x 1.25m pushpin board provides power.
- Initial applications:
  - Distributed audio stream storage.
  - Fault-tolerant holistic data (image) storage.
- Ongoing research:
  - Sensor networks.



# HARNNESS: Pluggable Heterogeneous Distributed Virtual Machine

## Exploring New Capabilities in Heterogeneous Distributed Computing

A Collaborative Research Effort Between Oak Ridge National Laboratory, University of Tennessee and Emory University

### Fault Tolerance

Petascale Approaches Beyond Standard Checkpoint/Restart

- Checksum Based (a la RAID)
- Localized State Neighborhoods
- Incremental Checkpointing



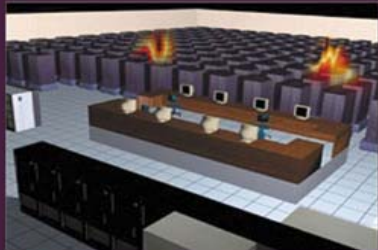
### Adaptability

New Dynamic Environments  
Collaborating and Personal VMs  
Pervasive Computing

### Multiple Plug-Ins and Parallel Paradigms

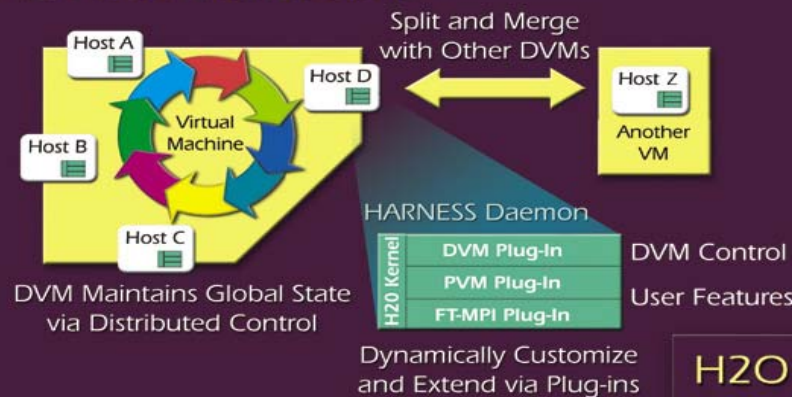


PVM Plug-In  
Application Monitoring  
Fault-Tolerant MPI Plug-In



FT-MPI Application Templates

### Harness Architecture



### GRID Lite

Personally Controlled (VM)  
Resource Sharing  
Minimum Modular Infrastructure  
Complements Existing DOE Data and Science Grids

### Near Stateless Computing

Task Communication  
Minimized Global State

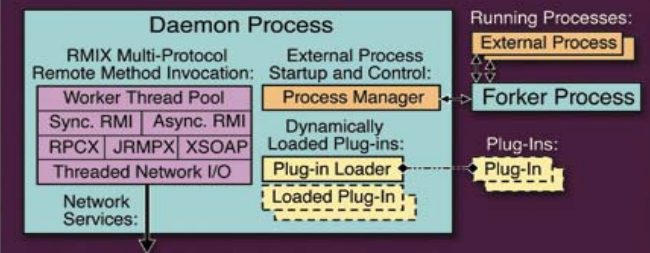
### Self-Assembling Virtual Machine

Parallel Plug-Ins Provide Capabilities  
Parallel Software Modules (Plug-Ins) for Flexibility and Dynamic Customization



### H2O Kernel

Implementations in C and Java  
Portable Multi-Threaded C Implementation



# OSCAR

Open  
Source  
Cluster  
Application  
Resources

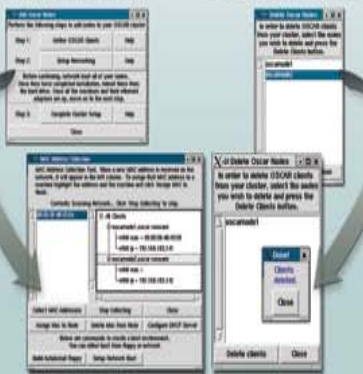
Installation



Management

Add Node Wizard

Delete Node Wizard



- Snapshot of "best known methods" for building, programming, and using clusters
- Consortium of academic/research and industry members
- Wizard-based cluster installation
  - Automatically configures headnode and services
  - Reduces time and expertise to build/install a cluster
  - Manages nodes and packages

- OSCAR - base build
- SSS-OSCAR - Scalable System Software build (SciDAC)
- Thin-OSCAR - diskless build
- HA-OSCAR - high availability cluster build
- SSI-OSCAR - single system image build based on Kerighed

[www.OpenClusterGroup.org/OSCAR](http://www.OpenClusterGroup.org/OSCAR)

# Cluster Power Tools

C3: Cluster, Command, and Control

- Scalable systems administrator
- Single system illusion (SSI) for clusters and multi-clusters
- Application and administration tools for secure cluster through multi-cluster access crossing administrative domains



Applications

C3 Command Line Cluster Power Tools

- cexec
- cpush
- cget
- cpushimage
- crm
- clist
- cname
- cnum
- ckill
- cshutdown

# MOLAR: Modular Linux and Adaptive Runtime Support

HEC Linux OS: modular, custom, lightweight

Kernel design				
Performance Observation	Monitoring	RAS	Testbeds	
Communications, IO	Extend/adapt runtime/OS	Root cause analysis	High availability	Provided

FastOS - Forum to Address Scalable Technology for Runtime and Operating Systems

Problem:

- Current OSs and runtime systems (OS/R) are unable to meet the various requirements to run large applications efficiently on future ultra-scale computers.

Goals:

- Development of a modular and configurable Linux framework.
- Runtime systems to provide a seamless coordination between system levels.
- Monitoring and adaptation of the operating system, runtime, and applications.
- Reliability, availability, and serviceability (RAS)
- Efficient system management tools

Impact:

- Enhanced support and better understanding of extremely scalable architectures
- Proof-of-concept implementation open to community researchers.

Participating organizations:



[www.csm.ornl.gov/ClusterPowerTools](http://www.csm.ornl.gov/ClusterPowerTools)

# Super-Scalable Algorithms: Developing a Foundation for Applications Scaling Beyond 100,000 Processors

<http://www.csm.ornl.gov/~gst>

## Self-healing Algorithms

Inherently fault-tolerant, these algorithms get the correct answer despite multiple task failures and without checkpointing.

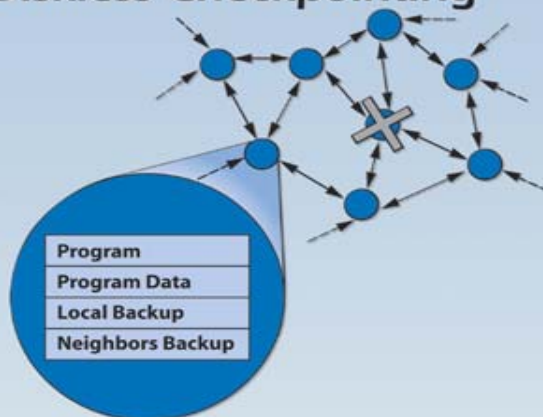
## Algorithms Developed:

- Finite difference and finite element
- Multigrid
- Global Maximum
- Monte Carlo

## Collaboration with IBM in Support of:

- design of Blue Gene\L
- creation of life sciences and nanotechnology applications for Blue Gene\L

## Diskless Checkpointing



- For non-healing algorithms, such as FFT, peer-to-peer diskless checkpointing improves scalability.
- Processors distribute their state to their logical neighbors using RAID like technology

## 1,000,000 Processor Simulator

Simulator allows testing of fault-tolerance and scalability beyond current systems.

## Simulator Features:

- Parallel Java running on Windows desktop to Linux cluster
- Adjustable topology: mesh, torus, random and manual
- Simulation of failures in single nodes or groups
- Supports native C and FORTRAN for scientific applications