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.
- Fault-tolerant heterogeneous metacomputing.
- High availability system software framework.
- Super-scalable algorithms for computing on 100,000 processors.

Research at Oak Ridge National Laboratory

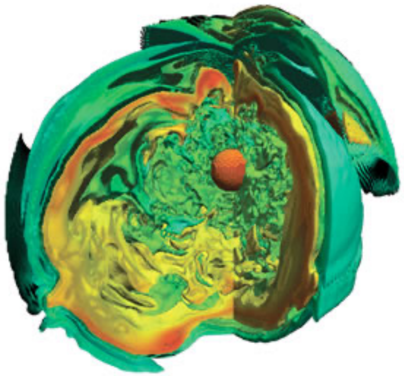
Christian Engelmann

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

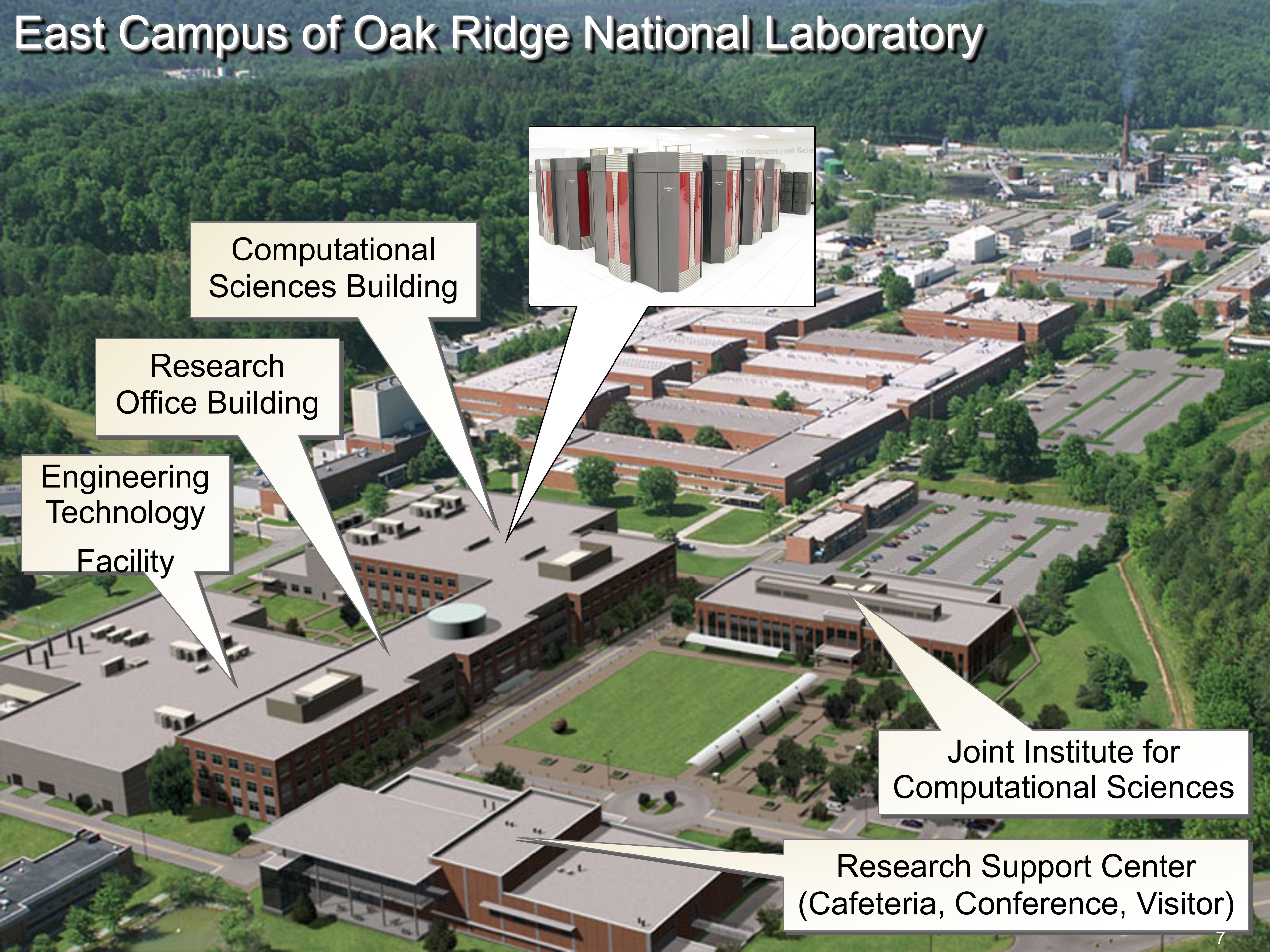
OAK RIDGE NATIONAL LABORATORY

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 DOE.
- Total land area: 58mi^2 (150km^2).
- ~3000 guest researchers each year.
- ~30,000 visitors each year.

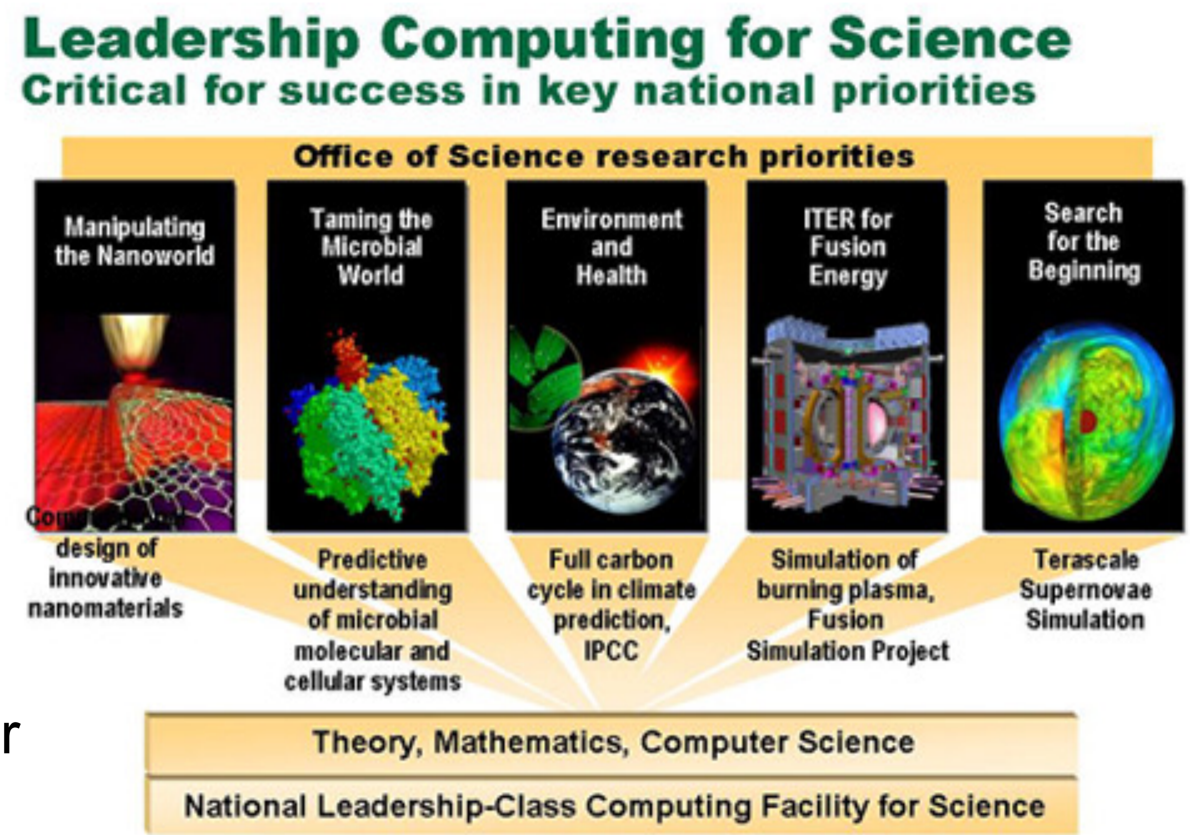


East Campus of Oak Ridge National Laboratory



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.



More information: www.nlcf.gov

Center for Computational Sciences

- Computer center with 40,000 ft^2 ($3700m^2$) floor space.
- 4 systems in the Top 500 List of Supercomputer Sites:
  - 11. Cray XT3, MPP with 5212P,10TB ⇨ 25 TFLOPS.
  - 50. Cray X1, Vector with 1024P, 4TB ⇨ 18 TFLOPS.
  - 143. IBM Power 4, Cluster with 864P, 1TB ⇨ 4.5 TFLOPS.
  - 362. SGI Altix, SSI with 256P, 2TB ⇨ 1.4 TFLOPS.

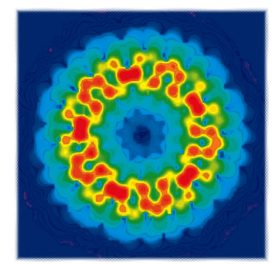
Leadership Computing Roadmap

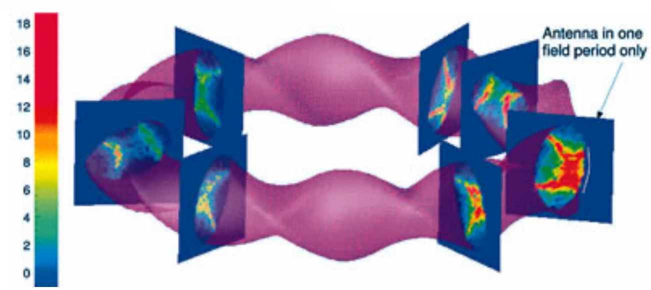
- Planned upgrades next year:
  - Cray XT3 to 20000P/40TB ⇨ 100 TFLOPS.
- Future roadmap:
  - ~ 2007 Upgrade Cray X1e to X2.
  - ~ 2007 Upgrade Cray XT3 to 250 TFLOPS.
  - ~ 2009 Installation of a 1 PFLOP system.

Cray Center of Excellence at ORNL

Computer Science Research Groups

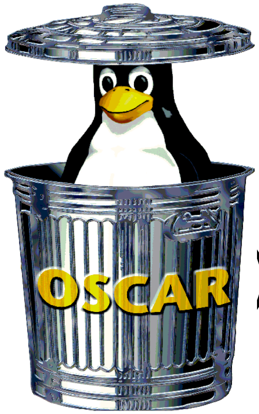
- Computer Science and Mathematics (CSM) Division.
  - Applied research focused on computational sciences, intelligent systems, and information technologies.
- CSM Research Groups:
  - Climate Dynamics
  - Computational Biology
  - Computational Chemical Sciences
  - Computational Materials Science
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





Network & Cluster Computing Projects

FT-MPI

- 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

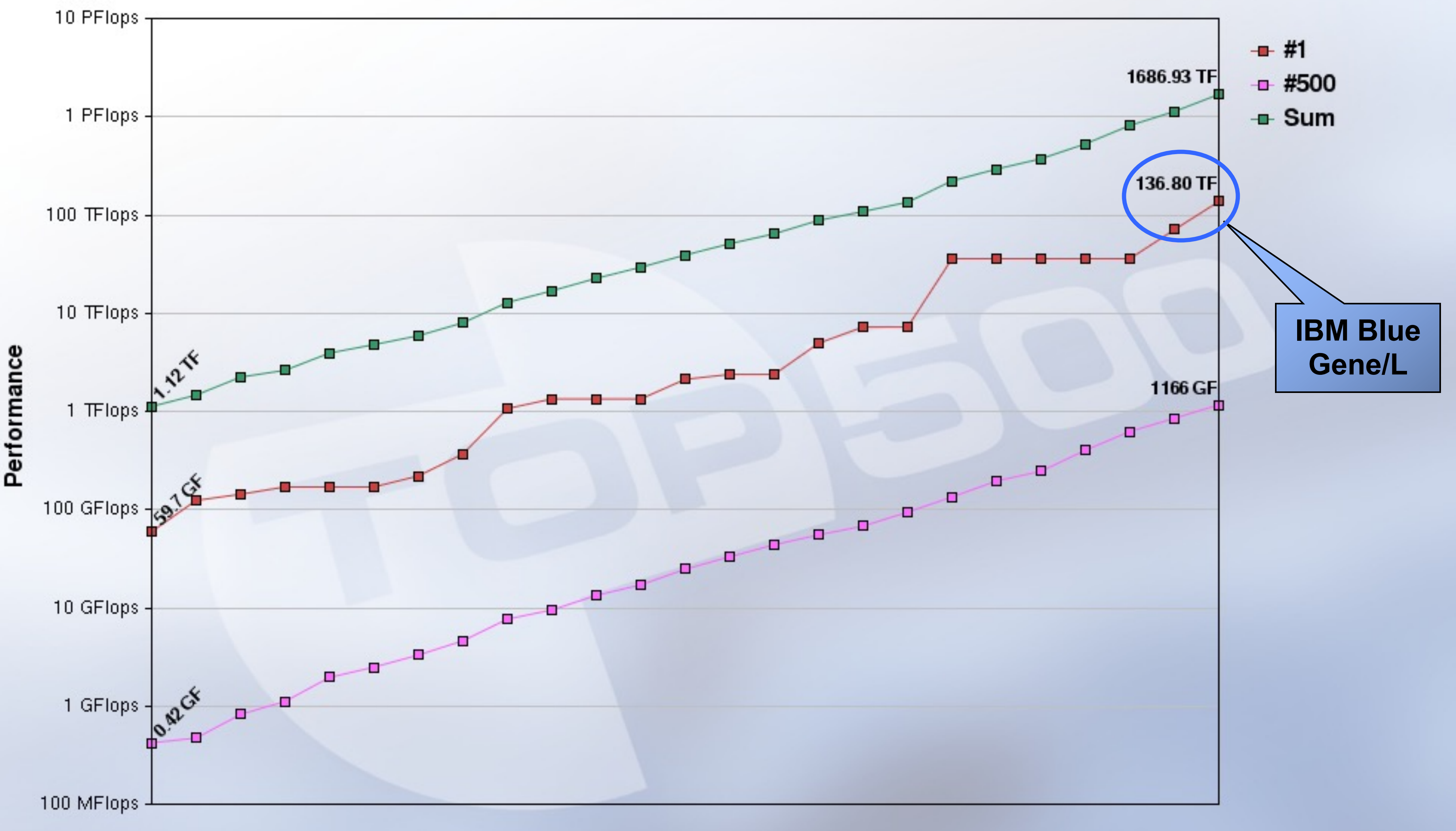
Christian Engelmann

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

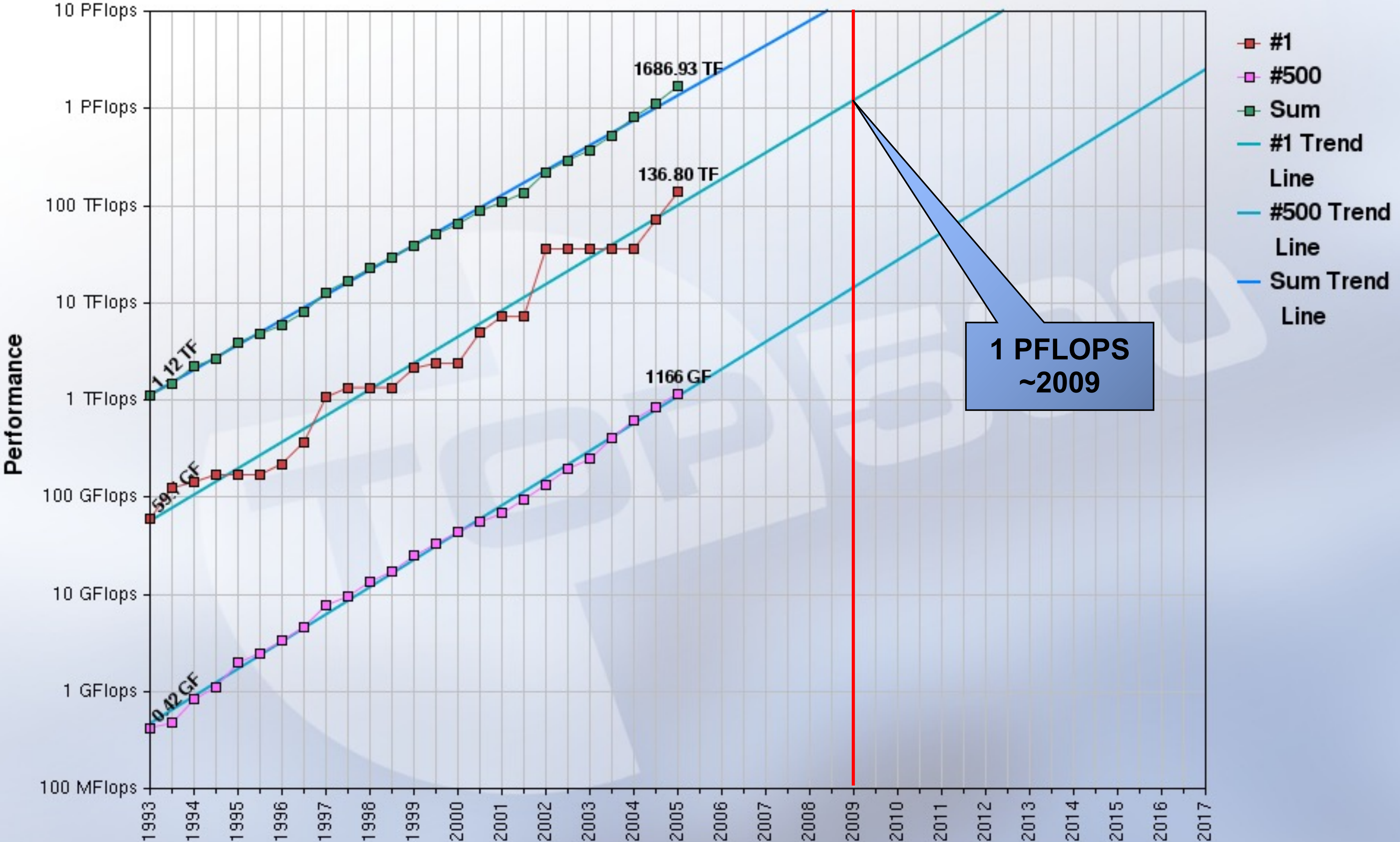
Scientific High-End Computing

- 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).

Performance Development



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.

Office of Science
U.S. DEPARTMENT OF ENERGY

OAK RIDGE NATIONAL LABORATORY
MANAGED BY UT-BATTELLE FOR THE DEPARTMENT OF ENERGY

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

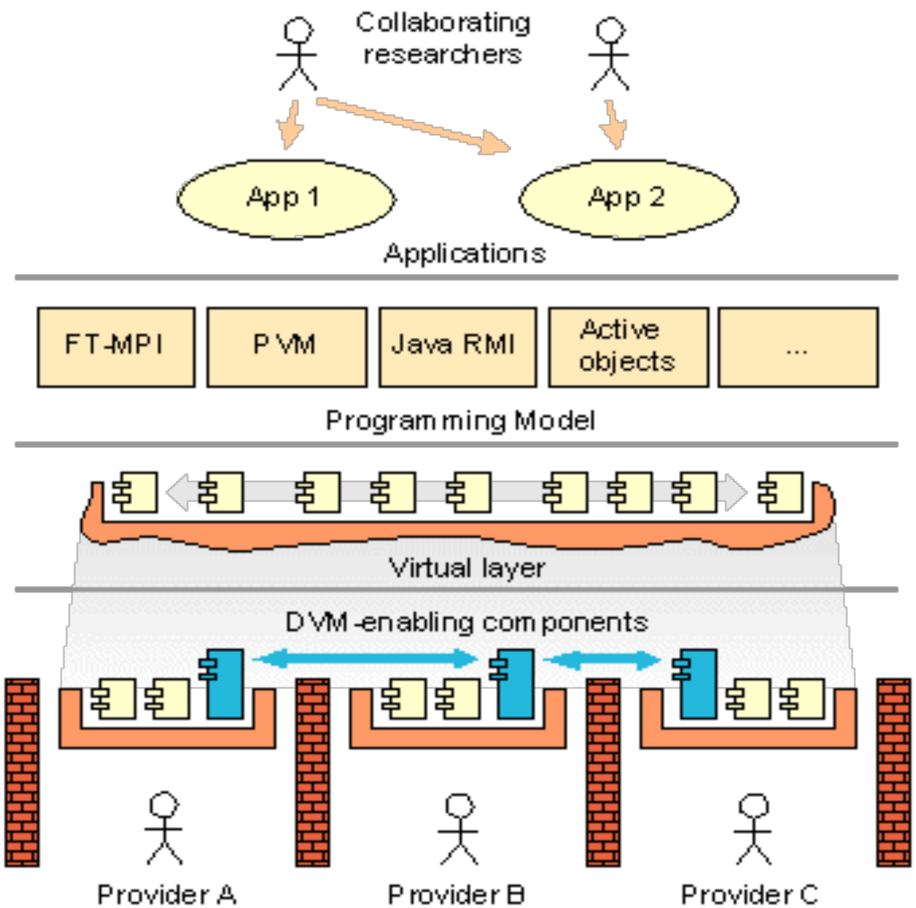
What is Harness

- 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

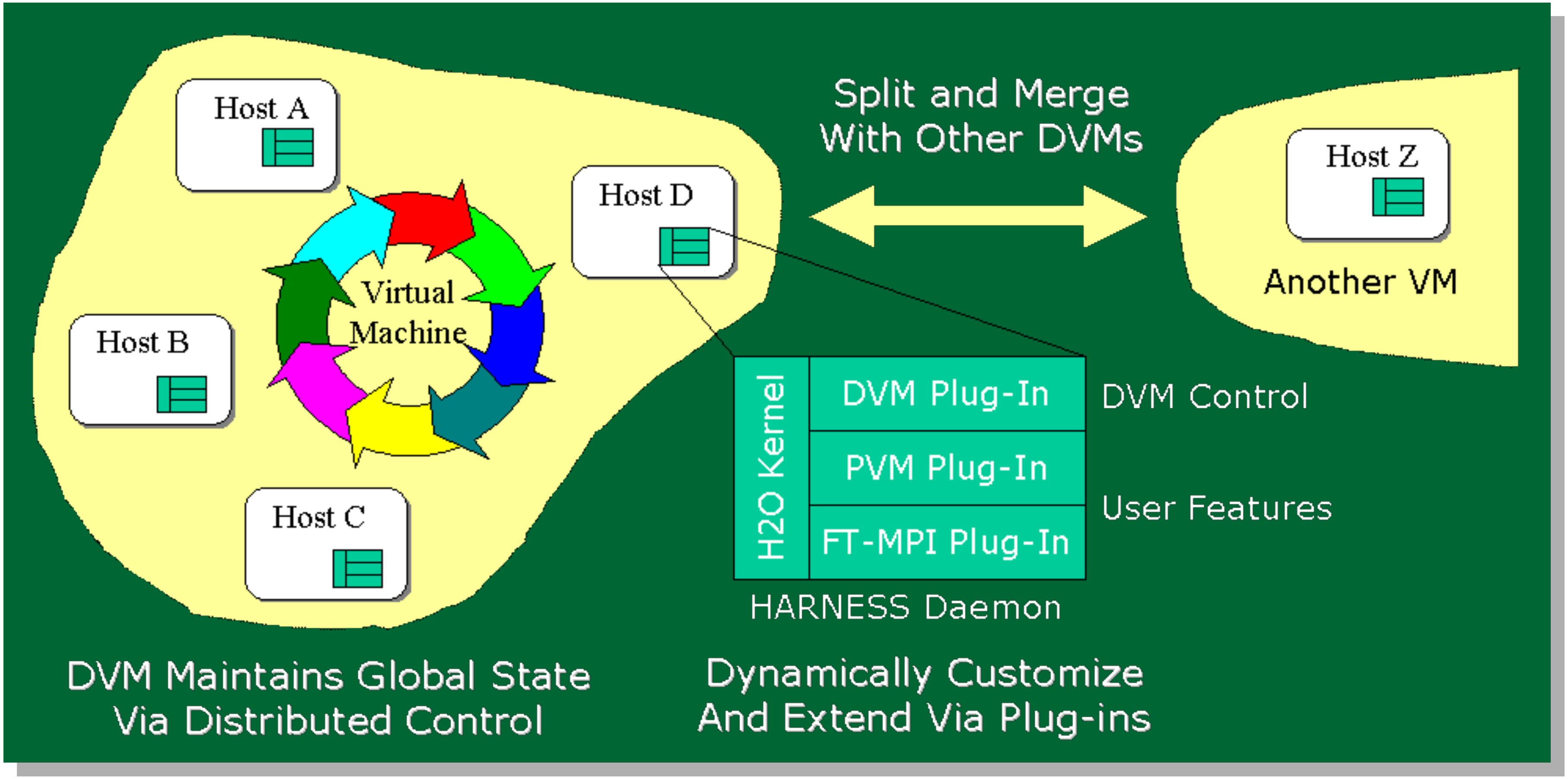
- Lightweight, pluggable software frameworks.
- Adaptive, reconfigurable runtime environments.
- Parallel plug-ins and diverse programming paradigms.
- Highly available distributed virtual machines (DVMs).
- Advanced ultra-scale approaches for fault tolerance.
- Fault-tolerant message passing (FT-MPI).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

Harness Architecture



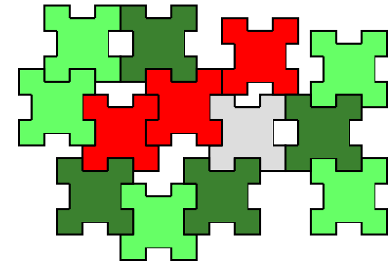
- Light-weight kernels share their resources.
- Plug-ins offer services.
- Support for diverse programming models.
- Distributed Virtual Machine (DVM) layer.
- Highly available DVM.
- Highly available plug-in services via DVM.

Harness DVM Architecture



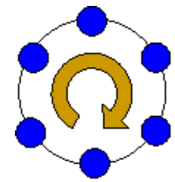
Harness Plug-ins

FT-MPI

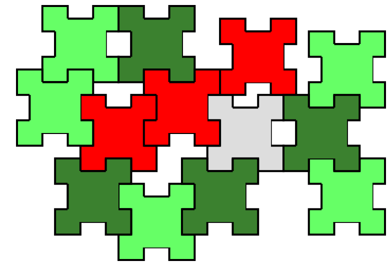


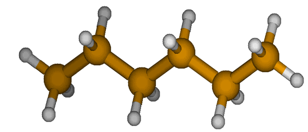
- PVM emulation plug-in:
  - Replaces the PVM daemon.
  - Allows users a seamless transition to Harness.
  - Plug-ins and applications just link libpvm.
  - PVM is controlled with the Harness console.
- Fault-tolerant MPI (FT-MPI) plug-in:
  - Combines several FT-MPI services in one plug-in.
  - Plug-ins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

Harness Plug-ins

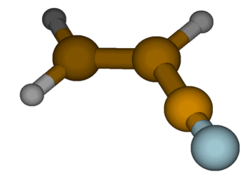


Distributed Control



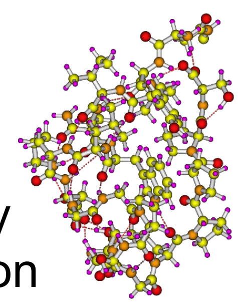


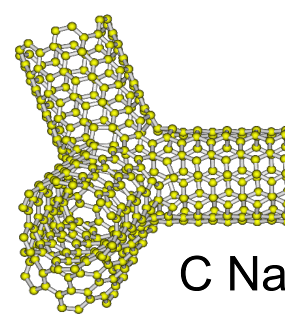
SCC-DFTB



- 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.

Geometry Optimization





C Nanotubes

High Availability System Software Framework

Christian Engelmann

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

Research Motivation

- 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.

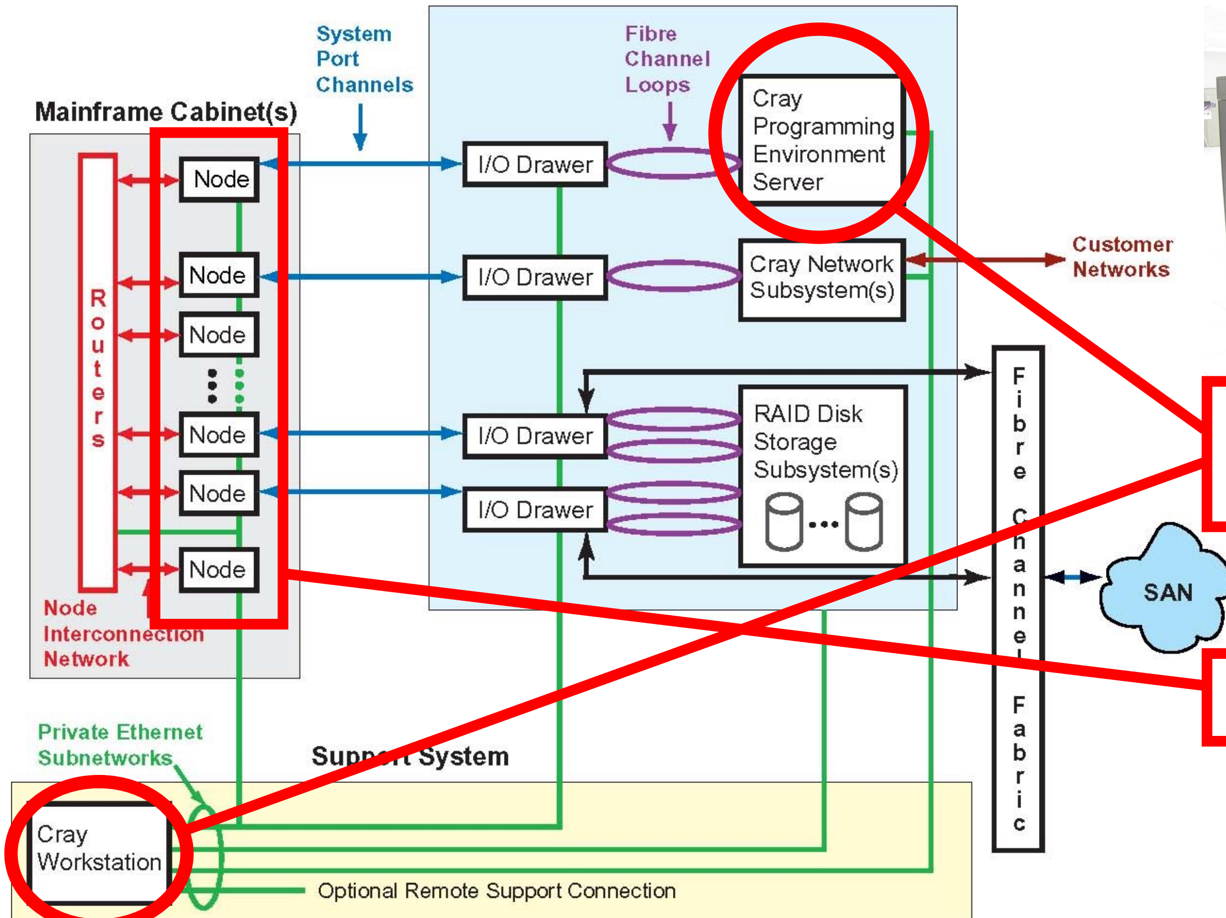
➢ *Why do we accept such significant system outages due to failures, maintenance or repair?*

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

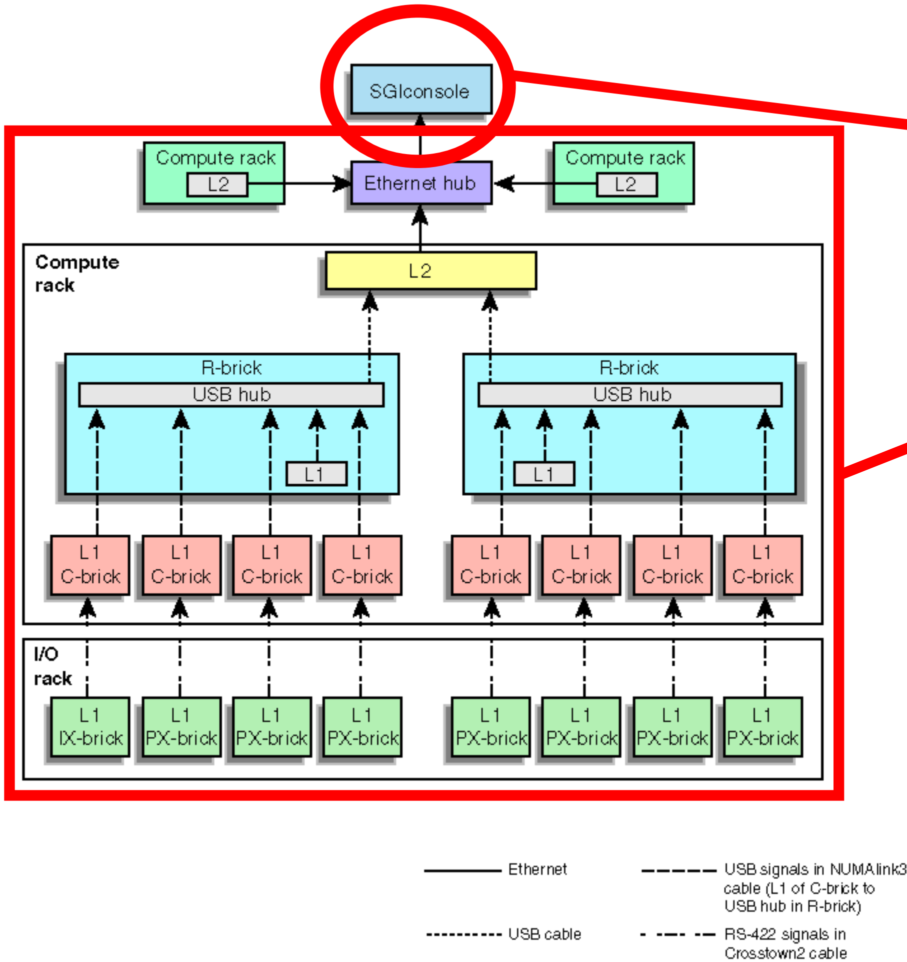
Vector Machines: Cray X1 (Phoenix)



Single Points of Failure, Single Points of Control

Single Points of Failure

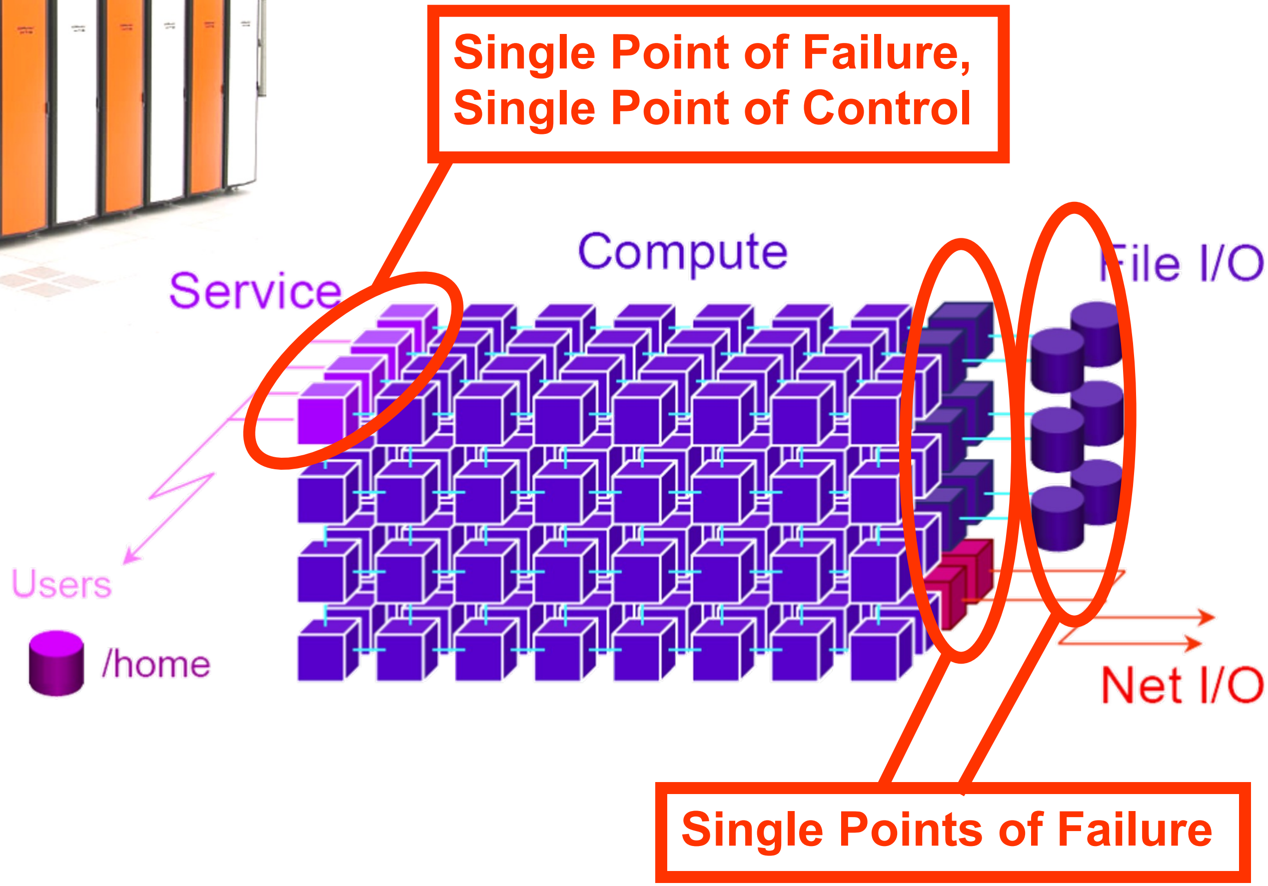
SSI Systems: SGI Altix (Ram)



Single Point of Failure, Single Point of Control

Single Points of Failure

MPPs: Cray XT3 (Jaguar)



Research Goals

- Provide high-level RAS capabilities similar to the IT/telecommunication industry (3-4 nines).
- Eliminate many of the numerous single-points of failure and control in today's terascale systems.
- Improve scalability and access to systems and data.
- *Development of techniques to enable terascale systems to run computational jobs 24x7.*
- *Development of proof-of-concept implementations as blueprint for production-type RAS solutions.*

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:

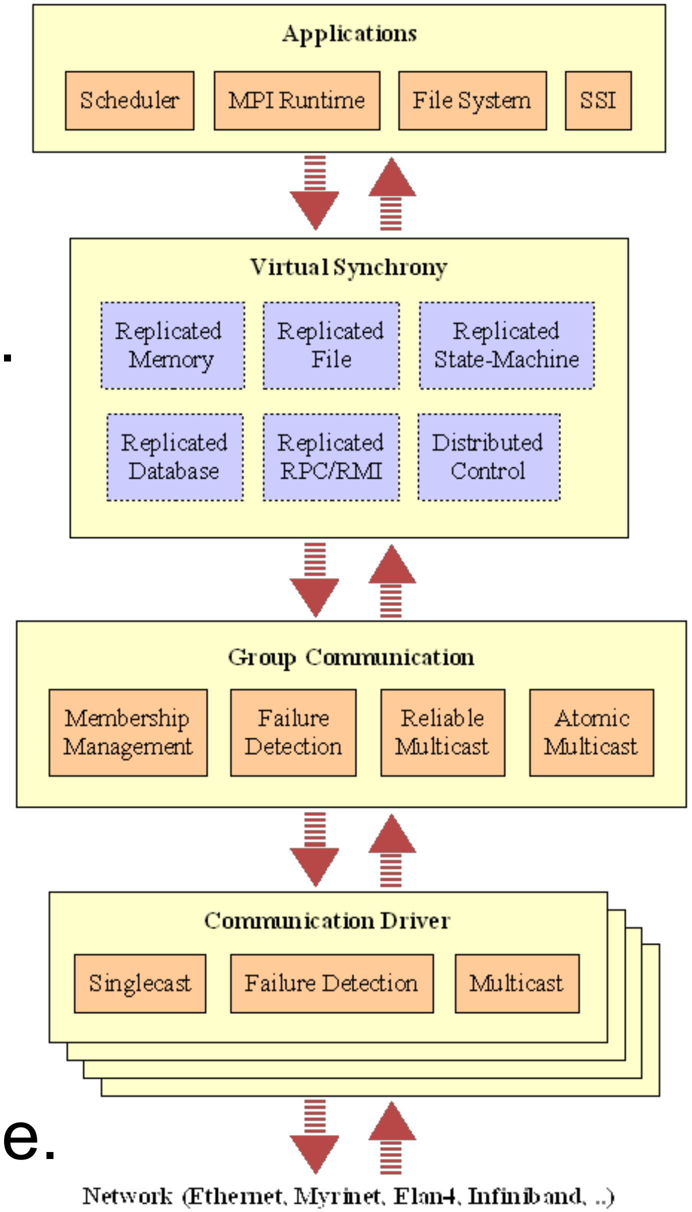
- HARNESS with symmetric distributed virtual machine.
- Similar projects: Cactus …
- Heterogeneous adaptable distributed middleware.
- No system level support.
- Solutions not flexible enough.

➢ *System-level data replication and distributed control service needed for active/active head node solution.*

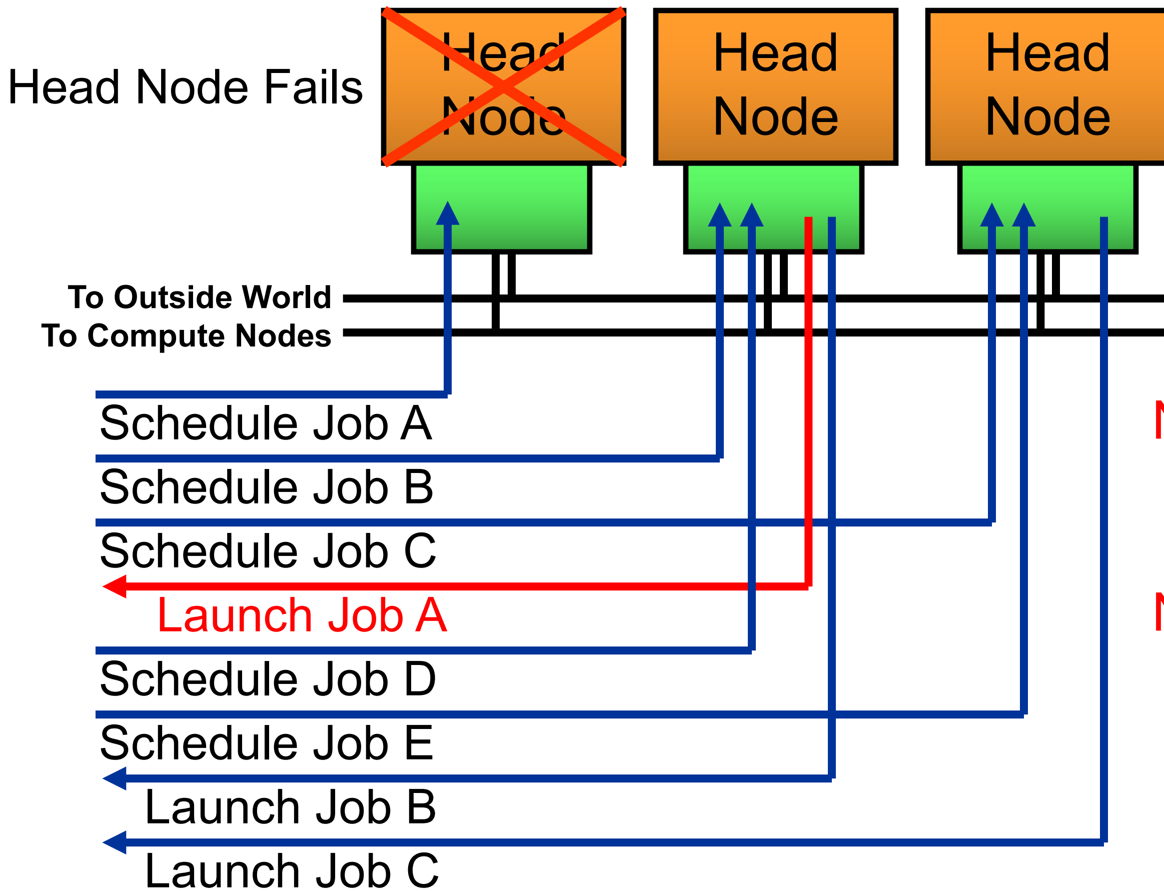
➢ *Reconfigurable framework similar to HARNESS needed to adapt to system properties and application needs.*

RAS Framework

- Pluggable component framework.
  - Communication drivers.
  - Group communication.
  - Virtual synchrony.
  - *Applications.*
- Interchangeable components.
- Adaptation to application needs, such as level of consistency.
- Adaptation to system properties, such as network and system scale.



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



No Single Point of Failure

No Single Point of Control

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

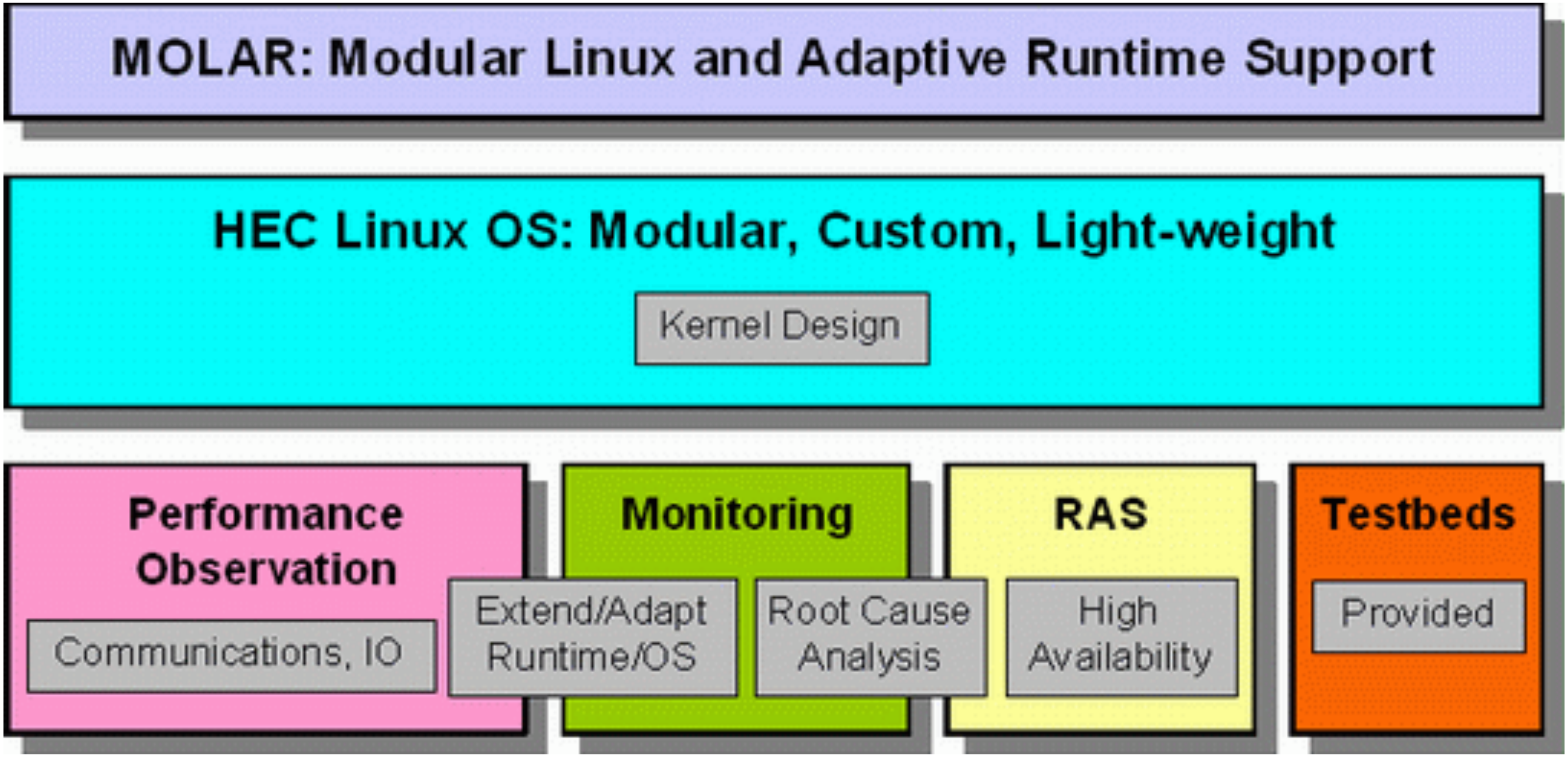
OHIO STATE

LOUISIANA TECH UNIVERSITY

The University of Reading

CRAY

MOLAR: HEC OS/R Research Map



Office of Science
U.S. DEPARTMENT OF ENERGY

OAK RIDGE NATIONAL LABORATORY
MANAGED BY UT-BATTELLE FOR THE DEPARTMENT OF ENERGY

Super-Scalable Algorithms for Computing on 100,000 Processors

Christian Engelmann

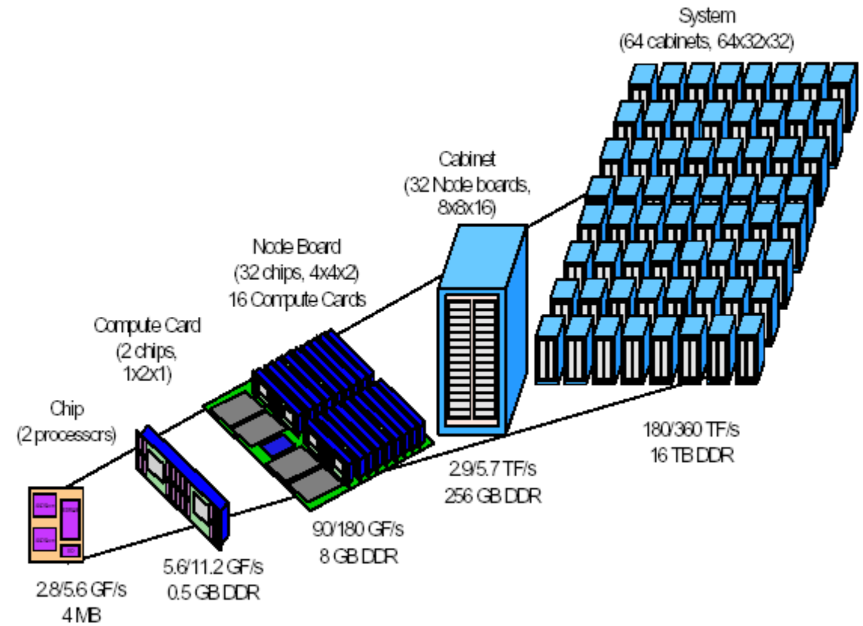
Network and Cluster Computing Group
Computer Science and Mathematics Division
Oak Ridge National Laboratory, Oak Ridge, USA

Super-Scale Architectures

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

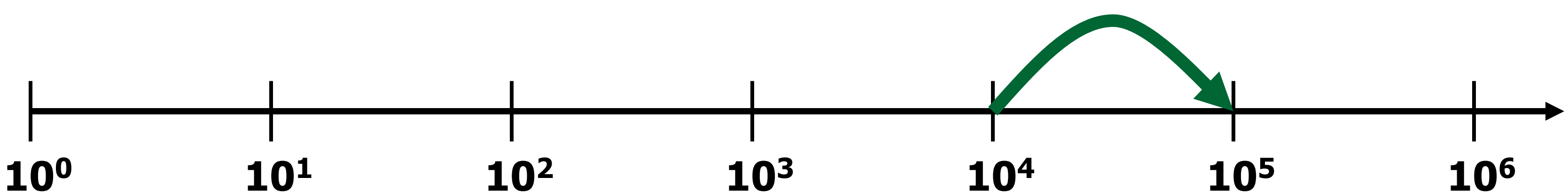
IBM Blue Gene/L

- 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.

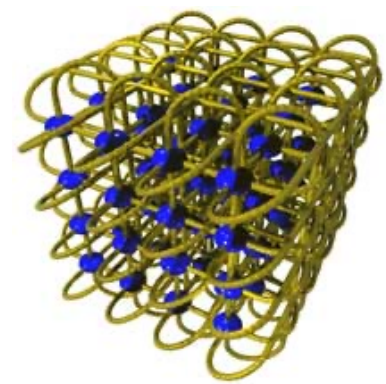


Fault-tolerance Issues

- How to survive on 100,000 processors?
- Failure rate grows with the system size.
- Mean time between failures (MBTF) 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.



- <u>Get scientists to think about scalability and fault-tolerance in super-scale systems!</u>

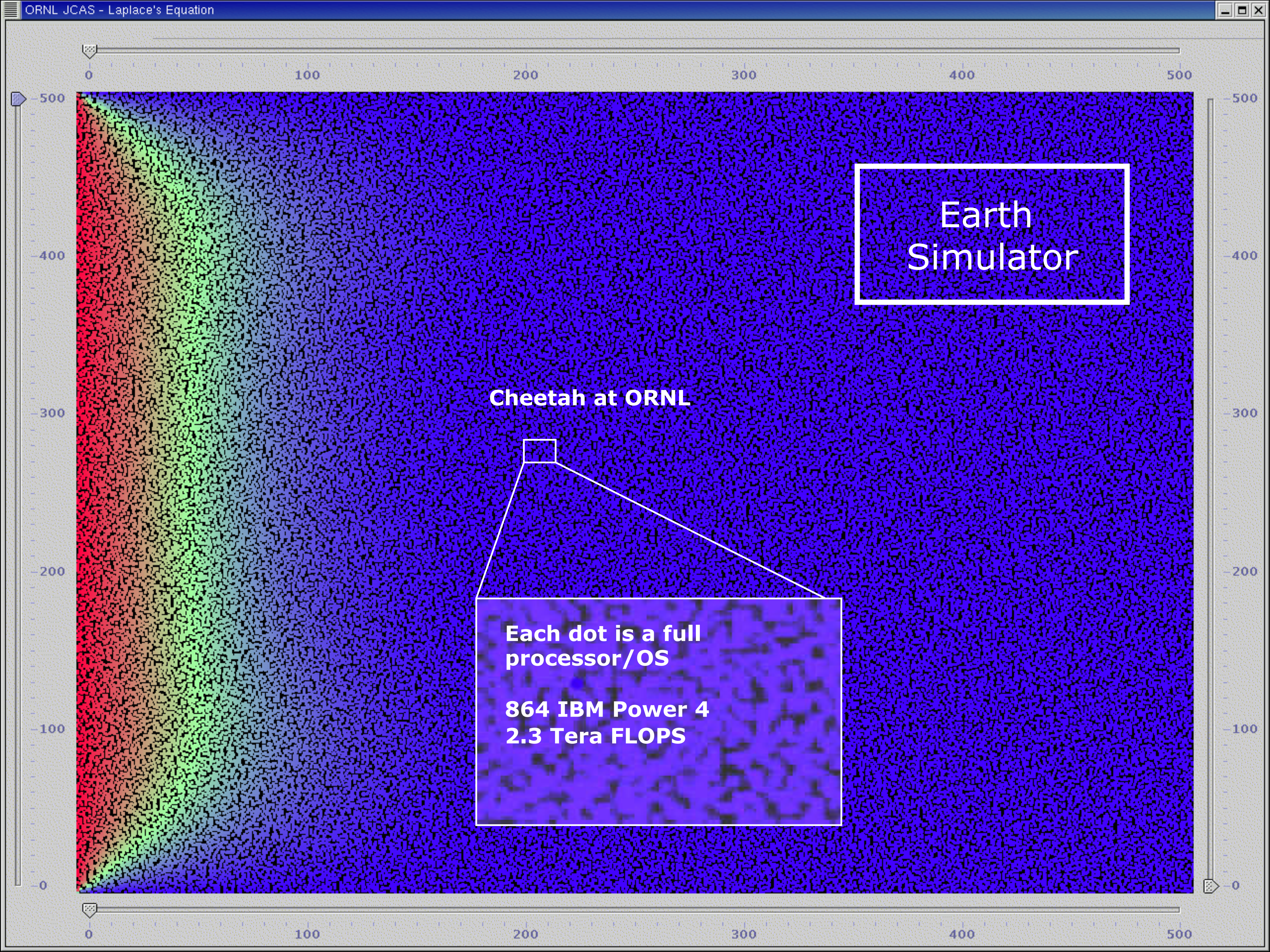
Cellular Algorithms Theory

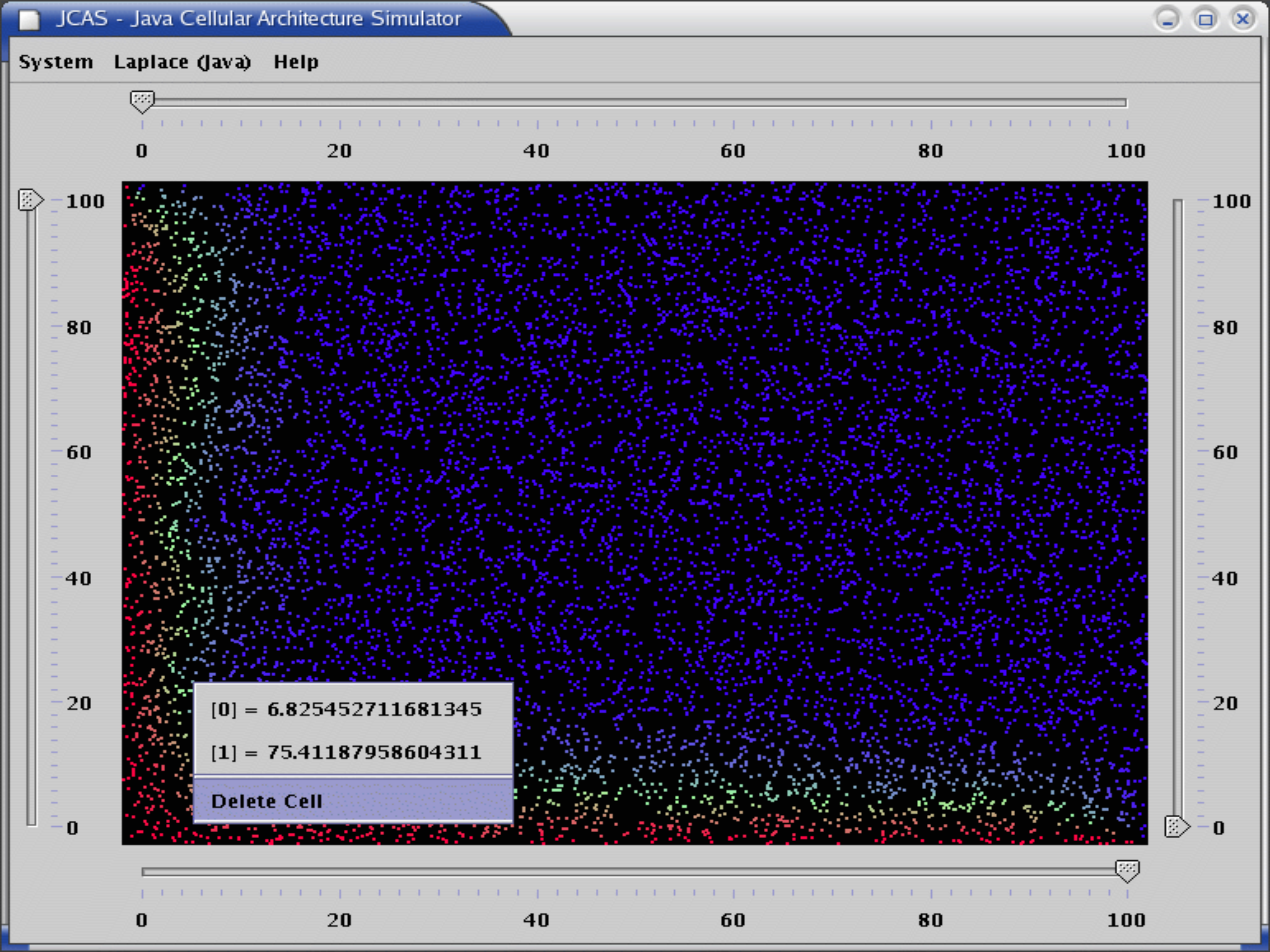
- Processes have only limited knowledge mostly about other processes in their neighborhood.
- Application is composed of local algorithms.
- Less inter-process dependencies, e.g not everyone needs to know when a process dies.
- Peer-to-peer communication with overlapping neighborhoods promotes scalability.

- 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.


ORNL JCAS - Laplace's Equation
Earth Simulator
Cheetah at ORNL
Each dot is a full processor/OS
864 IBM Power 4
2.3 Tera FLOPS
0
100
200
300
400
500


JCAS - Java Cellular Architecture Simulator
System
Laplace (Java)
Help
0
20
40
60
80
100
[0] = 6.825452711681345
[1] = 75.41187958604311
Delete Cell

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.

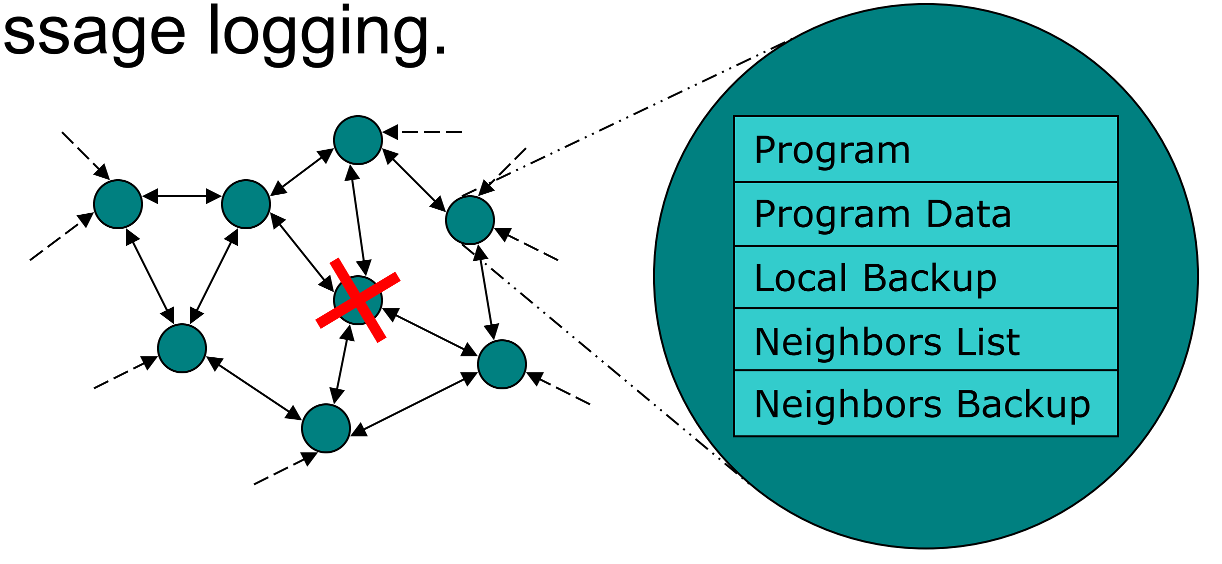
Super-scalable Fault Tolerance

- 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.



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