Advancing Reliability, Availability and Serviceability for High-Performance Computing

Stephen L. Scott and Christian Engelmann

Computer Science and Mathematics Division

Oak Ridge National Laboratory, Oak Ridge, TN, USA

Talk Outline

- Computer science research at Oak Ridge National Laboratory: Who we are and what we do…
- Reliability, availability and serviceability deficiencies of today's scientific high-end computing systems.
- High availability solutions for scientific high-end computing systems.

Computer Science Research at Oak Ridge National Laboratory: Who we are and what we do…

Stephen L. Scott

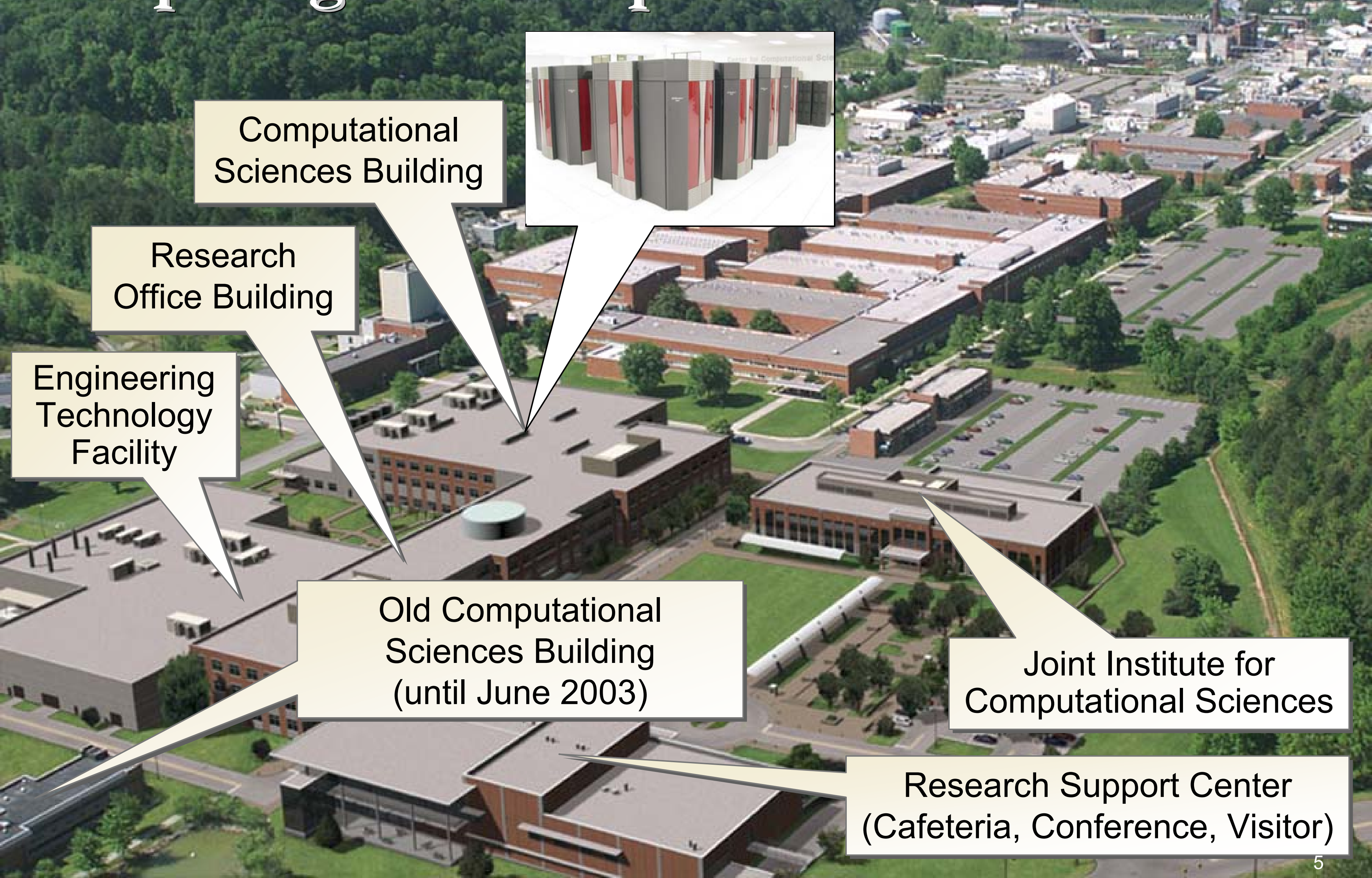
Computer Science and Mathematics Division

Oak Ridge National Laboratory, Oak Ridge, TN, USA

Largest Multipurpose Science Laboratory within the U.S. Department of Energy

- Privately managed for US DOE
- $1.06 billion budget
- 3,900 employees total
  - 1500 scientists and engineers
- 3,000 research guests annually
- 30,000 visitors each year
- Total land area 58mi^2 (150km^2)
- Nation's largest energy laboratory
- Nation's largest science facility:
  - The $1.4 billion Spallation Neutron Source
- Nation's largest concentration of open source materials research
- Nation's largest open scientific computing facility
- $300 million modernization in progress

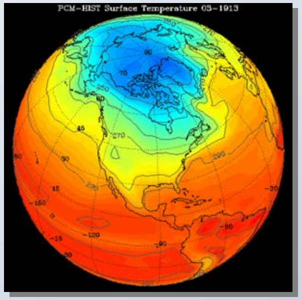
ORNL East Campus: Site of World Leading Computing and Computational Sciences



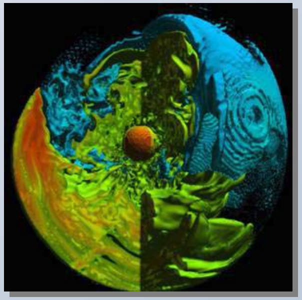
National Center for Computational Sciences

- 40,000 ft^2 (3700 m^2) computer center:
  - 36-in (~1m) raised floor, 18 ft (5.5 m) deck-to-deck
  - 12 MW of power with 4,800 t of redundant cooling
  - High-ceiling area for visualization lab:
    - 35 MPixel PowerWall, Access Grid, etc.
- 3 systems in the Top 500 List of Supercomputer Sites:
  - Jaguar: 10. Cray XT3, Cluster with 5212P,10TB ⇨ 25 TFLOPS.
  - Phoenix: 17. Cray X1E, Vector with 1024P, 4TB ⇨ 18 TFLOPS.
  - Cheetah: 283. IBM Power 4, Cluster with 864P, 1TB ⇨ 4.5 TFLOPS.
  - Ram: SGI Altix, SSI with 256P, 2TB ⇨ 1.4 TFLOPS.

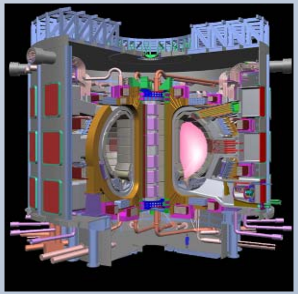
At Forefront in Scientific Computing and Simulation



- Leading partnership in developing the National Leadership Computing Facility
  - Leadership-class scientific computing capability
  - 100 TFLOPS in 2006
  - 250 TFLOPS in 2007
  - 1 PFLOP in 2009



- Attacking key computational challenges
  - Climate change
  - Nuclear astrophysics
  - Fusion energy
  - Materials sciences
  - Biology

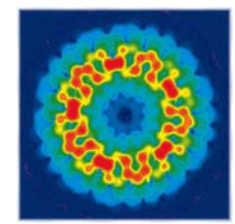


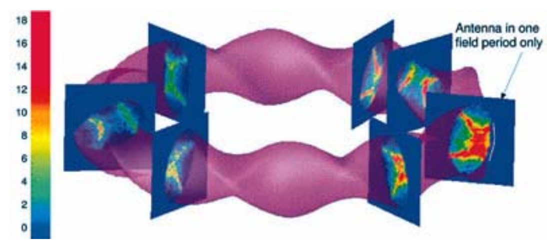
- Providing access to computational resources through high-speed networking (10Gbps)

Computer Science Research Groups

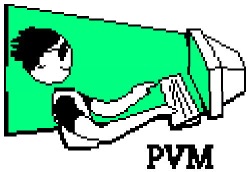
- Computer Science and Mathematics (CSM) Division.
  - Applied research focused on computational sciences, intelligent systems, and information technologies.

- CSM Research Groups:
  - Climate Dynamics
  - Complex Systems
  - Computational Chemical Sciences
  - Computational Materials Science
  - Future Technologies
  - Statistics and Data Science
  - Computational Mathematics
  - *Network and Cluster Computing (~23 researchers, ++)*



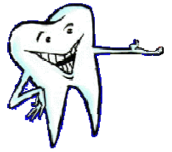


Network & Cluster Computing Projects

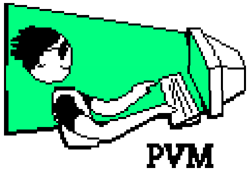


FT-MPI

- Parallel Virtual Machine (PVM).
- MPI Specification, FT-MPI and Open MPI.
- Common Component Architecture (CCA).
- Open Source Cluster Application Resources (OSCAR).
- Scalable cluster tools (C3).
- Scalable Systems Software (SSS).
- Fault-tolerant metacomputing (HARNESS).
- High availability for high-end computing (RAS/MOLAR).
- Super-scalable algorithms research.
- Parallel storage systems (Freeloader).



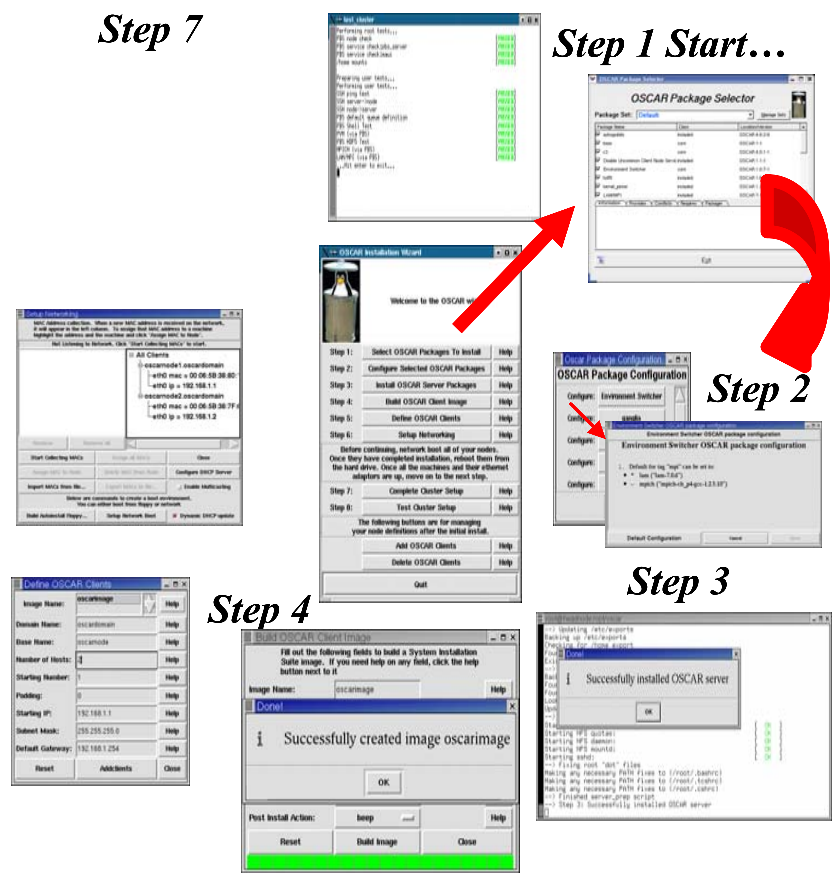
Network & Cluster Computing Projects



- Parallel Virtual Machine (PVM).
- MPI Specification, FT-MPI and Open MPI.
- Common Component Architecture (CCA).
- Open Source Cluster Application Resources (OSCAR).
- Scalable cluster tools (C3).
- Scalable Systems Software (SSS).
- Fault-tolerant metacomputing (HARNESS).
- High availability for high-end computing (RAS/MOLAR).
- Super-scalable algorithms research.
- Parallel storage systems (Freeloader).

Open Source Cluster Application Resources

- OSCAR Framework (cluster installation configuration and management)
  - Remote installation facility
  - Small set of "core" components
  - Modular package & test facility
  - Package repositories
- Use "best known methods"
  - Leverage existing technology where possible
- Wizard based cluster software installation
  - Operating system
  - Cluster environment
    - Administration
    - Operation
- Automatically configures cluster components
- Increases consistency among cluster builds
- Reduces time to build / install a cluster
- Reduces need for expertise



OSCAR Components

- Administration/Configuration
  - SIS, C3, OPIUM, Kernel-Picker, NTPconfig cluster services (dhcp, nfs, ...)
  - Security: Pfilter, OpenSSH
- HPC Services/Tools
  - Parallel Libs: MPICH, LAM/MPI, PVM
  - Torque, Maui, OpenPBS
  - HDF5
  - Ganglia, Clumon, … [monitoring systems]
  - *Other 3rd party OSCAR Packages*
- Core Infrastructure/Management
  - System Installation Suite (SIS), Cluster Command & Control (C3), Env-Switcher
  - OSCAR DAtabase (ODA), OSCAR Package Downloader (OPD)

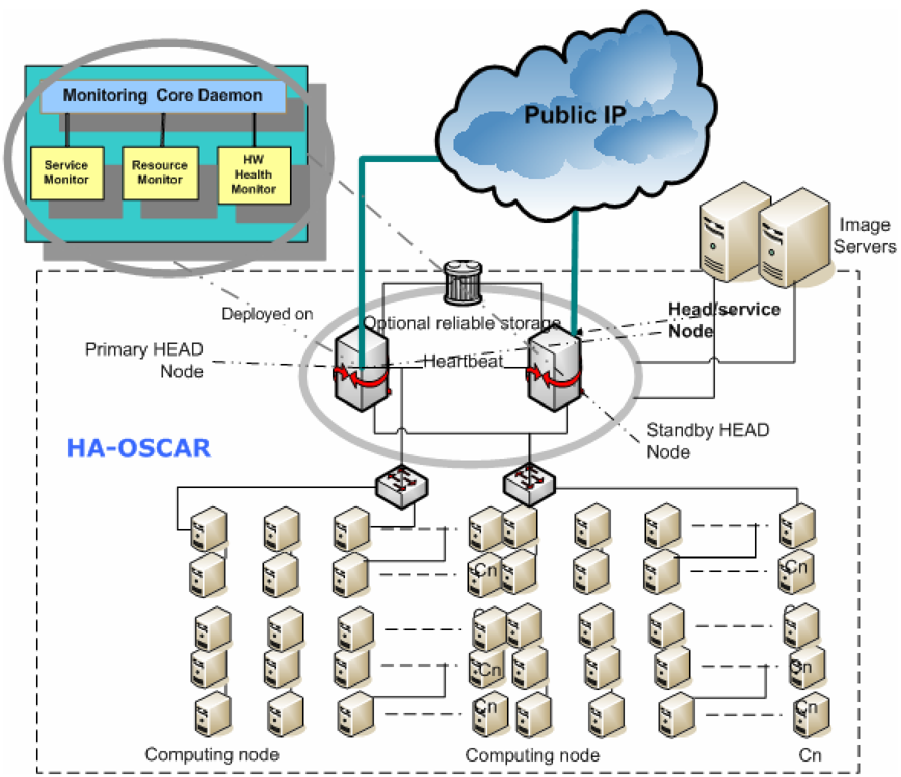
OSCAR Core Participants

- Intel
- Bald Guy Software
- Revolution Linux
- INRIA
- EDF
- Canada's Michael Smith Genome Sciences Center
- Indiana University
- Oak Ridge National Laboratory
- Louisiana Tech University
- NEC HPC Europe

SSI-OSCAR
Single System Image Open Source Application Resources

- Easy use thanks to SSI systems
  - SMP illusion
  - High Performance
  - Fault Tolerance
- Easy management thanks to OSCAR
  - Automatic cluster install / update

HA-OSCAR: RAS Management for Clusters



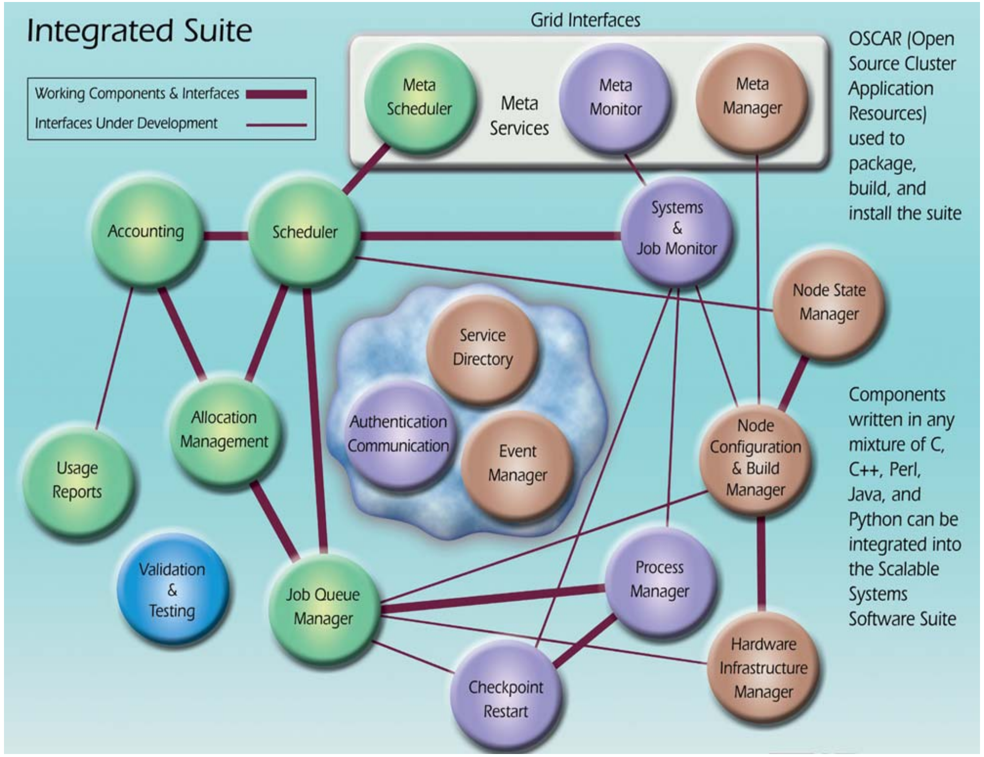
- The first open source HA Beowulf cluster release
- Self-configuration Multi-head Beowulf system
- HA and HPC clustering techniques to enable critical HPC infrastructure
- Active/Hot Standby
- Self-healing with 3-5 sec automatic failover time

LOUISIANA TECH UNIVERSITY

SSS-OSCAR: Scalable System Software

- Leverage OSCAR framework to package and distribute the Scalable System Software (SSS) suite, *SSS-OSCAR*.
- *SSS-OSCAR* – A release of OSCAR containing all SSS software in single downloadable bundle.

- **SSS project developing standard interface for scalable tools**
  - Improve interoperability
  - Improve long-term usability & manageability
  - Reduce costs for supercomputing centers
- **Map out functional areas**
  - Schedulers, Job Mangers
  - System Monitors
  - Accounting & User management
  - Checkpoint/Restart
  - Build & Configuration systems
- **Standardize the system interfaces**
  - Open forum of universities, labs, industry
  - Define component interfaces in XML
  - Develop communication infrastructure



C3 Power Tools

- Command-line interface for cluster system administration and parallel user tools.
- Parallel execution `cexec`
  - Execute across a single cluster or multiple clusters at same time
- Scatter/gather operations `cpush/cget`
  - Distribute or fetch files for all node(s)/cluster(s)
- Used throughout OSCAR and as underlying mechanism for tools like OPIUM's *useradd* enhancements.

C3 Building Blocks

- System administration
  - cpushimage - “push” image across cluster
  - cshutdown - Remote shutdown to reboot or halt cluster
- User & system tools
  - cpush - push single file -to- directory
  - crm - delete single file -to- directory
  - cget - retrieve files from each node
  - ckill - kill a process on each node
  - cexec - execute arbitrary command on each node
    - cexecs – serial mode, useful for debugging
  - clist – list each cluster available and it’s type
  - cname – returns a node name from a given node position
  - cnum – returns a node position from a given node name

Reliability, Availability and Serviceability Deficiencies of Today's Scientific High-End Computing Systems

Stephen L. Scott

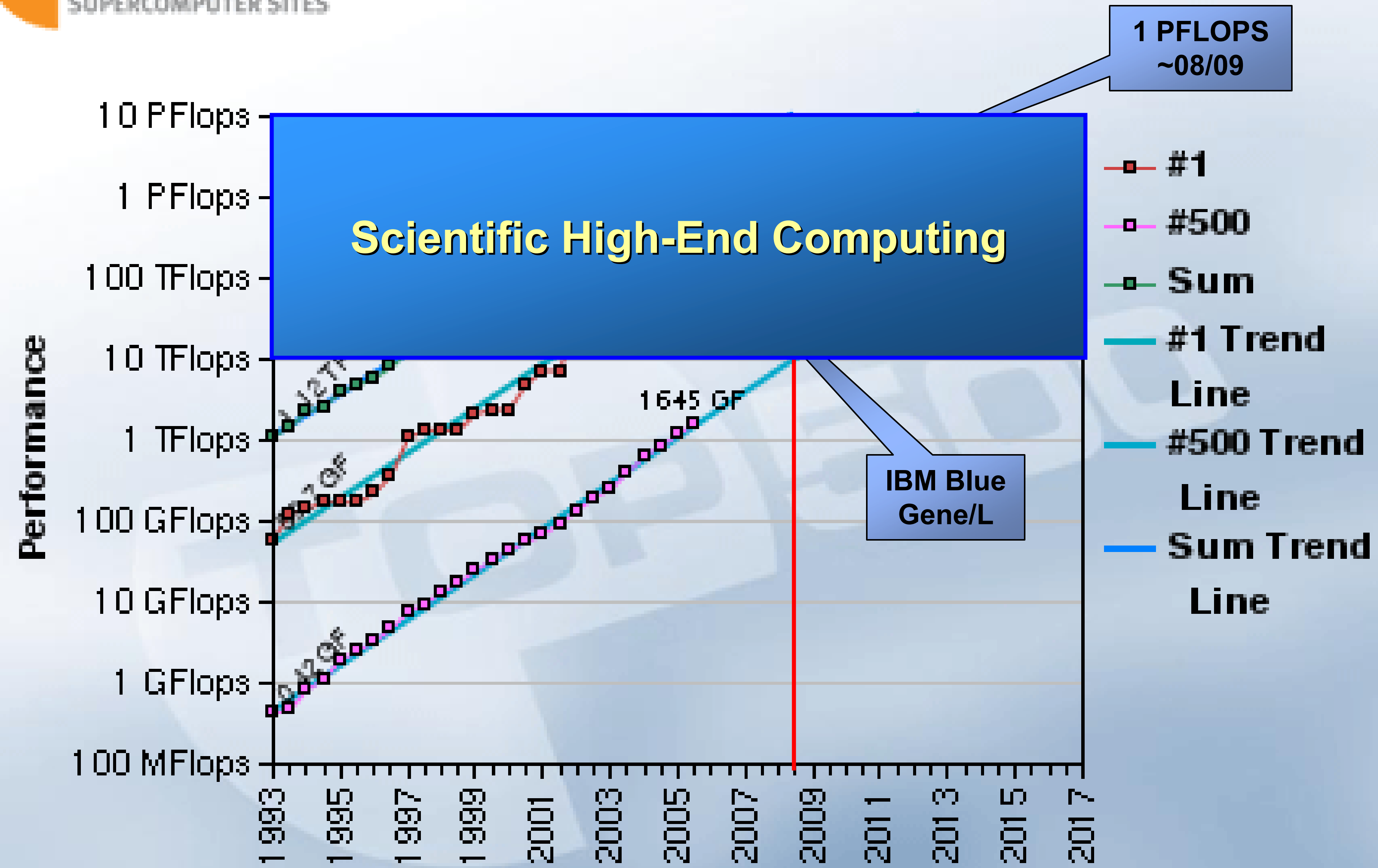
Computer Science and Mathematics Division

Oak Ridge National Laboratory, Oak Ridge, TN, USA

Scientific High-End Computing (HEC)

- Large-scale HPC systems.
  - Tens-to-hundreds of thousands of processors.
  - Current systems: IBM Blue Gene/L and Cray XT3
  - Next-generation systems: IBM Blue Gene/P and Cray XT4
- Computationally and data intensive applications.
  - 10 TFLOP – 1PFLOP with 10 TB – 1 PB of data.
  - Climate change, nuclear astrophysics, fusion energy, materials sciences, biology, nanotechnology, …
- Capability vs. capacity computing
  - Single jobs occupy large-scale high-performance computing systems for weeks and months at a time.

Projected Performance Development



Availability of Current Systems

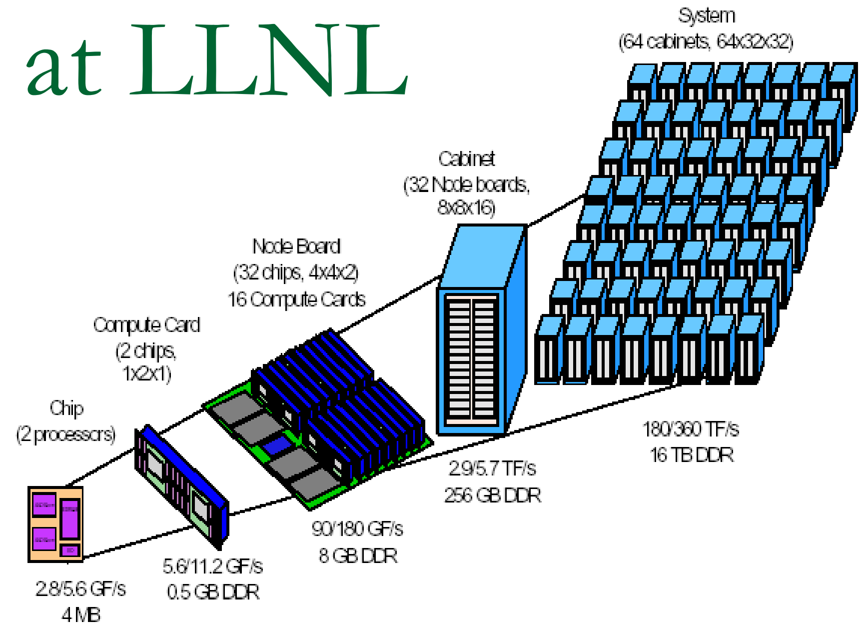
- Today's supercomputers typically need to reboot to recover from a single failure.
- Entire systems go down (regularly and unscheduled) for any maintenance or repair (MTBF = 40-50h).
- Compute nodes sit idle while their head node or one of their service nodes is down.
- Availability will get worse in 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

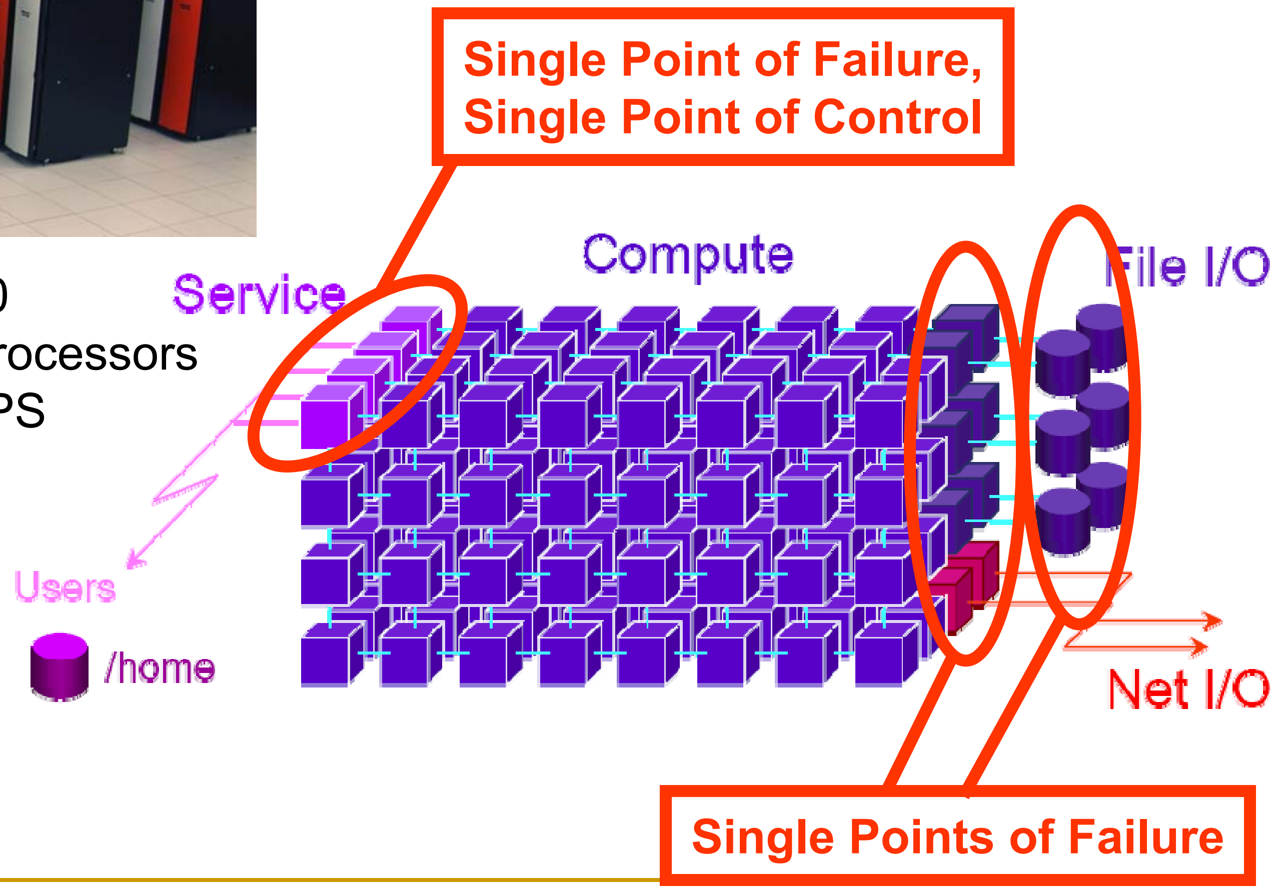
IBM Blue Gene/L at LLNL



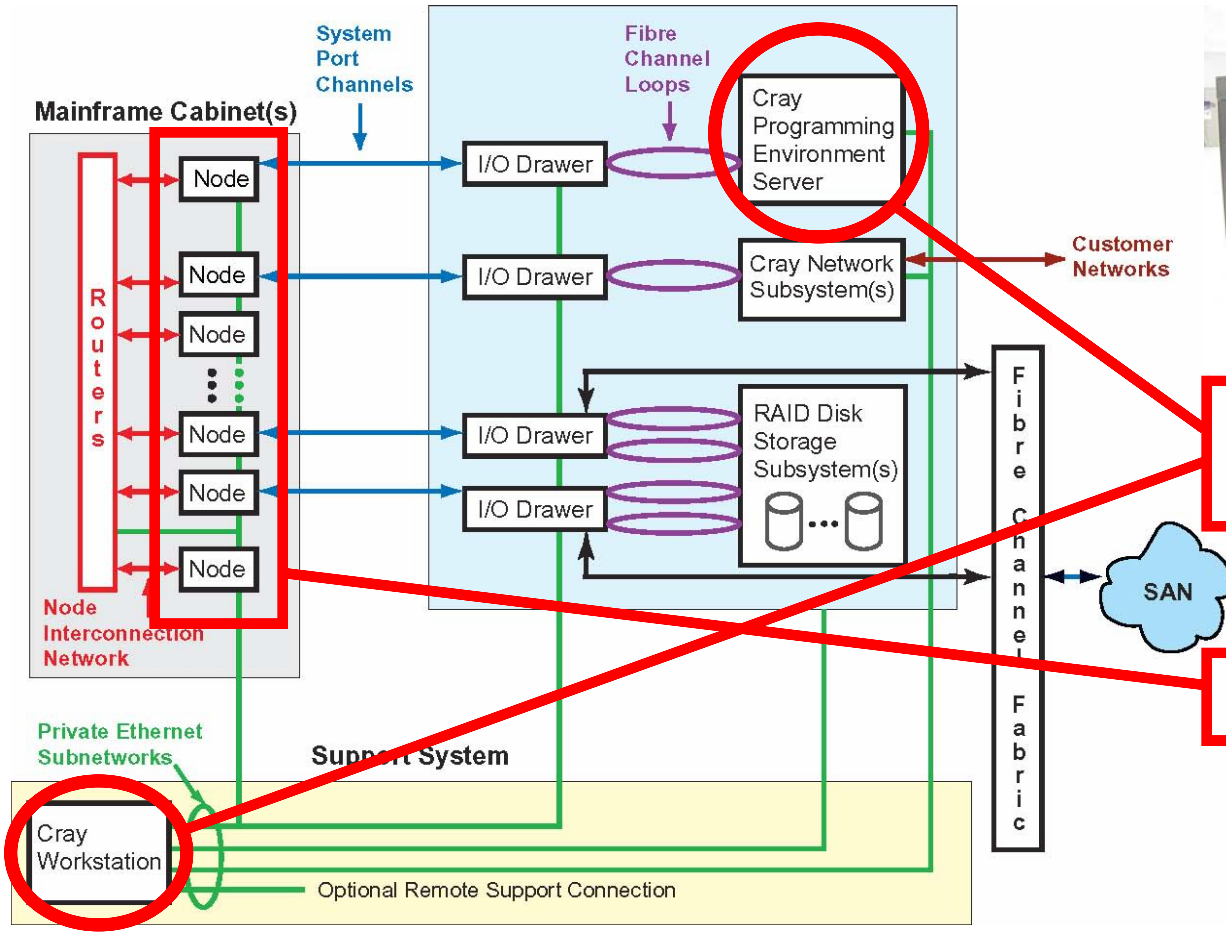
- #1 in Top 500.
- 367 TFLOPS.
- 131072 (700MHz) Power PC processors.
- 32 TB RAM.
- Partition (512 nodes) outage on single failure.
- MTBF = 40-50 hours.
- Weak I/O system prohibits checkpointing.

Clusters: Cray XT3 (Jaguar)

- Number 10 in Top 500
- 5212 AMD Opteron Processors
- 10TB RAM, 25 TFLOPS



Vector Machines: Cray X1 (Phoenix)



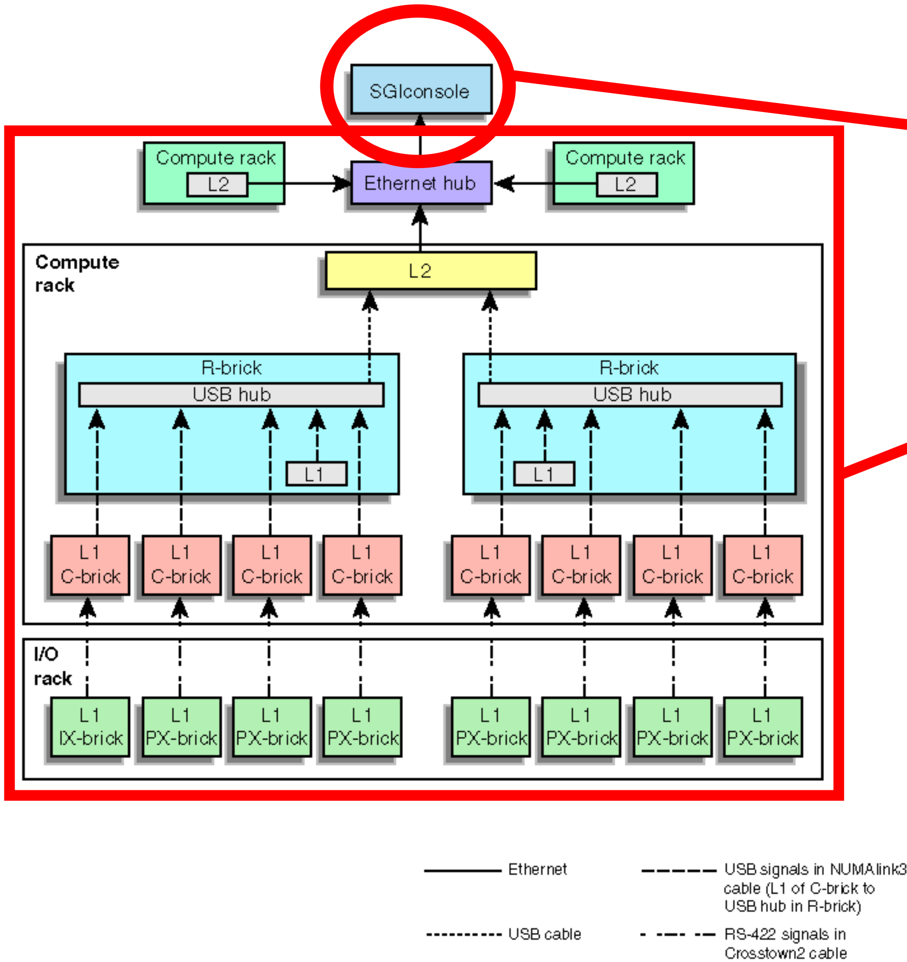
Single Points of Failure, Single Points of Control

Single Points of Failure

- Number 17 in Top 500
- 1024 Vector Processors
- 4TB RAM, 18 TFLOPS

SSI Clusters: SGI Altix (Ram)

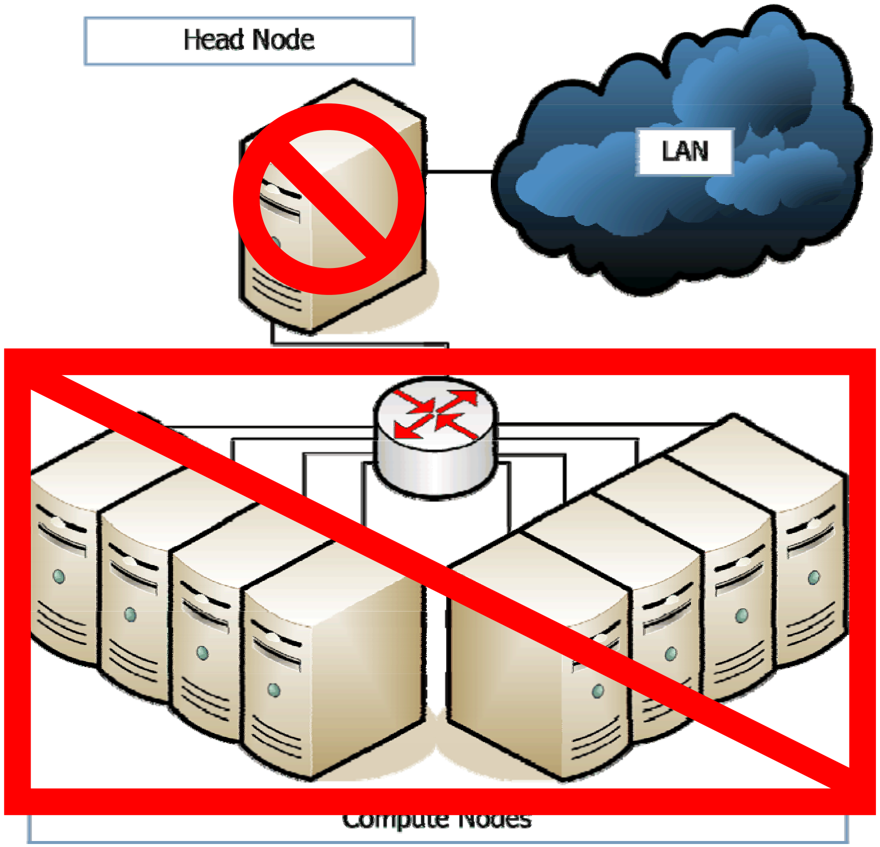
- 256 Itanium 2 Processors
- 2TB RAM, 1.4 TFLOPS



Single Point of Failure, Single Point of Control

Single Points of Failure

Single Head/Service Node Problem



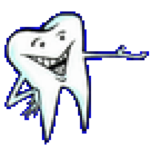
- Single point of failure.
- Compute nodes sit idle while head node is down.
- A = MTTF / (MTTR + MTTF)
- MTTF depends on head node hardware/software quality.
- MTTR depends on the time it takes to repair/replace node.
- MTTR = 0 ➔ A = 1.0 (100%) **continuous availability**.

High Availability though Redundancy

- High availability solutions are based on system component redundancy.
- If a component fails, the system is able to continue to operate using a redundant component.
- The level of availability depends on high availability model and replication strategy.
  - ➢ MTTR of a system can be significantly decreased.
  - ➢ Loss of state can be considerably reduced.
  - ➢ SPoF and SPoC can be completely eliminated.

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

- Addresses the challenges for operating and runtime systems to run large applications efficiently on future ultra-scale high-end computers.
- MOLAR is a collaborative research effort:



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

NC STATE UNIVERSITY

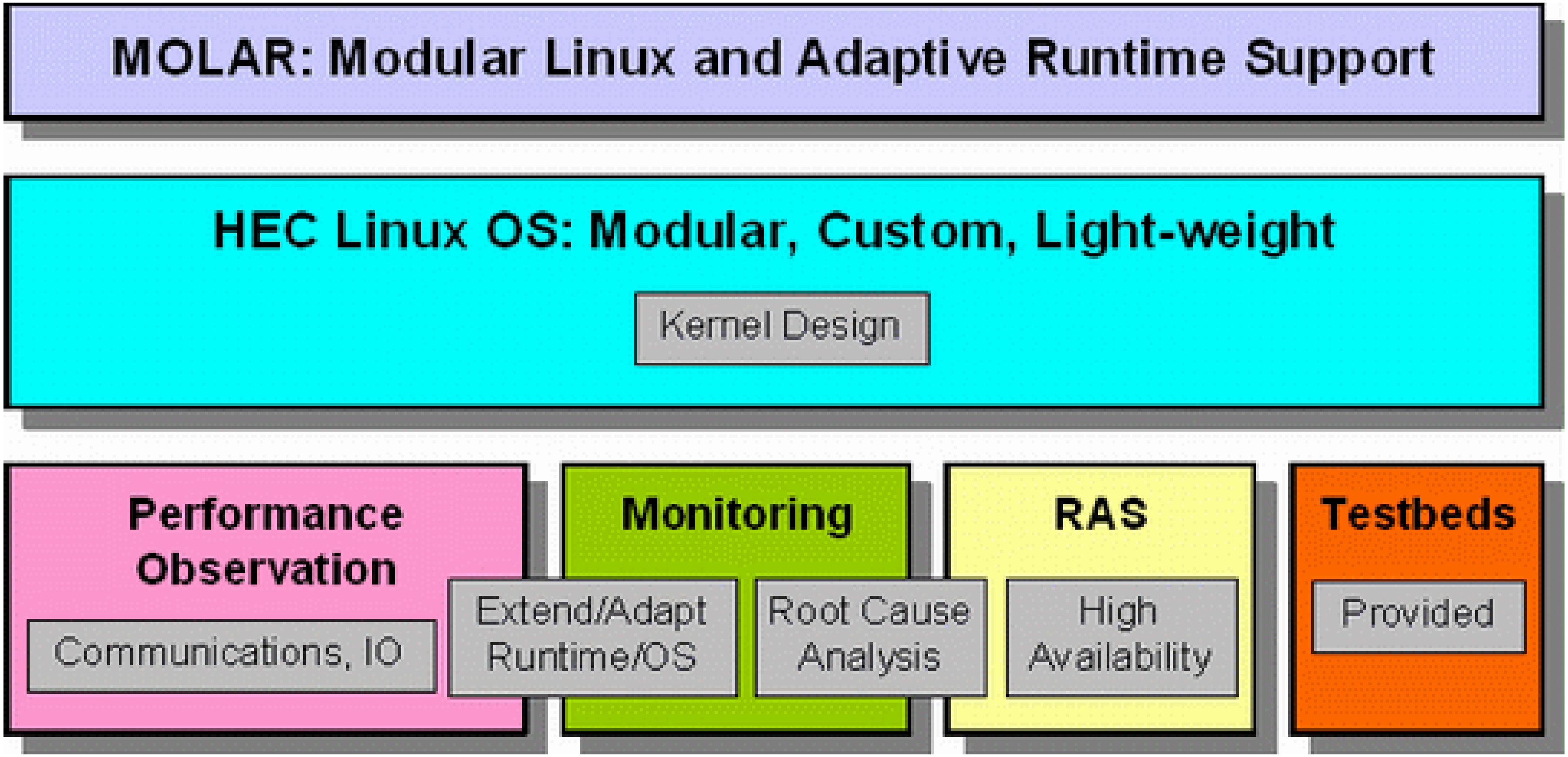
OHIO STATE

LOUISIANA TECH UNIVERSITY

The University of Reading

CRAY THE SUPERCOMPUTER COMPANY

MOLAR: HEC OS/R Research Map



Research and Development Goals

- Provide high-level RAS capabilities for current tera-scale and next-generation petascale HEC systems.
- Eliminate many of the numerous single-points of failure and control in today's HEC systems.
- *Development of techniques to enable HEC systems to run computational jobs 24x7.*
- *Development of proof-of-concept prototypes and production-type RAS solutions.*

High Availability Solutions for Scientific High-End Computing Systems

Christian Engelmann

Computer Science and Mathematics Division

Oak Ridge National Laboratory, Oak Ridge, TN, USA

High Availability Models

- Active/Standby:
  - For one active component at least one redundant inactive (standby) component.
  - Fail-over model with idle standby component(s).
  - Level of high-availability depends on replication strategy.
- Active/Active:
  - Multiple redundant active components.
  - No wasted system resources.
  - State change requests can be accepted and may be executed by every member of the component group.

Active/Warm-Standby

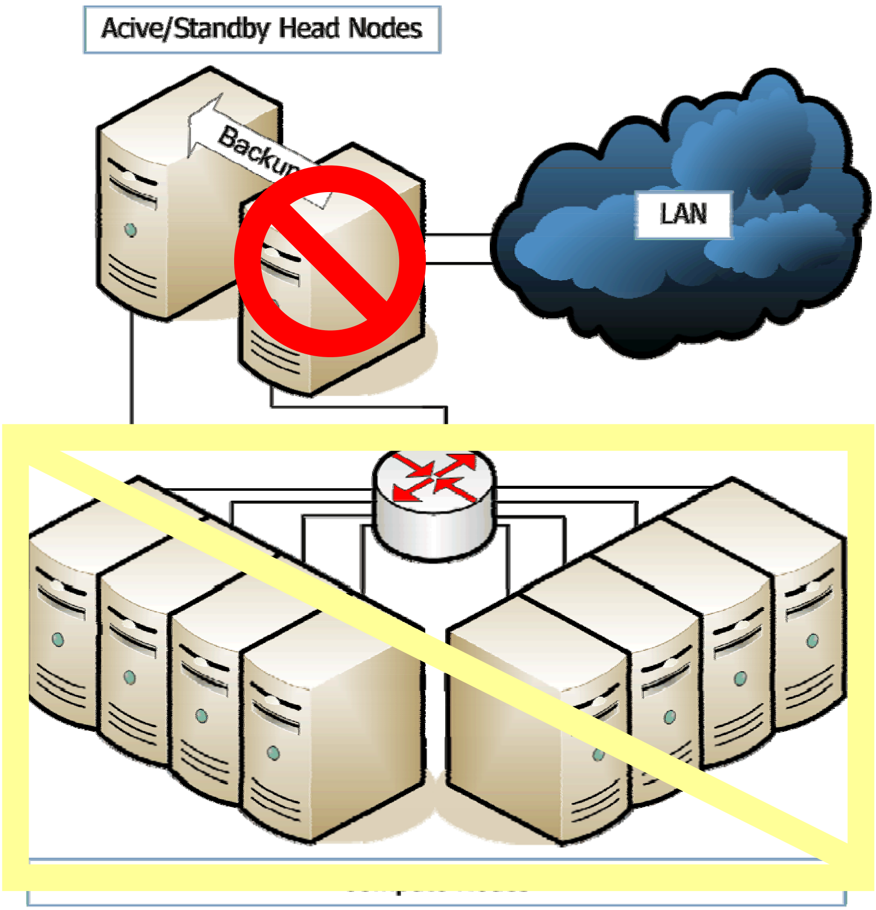
- Hardware and software redundancy.
- State is regularly replicated to the standby.
- Standby component automatically replaces the failed component and continues to operate based on the previously replicated state.
- Only those component state changes are lost that occurred between the last replication and the failure.
- Component state is copied using *passive replication*, i.e. in intervals or after a state change took place.

Active/Hot-Standby

- Hardware and software redundancy.
- State is replicated to the standby <u>during</u> change.
- Standby component automatically replaces the failed component and continues to operate based on the <u>current</u> state.
- Component state is copied using *active replication*, i.e. by commit protocols that ensure consistency.

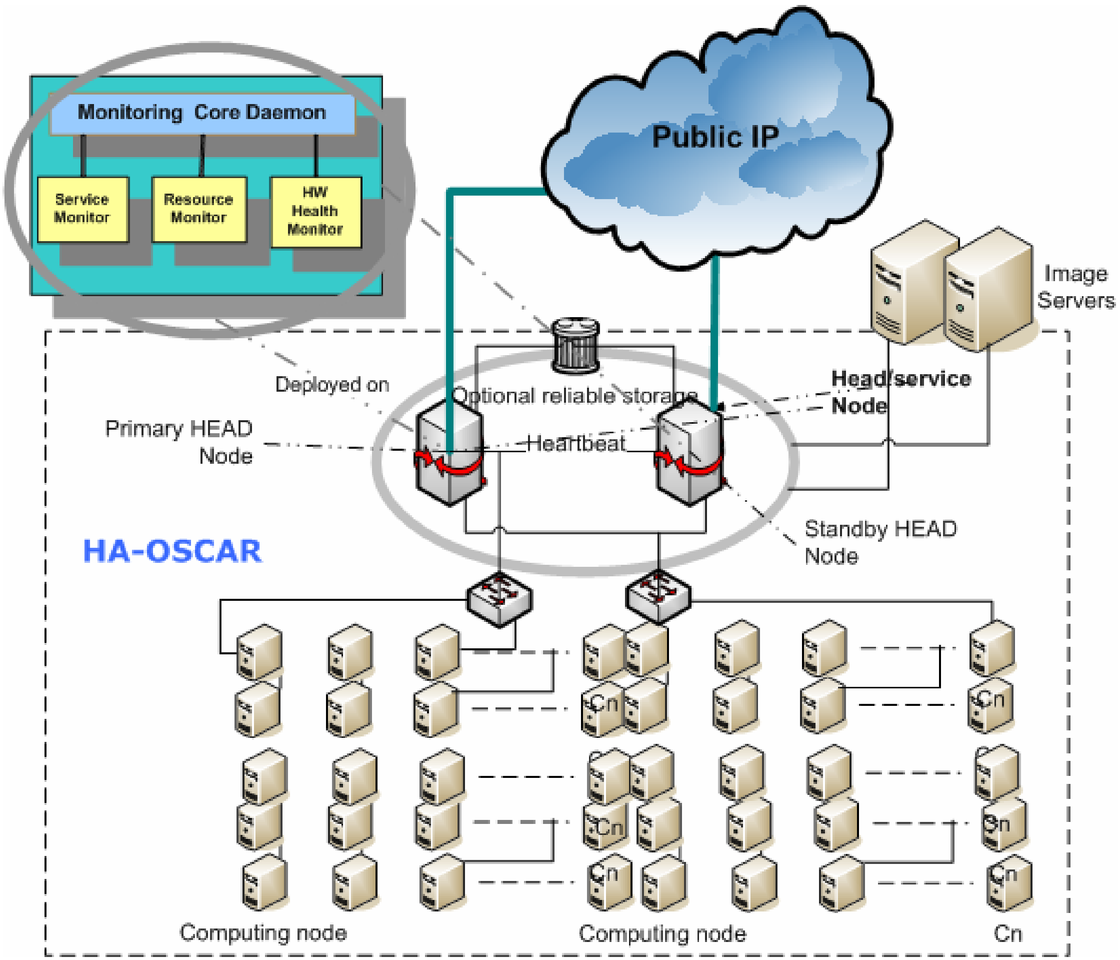
➢ Continuous availability without any interruption.

Active/Standby Head/Service Nodes



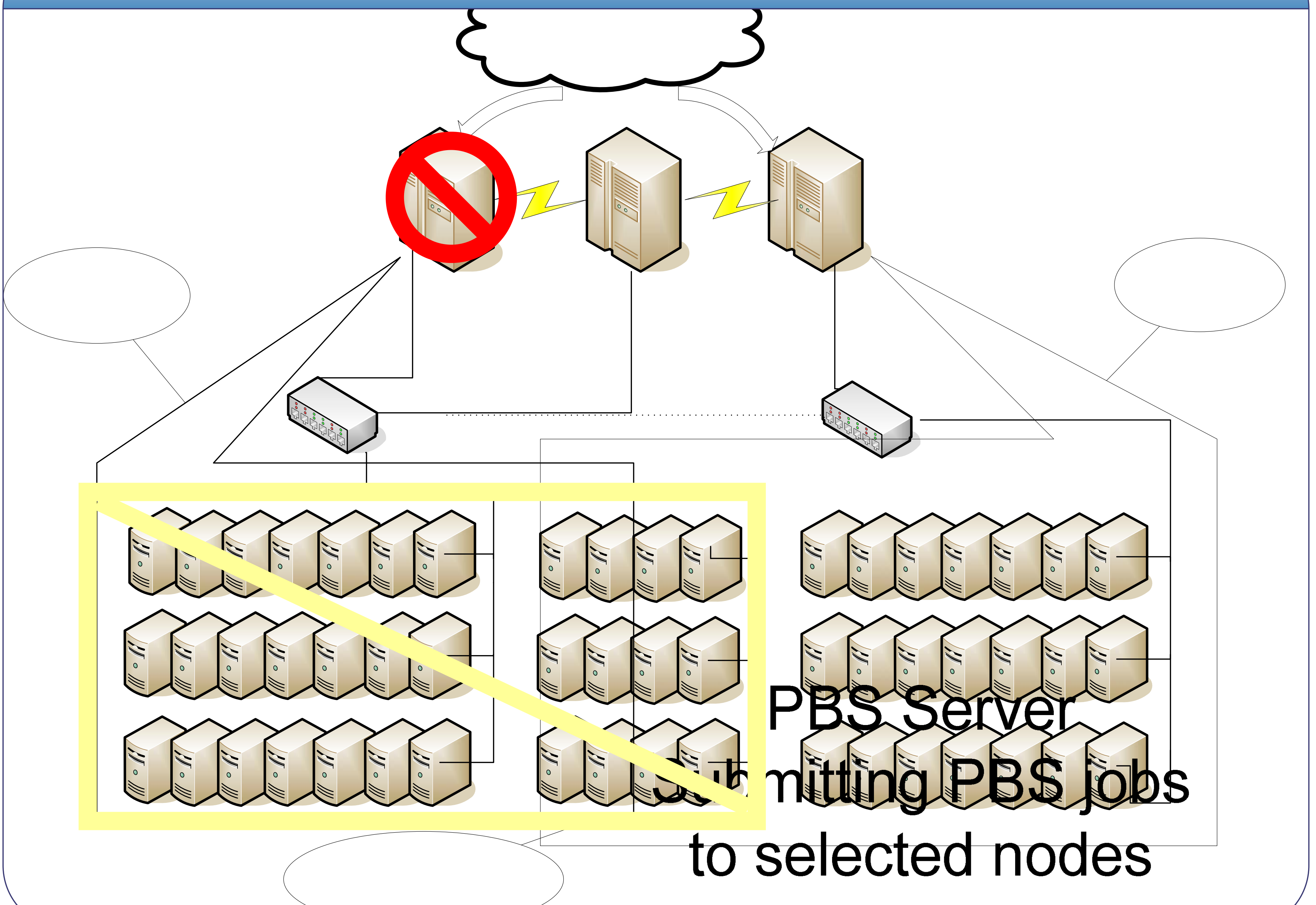
- Single active head node.
- Backup to shared storage.
- Simple checkpoint/restart.
- Fail-over to standby node.
- Idle standby head node.
- Rollback to backup.
- Service interruption for the time of the fail-over.
- Service interruption for the time of restore-over.

Active/Standby PBS with HA-OSCAR



Asymmetric Active/Active

- Hardware and software redundancy.
- However, no component state replication.
- Multiple uncoordinated redundant active system components that do not share state.
- In case of a failure, all other active system components continue to operate.
- Stateful components loose all of their state.
- Additional hot-standby components may offer continuous availability.

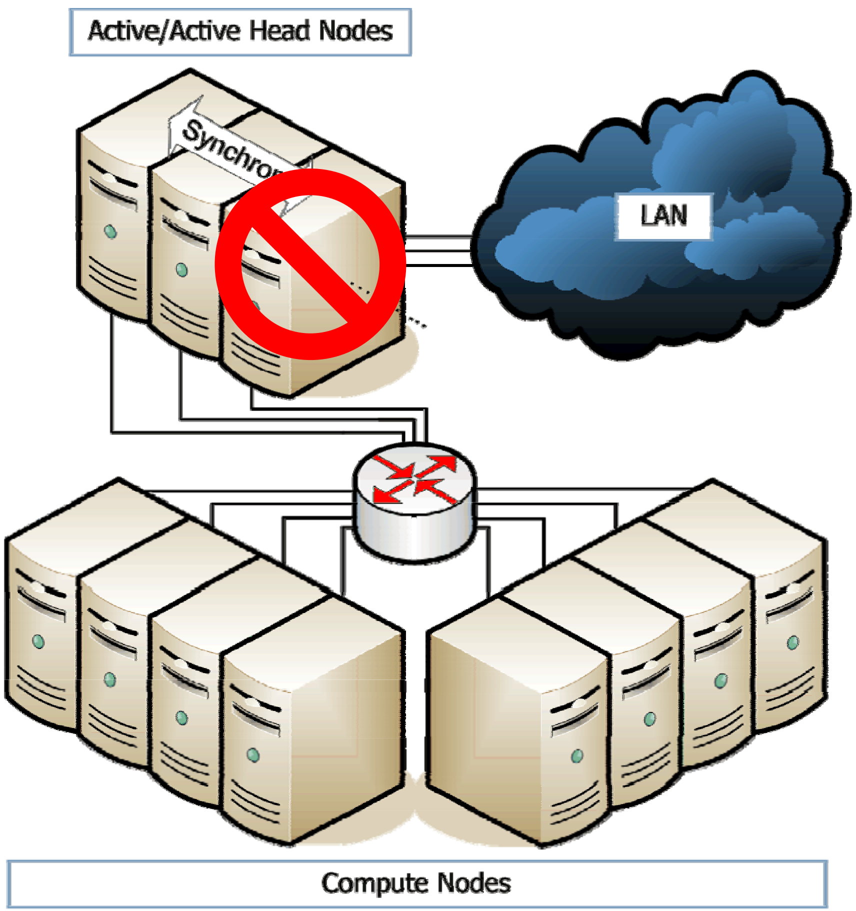


On fly selection of

Symmetric Active/Active

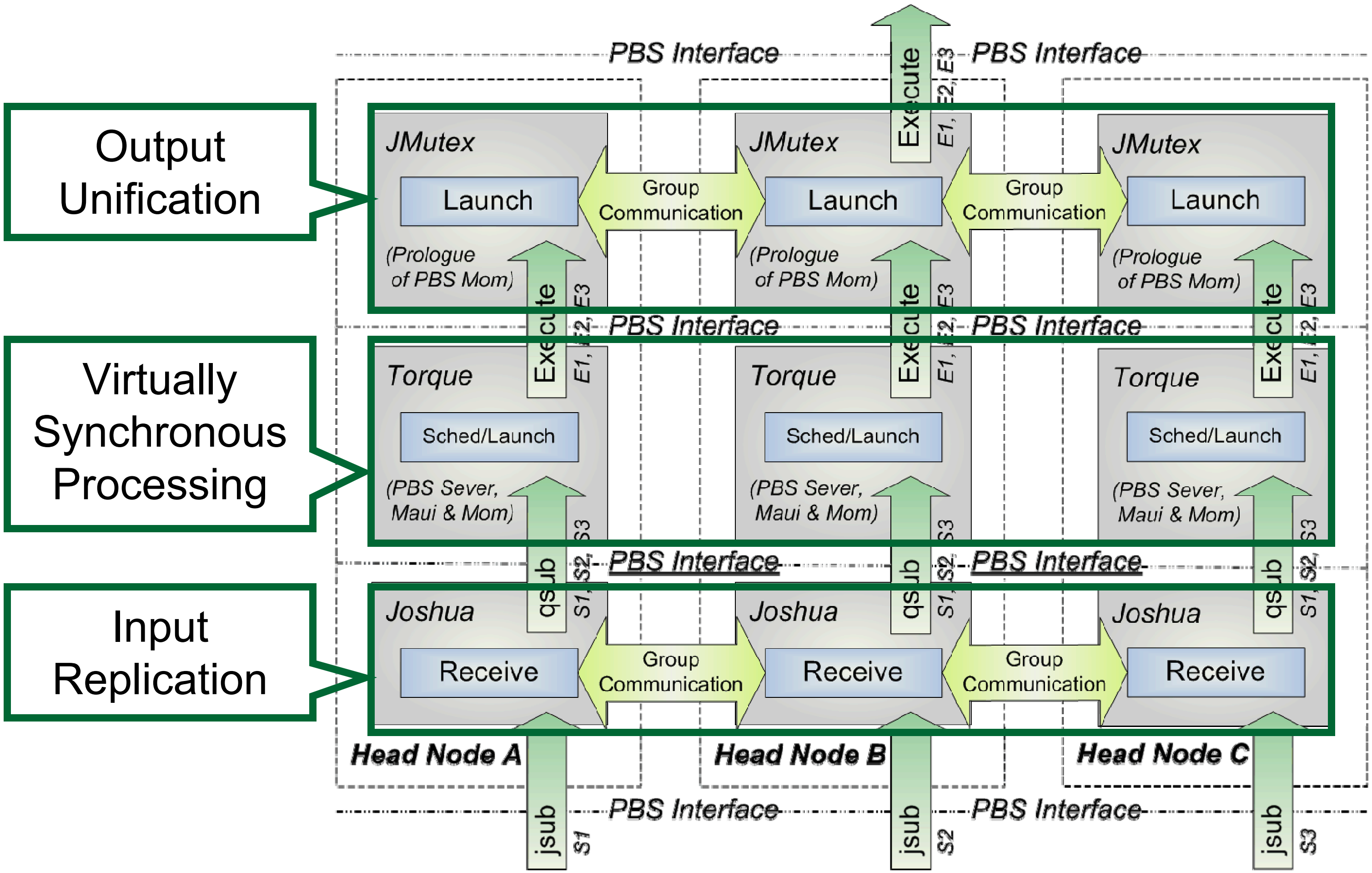
- Hardware and software redundancy.
- Component state is *actively replicated* within an active component group using advanced commit protocols (*distributed control*, *virtual synchrony)*.
- All other active system components continue to operate using the current state.
- Component state is shared in form of *global state.*
- Continuous availability without any interruption and without wasting resources.

S-Active/Active Head/Service Nodes

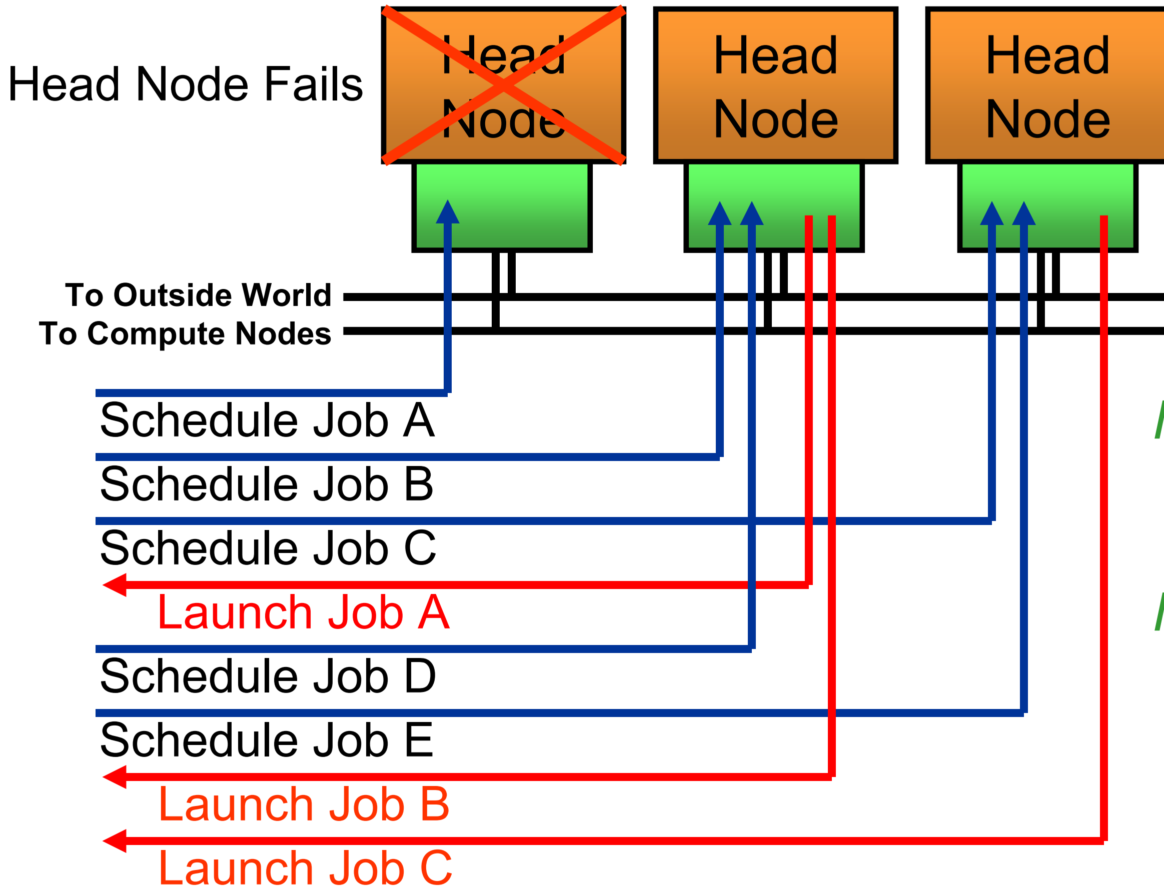


- Many active head nodes.
- Work load distribution.
- Symmetric replication between head nodes.
- Continuous service.
- Always up-to-date.
- No fail-over necessary.
- No restore-over necessary.
- Virtual synchrony model.
- Complex algorithms.

S-Active/Active Torque with JOSHUA



S-Active/Active Torque with JOSHUA



No Single Point of Failure

No Single Point of Control

Active/Active Redundancy for Nines

$$A_{component} = MTBF / (MTBF + MTTR)$$

$$A = 1 - (1 - A_{component})^n$$

$$T_{down} = 8760 * (1 - A)$$

No. HN	Availability	Downtime
1	98.580441640%	5d 4h 21m

Based on: MTBF 5000-hours, MTTR 72-hours

Active/Active Redundancy for Nines

$$A_{component} = MTBF / (MTBF + MTTR)$$

$$A = 1 - (1 - A_{component})^n$$

$$T_{down} = 8760 * (1 - A)$$

No. HN	Availability	Downtime
1	98.580441640%	5d 4h 21m
2	99.979848540%	1h 45m

Based on: MTBF 5000-hours, MTTR 72-hours

Active/Active Redundancy for Nines

$$A_{component} = MTBF / (MTBF + MTTR)$$

$$A = 1 - (1 - A_{component})^n$$

$$T_{down} = 8760 * (1 - A)$$

No. HN	Availability	Downtime
1	98.580441640%	5d 4h 21m
2	99.979848540%	1h 45m
3	99.999713938%	1m 30s

Based on: MTBF 5000-hours, MTTR 72-hours

Active/Active Redundancy for Nines

$$A_{component} = MTBF / (MTBF + MTTR)$$

$$A = 1 - (1 - A_{component})^n$$

$$T_{down} = 8760 * (1 - A)$$

No. HN	Availability	Downtime
1	98.580441640%	5d 4h 21m
2	99.979848540%	1h 45m
3	99.999713938%	1m 30s
4	99.999995939%	1s

Based on: MTBF 5000-hours, MTTR 72-hours

Active/Active Redundancy for Nines

$$A_{component} = MTBF / (MTBF + MTTR)$$

$$A = 1 - (1 - A_{component})^n$$

$$T_{down} = 8760 * (1 - A)$$

No. HN	Availability	Downtime
1	98.580441640%	5d 4h 21m
2	99.979848540%	1h 45m
3	99.999713938%	1m 30s
4	99.999995939%	1s
5	99.999999942%	18ms

Based on: MTBF 5000-hours, MTTR 72-hours

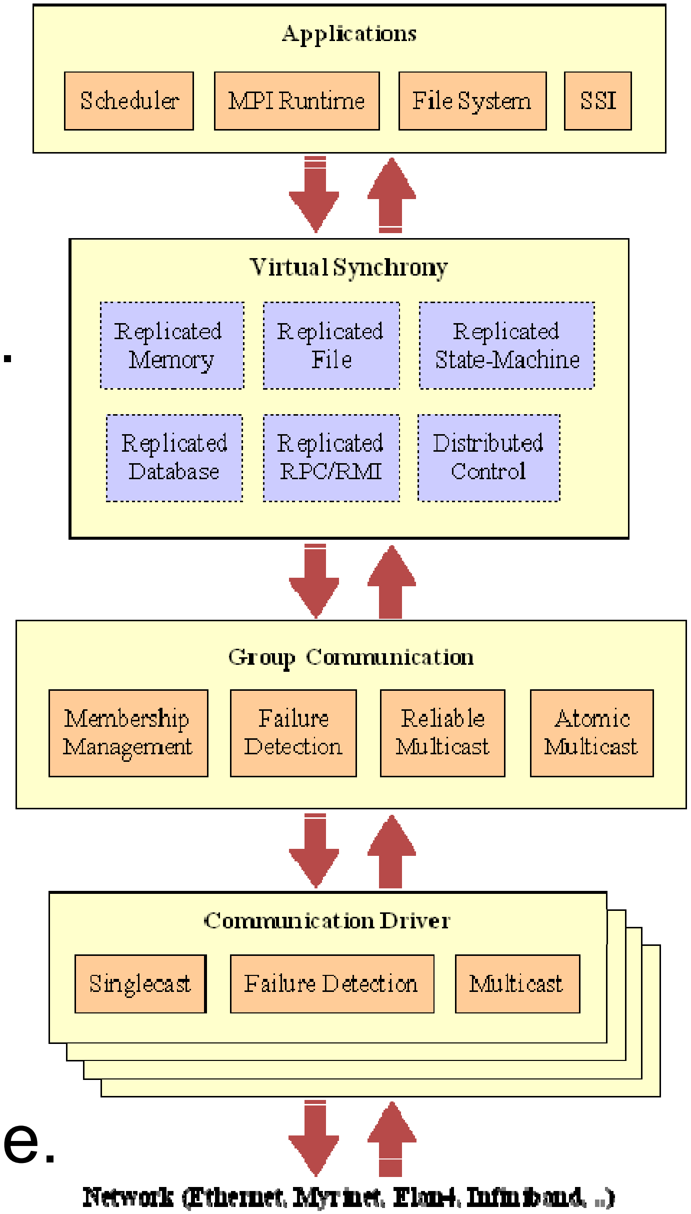
Generic High Availability Framework

- HA-OSCAR:
  - Heartbeat for monitoring and IP-failover.
  - PBS specific scripts for replication to standby.
- JOSHUA:
  - Transis for group communication.
  - TORQUE specific commands for input replication.
  - TORQUE specific scripts for output unification.

> *How can we provide active/stand-by and active/active high availability solutions for services in a **generic**, **modular** and **configurable** fashion?*

PANACEA 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.



PANACEA Prototype

- Unique, flexible, dynamic, C-based component framework: Adaptive Runtime Environment (ARTE).
- Dynamic component loading/unloading on demand.
- XML as interface description language (IDL).
- “Everything” is a component:
  - Communication driver modules.
  - Group communication layer modules.
  - Virtual synchrony layer modules.
- PANACEA = ARTE + RAS components

Further Information

- C3: www.csm.ornl.gov/torc/C3
- OSCAR: www.OpenClusterGroup.org
- HA-OSCAR: xcr.cenit.latech.edu/ha-oscar
- MOLAR: www.fastos.org/molar