Office of Science
U.S. DEPARTMENT OF ENERGY

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

High Availability for Ultra-Scale Scientific High-End Computing

Christian Engelmann [1,2]

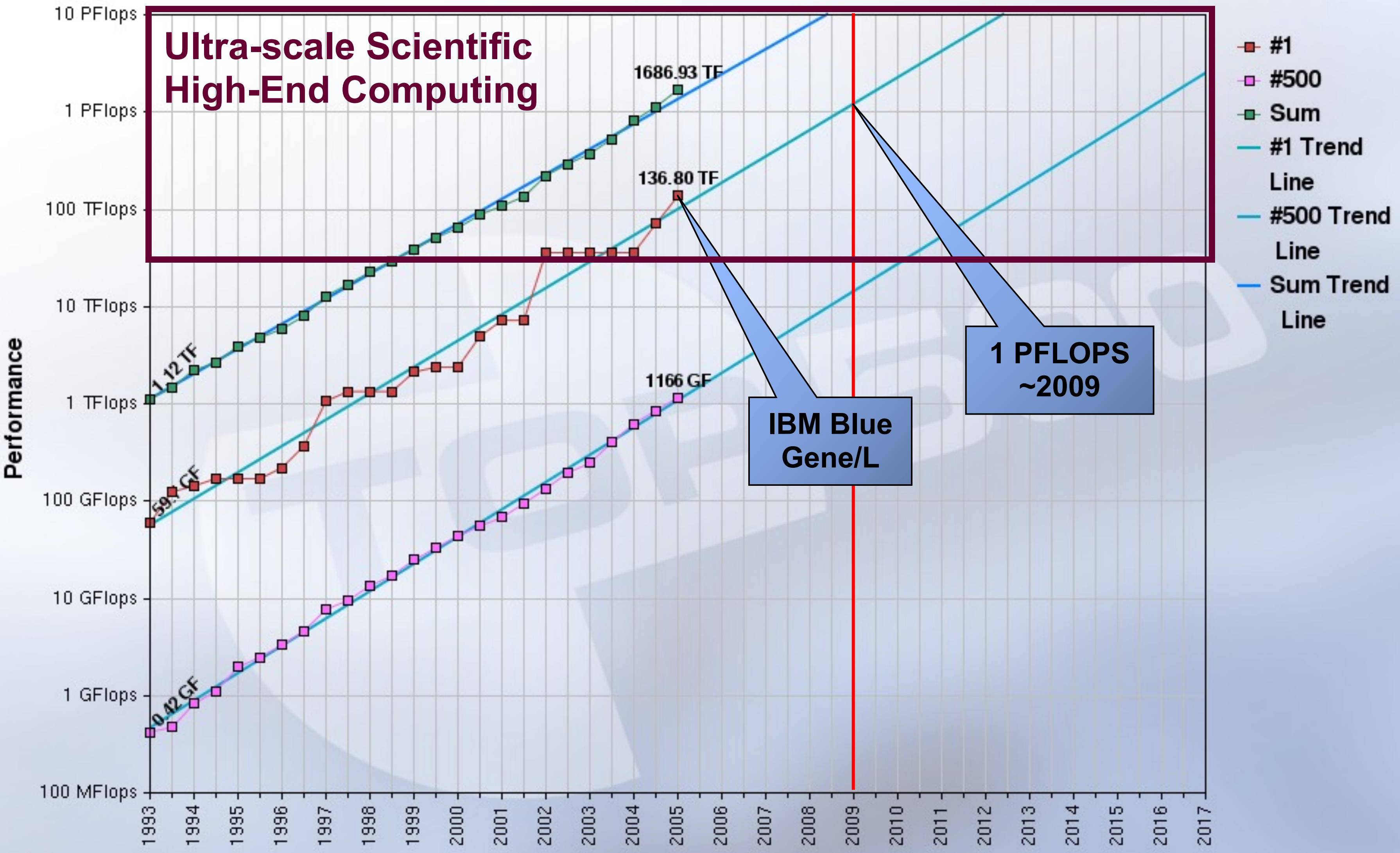
[1] *Research and Development Staff Member*
Computer Science and Mathematics Division
Oak Ridge National Laboratory, Oak Ridge, USA

[2] *Research Assistant*
Department of Computer Science
The University of Reading, Reading, UK

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



Systems at Oak Ridge National Lab

- 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 X1e, 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

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.

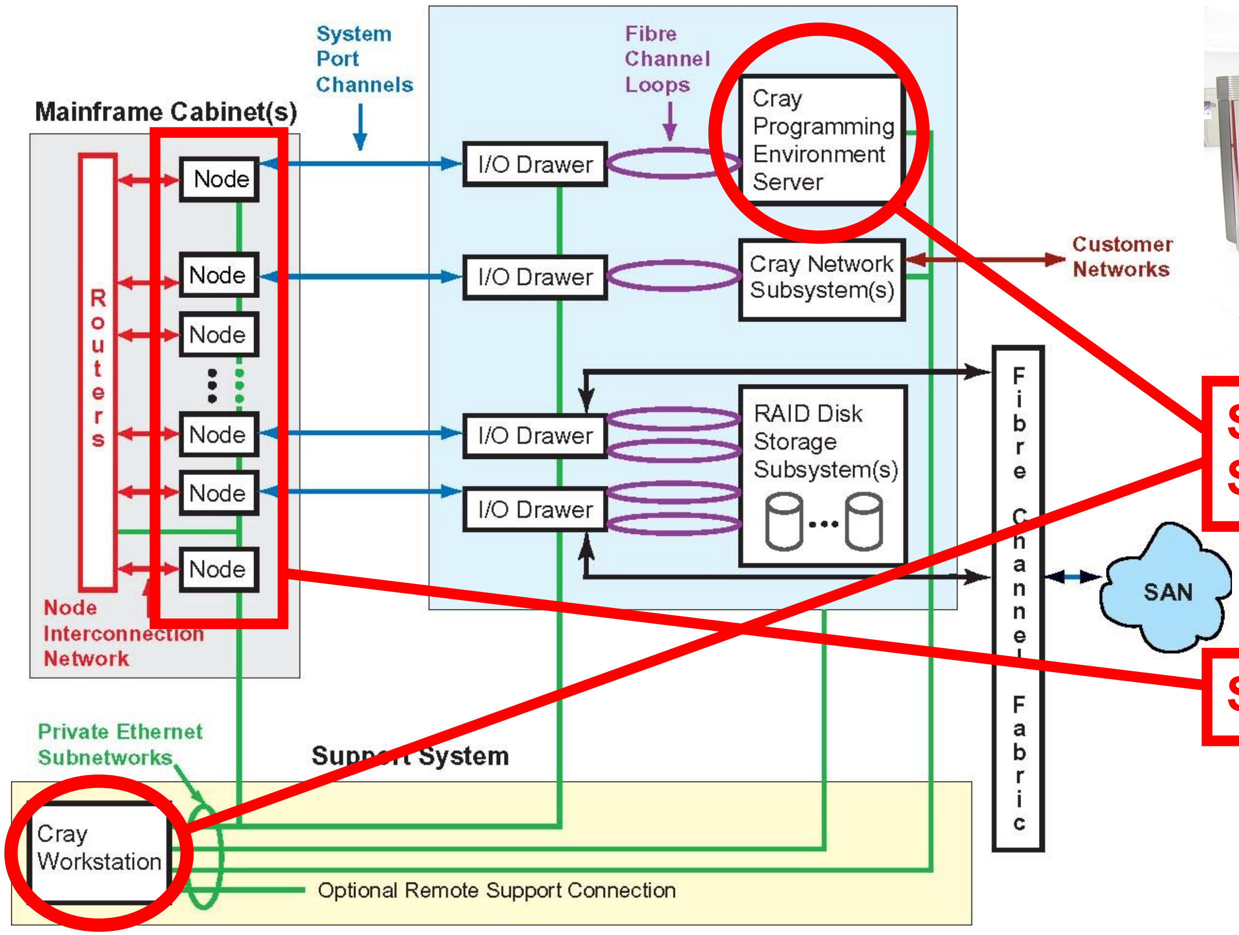
➢ *Why do we accept such significant system outages due to failures, maintenance or repair?*

Availability Measured by the Nines

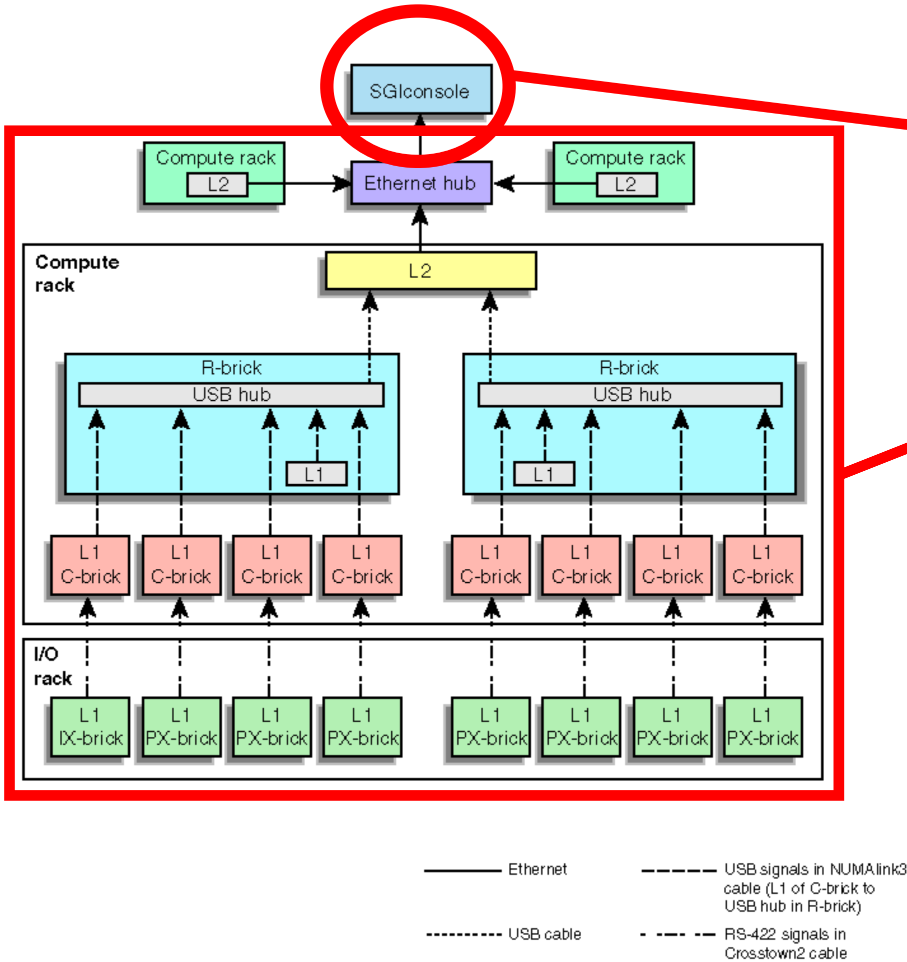
9's	Availability	Downtime/Year	Examples
1	90.0%	36 days, 12 hours	Personal Computers
2	99.0%	87 hours, 36 min	Entry Level Business
3	99.9%	8 hours, 45.6 min	ISPs, Mainstream Business
4	99.99%	52 min, 33.6 sec	Data Centers
5	99.999%	5 min, 15.4 sec	Banking, Medical
6	99.9999%	31.5 seconds	Military Defense

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

Vector Machines: Cray X1e (Phoenix)



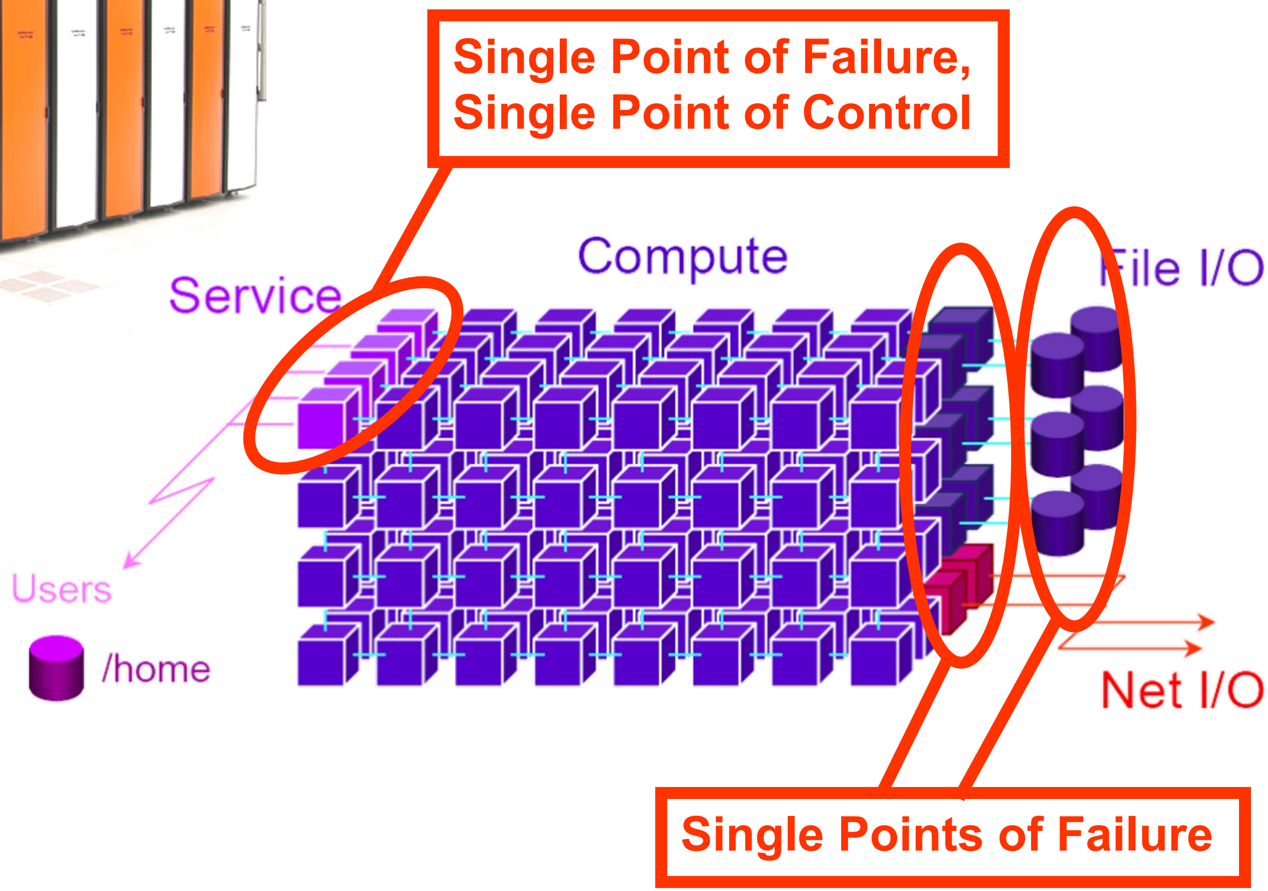
SSI Systems: SGI Altix (Ram)



Single Point of Failure, Single Point of Control

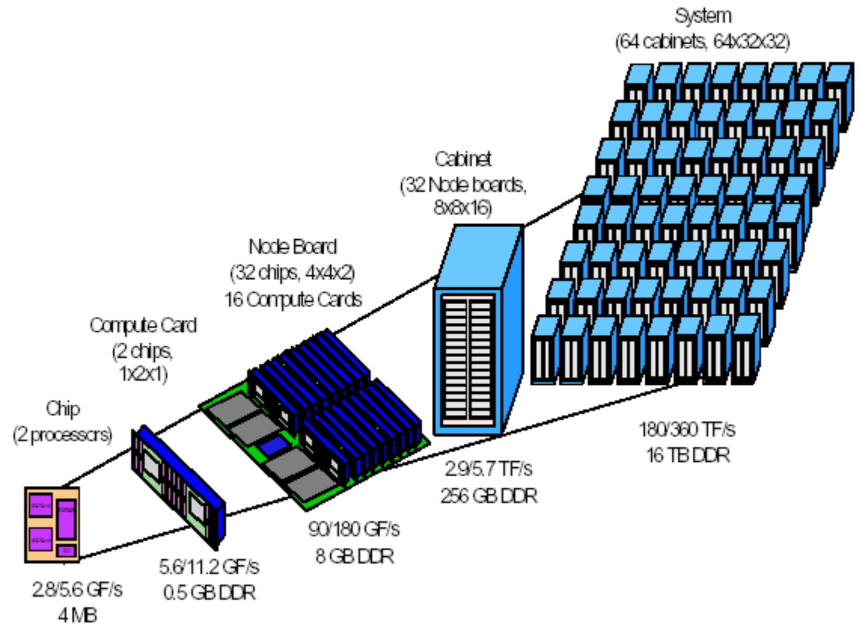
Single Points of Failure

MPPs: Cray XT3 (Jaguar)



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.
- Full capacity in fall 2005.
- Partition (512 nodes) outage on single failure.
- MTBF = 40-50 hours.



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

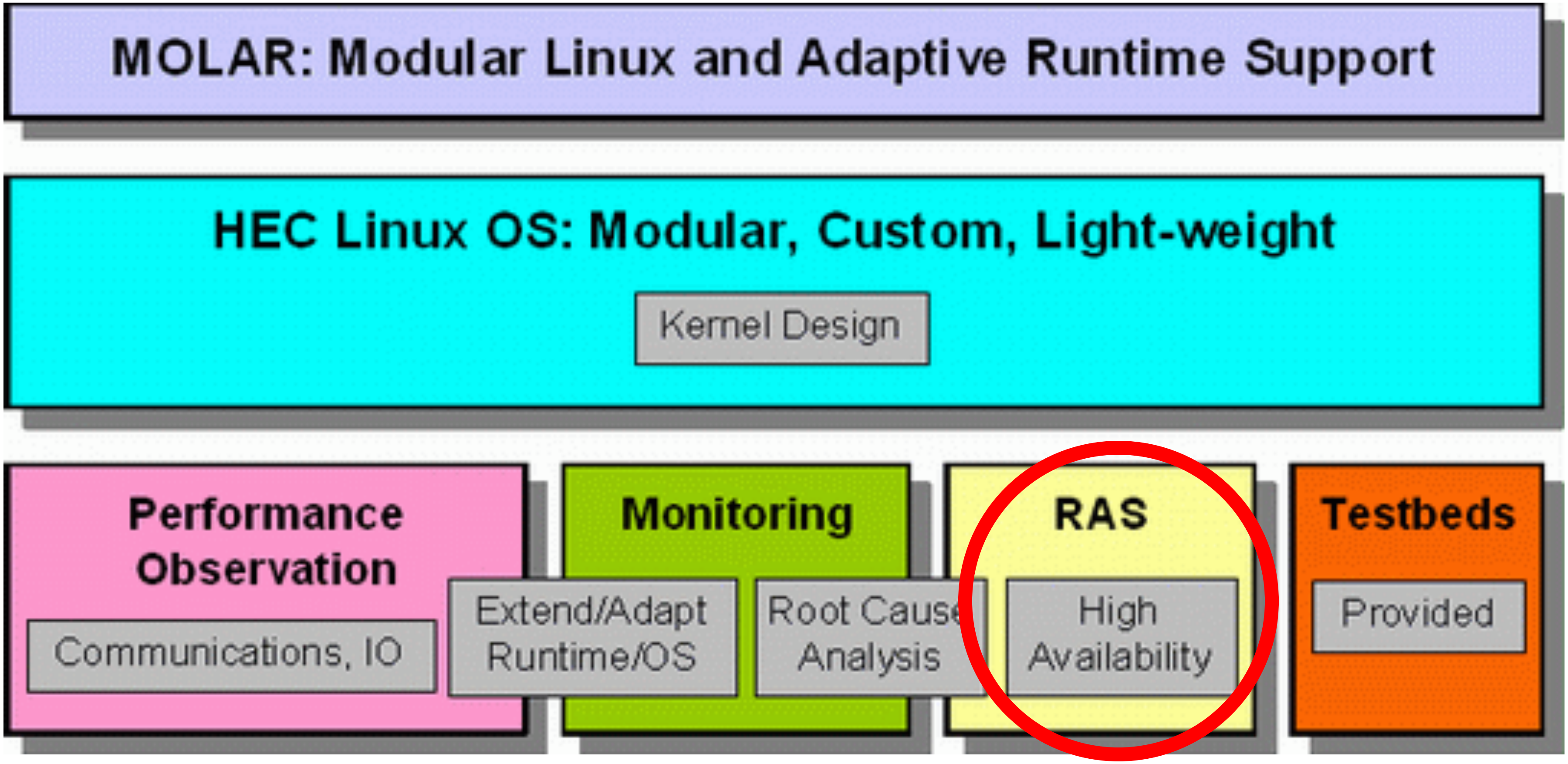
OHIO STATE

LOUISIANA TECH UNIVERSITY

The University of Reading

CRAY THE SUPERCOMPUTER COMPANY

MOLAR: HEC OS/R Research Map



Research 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 HEC systems.
- Improve scalability and access to systems and data.
- *Development of techniques to enable HEC systems to run computational jobs 24x7.*
- *Development of proof-of-concept implementations as blueprint for production-type RAS solutions.*

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.

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/Warm-Standby

- Warm-standby HA for compute nodes involves replication to backup storage.
- Examples:
  - Checkpoint/restart mechanisms (e.g. BLCR).
  - Diskless checkpointing.
  - HA-OSCAR.
  - SLURM.

Active/Hot-Standby

- Hardware and software redundancy.
- State is replicated to the standby during change.
- Standby component automatically replaces the failed component and continues to operate based on the current state.
- Component state is copied using *active replication*, i.e. by commit protocols that ensure consistency.
- ➢ Continuous availability without any interruption.

Active/Hot-Standby

- Hot-standby HA for compute nodes may involve a significant replication overhead.
- Examples:
  - PBSPro for the Cray XT3.
  - MPICH-V message logging facility.

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.

Symmetric Active/Active

- Symmetric active/active HA for compute nodes may involve an even more significant overhead.
- Examples:
  - Group communication systems, e.g. Transis.
  - Distributed virtual machines (DVMs), e.g. Harness.
  - Stock market exchange systems.
  - Military: AEGIS battle radar system.

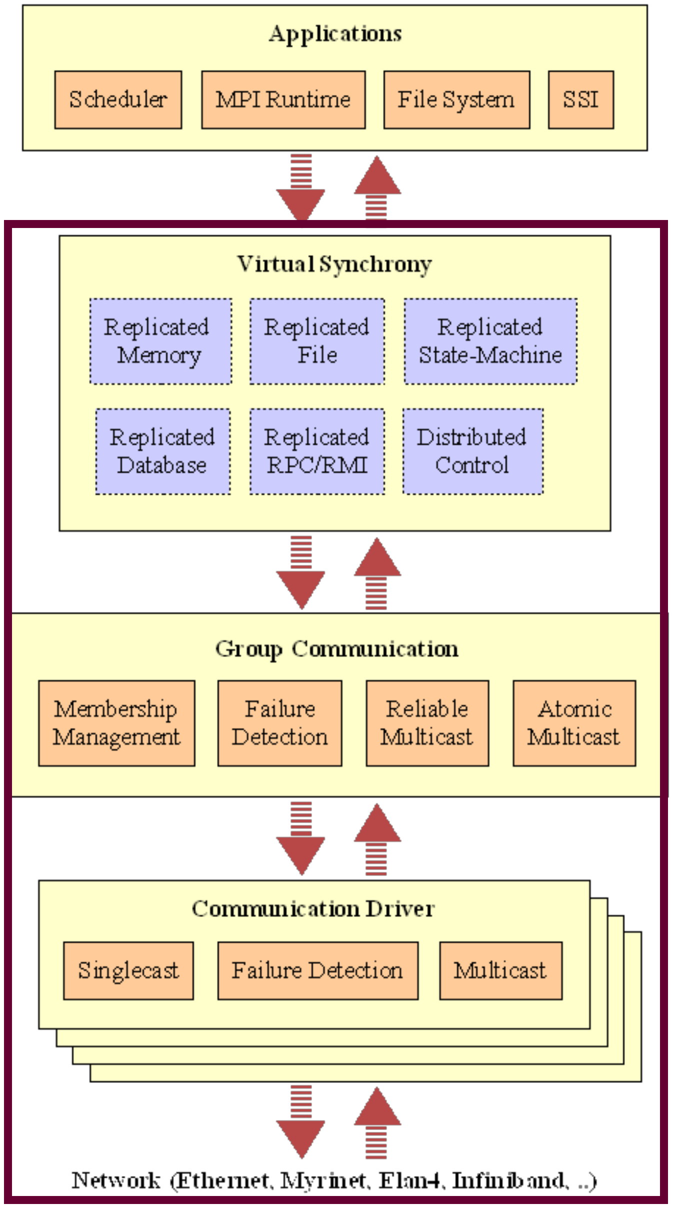
Research in HA for HEC

- We have analyzed current HEC systems and identified their high availability deficiencies.
- We have presented several HA concepts, and explained how they can be applied to HEC systems.
- Main focus of future efforts needs to be on active/hot-standby and active/active solutions that include all system services as well as applications.

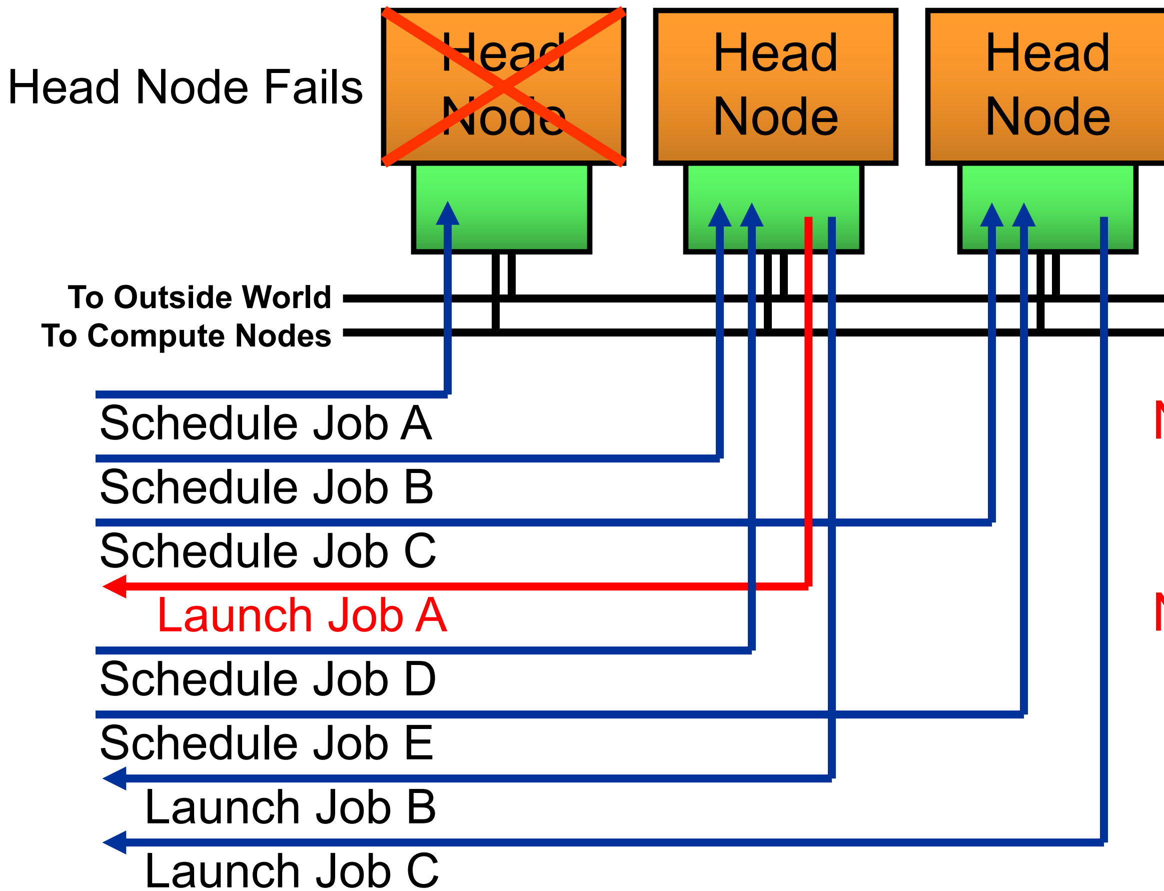
➢ *Flexible, modular software framework is needed to simplify deployment of high availability solutions.*

Framework Prototype

- 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 scale.



Framework Prototype on Active/Active Head Nodes: Scheduler Example



No Single Point of Failure

No Single Point of Control

Plan For The Next Year

- Framework specification - publication.
  - Clearly define individual services and their interfaces.
- Final framework implementation.
  - Implement basic services that others depend on first.
  - Implement one protocol to allow application testing.
- Further exploration of applications and use cases.
- Investigate Open MPI collaboration.
- !!! Come up with a nice project name !!!
- Journal publication after initial framework release.

Further Research Opportunities

- Scalable group communication algorithms.
  - 100,000 processors and beyond.
  - Peer-to-peer diskless checkpointing.
  - SSI: Kerrighed.
- Runtime adaptation to system changes.
  - Weak vs. strong protocols.
  - Mobile wireless devices.
- Automatic framework configuration.
  - Simplify ease of use by “automagic” adaptation.
- Carrier Grade Linux.

More Detailed Information

- C. Engelmann and S. L. Scott. *"High Availability for Ultra-Scale High-End Scientific Computing"*. Proceedings of COSET-2, June 2005.
- C. Engelmann and S. L. Scott. *"Concepts for High Availability in Scientific High-End Computing"*. Proceedings of HAPCW, October 2005.

- http://www.csm.ornl.gov/~engelman/
- http://fastos.org/molar

High Availability for Ultra-Scale Scientific High-End Computing

Christian Engelmann [1,2]

[1] *Research and Development Staff Member*
Computer Science and Mathematics Division
Oak Ridge National Laboratory, Oak Ridge, USA

[2] *Research Assistant*
Department of Computer Science
The University of Reading, Reading, UK