JOSHUA: Symmetric Active/Active Replication for Highly Available HPC Job and Resource Management

Kai Uhlemann[1,2], Christian Engelmann[1,2], and Stephen L. Scott[2]

1 Department of Computer Science
The University of Reading, Reading, UK

2 Computer Science and Mathematics Division,
Oak Ridge National Laboratory, Oak Ridge, USA

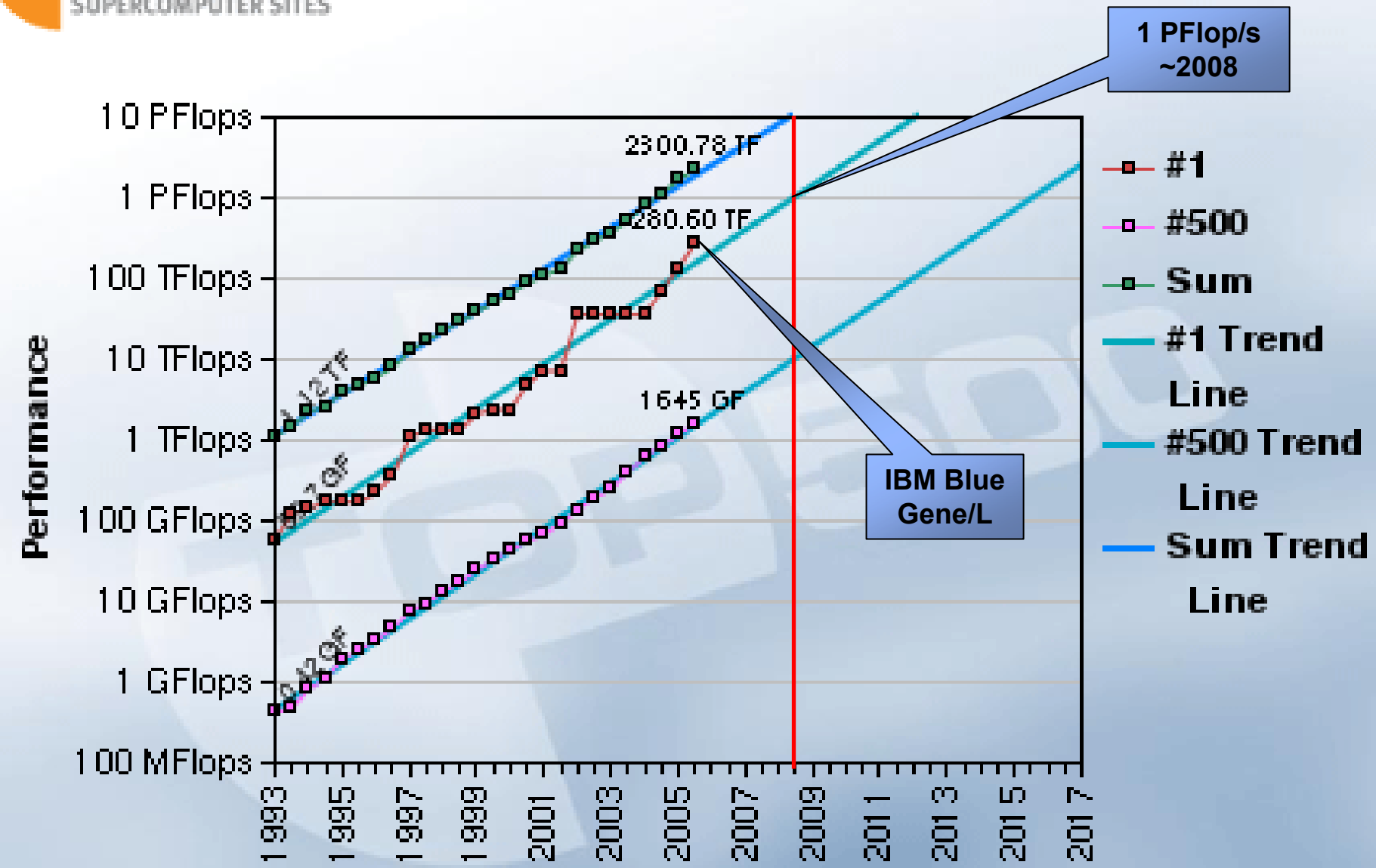
Presentation Overview

- Motivation & background
  - Current HPC systems and their availability
  - Single head/service node problem
- High availability models
  - Active/standby
  - Active/active
- Prototype implementation
  - Symmetric active/active replication
  - Software architecture
  - Introduced overhead and gained availability
  - Remaining issues

Scientific High-End Computing

- Large-scale HPC systems
  - 10,000 to 100,000 to 1,000,000 processor cores
  - Current systems: Cray XT3 and IBM Blue Gene/L
  - Next-generation: petascale Cray XT and IBM Blue Gene
- Computationally and data intensive applications
  - 10 TFlops to 100 TFlops to 1PFlops (sustained)
  - Climate change, nuclear astrophysics, fusion energy, materials sciences, biology, nanotechnology, …
- Capability computing
  - Single computational jobs occupy entire large-scale HPC systems for weeks and months at a time

Projected Performance Development

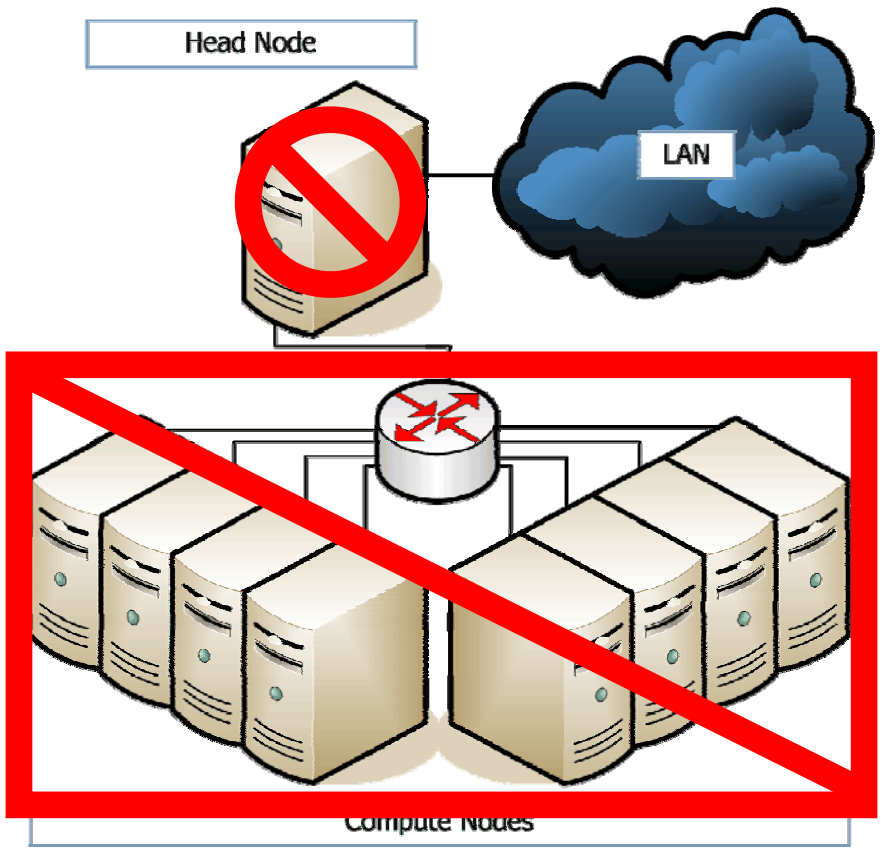


Availability Measured by the Nines

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

Single Head/Service Node Problem



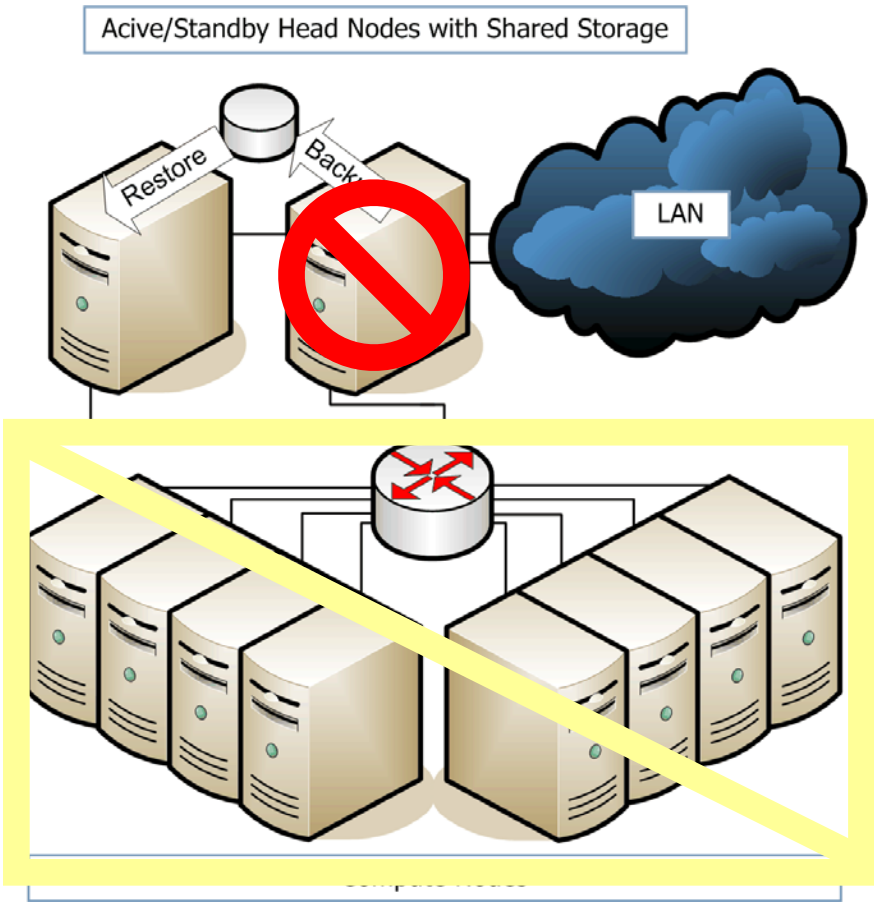
- Single point of failure
- Compute nodes sit idle while head node is down
- A = MTTF / (MTTF + MTTR)
- MTTF depends on head node hardware/software quality
- MTTR depends on the time it takes to repair/replace node

➢ MTTR = 0 ➔ A = 1.00 (100%) **continuous availability**

High Availability Models

- Active/Standby (Warm or Hot):
  - 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 (Asymmetric or Symmetric):
  - 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

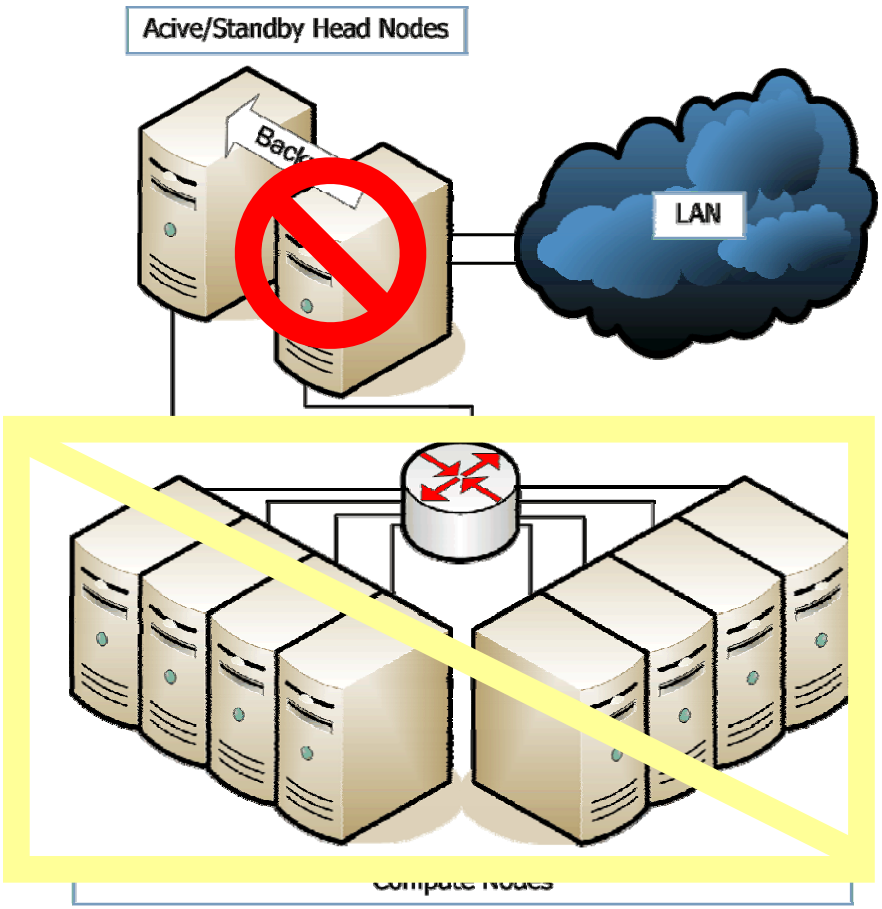
Active/Standby with Shared Storage



- Single active head node
- Backup to shared storage
- Simple checkpoint/restart
- Fail-over to standby node
- Possible corruption of backup state when failing during backup
- Introduction of a new single point of failure
- Correctness and availability are NOT guaranteed

➔ SLURM, meta data servers of PVFS and Luste

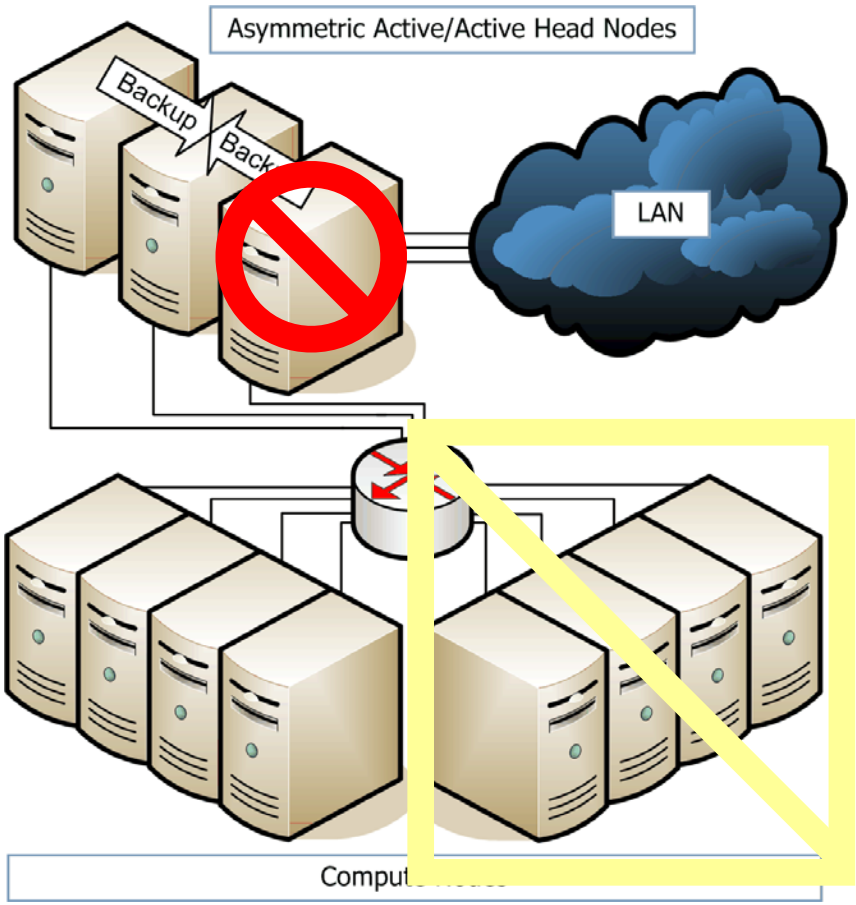
Active/Standby Redundancy



- Single active head node
- Backup to standby node
- Simple checkpoint/restart
- Fail-over to standby node
- Idle standby head node
- Rollback to backup
- Service interruption for fail-over and restore-over

➔ HA-OSCAR, Torque on Cray XT

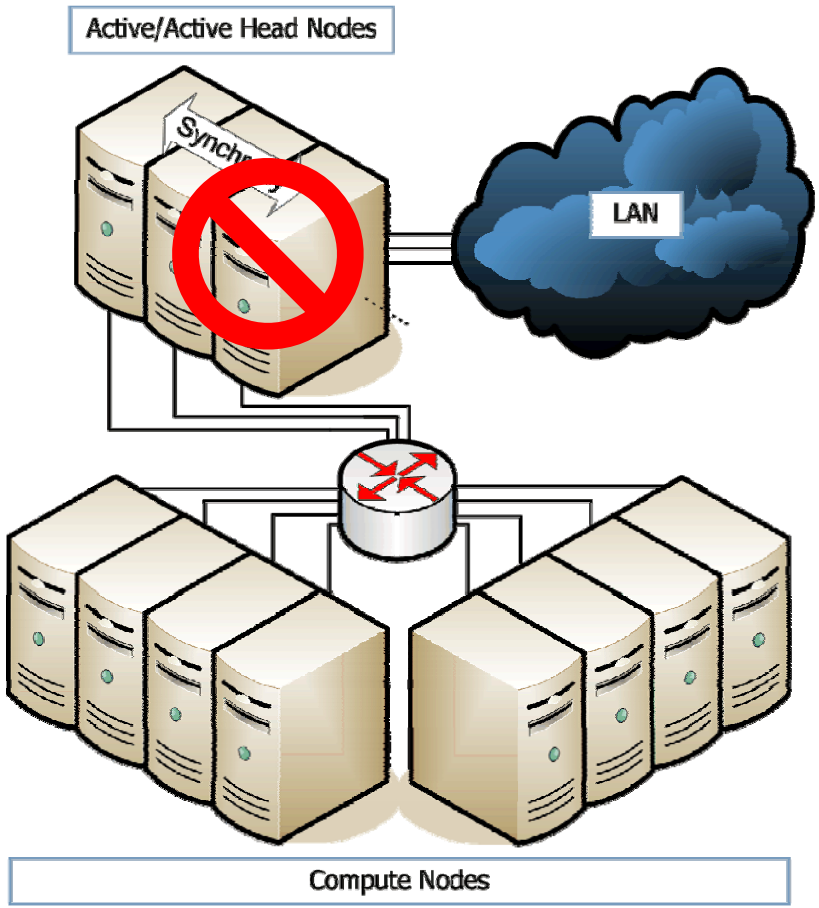
Asymmetric Active/Active Redundancy



- Many active head nodes
- Work load distribution
- Optional fail-over to standby head node(s) (*n+1* or *n+m*)
- No coordination between active head nodes
- Service interruption for fail-over and restore-over
- Loss of state w/o standby
- Limited use cases, such as high-throughput computing

➔ Prototype based on HA-OSCAR

Symmetric Active/Active Redundancy

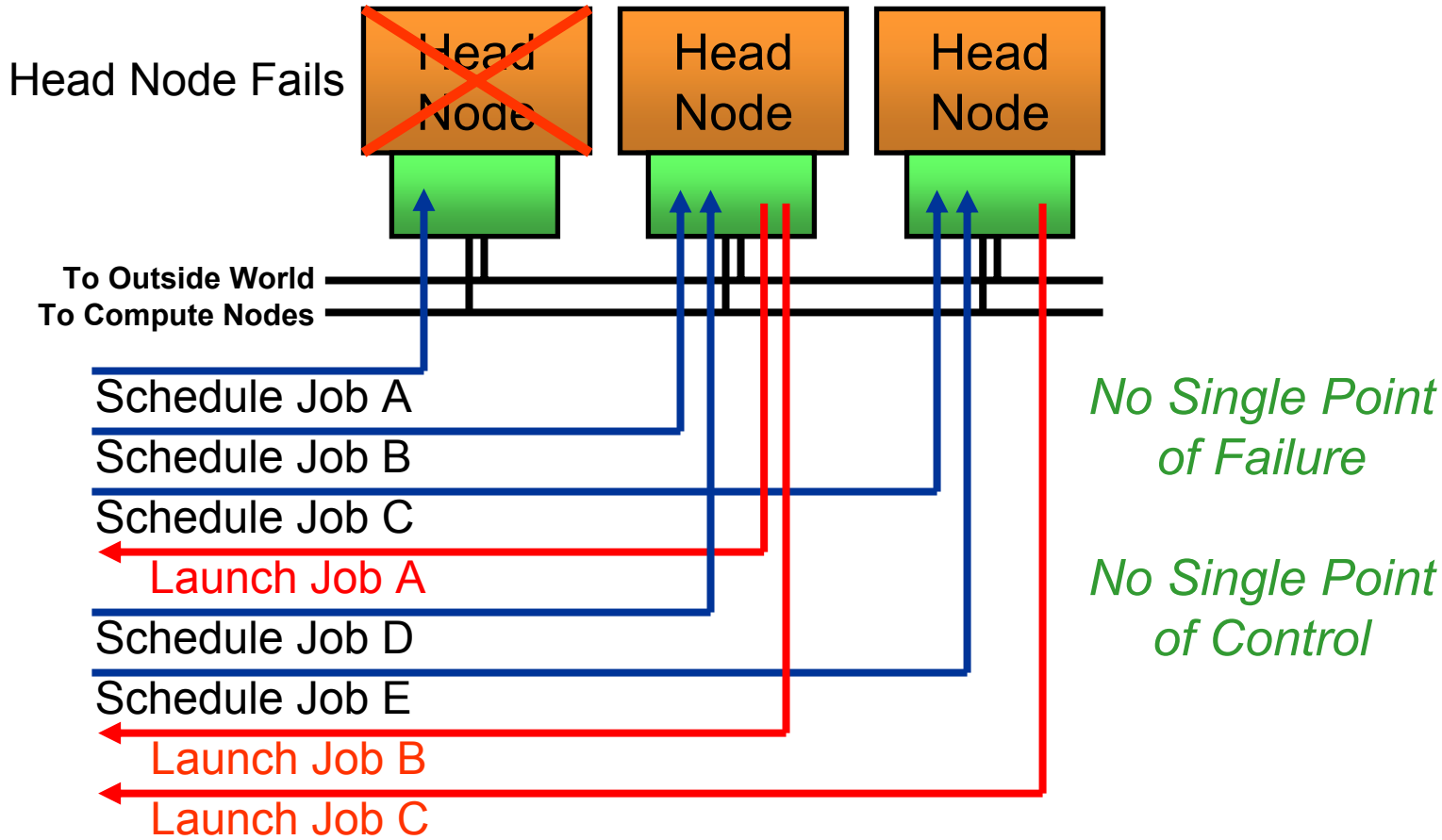


- Many active head nodes
- Work load distribution
- Symmetric replication between head nodes
- Continuous service
- Always up-to-date
- No fail-over necessary
- No restore-over necessary
- Virtual synchrony model
- Complex algorithms
- JOSHUA prototype for Torque

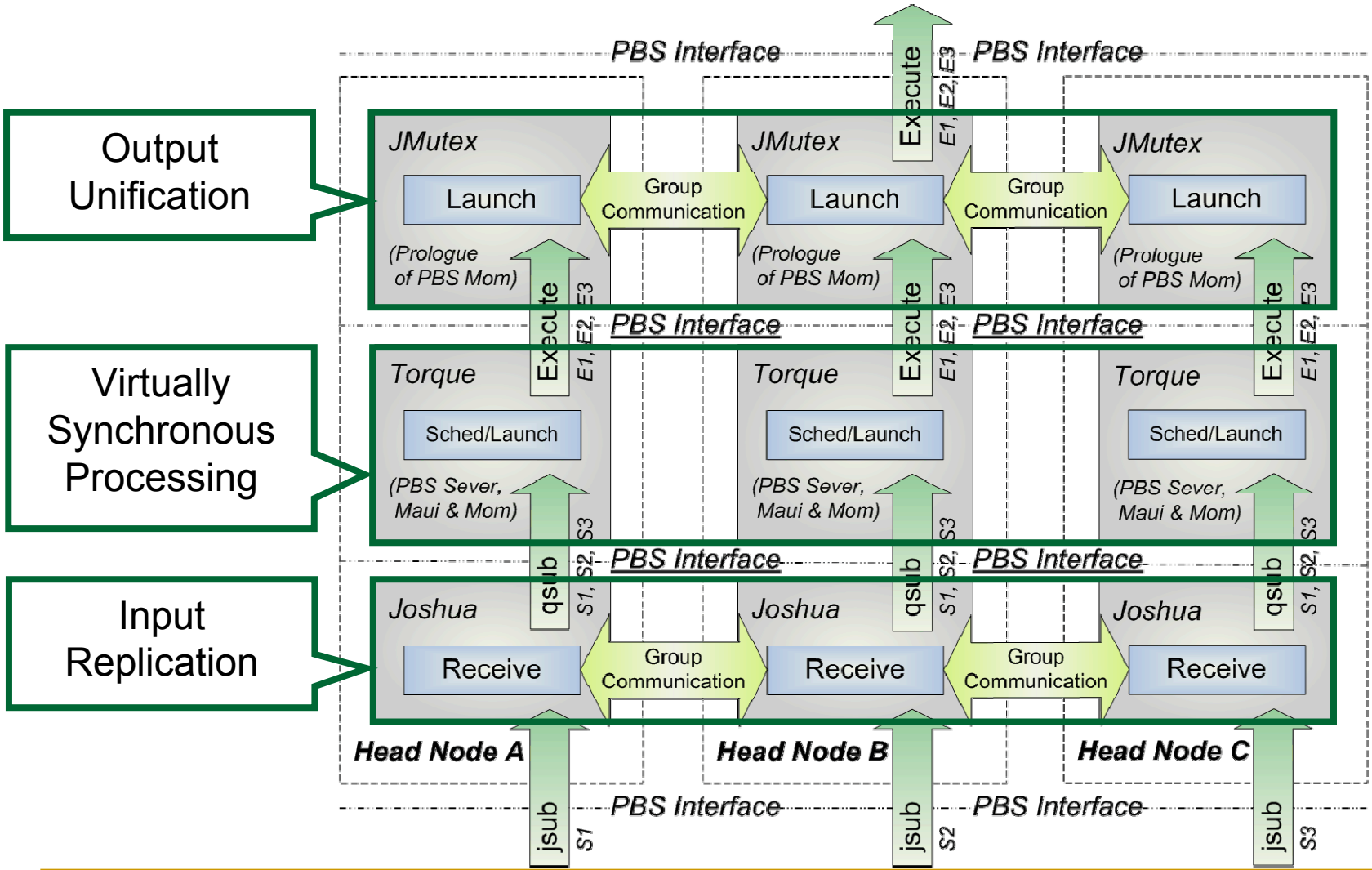
Prototype Implementation

- University of Reading Master's thesis project
- Based on external symmetric active/active replication using a group communication system
- Transis v1.03 for group communication
- TORQUE v2.0p5 as queue manager
- Maui v3.2.6p13 as job scheduler

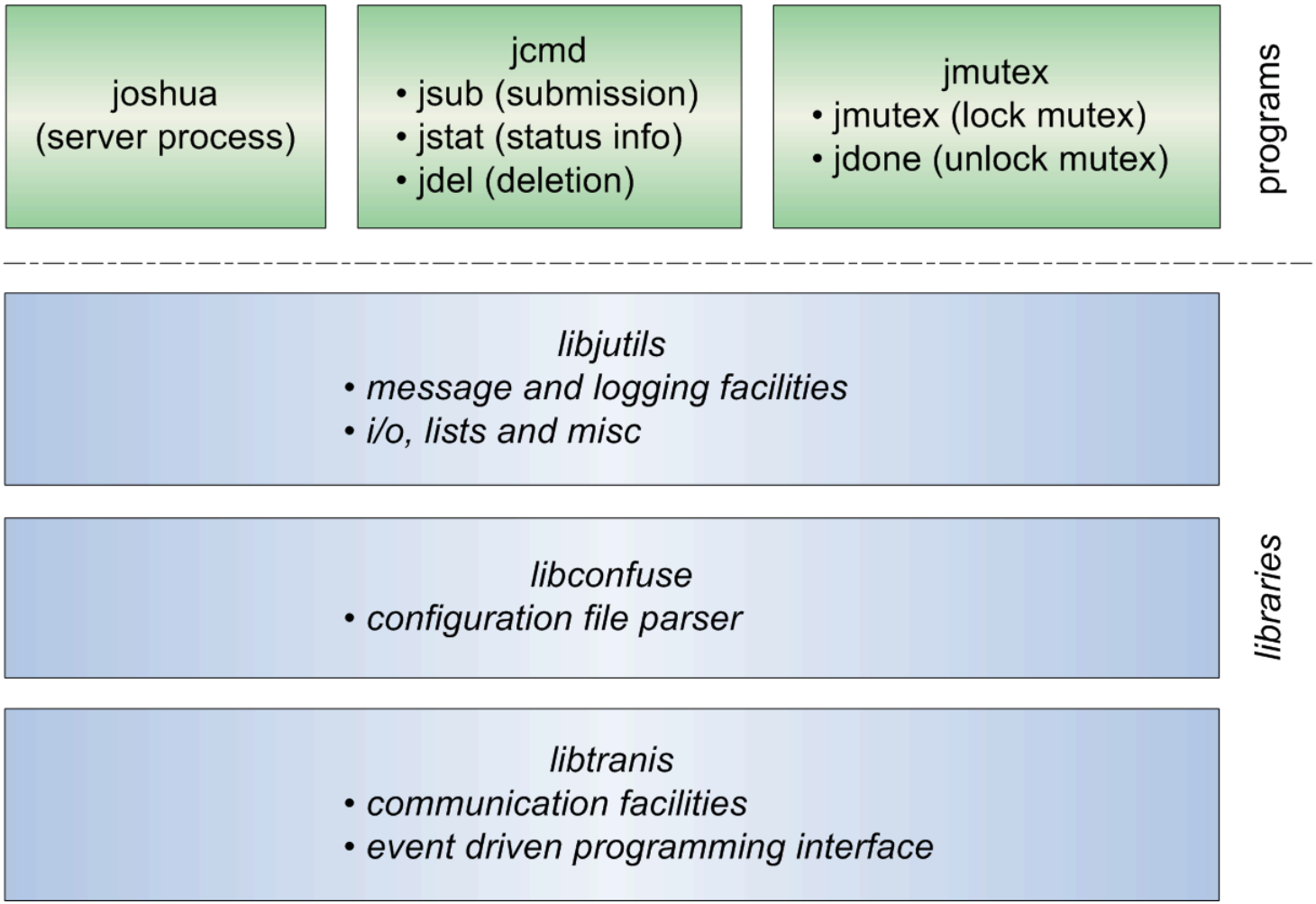
JOSHUA: Symmetric Active/Active Replication for PBS Torque



Symmetric Active/Active Replication



JOSHUA Software Architecture



Introduced Overhead

- Group communication system adds overhead for reliable and atomic multicast
- Latency increases with number of active nodes
- Throughput decreases with number of active nodes
- Overhead in acceptable range for this scenario

  - Nodes: Pentium III 450MHz on 100MBit/s Ethernet

System	#	Latency	Overhead
TORQUE	1	98 ms	
JOSHUA/TORQUE	1	134 ms	36 ms / 37%
JOSHUA/TORQUE	2	265 ms	158 ms / 161%
JOSHUA/TORQUE	3	304 ms	206 ms / 210%
JOSHUA/TORQUE	4	349 ms	251 ms / 256%

Job Submission Latency Overhead

System	#	10 Jobs	50 Jobs	100 Jobs
TORQUE	1	0.93s	4.95s	10.18s
JOSHUA/TORQUE	1	1.32s	6.48s	14.08s
JOSHUA/TORQUE	2	2.68s	13.09s	26.37s
JOSHUA/TORQUE	3	2.93s	15.91s	30.03s
JOSHUA/TORQUE	4	3.62s	17.65s	33.32s

Job Submission Throughput Overhead

Symmetric Active/Active High Availability for Head and Service Nodes

- $A_{component}$ = MTTF / (MTTF + MTTR)
- A_{system} = 1 - (1 - $A_{component}$) n
- T_{down} = 8760 hours * (1 – A)
- Single node MTTF: 5000 hours
- Single node MTTR: 72 hours

Nodes	Availability	Est. Annual Downtime
1	98.58%	5d 4h 21m
2	99.97%	1h 45m
3	99.9997%	1m 30s
4	99.999995%	1s

Single-site redundancy for 7 nines does not mask catastrophic events.

Remaining Issues

- Stability problems with Transis group communication system (crashed after 3-5 days of stress test)
- PBS mom servers did not simply ignore a failed head node, but rather kept the current job in running status until it returned to service
- PBS mom server and JOSHUA scripts run on compute nodes, where failures are not tolerated
- Room for performance improvement within group communication system

Future Work

- Fix remaining issues for production-type solution
- Provide similar solutions for other critical HPC system services, such as:
  - Parallel Virtual File System (PVFS) metadata
  - Lustre Cluster File System metadata
  - Others: …

MOLAR: 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.
- Part of the Forum to Address Scalable Technology for Runtime and Operating Systems (FAST-OS).
- MOLAR is a collaborative research effort (*www.fastos.org/molar*):

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

Office of Science
U.S. DEPARTMENT OF ENERGY

NC STATE UNIVERSITY

OHIO STATE

LOUISIANA TECH UNIVERSITY

The University of Reading

CRAY THE SUPERCOMPUTER COMPANY

JOSHUA: Symmetric Active/Active Replication for Highly Available HPC Job and Resource Management: Questions or Comments?

Kai Uhlemann[1,2], Christian Engelmann[1,2], and Stephen L. Scott[2]

1 Department of Computer Science
The University of Reading, Reading, UK

2 Computer Science and Mathematics Division,
Oak Ridge National Laboratory, Oak Ridge, USA