Symmetric Active/Active High Availability for High-Performance Computing System Services: *Accomplishments and Limitations*

Christian Engelmann[1,2], Stephen L. Scott[1],
Chokchai (Box) Leangsuksun[3], Xubin (Ben) He[4]

[1] Oak Ridge National Laboratory, Oak Ridge, USA
[2] The University of Reading, Reading, UK
[3] Louisiana Tech University, Ruston, USA
[4] Tennessee Tech University, Cookeville, USA

Overview

- Overall background
  - Scientific high-performance computing
  - Availability issues in high-performance computing systems
- Service-level availability taxonomy
- Symmetric active/active replication
  - Model, algorithms, architecture
- Symmetric active/active prototypes
  - PBS TORQUE job and resource management service
  - Parallel Virtual File System metadata service
- Symmetric active/active replication framework

Scientific High-Performance Computing

- Large-scale high-performance computing
  - Tens-to-hundreds of thousands of processors
  - Current systems: IBM BG/L and Cray XT5
  - Next-generation: Petascale IBM BG/P, Cray Baker
- Computationally and data intensive applications
  - 100 TFlops - 1 PFlops with 100 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

Availability Measured by the Nines

see <http://www.nccs.gov/computing-resources/systems-status/> for current ORNL system status

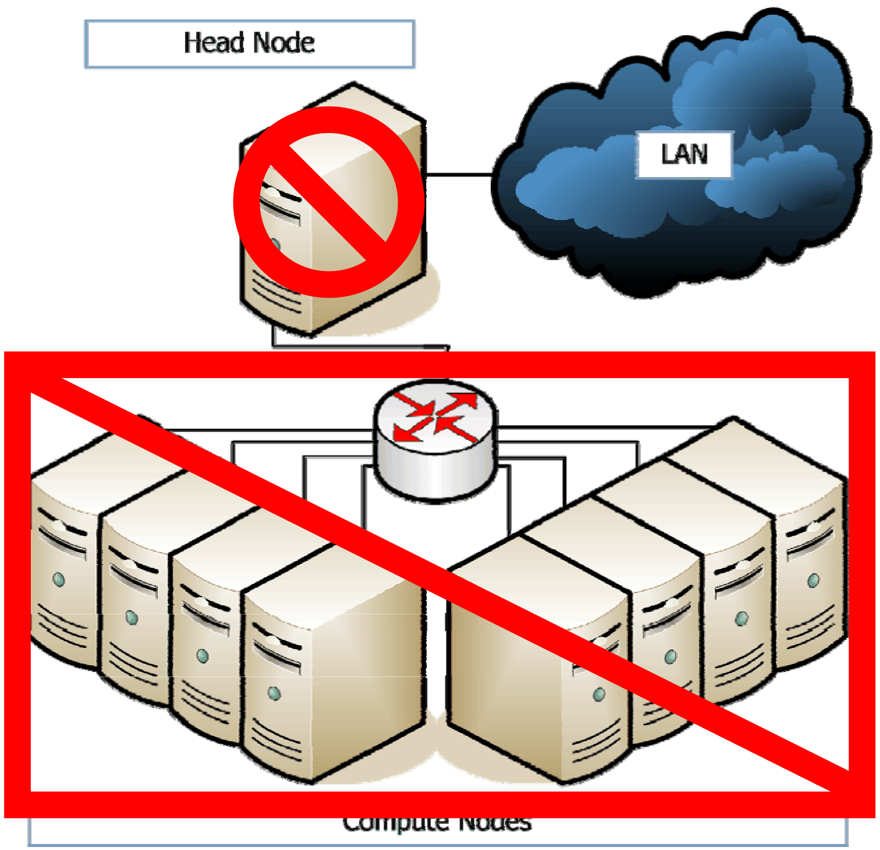
9's	Availability	Downtime/Year	Examples
1	90.0%	36 days, 12 hours	Personal Computers
2	99.0%	87 hours, 36 min	Entry Level Business
3	99.9%	8 hours, 45.6 min	ISPs, Mainstream Business
4	99.99%	52 min, 33.6 sec	Data Centers
5	99.999%	5 min, 15.4 sec	Banking, Medical
6	99.9999%	31.5 seconds	Military Defense

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

Typical Failure Causes in HPC Systems

- Overheating (design errors - specification vs. usage)
- Memory and network errors (soft errors)
- Hardware failures due to wear/age of:
  - Hard drives, memory modules, network cards, processors
- Software failures due to bugs in:
  - Operating system, middleware, applications
- ➔ Different scale requires different solutions:
  - ➔ Compute nodes (up to ~200,000)
  - ➔ Front-end, service, and I/O nodes (1 to ~200)

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

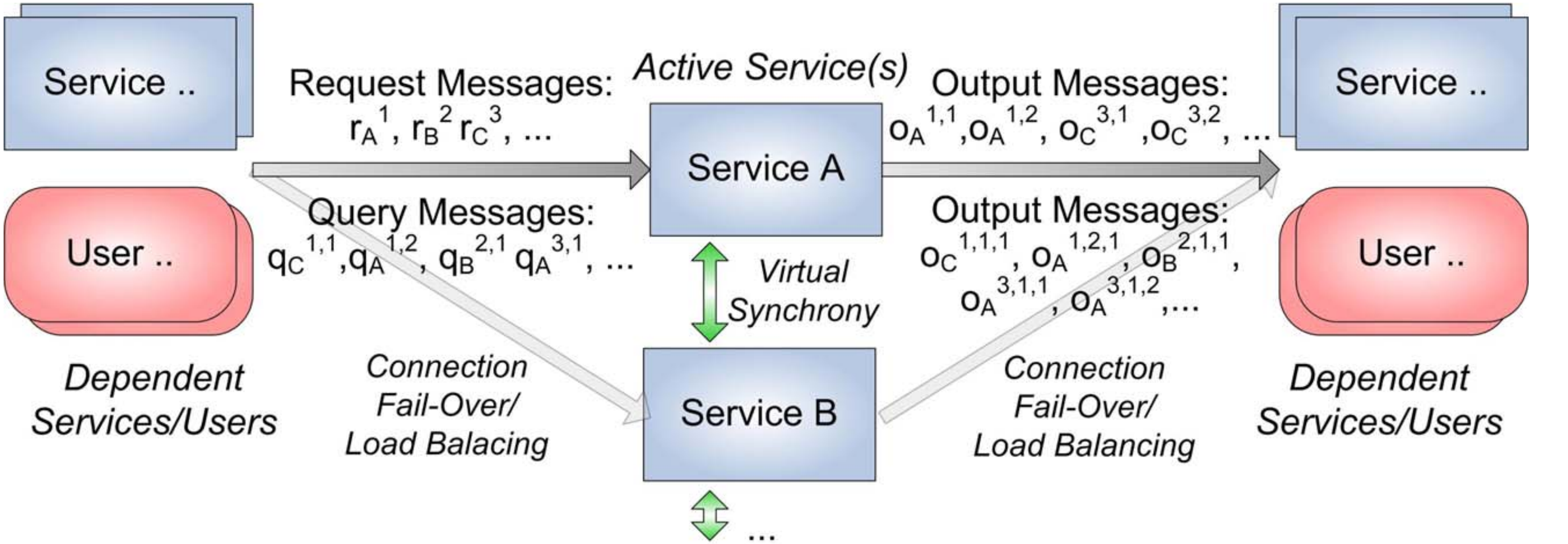
➢ MTTR = 0 ➔ A = 1.00 (100%) continuous availability

➢ Fail-stop model

Service-level Availability Taxonomy

- No redundancy → Manual masking
- Hardware redundancy only → Active/cold standby
- Hardware and software redundancy:
  - Active/warm standby → Replication in intervals, *1+m* service nodes
  - Active/hot standby → Replication on change, *1+m* service nodes
  - Asymmetric active/active → High availability clustering, *n+m* service nodes
  - Symmetric active/active → State-machine replication, *n* service nodes

Symmetric Active/Active Replication



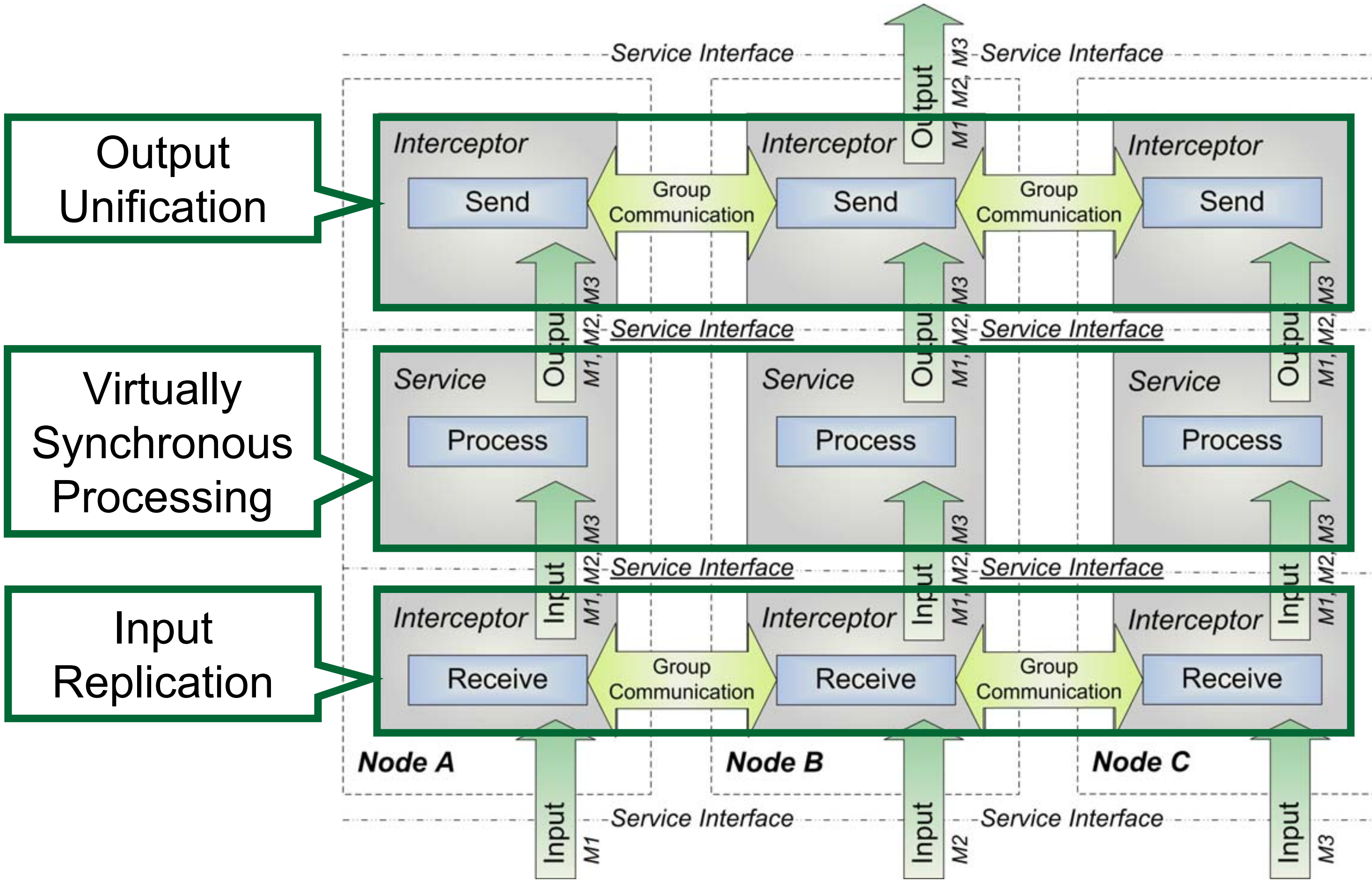
- Replication of service capability via multiple active services
- Replication of state among active services
- Virtual synchrony (state-machine replication) model

Comparison of Replication Methods

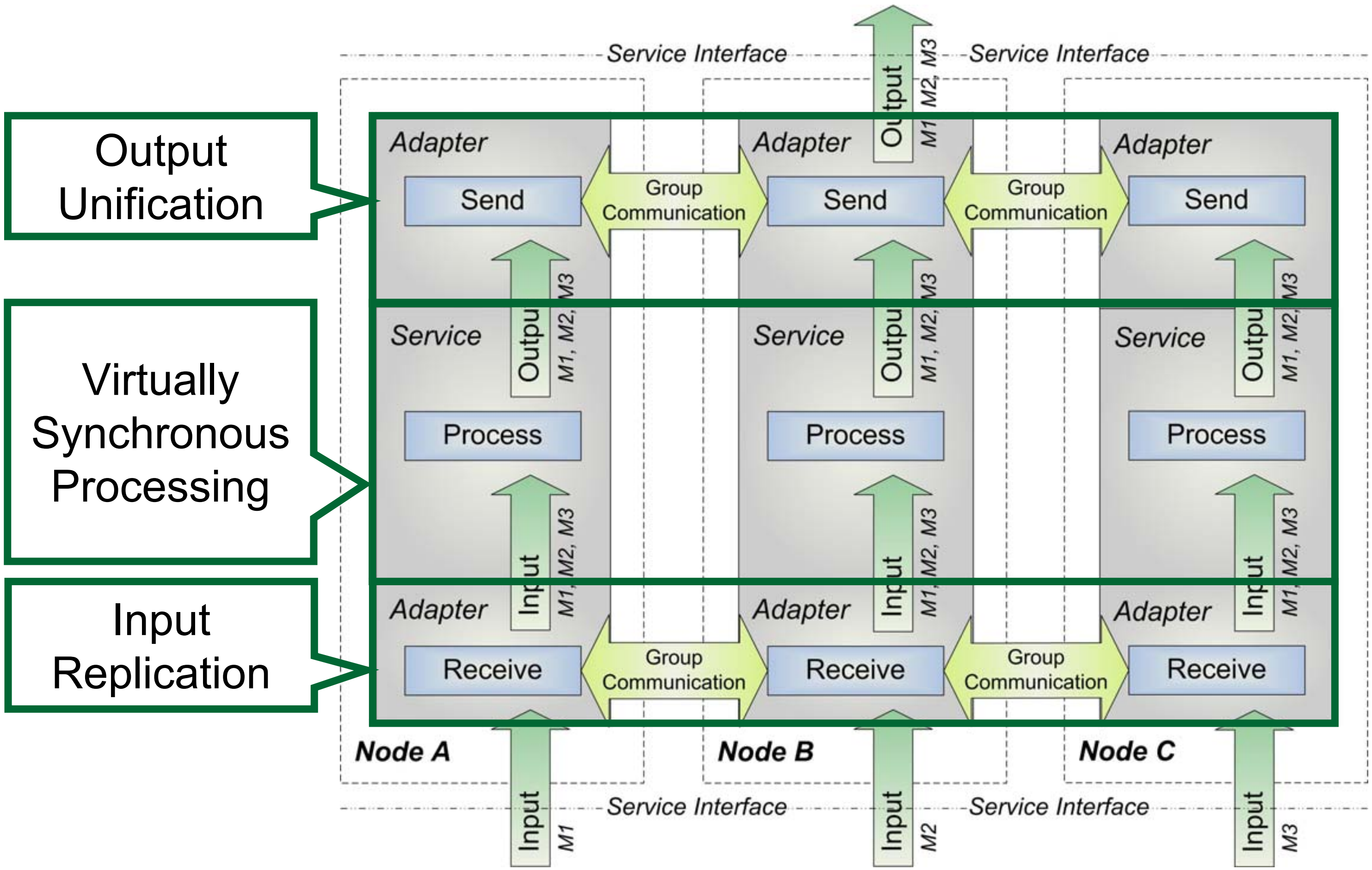
Method	$MTTR_{recovery}$	Latency Overhead
Warm-Standby	$T_d + T_f + T_r + T_c$	0
Hot-Standby	$T_d + T_f + T_r$	$2l_{A,B}$, $O(log_2(n))$, or worse
Asymmetric with Warm-Standby	$T_d + T_f + T_r + T_c$	0
Asymmetric with Hot-Standby	$T_d + T_f + T_r$	$2l_{A,\alpha}$, $O(log_2(n))$, or worse
Symmetric	$T_d + T_f + T_r$	$2l_{A,B}$, $O(log_2(n))$, or worse

T_d, time between failure occurrence and detection
T_f, time between failure detection and fail-over
T_c, time to recover from checkpoint to previous state
T_r, time to reconfigure client connections
$l_{A,B}$ and $l_{A,\alpha}$, communication latency between A and B, and A and α

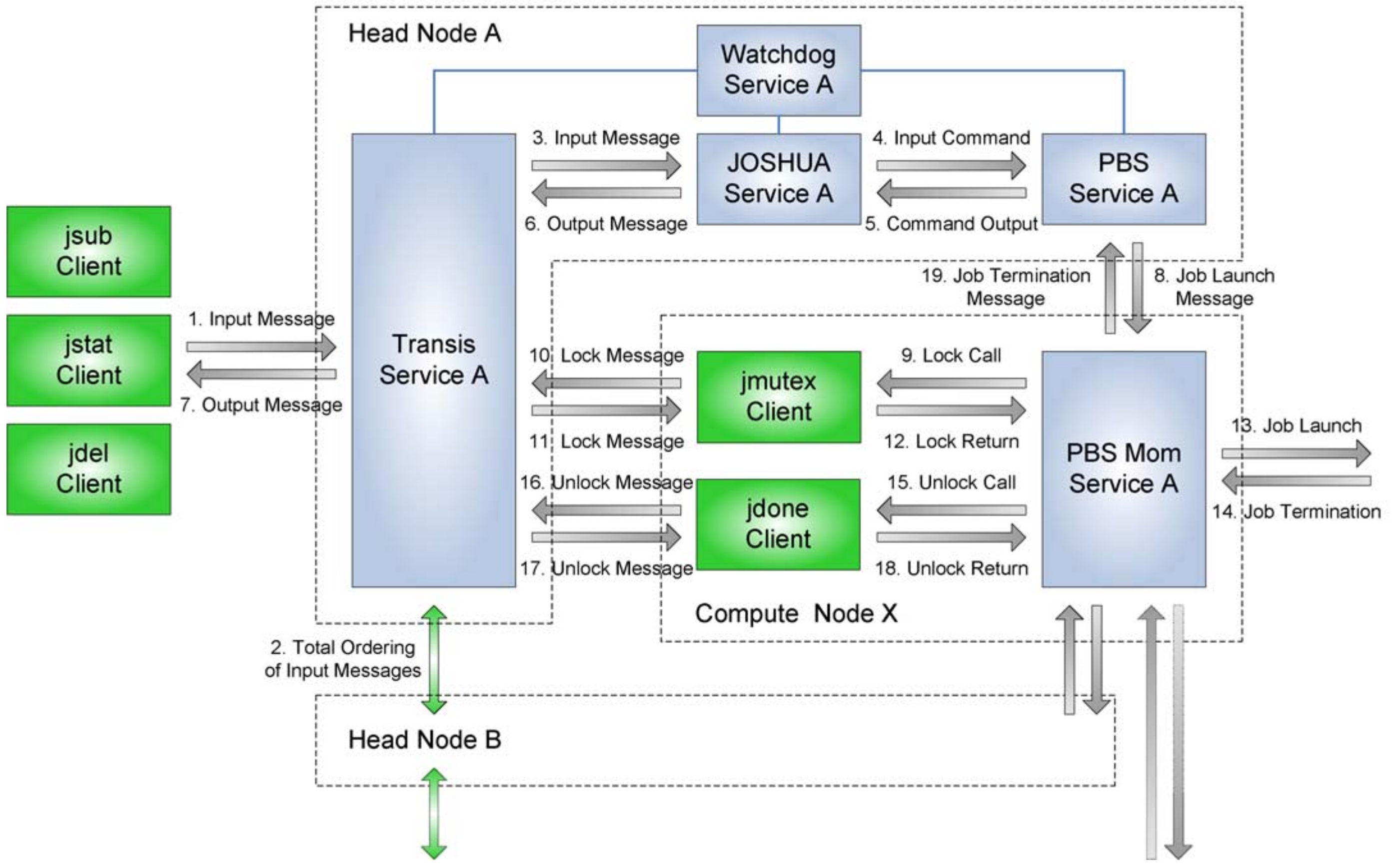
External Symmetric Active/Active Replication



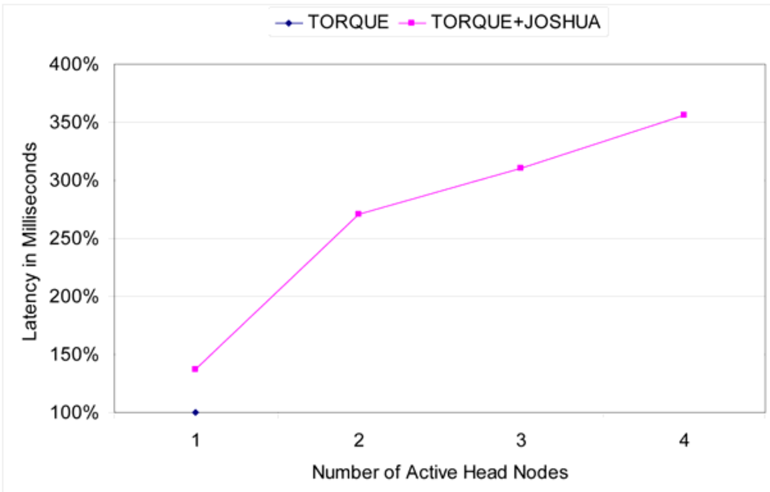
Internal Symmetric Active/Active Replication



Symmetric Active/Active PBS Torque



Symmetric Active/Active PBS Torque

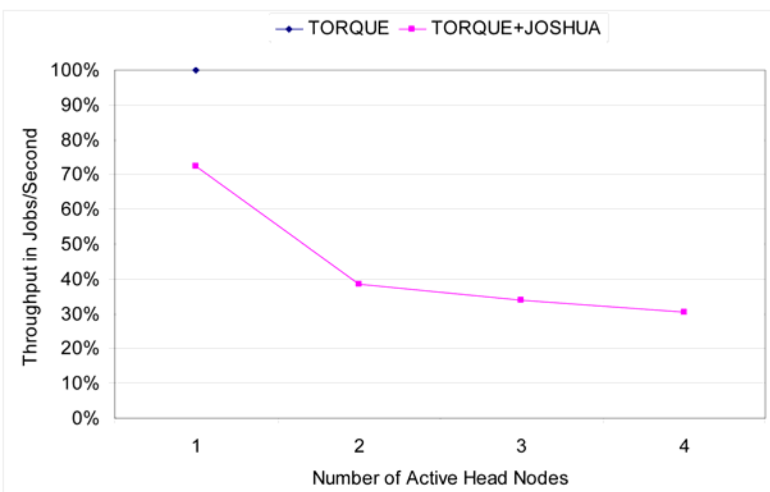


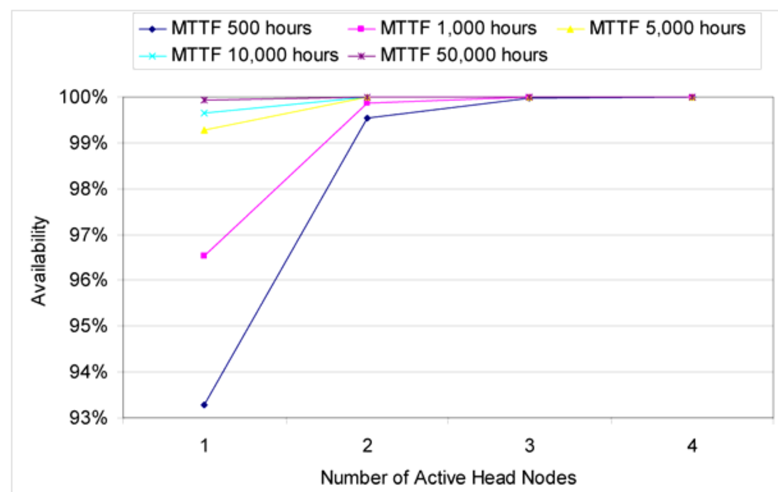
$$A_{redundancy} = [1 - (1 - A_{component})^n][1 - (1 - A_{redundant})^n]$$

$$A_{component} = \frac{MTTF_{component}}{MTTF_{component} + MTTR_{component}}$$

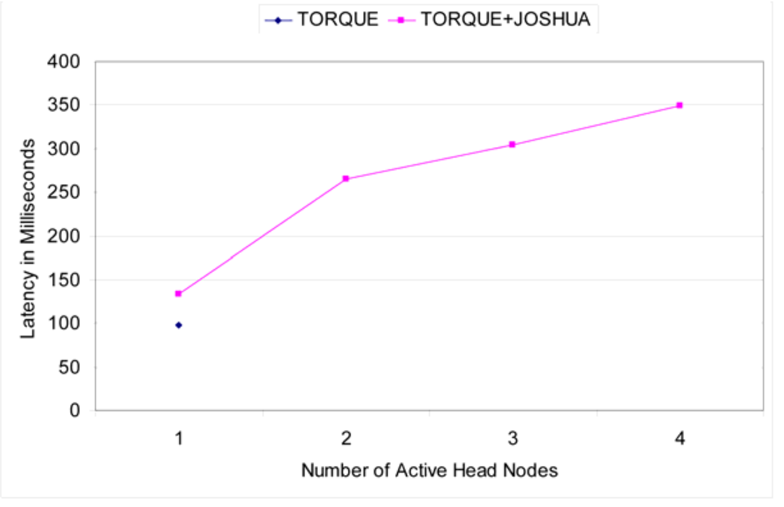
$$A_{redundant} = \frac{MTTF_{component}}{MTTF_{component} + MTTR_{recovery}}$$

$MTTR_{recovery}$ = *500 milliseconds*
$MTTR_{component}$ = *36 hours*





Symmetric Active/Active PBS Torque



$$A_{redundancy} = [1 - (1 - A_{component})^n][1 - (1 - A_{redundant})^n]$$

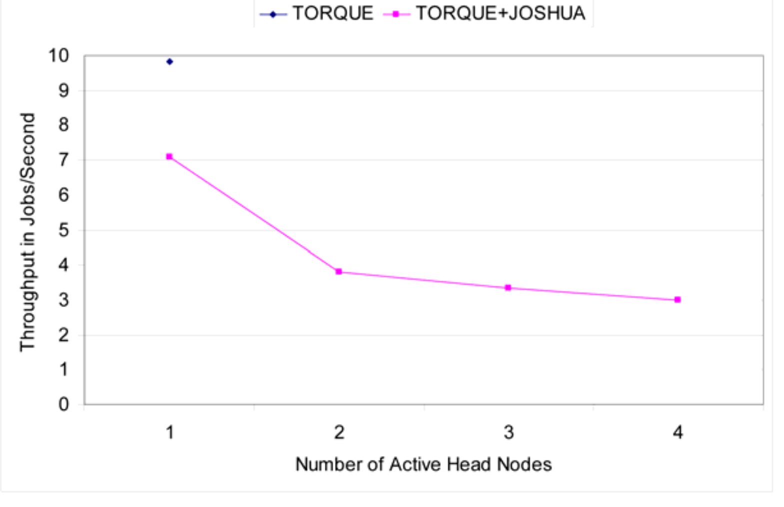
$$A_{component} = \frac{MTTF_{component}}{MTTF_{component} + MTTR_{component}}$$

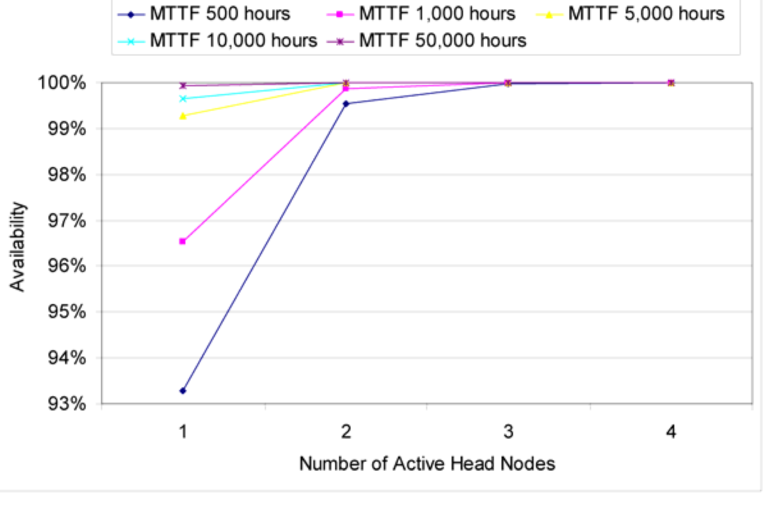
$$A_{redundant} = \frac{MTTF_{component}}{MTTF_{component} + MTTR_{recovery}}$$

$\Downarrow$

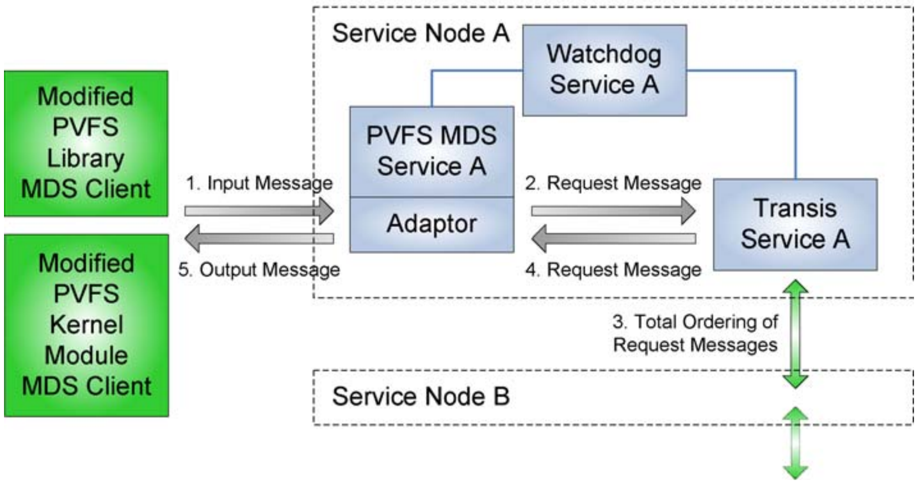
$MTTR_{recovery}$ = *500 milliseconds*
$MTTR_{component}$ = *36 hours*

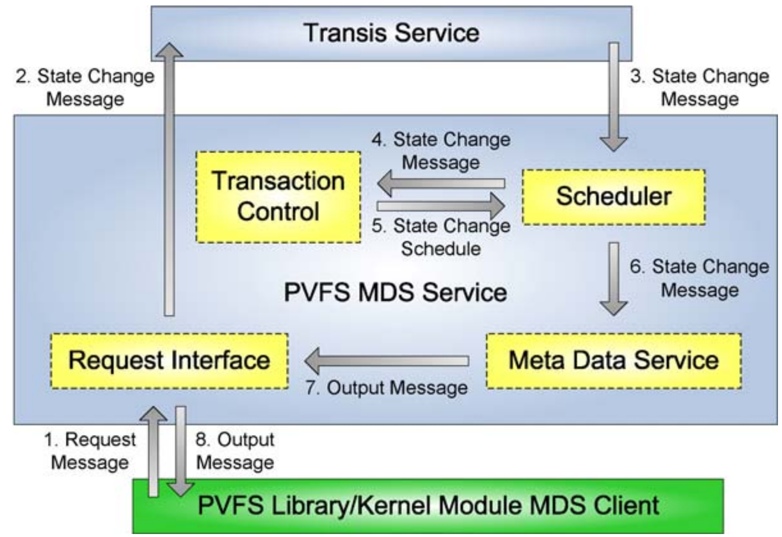
$\Downarrow$

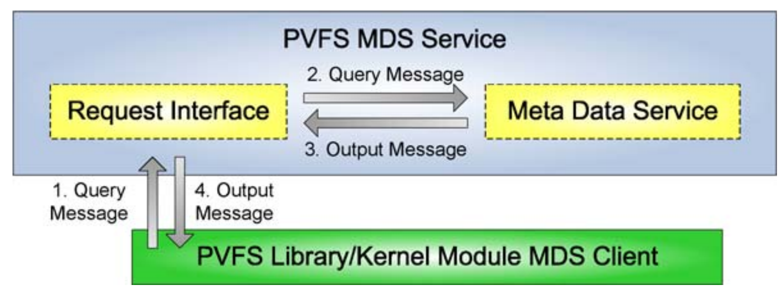




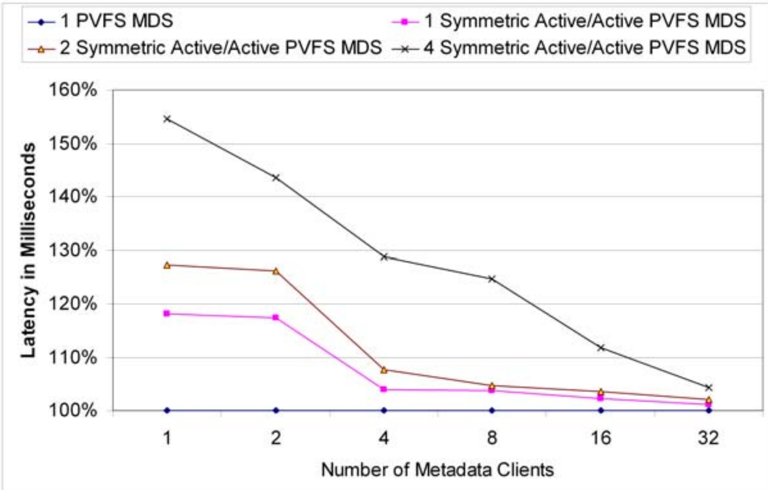
Symmetric Active/Active PVFS MDS

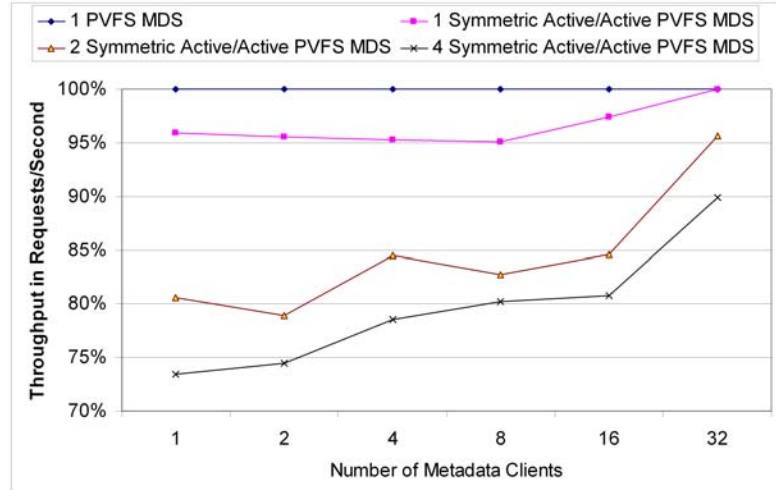


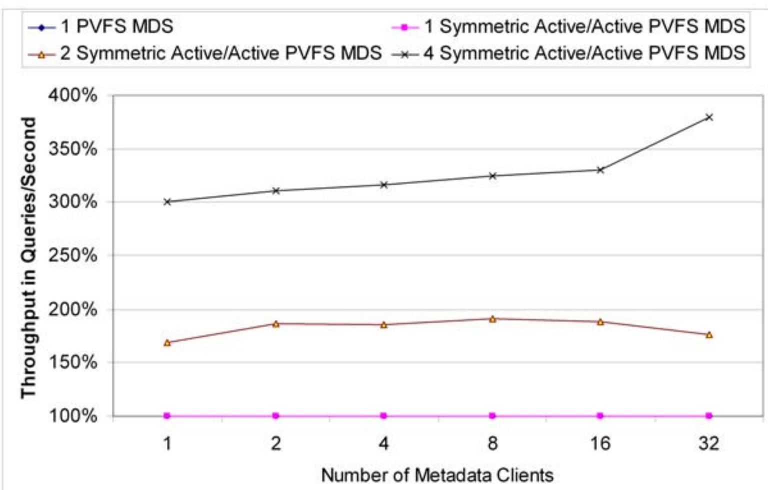


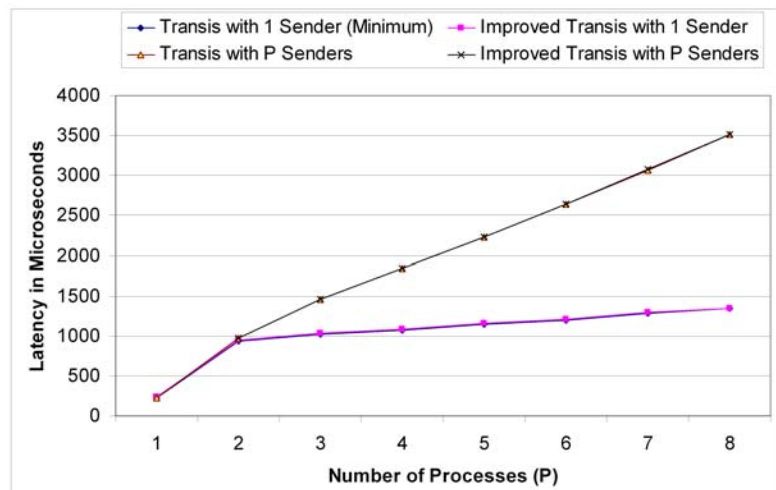


Symmetric Active/Active PVFS MDS

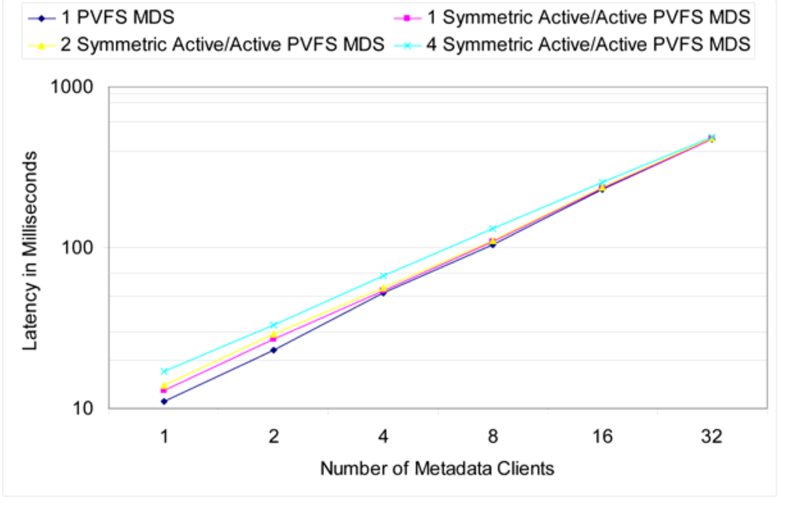


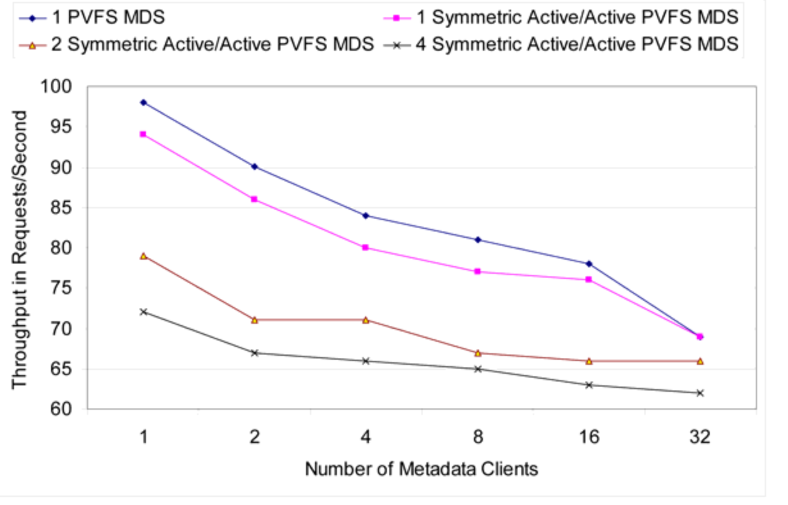


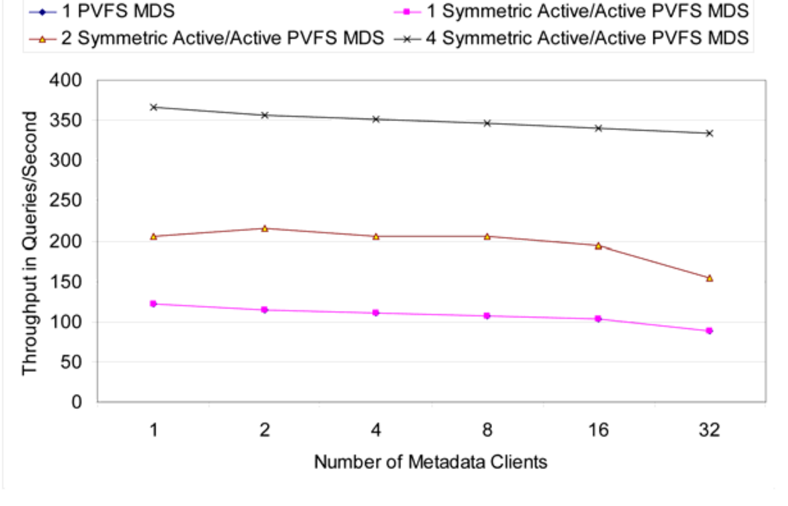


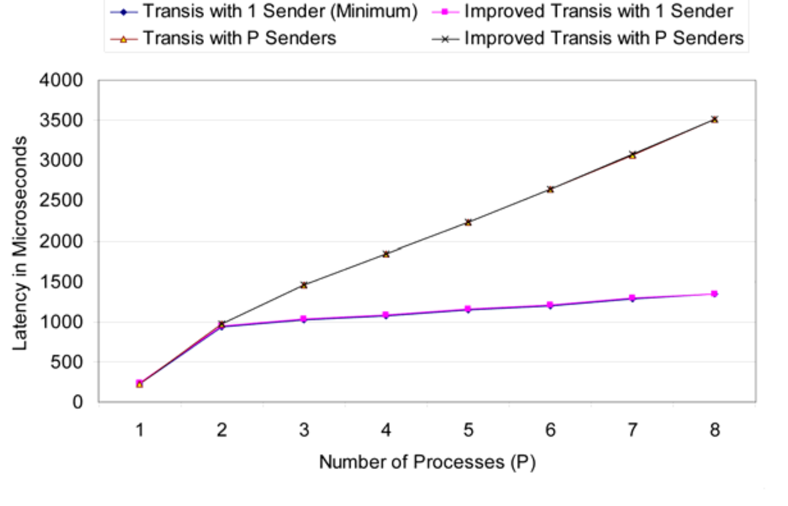


Symmetric Active/Active PVFS MDS

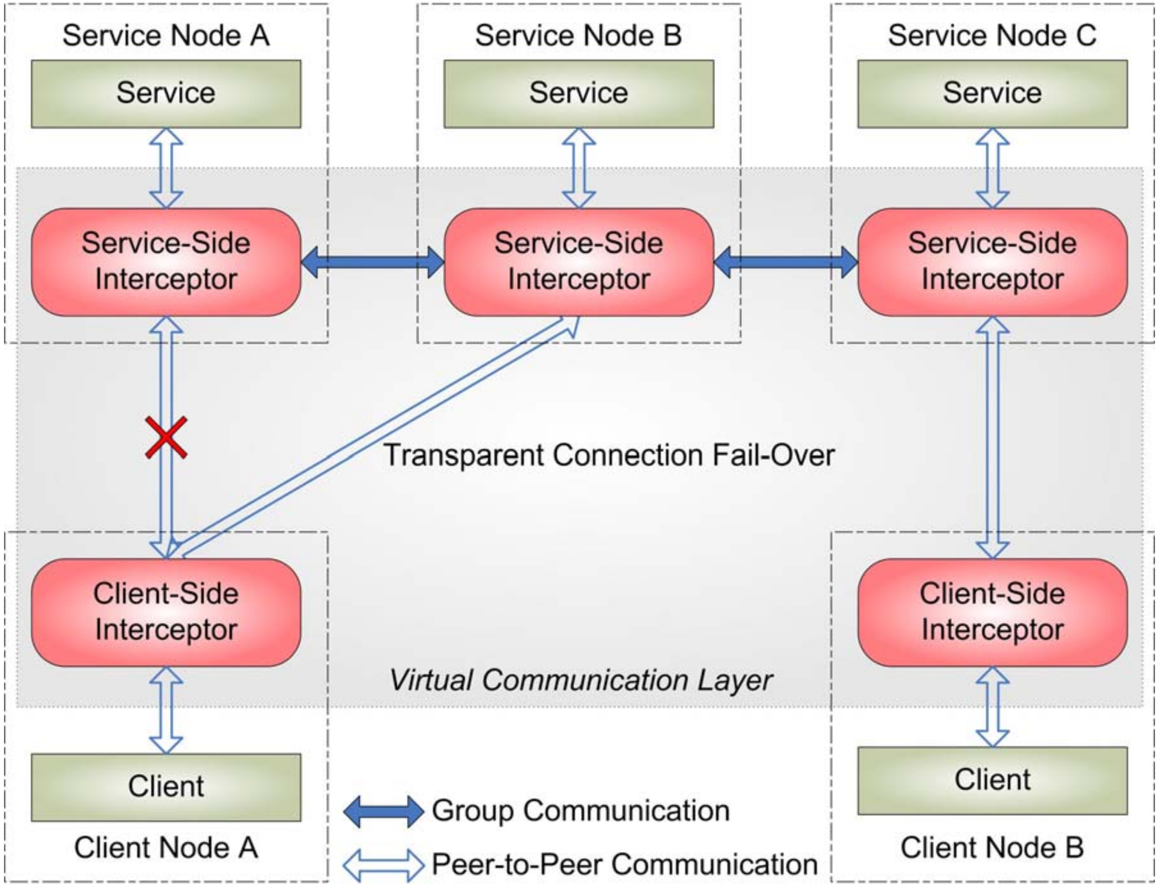




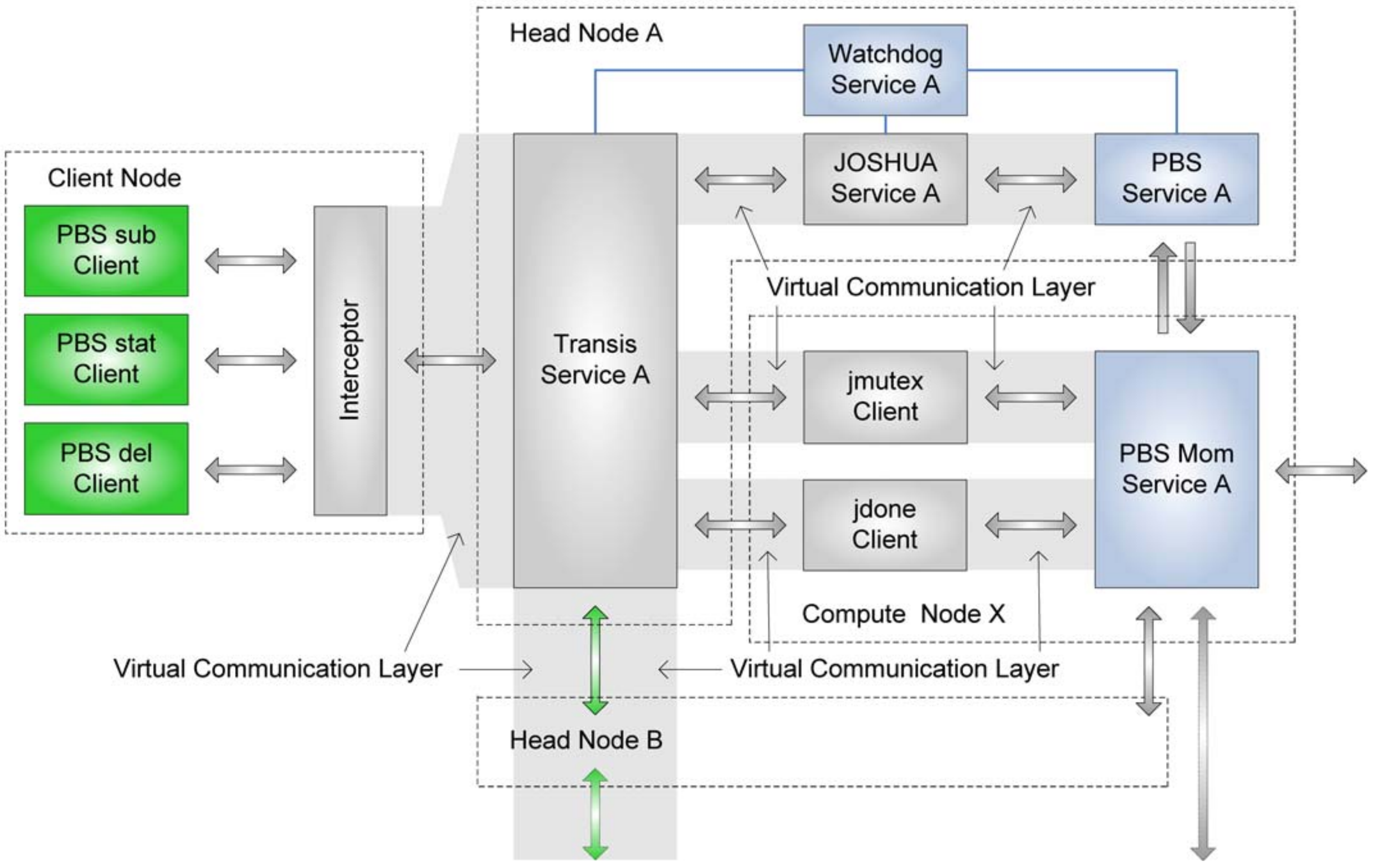




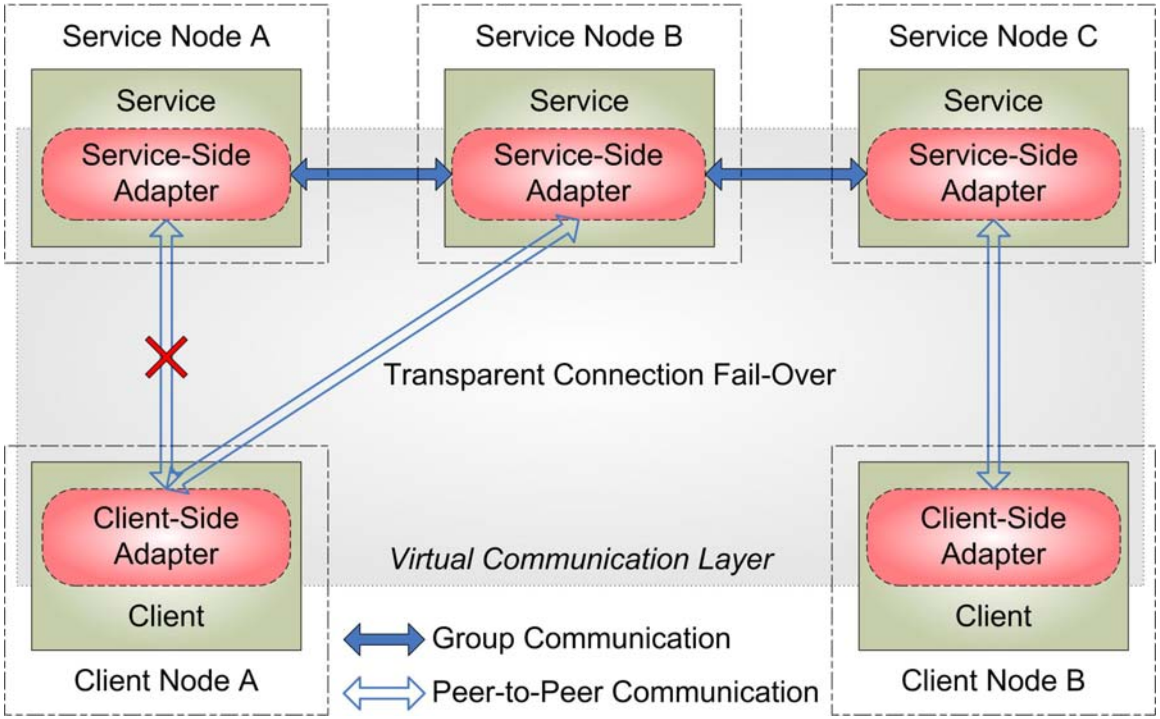
Transparent External Symmetric Active/Active Replication for Client/Service Scenarios



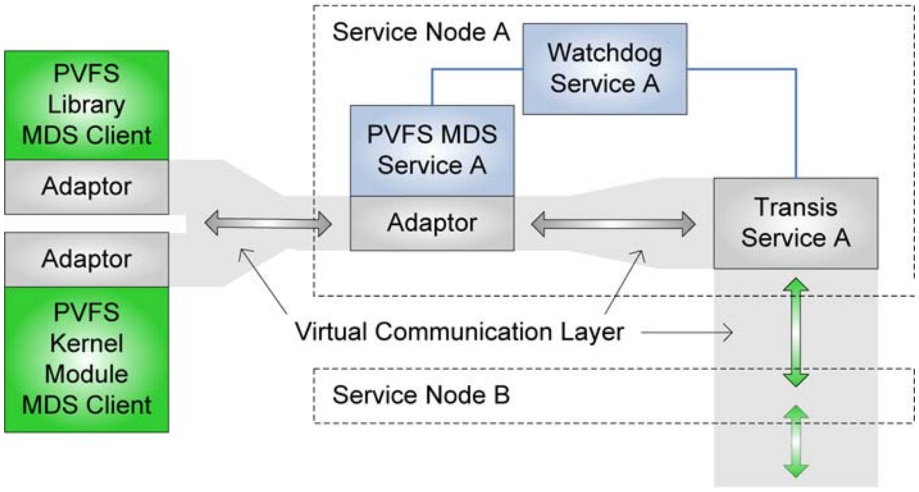
Transparent External Symmetric Active/Active Replication: PBS TORQUE Example



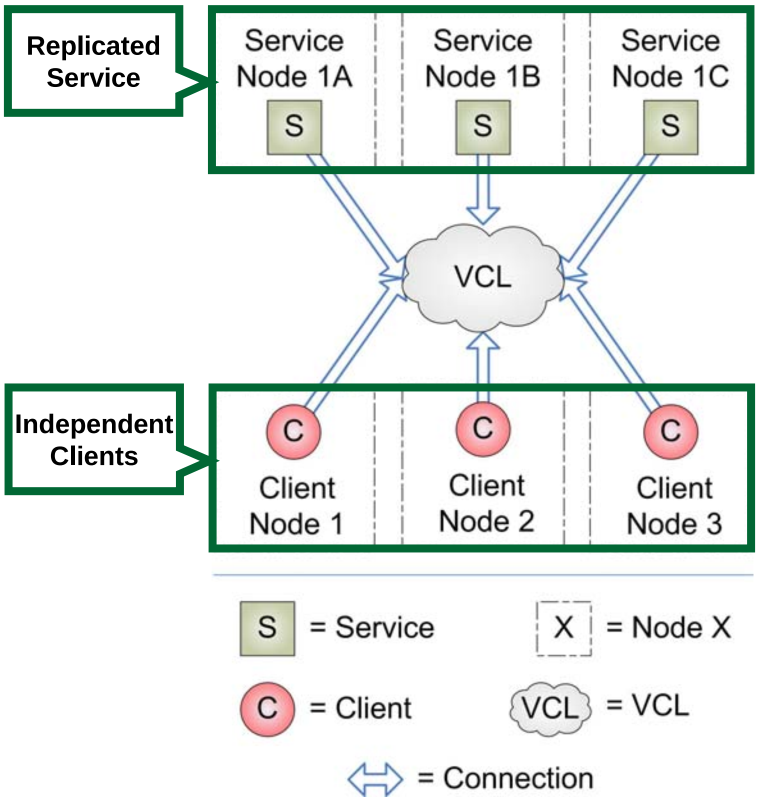
Transparent Internal Symmetric Active/Active Replication for Client/Service Scenarios



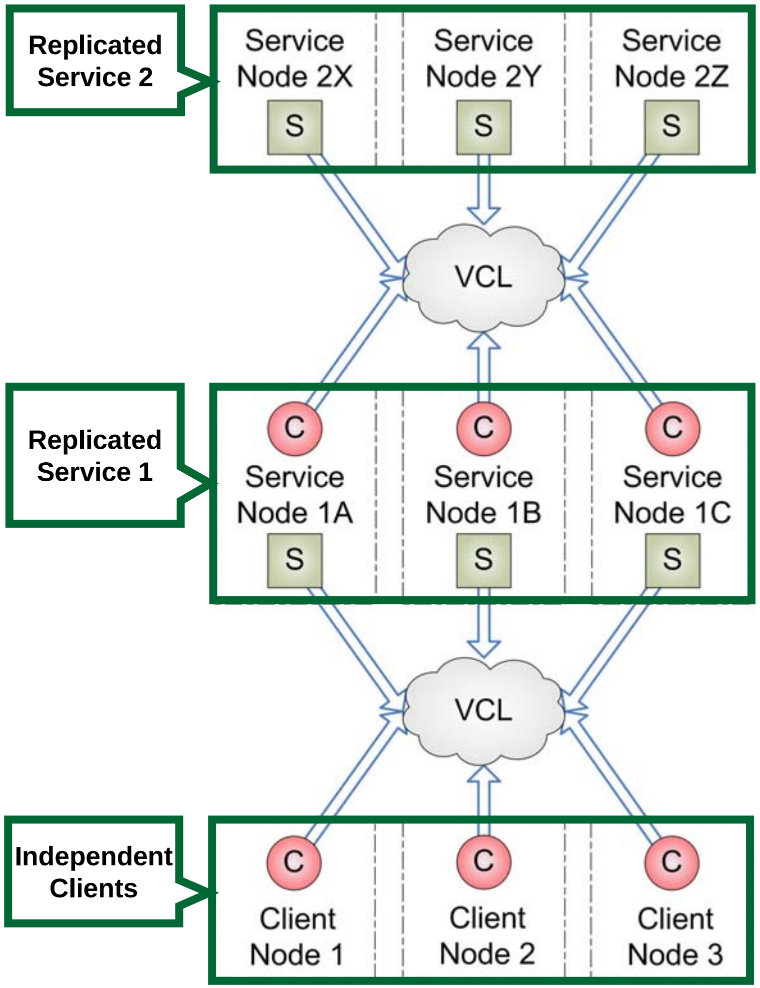
Transparent Internal Symmetric Active/Active Replication: PVFS MDS Example



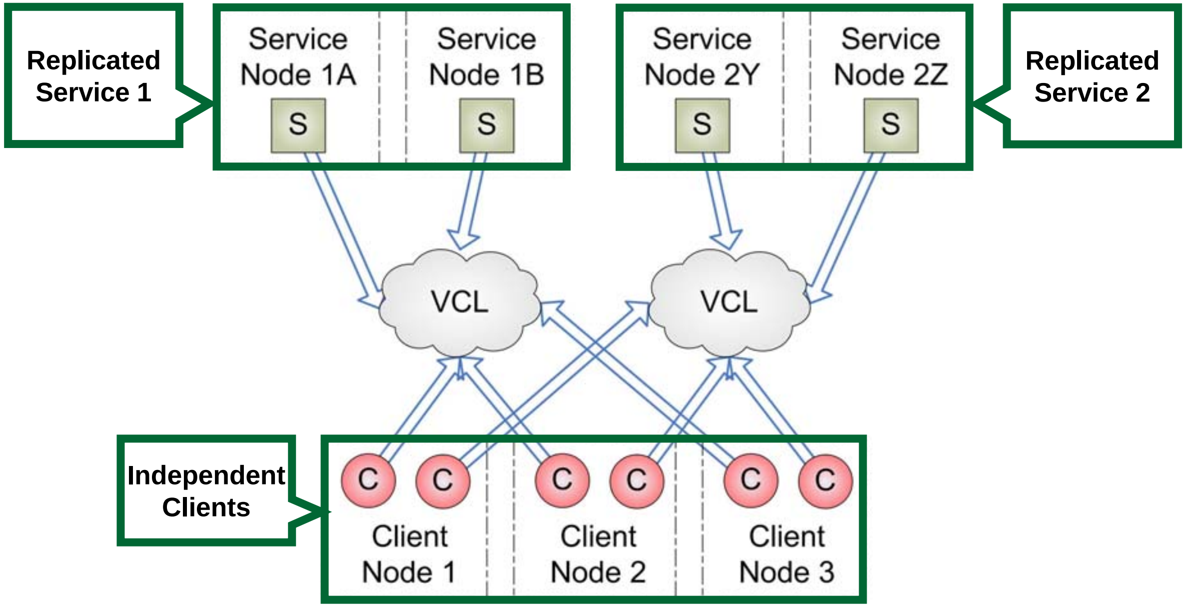
Transparent Symmetric Active/Active Replication for Client/Service Scenarios – High-Level Abstraction



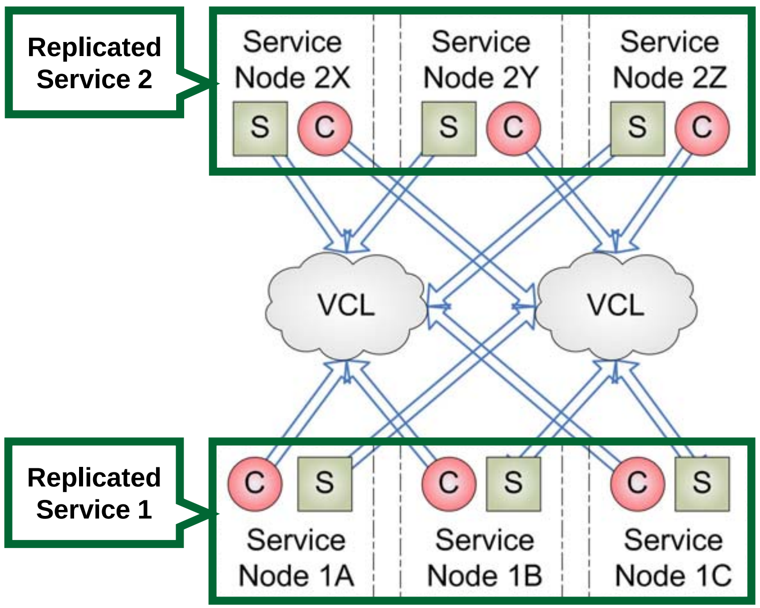
Transparent Symmetric Active/Active Replication for Client/Client+Service/Service Scenarios



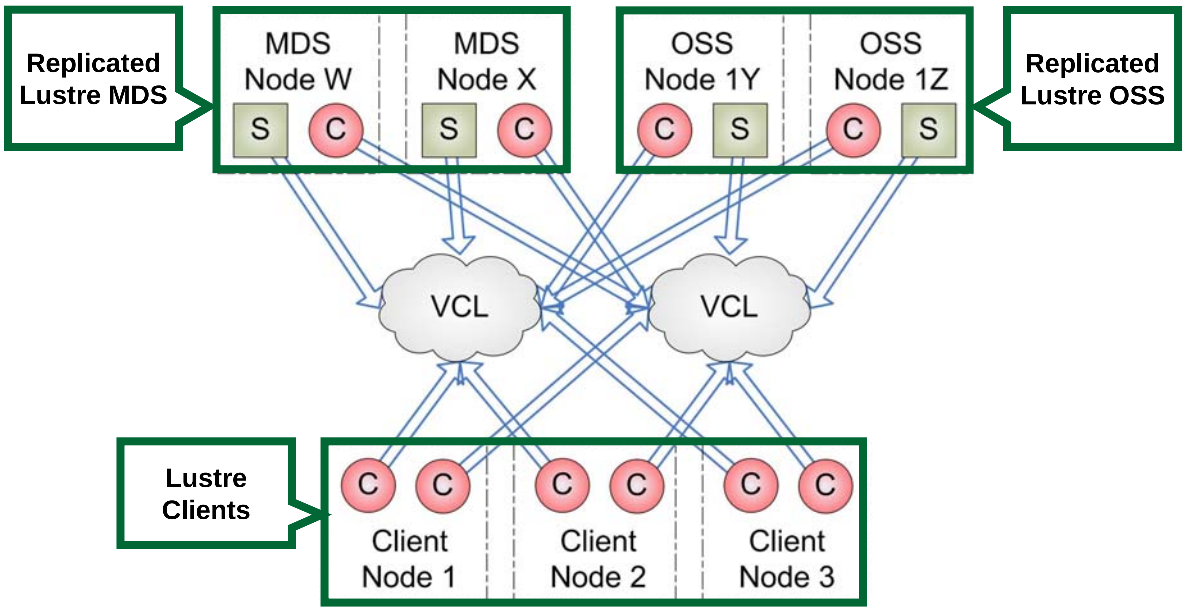
Transparent Symmetric Active/Active Replication for Client/2 Services Scenarios



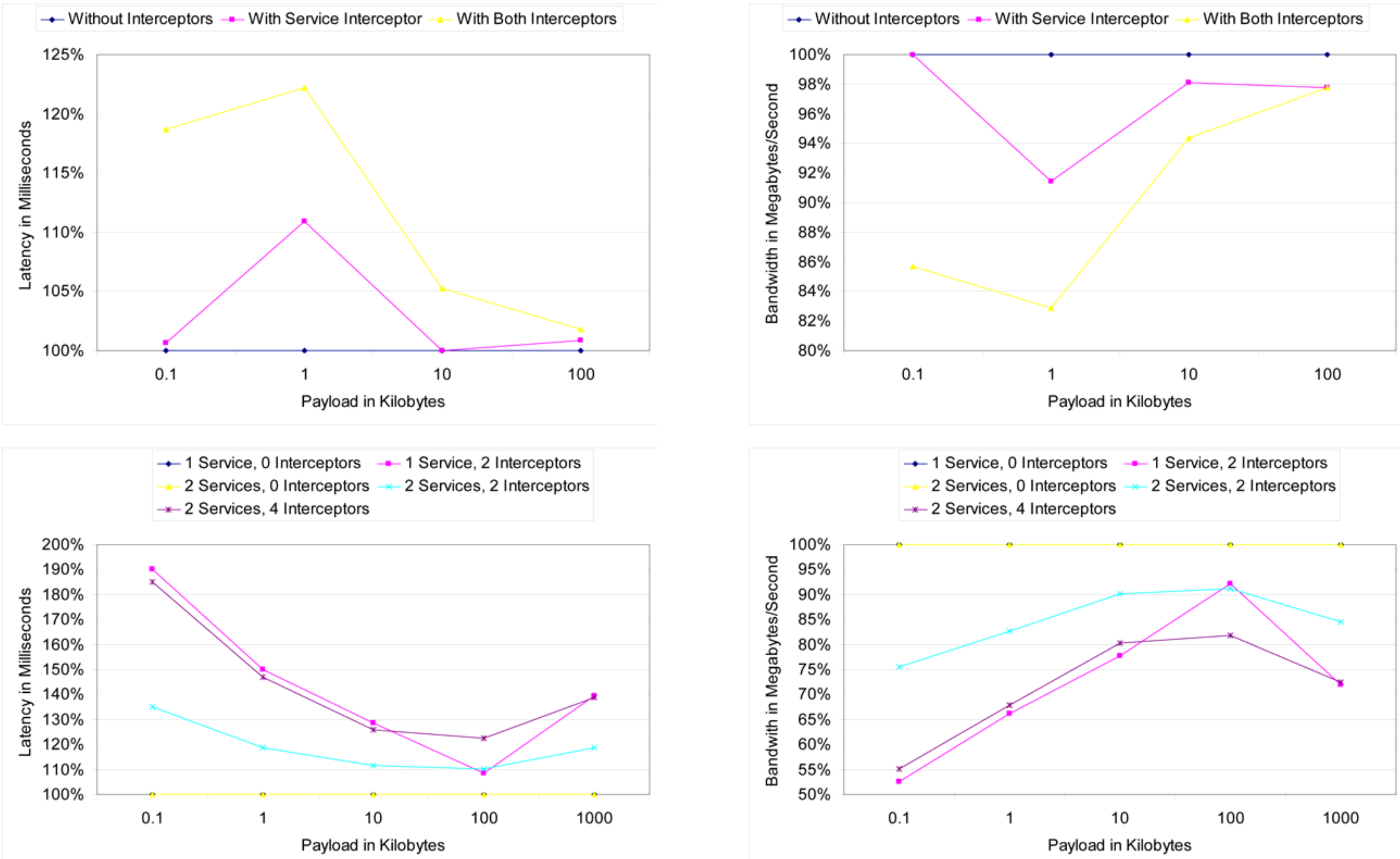
Transparent Symmetric Active/Active Replication for Service/Service Scenarios



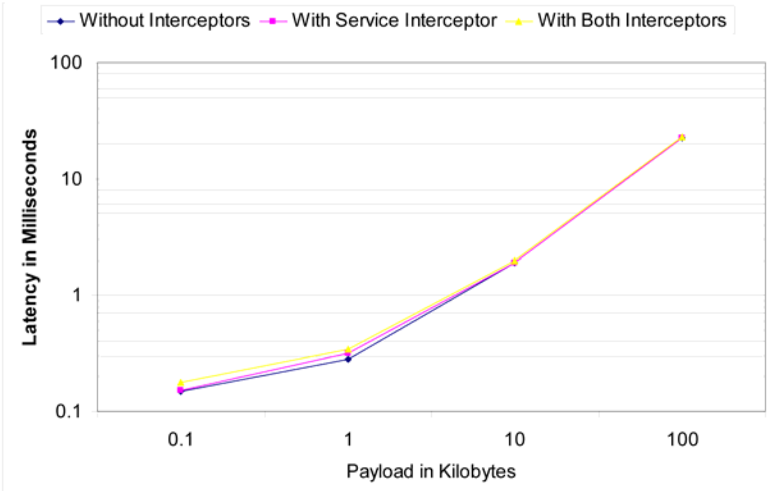
Example: Transparent Symmetric Active/Active Replication for the Lustre Cluster File System

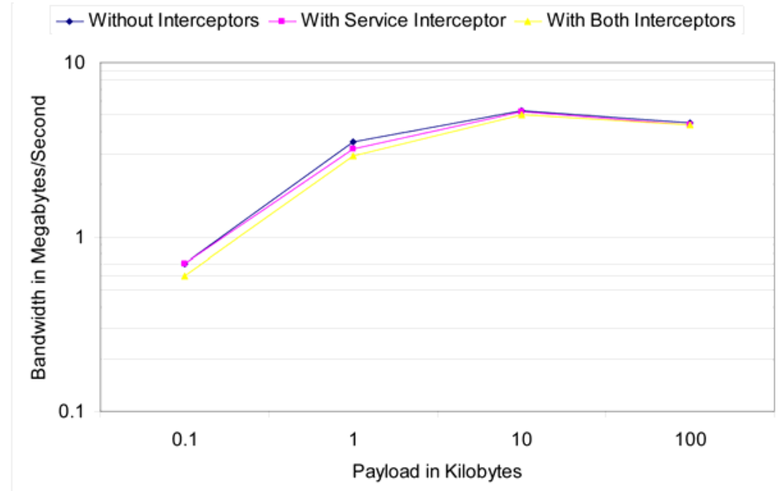


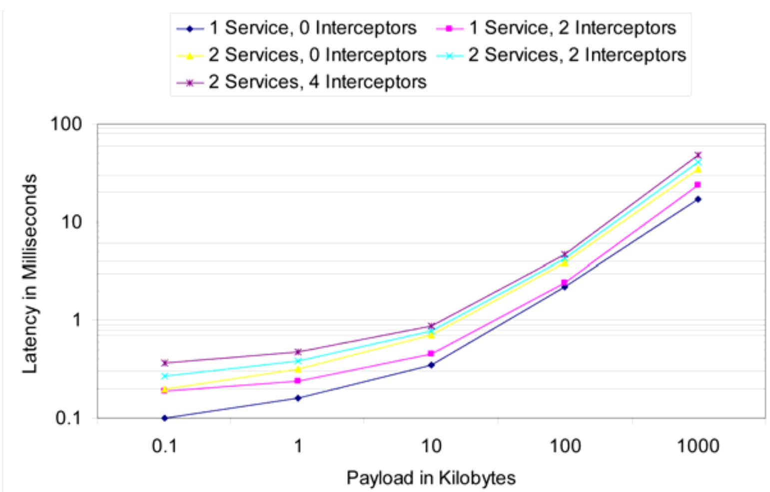
Interceptor Communication Overhead

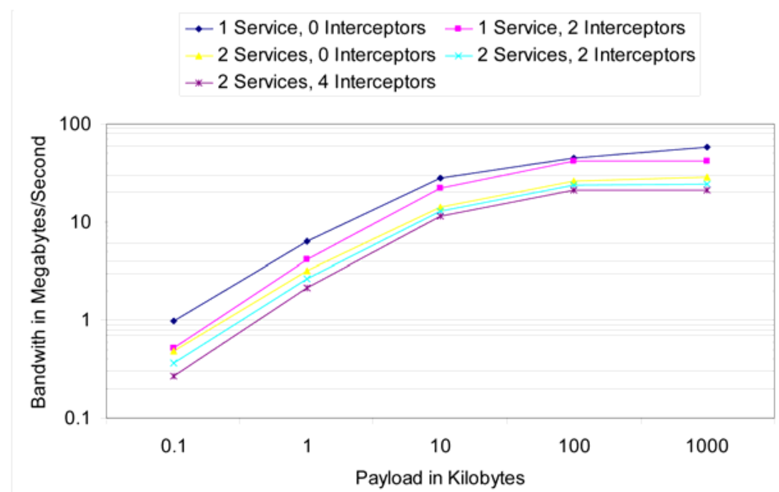


Interceptor Communication Overhead









Accomplishments

- Examined past and ongoing work in high availability for:
  - HPC, distributed systems, and IT/telco services
- Provided a modern service high availability taxonomy
- Generalized HPC system architectures
- Identified specific HPC system availability deficiencies
- Defined and compared service high availability methods
- Developed symmetric active/active replication prototypes:
  - HPC job and resource management service (PBS TORQUE)
  - HPC parallel file system metadata service (PVFS MDS)
  - Transparent replication software framework (prelim. prototype)

Limitations and Possible Future Work

- Development of a production-type symmetric active/active replication software infrastructure
- Development of production-type high availability support for HPC system services
- Extending the replication software framework to support active/standby and asymmetric active/active
- Extending the replication software framework to support non-IP communication networks
- Extending the lessons learned to other service-oriented or service-dependent architectures

Symmetric Active/Active High Availability for High-Performance Computing System Services: *Accomplishments and Limitations*

Christian Engelmann[1,2], Stephen L. Scott[1], Chokchai (Box) Leangsuksun[3], Xubin (Ben) He[4]

[1] Oak Ridge National Laboratory, Oak Ridge, USA
[2] The University of Reading, Reading, UK
[3] Louisiana Tech University, Ruston, USA
[4] Tennessee Tech University, Cookeville, USA