Minimizing Network Contention in InfiniBand Clusters with a QoS-Aware Data-Staging Framework

Raghunath Rajachandrasekar, Jai Jaswani, Hari Subramoni
&
Dhabaleswar K. (DK) Panda

Network-Based Computing Laboratory
Department of Computer Science and Engineering
The Ohio State University

Outline

- Motivation
- Background Concepts
- Design Alternatives
- Proposed Design
- Experimental Evaluation
- Summary and Future Work

Motivation

- Rapid growth in scale and complexity of HPC systems
- Abundance of resources – millions of cores, high-bandwidth networks, etc.
- Time-shared resources heavily under-utilized
- Space-sharing mechanisms face several challenges
- CPU, memory, interconnect and other resources shared amongst software components causing contention!

Competing for the Interconnect

- Several communication-intensive parallel applications
- Constant need to persist application data
  - Checkpoints, out-of-core data, log files, etc.
- Parallel filesystems – client/server model
  - clients, object-storage servers, metadata server
- Heavy data-movement over the standard interconnect
  - Contention with applications that use the same network fabric
- Need to multiplex I/O and application communication over the interconnect efficiently!

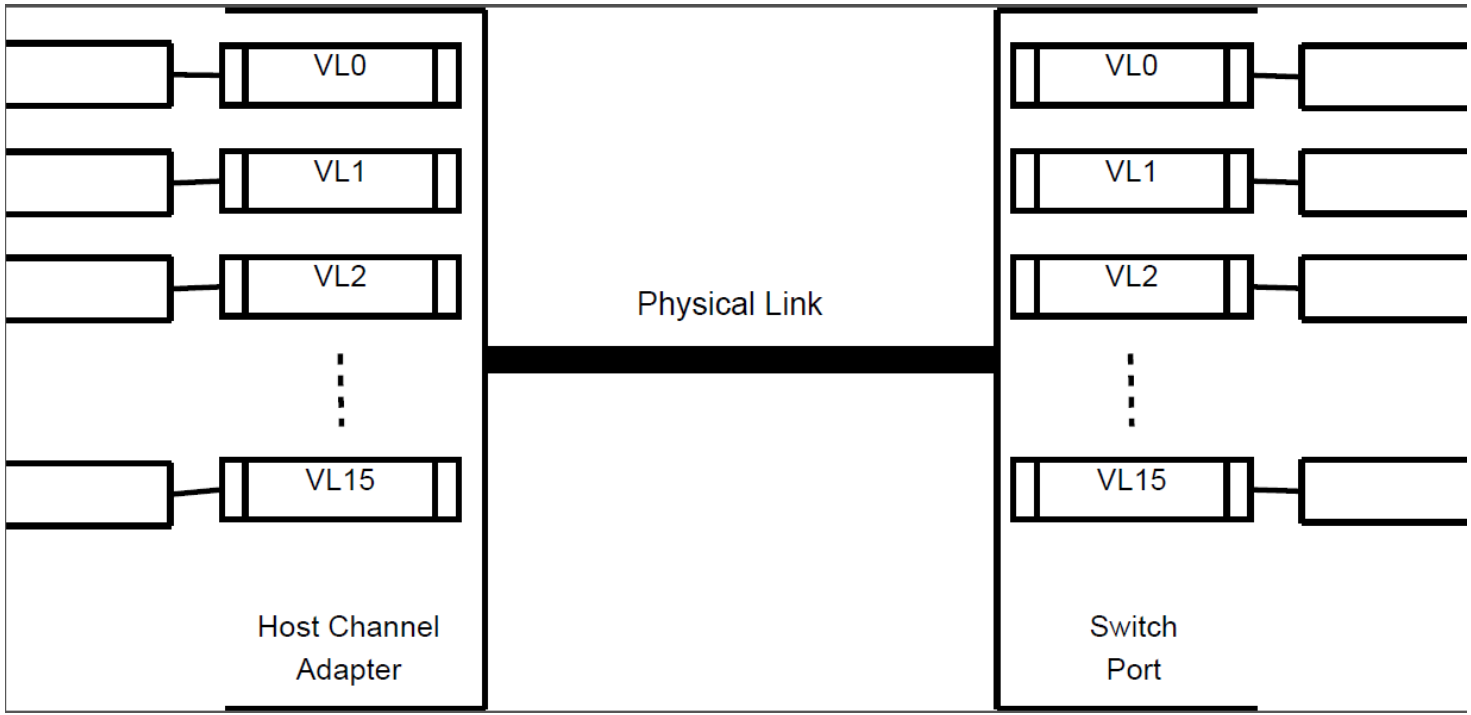
Problem Statement

- Can the performance impact of network space-sharing between parallel applications and filesytems be characterized?
- How can the Quality-of-Service capabilities of cutting-edge interconnects be leveraged to alleviate network contention?
- Can existing HPC middleware benefit from such a QoS-enabled parallel filesystem?

Quality-of-Service Support in InfiniBand

- InfiniBand (IB): a widely used high-speed interconnect
  - Over 41% of commodity clusters in Top500 ranking use IB
  - Provides send/recv and memory-based semantics
  - RDMA capability to allow remote memory access
- QoS to increase/limit priority of different data-flows
- Supports multiple Virtual Lanes (VL) to let different data-flows share the same physical link
- Exclusive buffering and flow-control resources for each VL
- Abstractions to configure priority
  - Service Level (SL) at the switch-level
  - Traffic Class (TClass) at the router-level

Quality-of-Service Support in InfiniBand



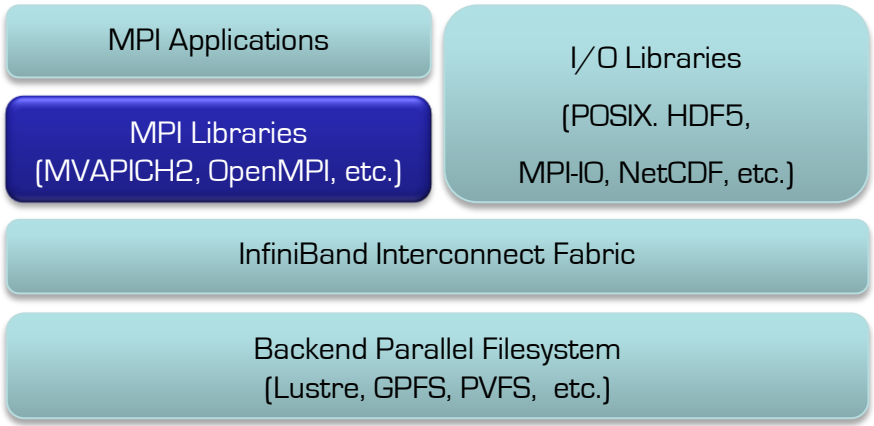
Configuring the IB Subnet

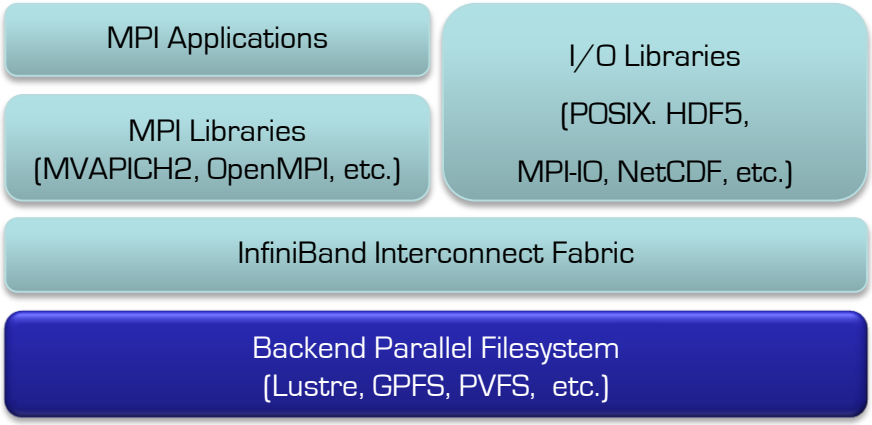
- OpenSM used as the subnet manager
  - SM initiates and configures IB subnet
  - SM Agent (SMA) deployed on every device port
  - SMs and SMAs communicate with SM Packets (SMPs)
  - VL15 used exclusively for management traffic
- **opensm.conf**: SL2VL mapping and VL arbitration table

Configuring OpenSM

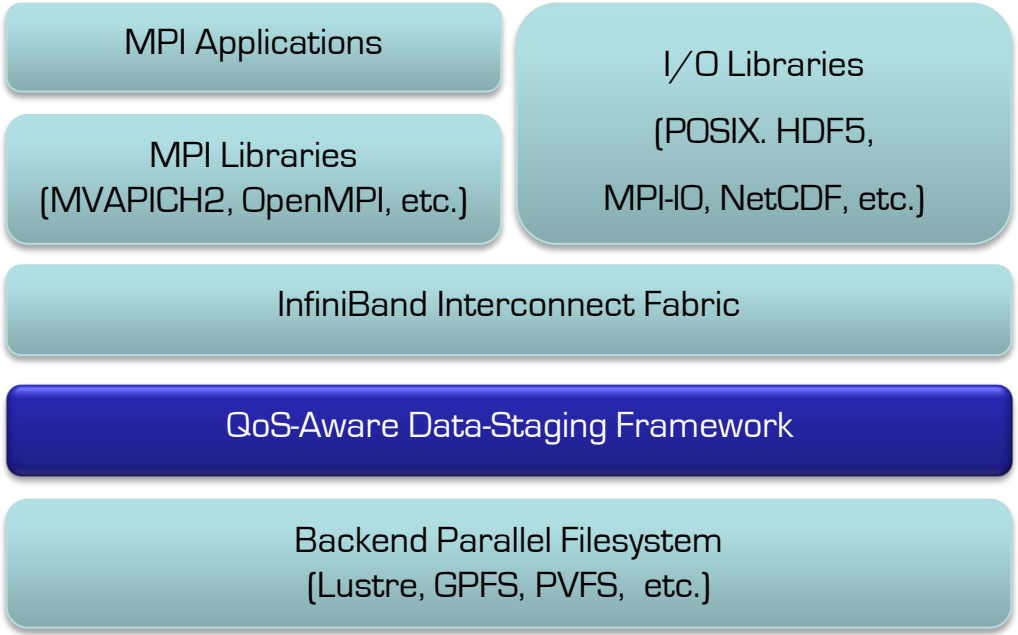
```
qos_ca_high_limit
    255
qos_ca_vlarb_high
    0:40, 1:0, 2:0, 3:0, 4:0, 5:0, 6:0, 7:0, 8:0, 9:0, 10:0, 11:0, 12:0, 13:0, 14:0
qos_ca_vlarb_low
    0:0, 1:2, 2:4, 3:8, 4:16, 5:32, 6:64, 7:128
qos_ca_sl2vl
    0,1,2,3,4,5,6,7,15,15,15,15,15,15,15,15
```

Design Alternatives

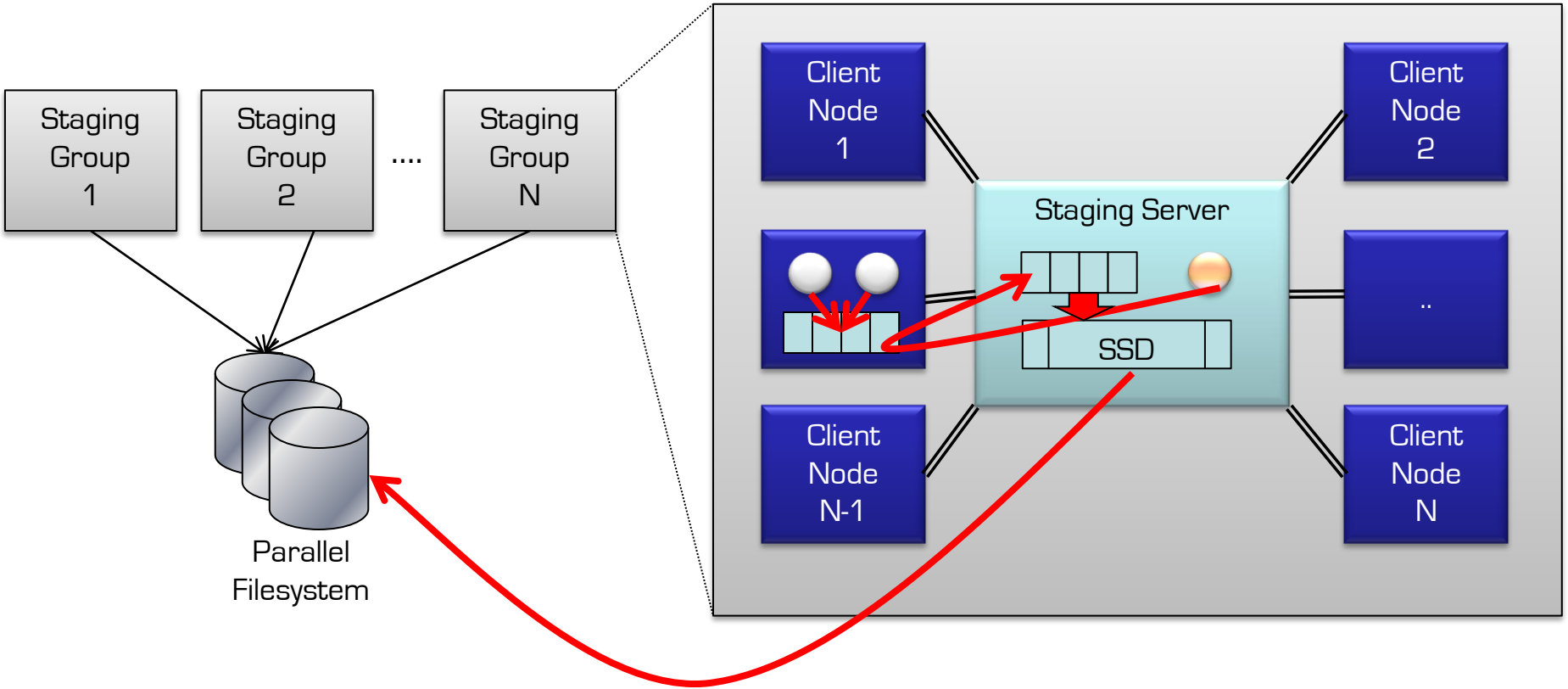




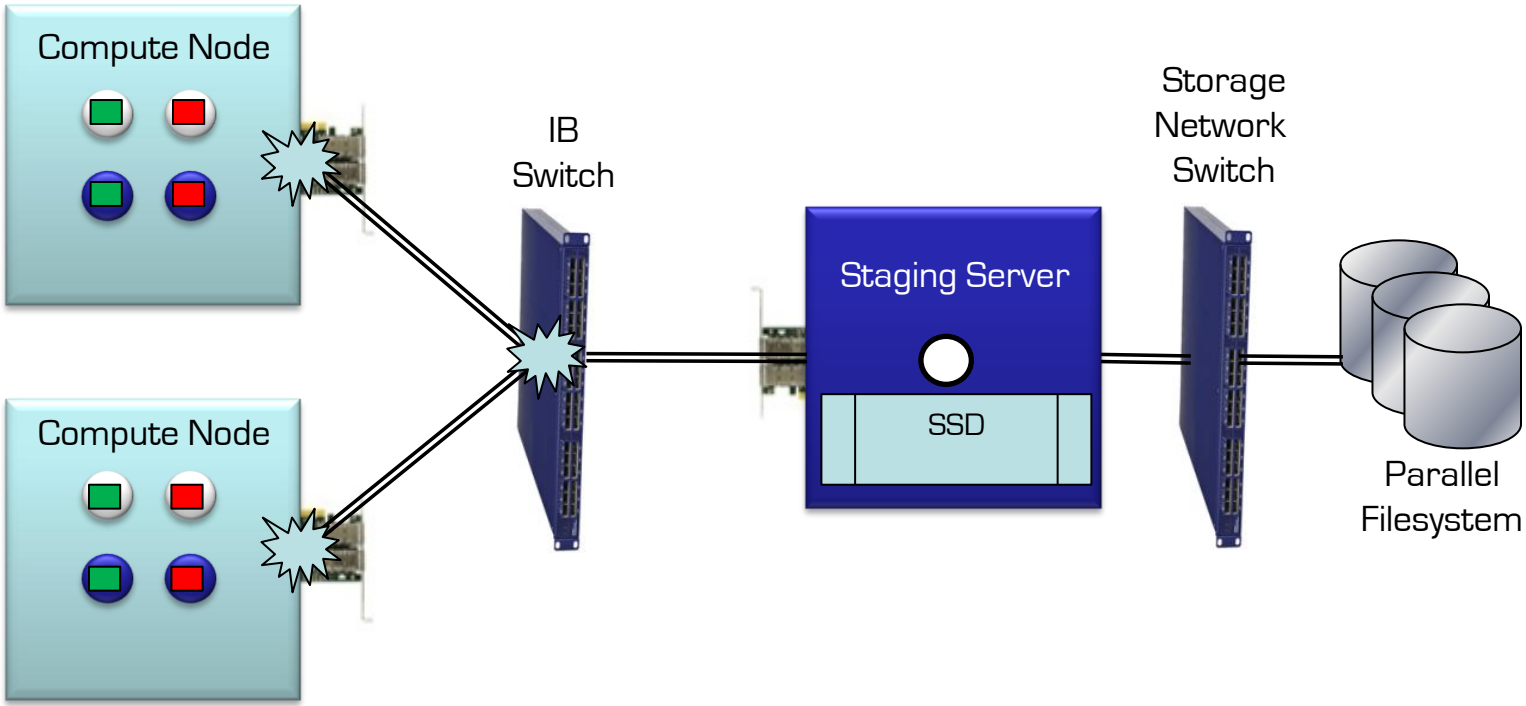
Design Alternatives



Hierarchical Data-Staging



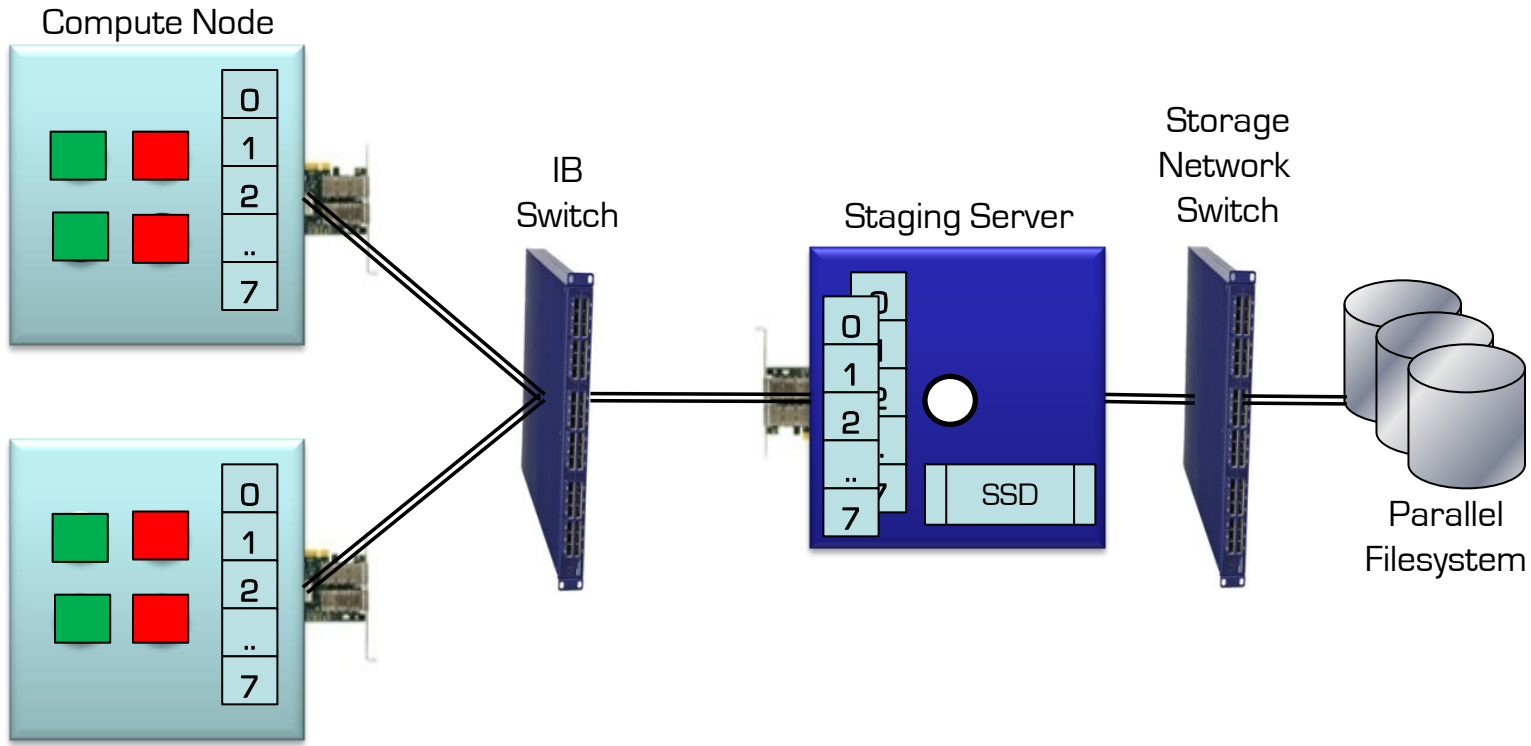
Hierarchical Data-Staging



Enabling QoS within the Filesystem

- Goals
  - To give highest priority to the parallel application
  - To overlap I/O operations with application compute cycles
  - To allow dynamic priority selection
  - To allow per-file QoS
- Persistent QPs in the data-staging architecture
- No end-points amongst client nodes

Enabling QoS within the Filesystem



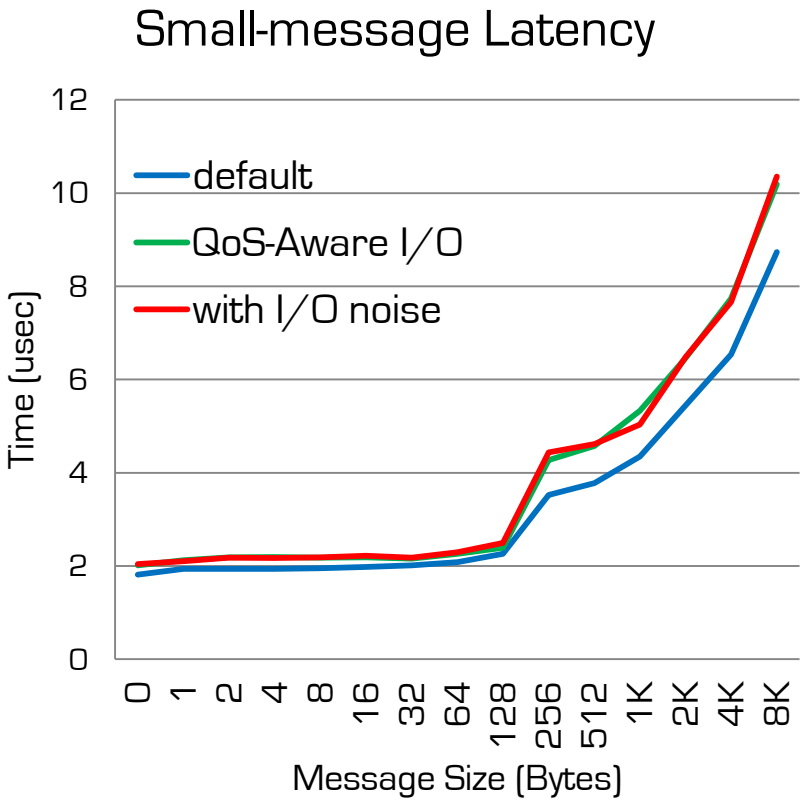
Experimental Evaluation

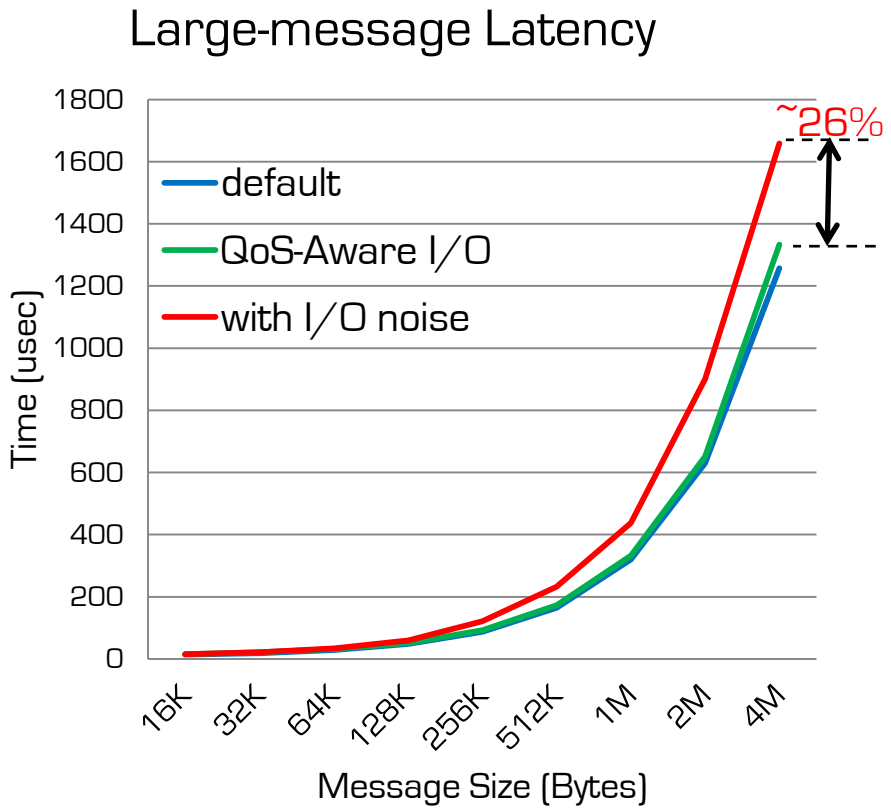
- Micro-benchmark and application-level evaluation
- 160-node Linux cluster running RHEL6 (2.6.32 kernel)
- Compute nodes
  - 8-core dual-socket Intel Xeon CPU
  - 12GB physical memory
  - InfiniBand QDR ConnectX-2 HCAs
- Staging Nodes
  - 24GB physical memory
  - 300GB OCZ VeloDrive PCIe SSD

Experimental Evaluation

- OSU Micro-Benchmark suite used for latency and bandwidth experiments
- I/O noise generated with GNU 'dd' utility
- Direct I/O mode used to avoid kernel page cache buffering
- MPI processes and I/O threads mapped to different CPU cores

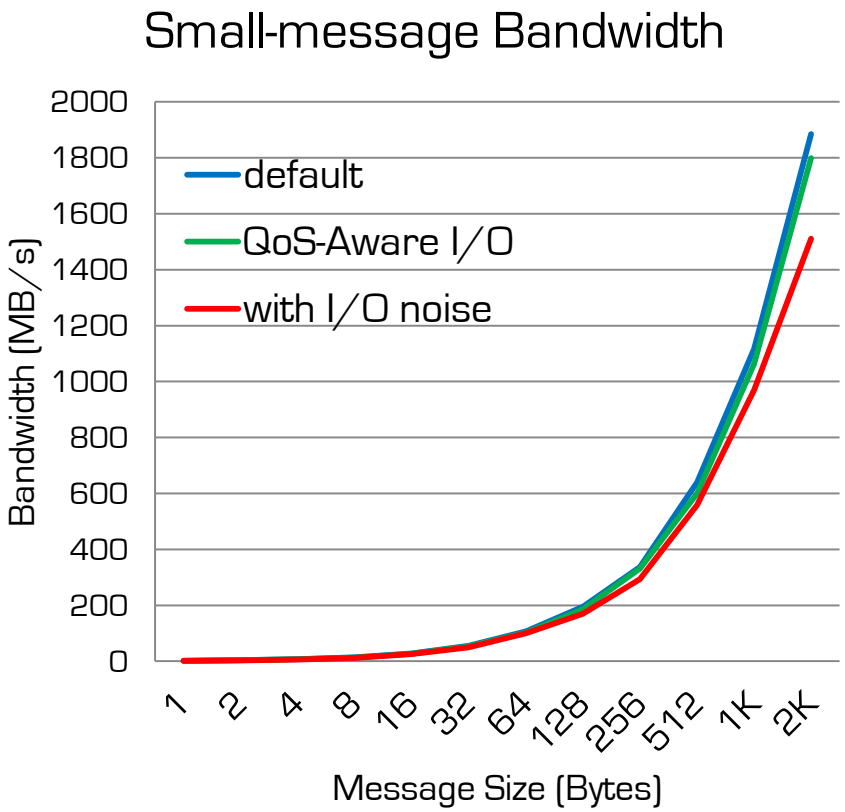
Impact on MPI Pt-to-Pt Latency

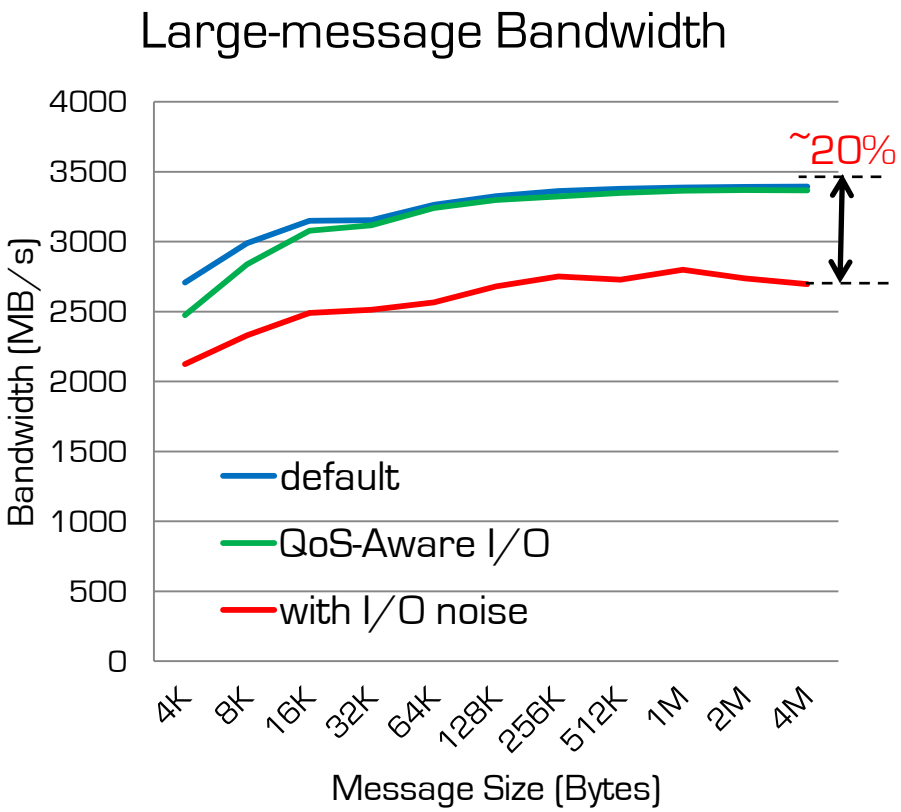




4MB message latency reduced by 320 usec

Impact on MPI Pt-to-Pt Bandwidth

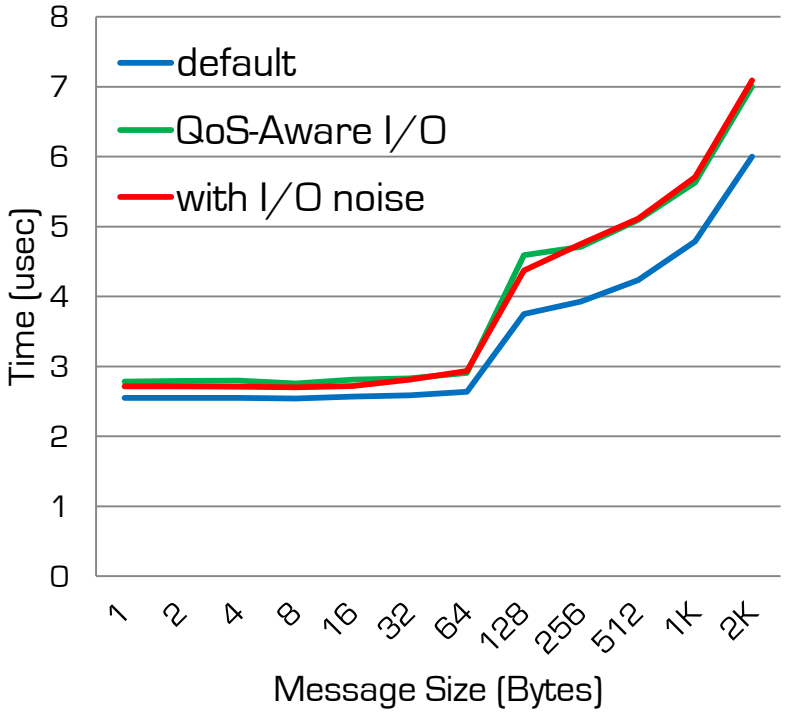




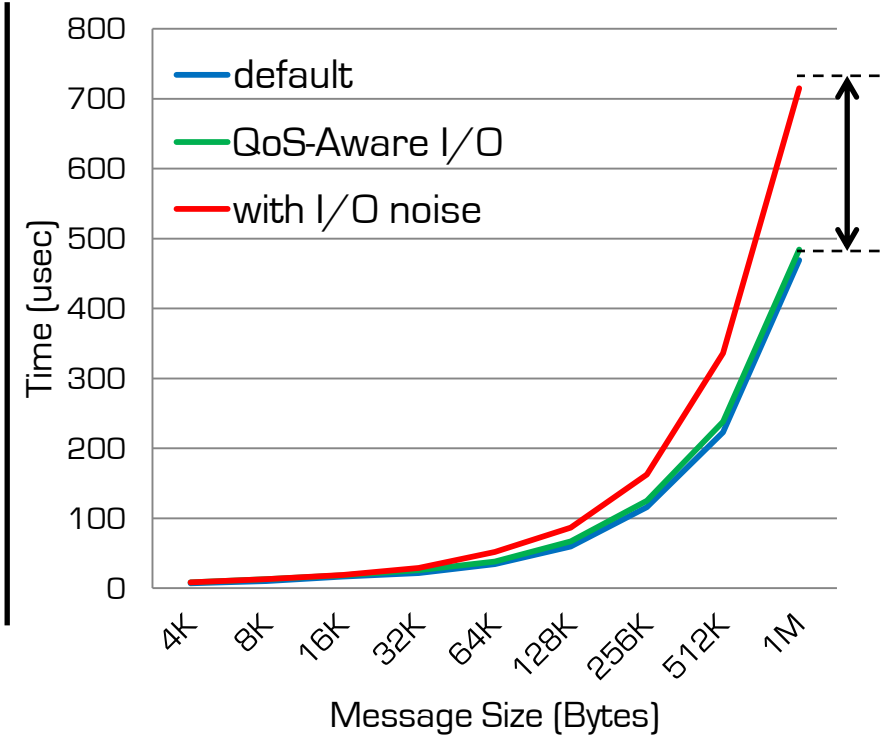
4MB message bandwidth increased by 674 MB/s

Impact on MPI Collectives

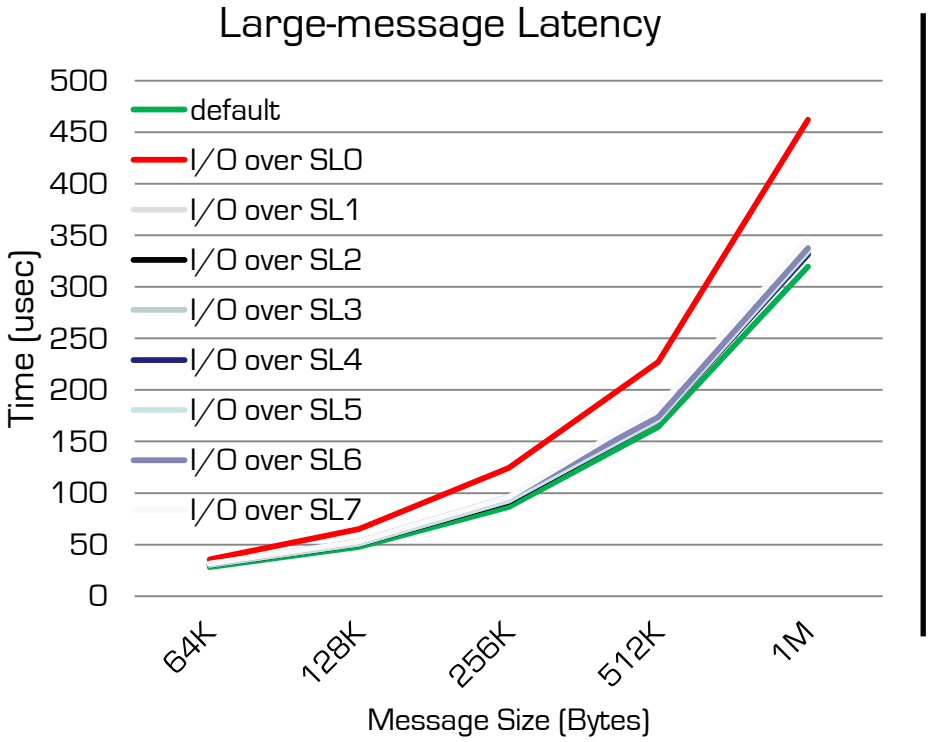
Small-message AlltoAll Latency

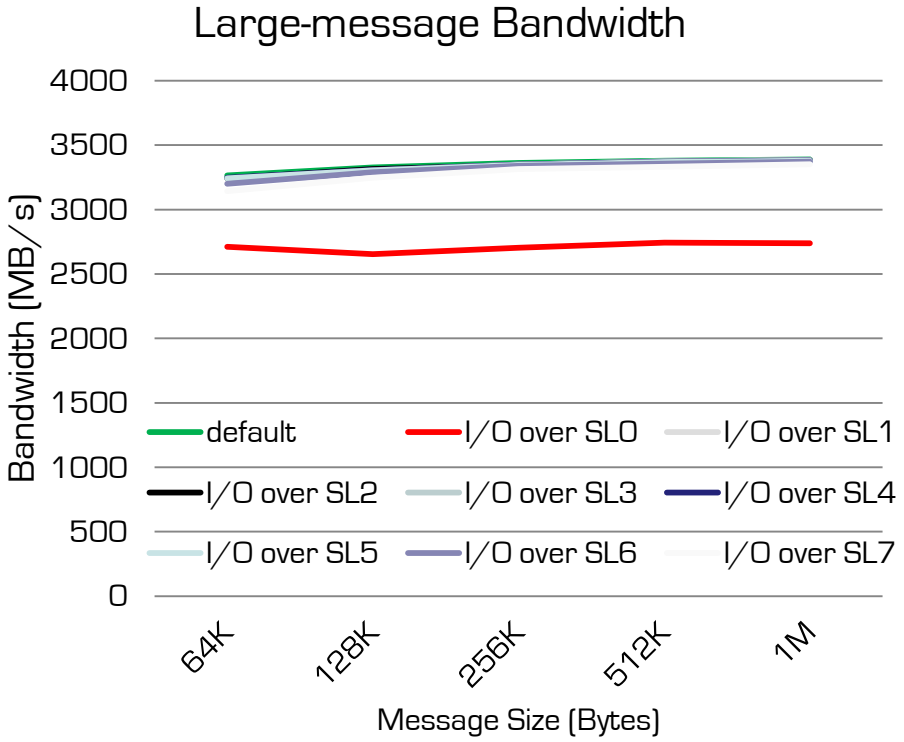


Large-message AlltoAll Latency

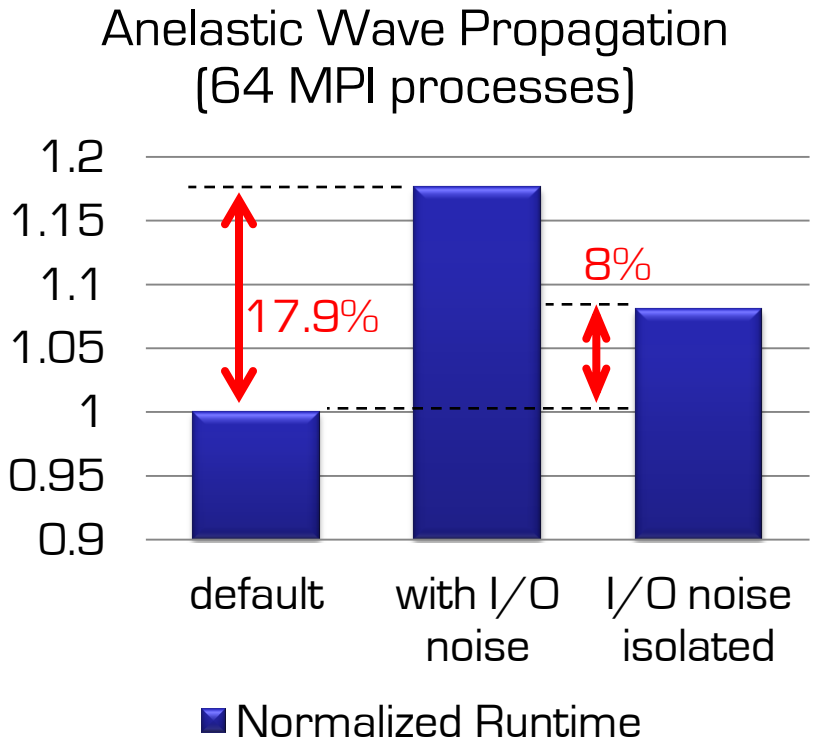


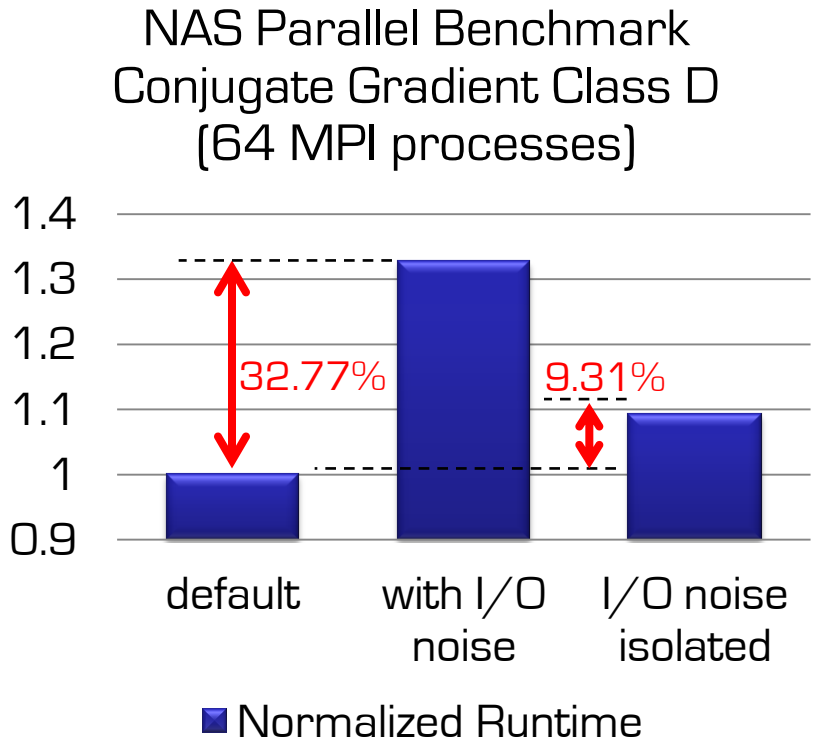
Impact of SL Weights





Impact on Applications





Summary and Future Work

- Designed a QoS-Aware filesystem to reduce n/w contention
- Studied the impact of the design with low-level benchmarks and communication-intensive applications

- Future work
  - Large-scale studies with different classes of I/O and MPI workloads
  - Study the memory-overheads induced by the design
  - Impact of solution on I/O libraries like MPI-IO, NetCDF, HDF5, etc.
  - Identify balanced thresholds to reduce both MPI and I/O contention

Thank you!

Network-Based Computing Laboratory

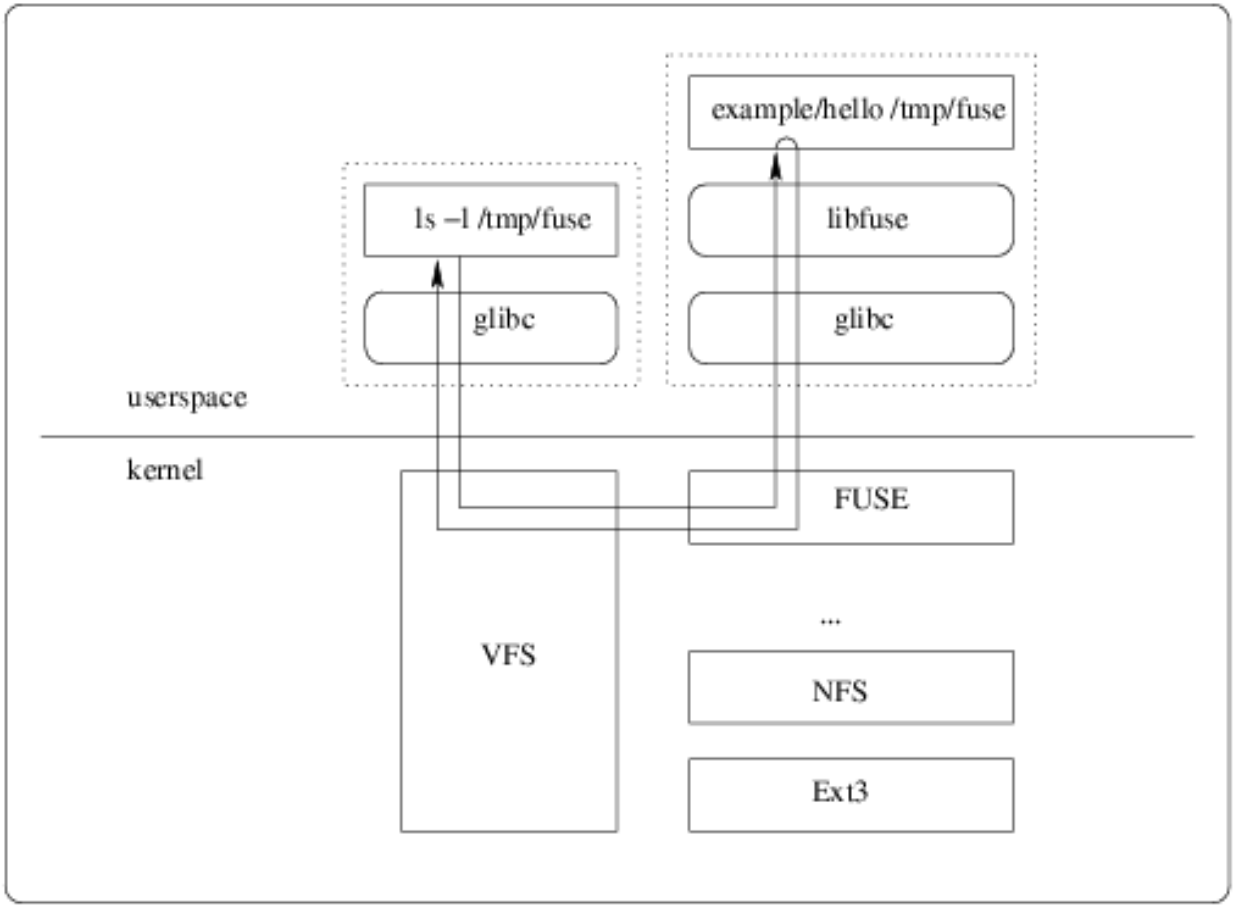
http://nowlab.cse.ohio-state.edu

http://cse.osu.edu/~rajachan

rajachan@cse.ohio-state.edu

Backup Slides

How FUSE works?



Related Work

- QoS-Aware disk scheduling (Aref et. al., '02; Bruno et al., '99)
- Machine-learning based I/O throughput QoS (Zhang et al., '11)
- Differential services using proxy servers (Xu et al., '10)

- IB QoS to avoid Head-of-Line blocking (Subramoni et al., '10)
- Optimal configuration for VL arbitration tables (Alfaro et al., 02)
- Formal model to manage VL arbitration tables (Alfaro et al., '07)
- QoS over advanced switching networks (Martinez et al., '08)