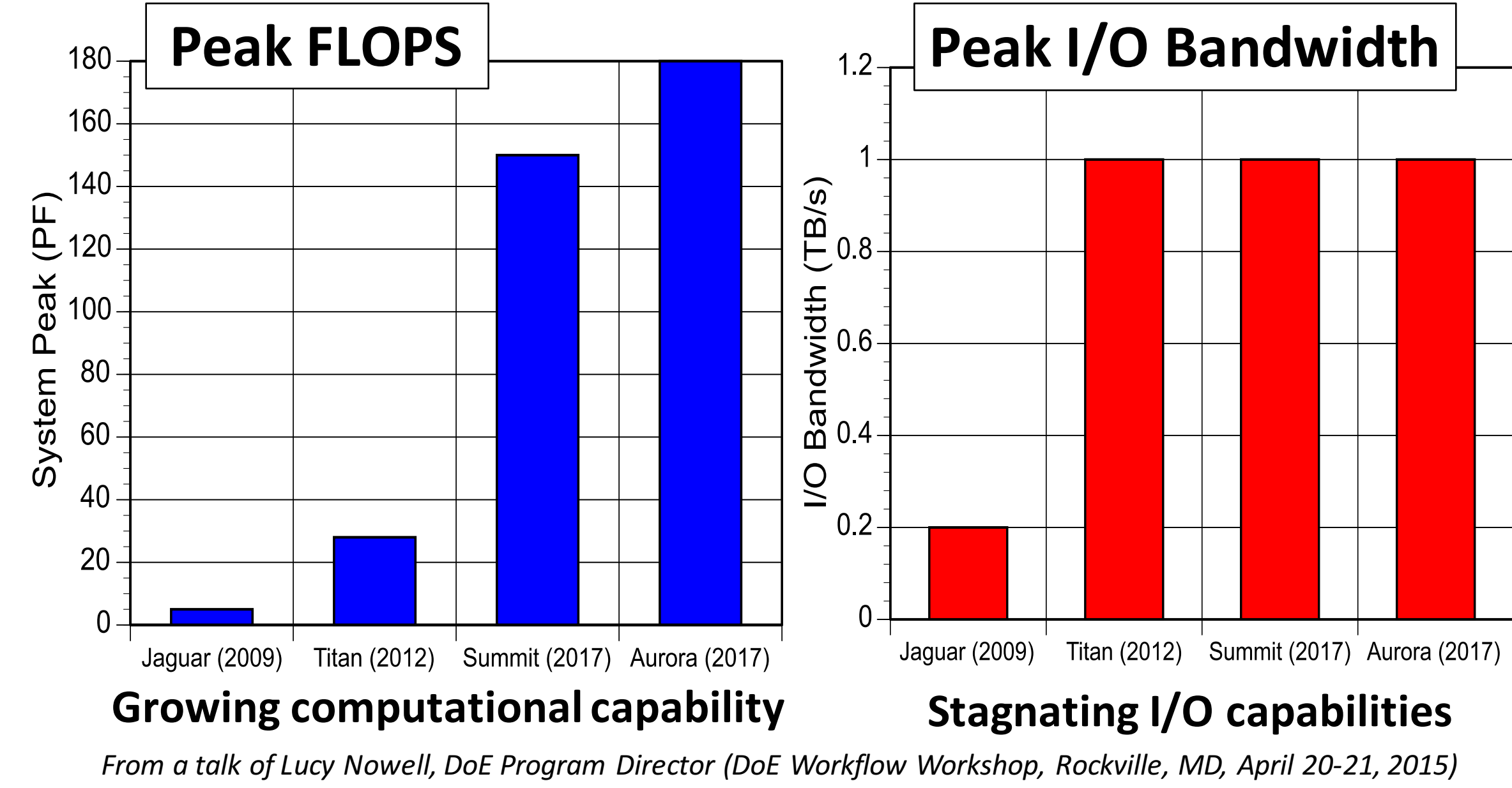
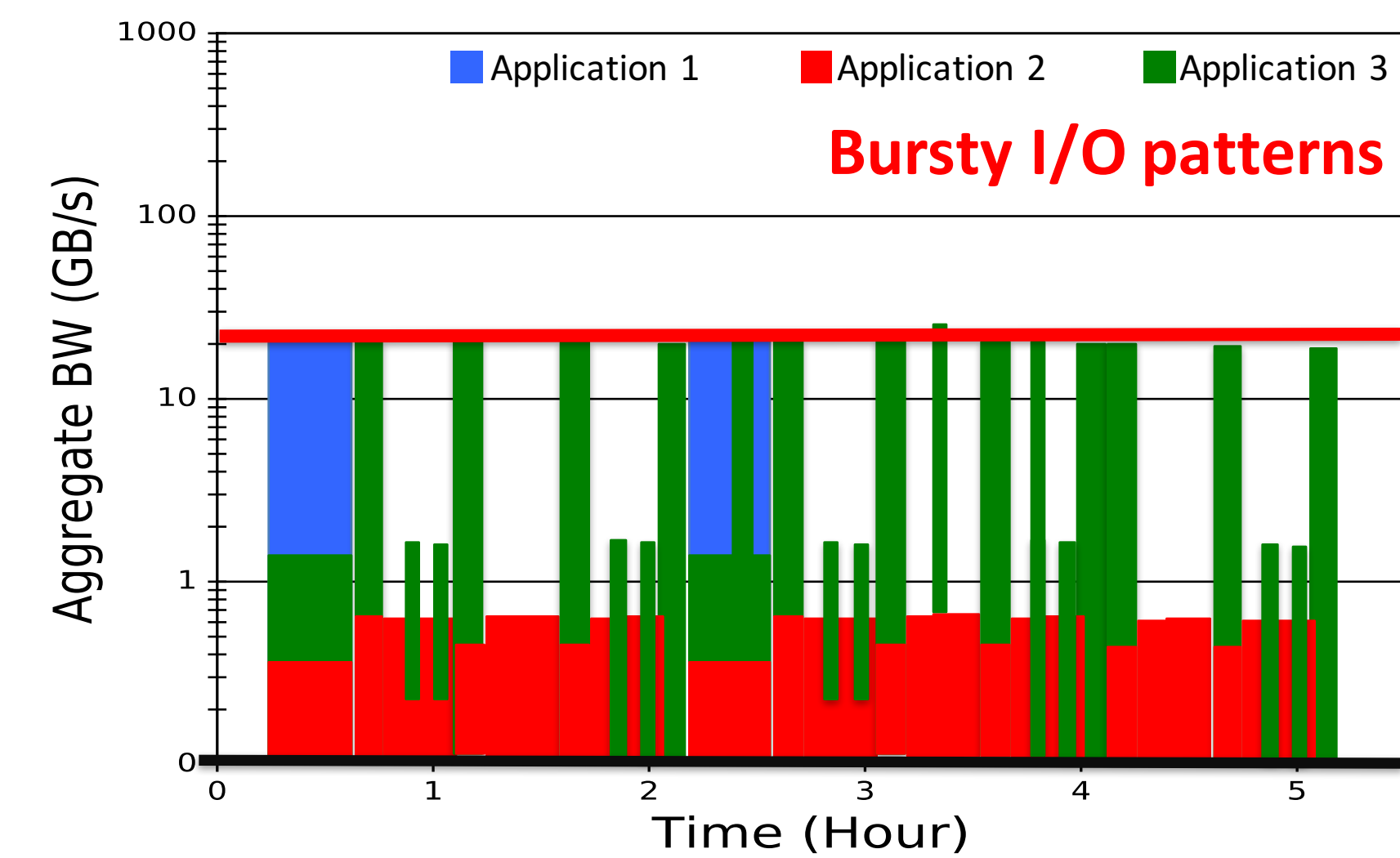




Motivation

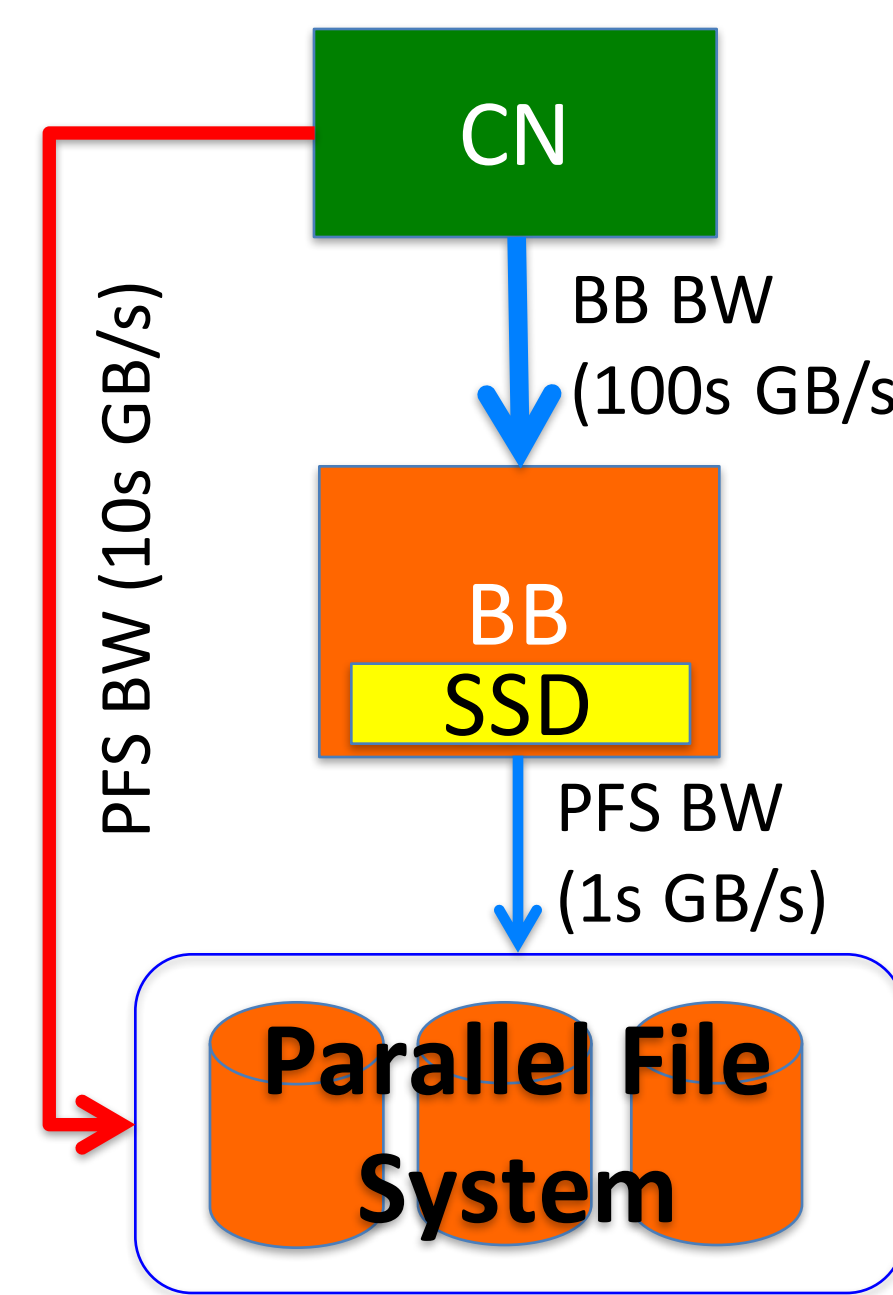


- The FLOPS vs. I/O imbalance can cause I/O contention
- Burst buffers (BB) and smart staging postpone contention
- Parallel file systems (PFSes) remain the main bottleneck

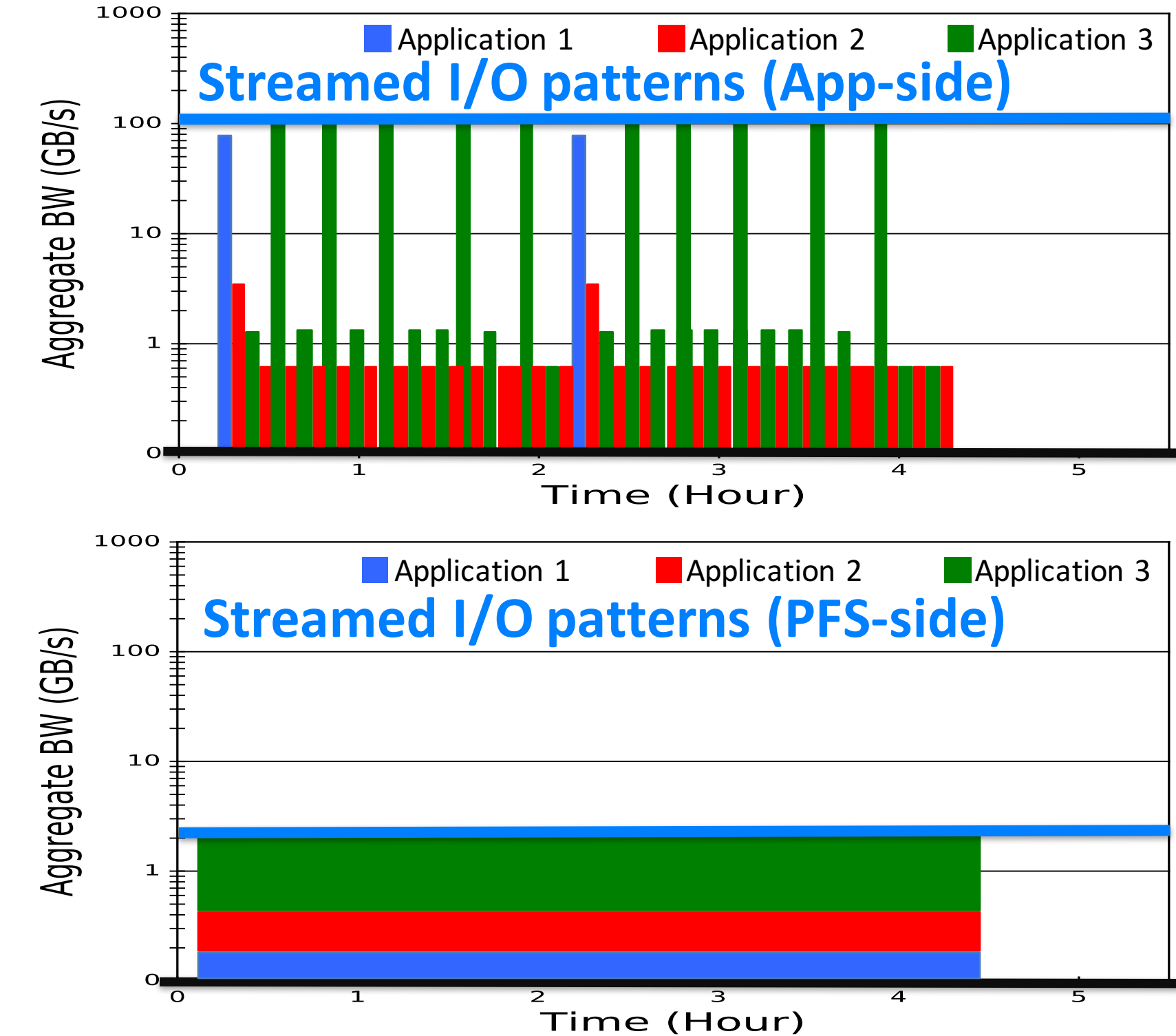


Based on: Liu, N, Cope, J, Carns, P, Carothers, C, Ross, R, Grider, G, Crume, A, Maltzahn, C. "On the Role of Burst Buffers in Leadership-class Storage Systems" MSST/SNAPI 2012

- Without BBs, the bursty I/O goes straight to the PFS
- With BBs, the application sees much higher I/O BWs
- With BBs, the I/O to the PFS is a constant stream
- PFS is now provisioned for avg. I/O load (not max load)



We propose a novel, I/O-aware batch scheduling algorithm that can manage I/O contention at the PFS level [1]



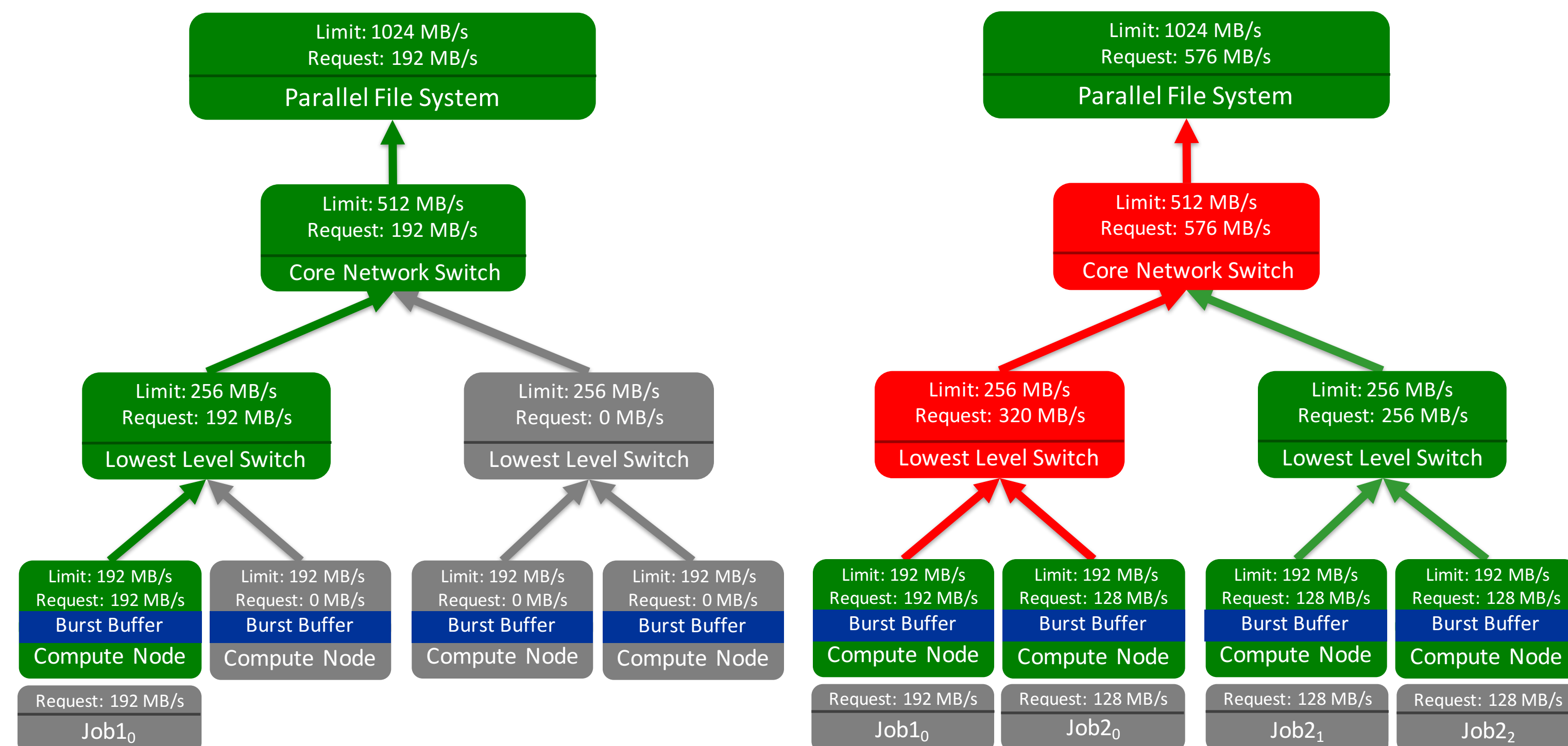
Making the Scheduler I/O-aware

- I/O-aware means using I/O as a key constraint when scheduling jobs
 - Jobs are delayed if they would cause contention in the I/O subsystem
- I/O-aware schedulers keep track of I/O allocations and predict potential I/O contention using both the I/O subsystem and I/O contention models

Flux framework's global system view and resource description language enable the use of I/O subsystem and contention models in a scheduler

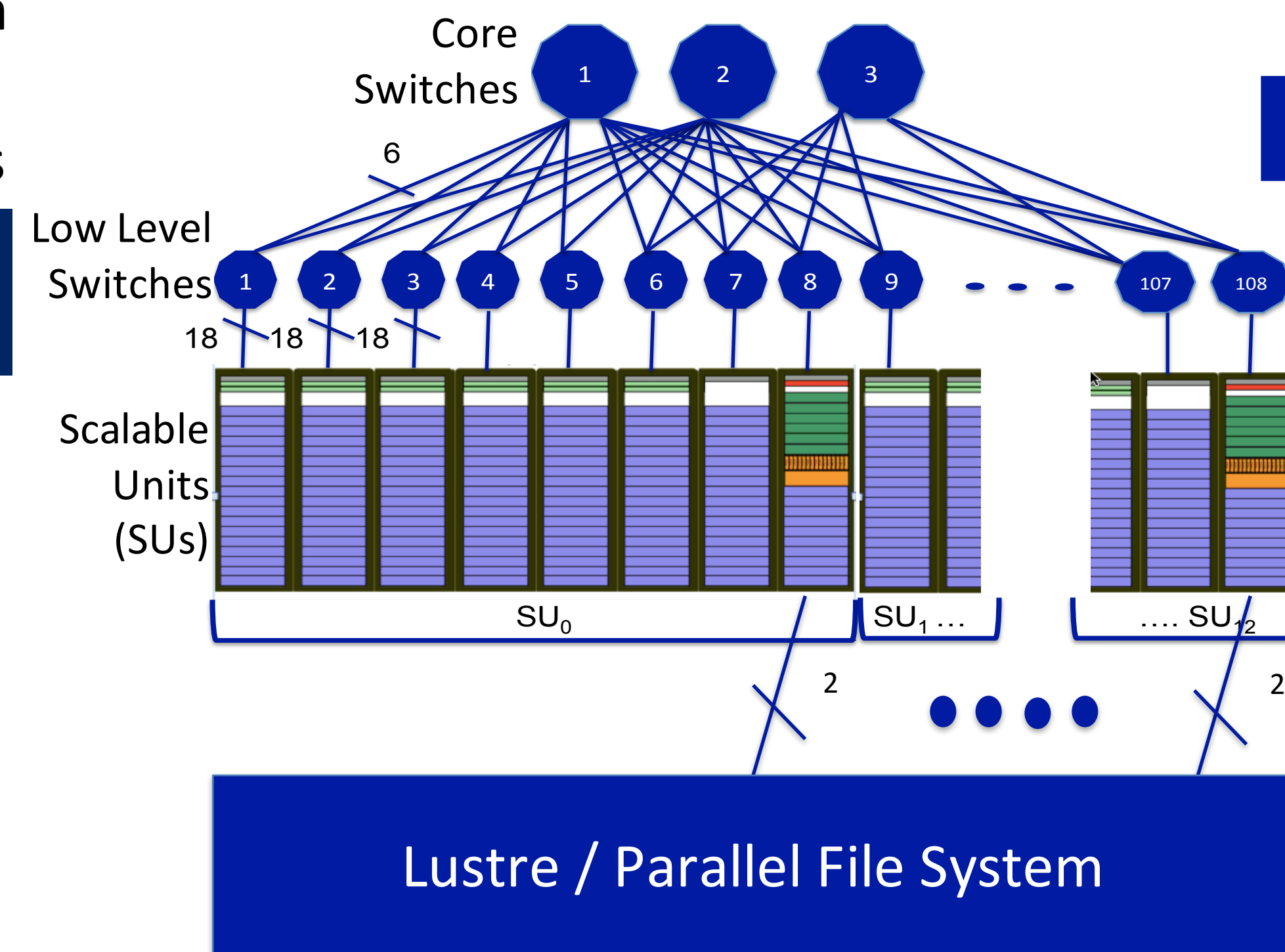
I/O-aware Scheduling Scenarios

- Job 1, by itself, can be scheduled on the system
- Job 2 requests too much BW and can cause contention with Job 1
 - Job 2 is delayed until more BW is available (i.e., when Job 1 completes)

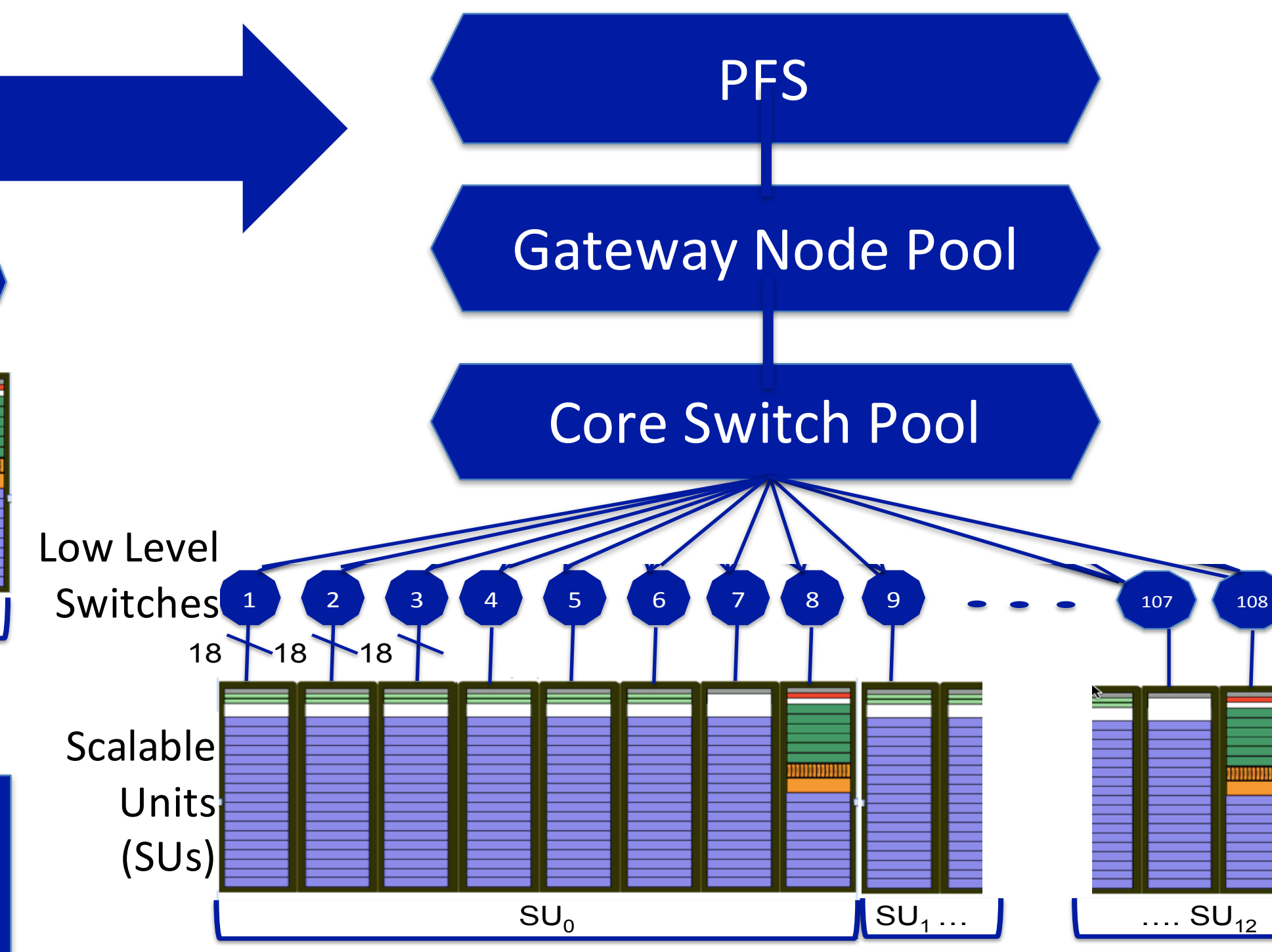


Modeling the I/O Subsystem

From a complex resource graph...



To a simple resource tree



- Modeled system:
 - A 1944 node/12 SU cluster
 - I/O routed round-robin across core switches and gateway nodes

- Key simplifications:
 - Merge core switches and gateway nodes
 - Leverage BBs to model I/O as a constant stream rather than variable bursts

Modeling the I/O Contention

- Two scenarios are modeled:
 - All jobs get their requested BW and extra BW remains
 - Smaller I/O requests are satisfied, larger requests contend for BW; no extra BW remains
- Contention occurs in case two and is modeling using an *Interference Factor* defined in [2]

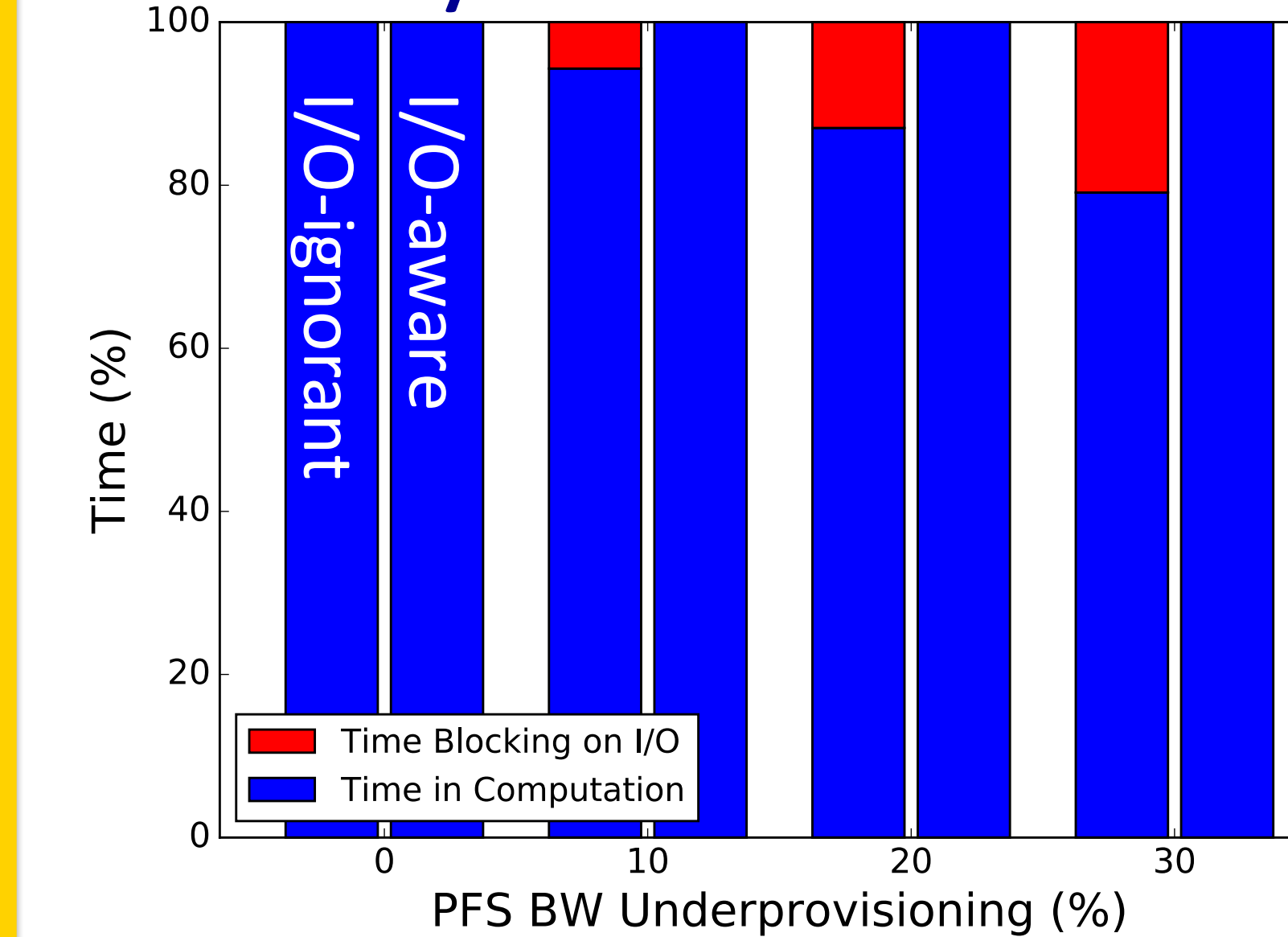
Test Configuration

- 2,500 jobs sampled from LLNL's workloads
 - Constant job I/O rate of 18 MB/s
- 3,888 node system model from LLNL's CTS-1
- I/O-aware/ignorant versions of EASY backfilling
 - Emulated using the Flux framework emulator
- Four levels of PFS provisioning
 - 0% (70GB/s), 10% (63 GB/s), 20% (56 GB/s), and 30% (49 GB/s)
 - Simulates a small PFS or a reservation of BW for external sources of I/O

Critical Questions

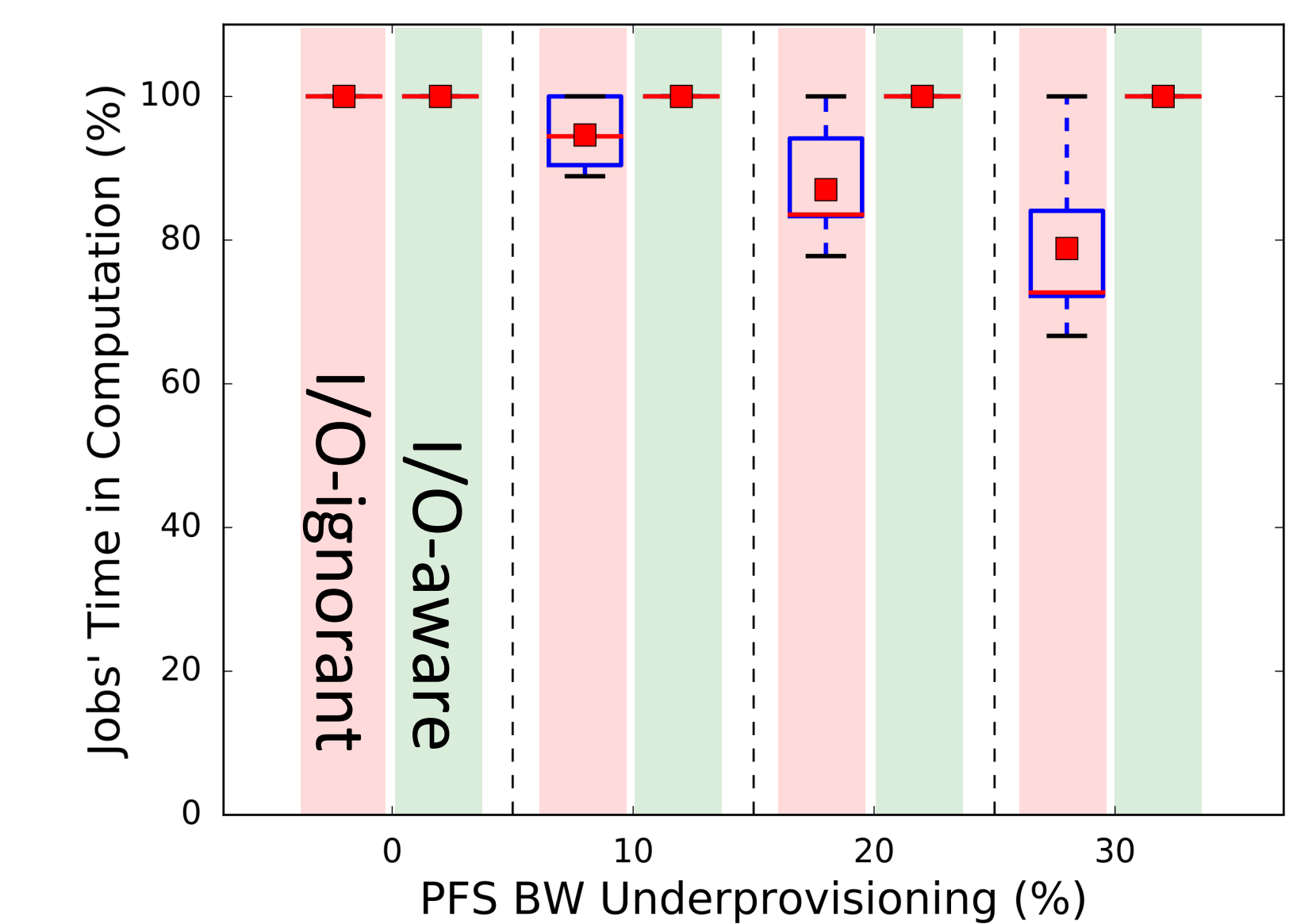
- Does I/O-aware scheduling:
 - Impact percentage of time that nodes spend in computation?
 - Impact the variability of each individual job's performance?
 - Affect the time to make a scheduling decision?
- What is the trade-off between system efficiency and turnaround time?

Total System Performance



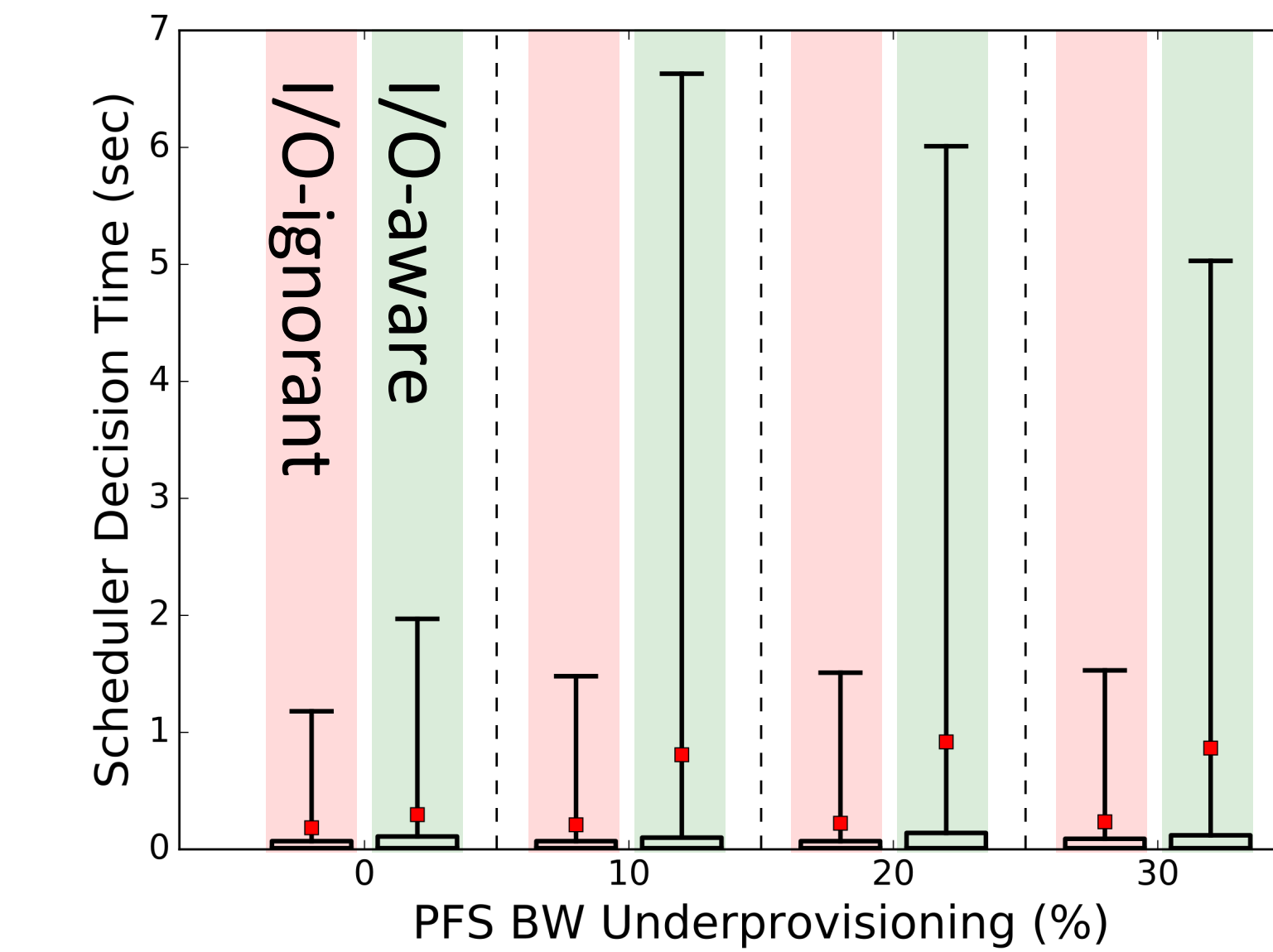
I/O-aware scheduling keeps allocated nodes in computation 100% of the time

Individual Job Performance



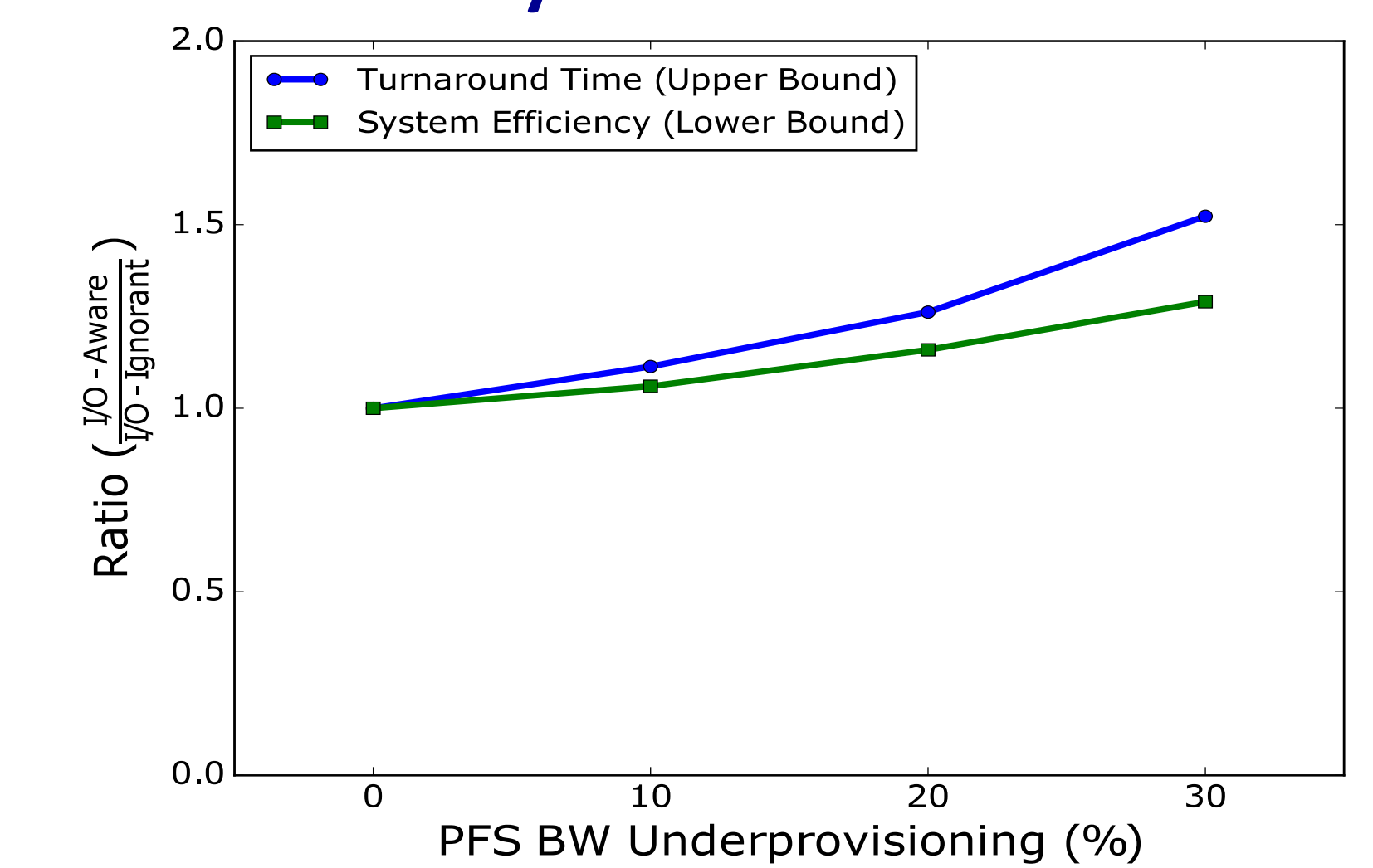
I/O-aware scheduling eliminates variability in job performance due to I/O contention

Scheduler Decision Time



I/O-aware scheduling is still viable for online batch job scheduling

Efficiency vs. Turnaround



I/O-aware scheduling increases science (>1.29x) in exchange for increasing turnaround time (<1.52x)

References: [1] S. Herbein, D. H. Ahn, D. Lipari, T. R. Scogland, M. Stearman, M. Grondona, J. Garlick, B. Springmeyer, and M. Taufer, "Scalable I/O-Aware Job Scheduling for Burst Buffer Enabled HPC Clusters," in Proc. of the 25th International Symposium on High-Performance Parallel and Distributed Computing (HPDC), 2016.
[2] M. Dorier, G. Antoniu, R. Ross, D. Kimpe, and S. Ibrahim. CALCIOM: Mitigating I/O Interference in HPC Systems Through Cross-Application Coordination. In Proc. of the 2014 IEEE 28th International Parallel and Distributed Processing Symposium (IPDPS), May 2014.