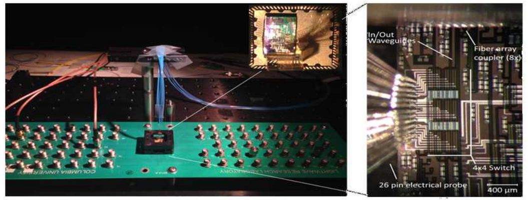
SAND2020-9400C

Opportunities and limitations of Quality-of-Service in Message Passing applications on adaptively routed Dragonfly and Fat Tree networks



PRESENTED BY

Jeremiah Wilke, Sandia National Labs, Livermore, CA

Contributors: Joseph P. Kenny

Cluster 2020, Virtual Kobe

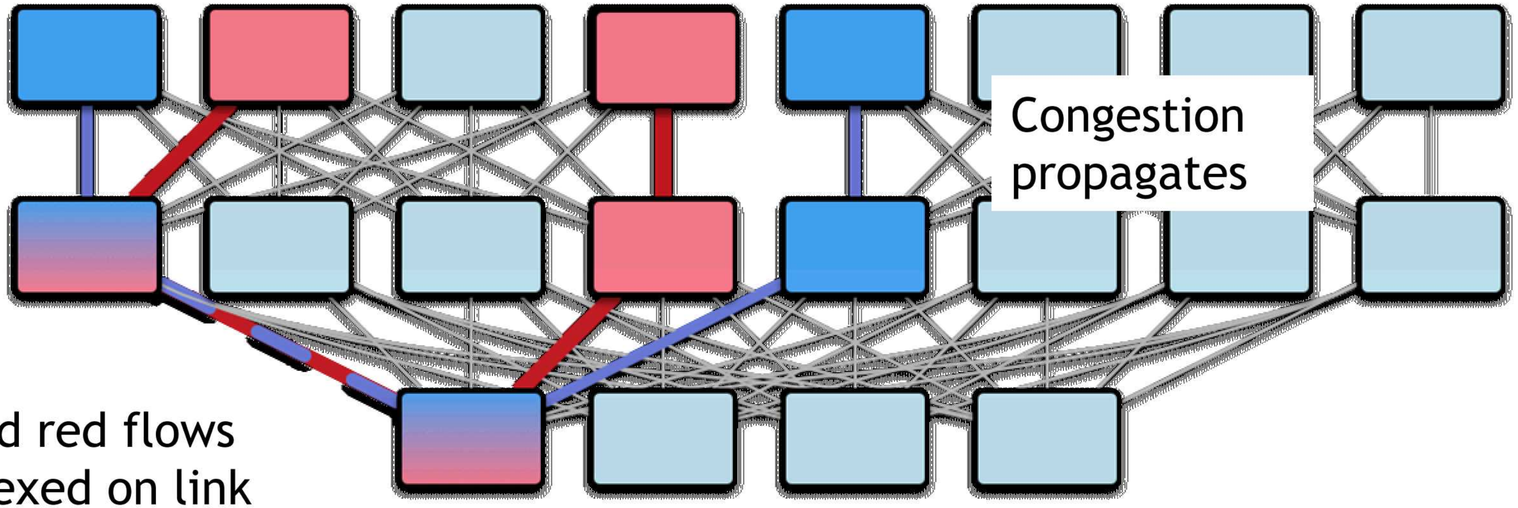
Sandia National Laborat
laboratory managed and
Technology & Engineeri
LLC, a wholly owned su
International Inc., for th
Energy's National
Administration under co

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Quality-of-Service (QoS) mitigates head-of-line blocking in congested networks

Congestion blocks flow

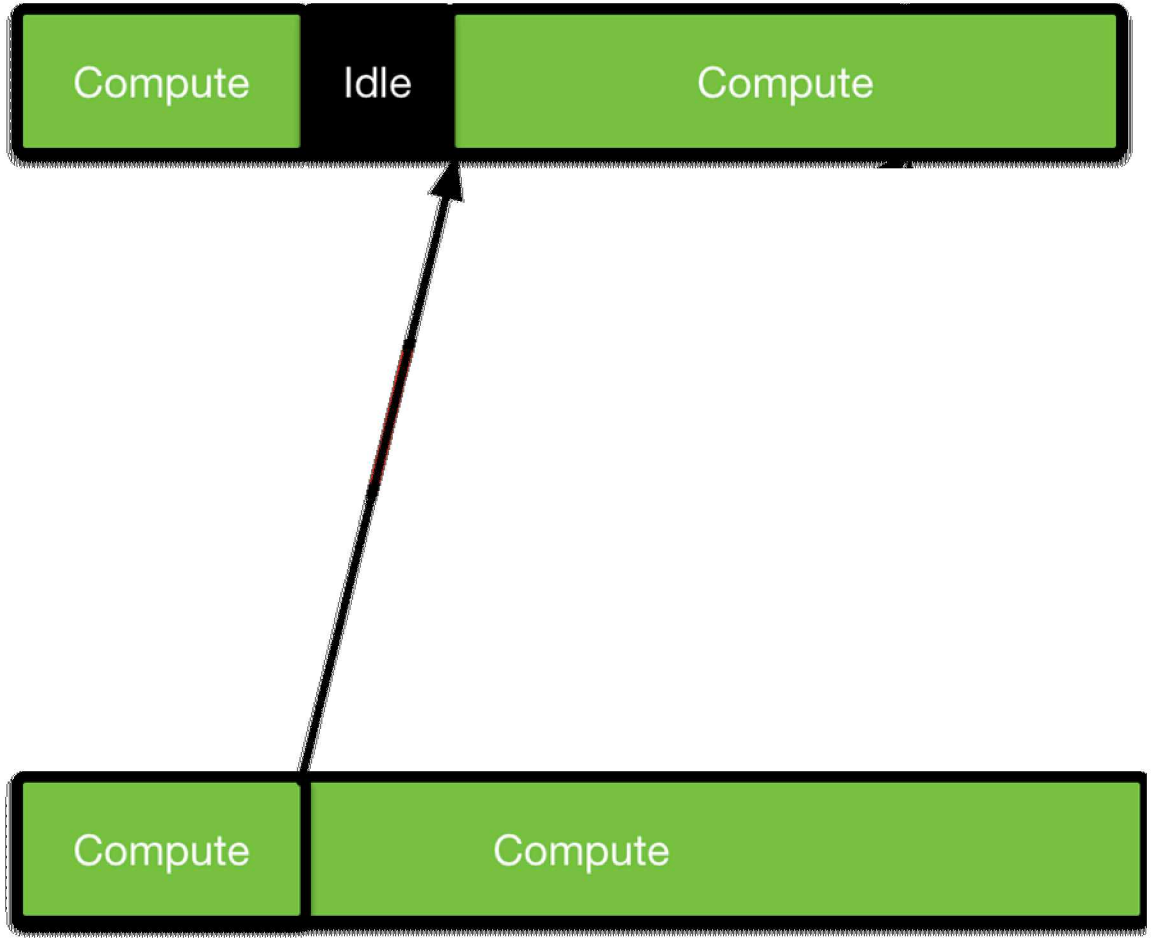
Split physical link into “virtual” lanes that allow red flow to make progress



Blue and red flows multiplexed on link

Both flows now blocked

Tightly coupled processes in high-performance computing want to avoid ***unexpected*** delays



Short delay in communication between processes leads to small idle time

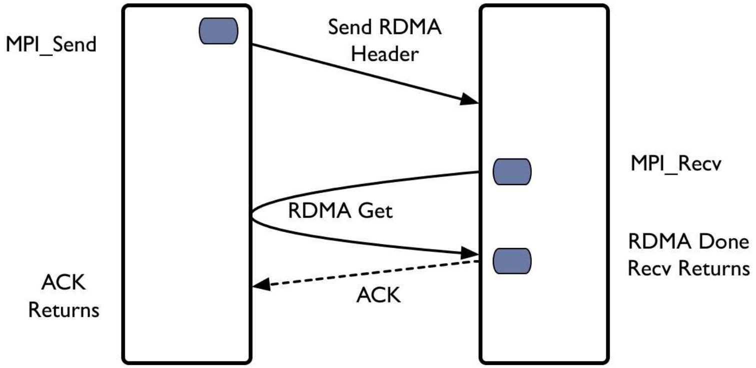
Long delay in communication between processes leads to large idle time

Long delay could be due to network contention

Prioritizing traffic reduces idle time, has little performance impact on low-priority traffic

Several types of traffic have more severe performance consequences when encountering network contention

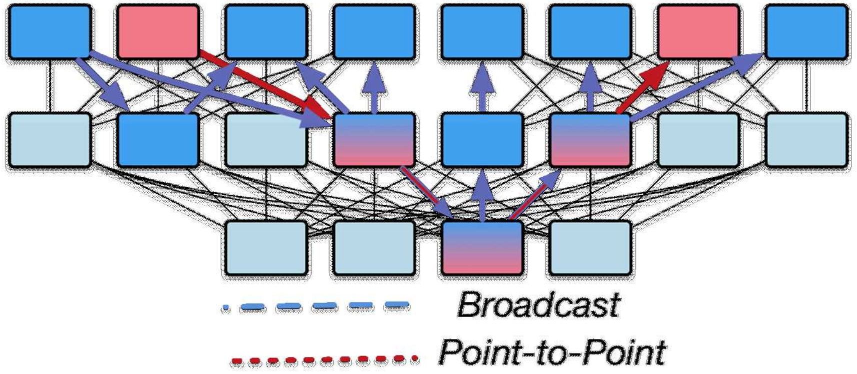
Pt-2-Pt Rendezvous Protocol



Control messages (header,ack) should not be delayed by RDMA Get bulk transfers

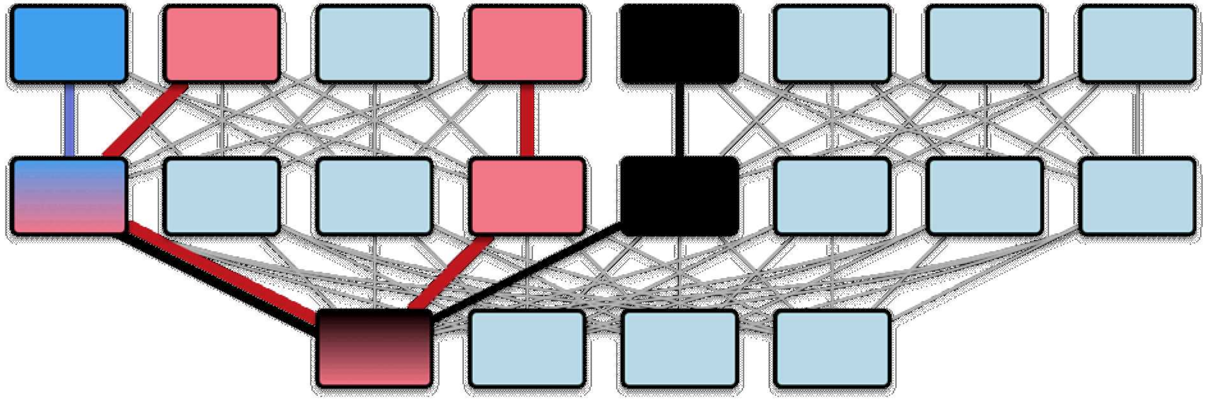
Latency-sensitive, small messages should not be delayed by bandwidth-hungry, large messages

Entire collective should not be delayed by single point-to-point

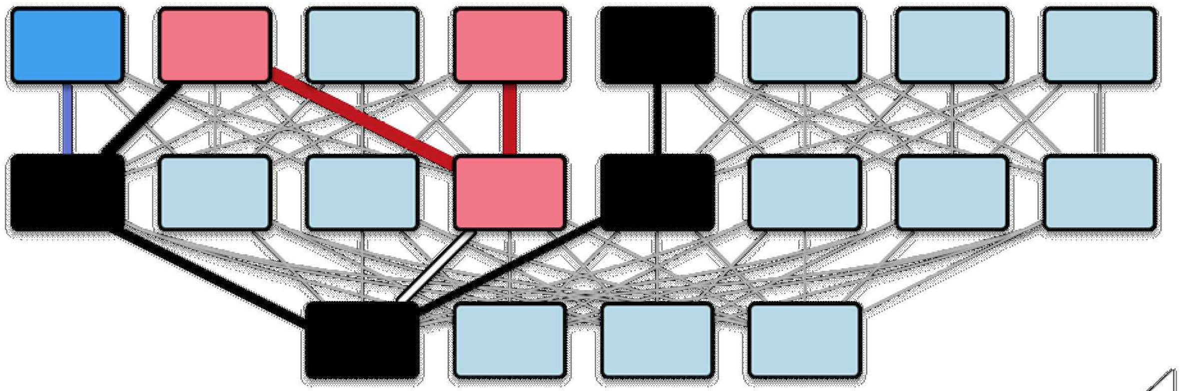


Adaptive routing on topology avoids congestion, QoS mitigates negative effects when congestion occurs

With heavy traffic on network, avoidance may be impossible! Need mitigation.
(e.g. minimal routing is best for uniform random traffic)

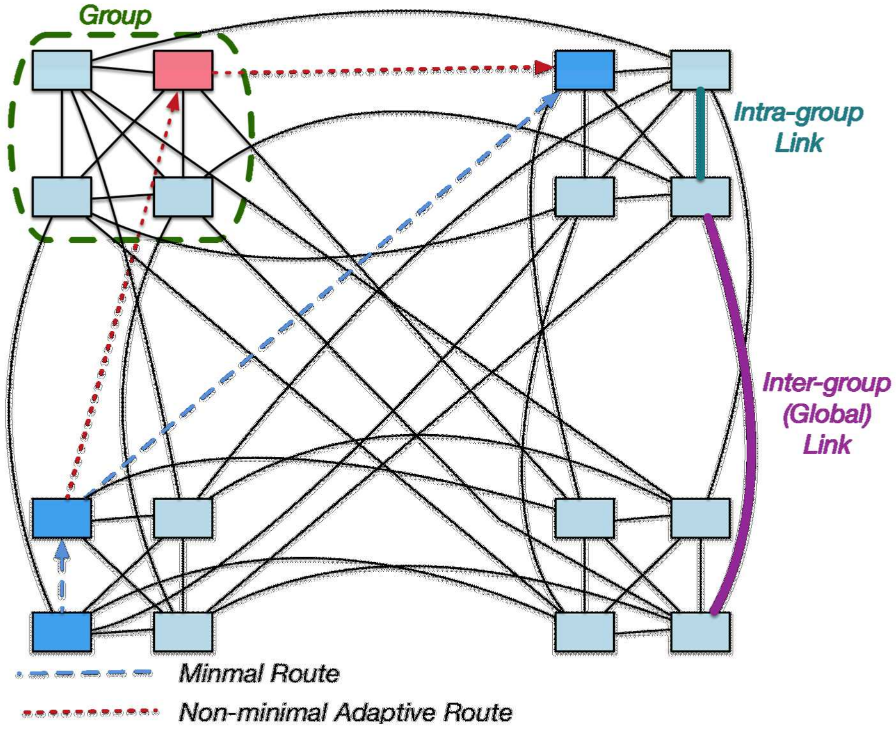


QoS prevents starvation in the presence of congestion

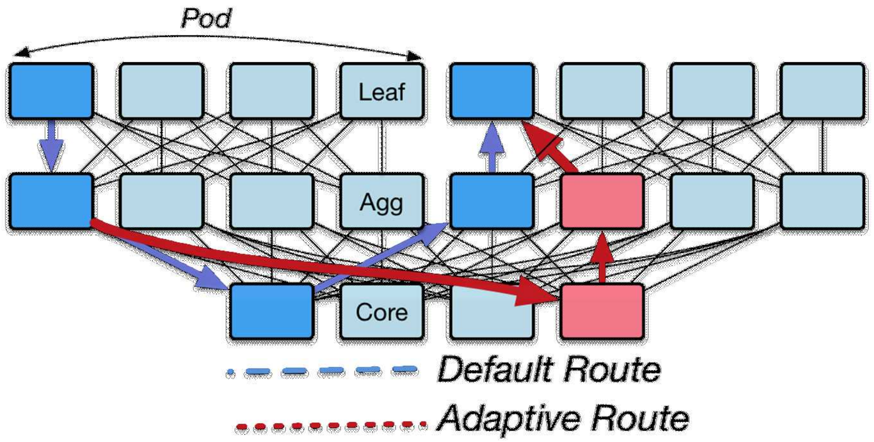


Adaptive routing avoids congestion

Fat-Tree and Dragonfly have different ways of linking within groups/pods and adding global links across groups/pods

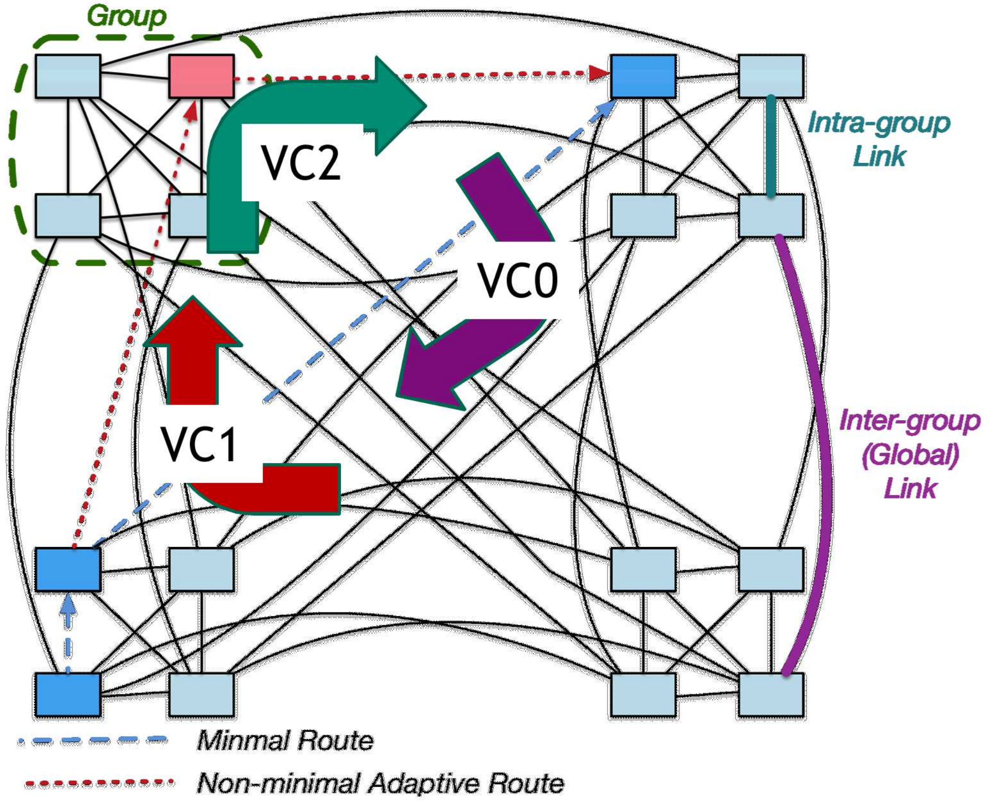


- Organized into groups
- All-to-all connectivity within group
- Every group has at least one connection to all other groups
- Less expensive: fewer switches

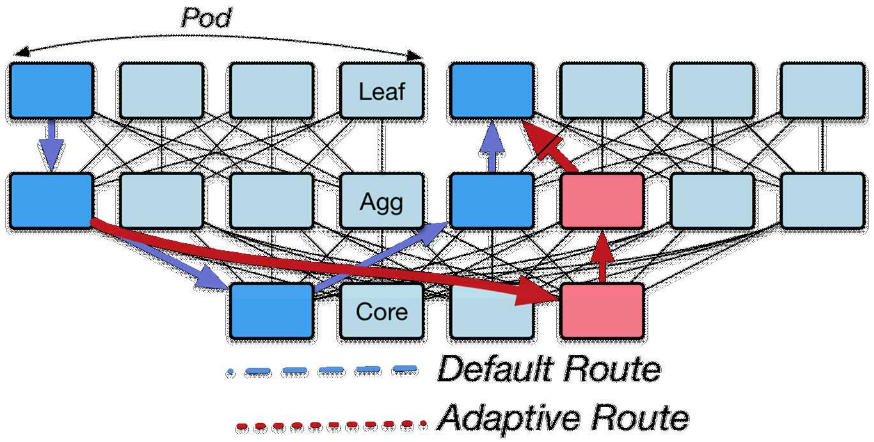


- Leaf, aggregation, and core switches form 3-level fat tree
- Interconnected leaf and aggregation switches form a "pod"
- More expensive: extra core switches and global links

Fat-Tree and Dragonfly have different tradeoffs in path length, path diversity, deadlock cycles



- Minimal paths are at most 1-3 hops
- Adaptive paths are non-minimal
- Adaptive paths create potential “deadlock cycles” in link graph
- Multiple virtual channels to break deadlock



- Minimal paths are at 2-4 hops
- Adaptive paths are minimal
- No cycles in link graph
- Single virtual channel to avoid deadlock

Adaptive routing, QoS, Dragonfly, and Fat Tree create a complex mix of tradeoffs

- Quality-of-Service:
  - Want different service levels (SLs) to distinguish traffic classes and avoid HoL blocking, but SLs use up scarce virtual lanes in the switch
- Adaptive Routing:
  - Want different virtual channels (VCs) for more path diversity with deadlock freedom, but VCs use up scarce virtual lanes in the switch
- Fat Tree:
  - Want to exploit simple routing for better adaptive routing, QoS, but reduce the cost
- Dragonfly:
  - Want to exploit the low diameter, low cost, and path diversity but reduce VC requirements

Dragonfly proposal: allow more flexible QoS by using minimal routing selectively to reduce virtual channel requirements

Dragonfly adaptive routing requires too many virtual channels (6 in most flexible case) to fit within realistic limits of available VLs (8-16 in most cases)

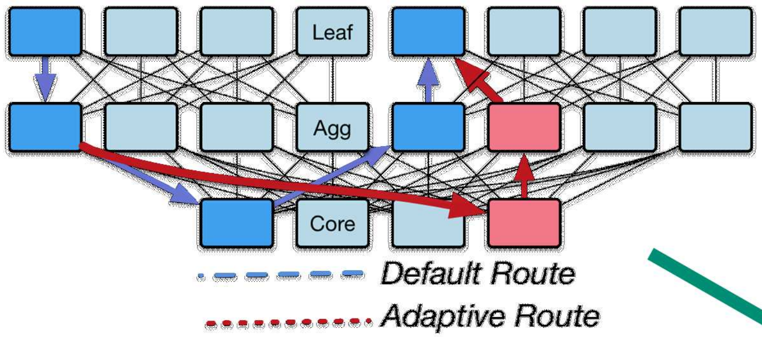
- ***Proposal (novel contribution in this work):***
  - Distinguish elephant (large)/mice (small), collective/pt-2-pt
  - Use high priority, minimal routing for mice
  - Use medium priority, adaptive routing for elephant collectives
  - Use low priority, adaptive routing for elephant pt-2-pt

	Large Pt-2-Pt						*Large Collective*						*Small Pt-2-Pt*		*Small Collective*	
	SL 0						SL 1						SL 2		SL 3	
VC	0	1	2	3	4	5	0	1	2	3	4	5	0	1	0	1
VL	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	Adaptive						*Adaptive*						*Minimal*		*Minimal*	

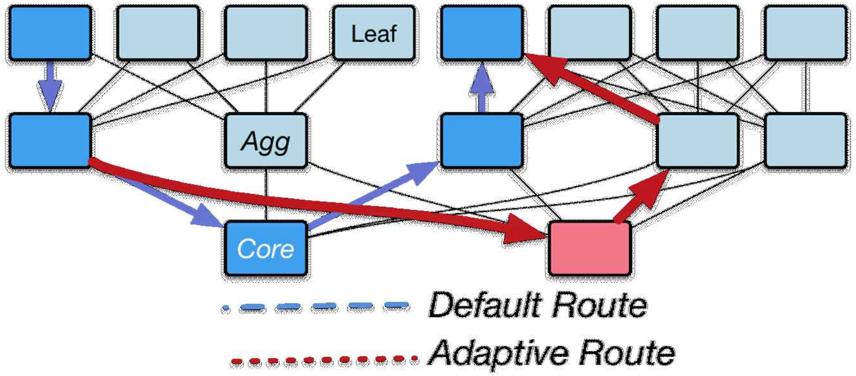
Mapping of service levels and virtual channels to virtual lanes

Fat Tree proposal: reduce cost to match Dragonfly by tapering

Full Fat Tree with Equal Cost MultiPath (ECMP) routing

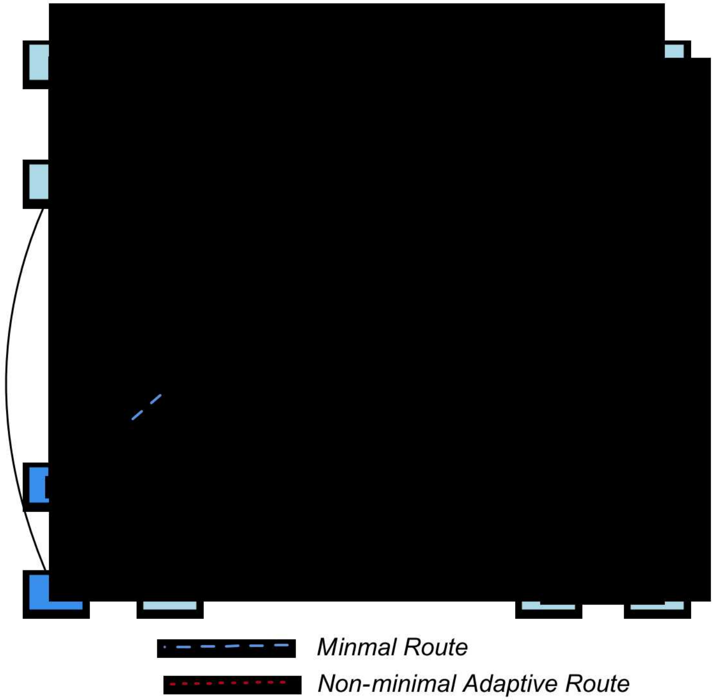


Tapered Tree with Equal Cost MultiPath (ECMP) routing

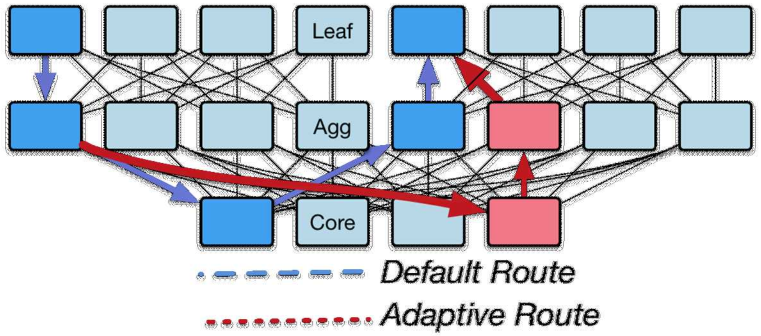


Goal: Find best combination of QoS and adaptive routing between Dragonfly and Fat Tree

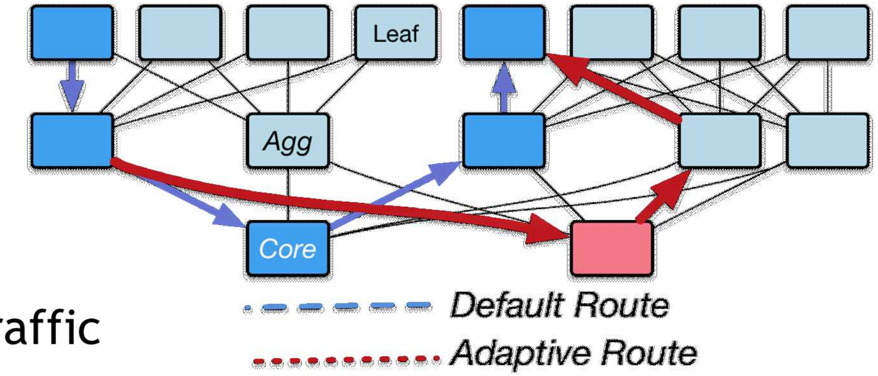
Use progress adaptive routing (PAR) which allows misrouting at each network hop



Full Fat Tree with Equal Cost MultiPath (ECMP) routing

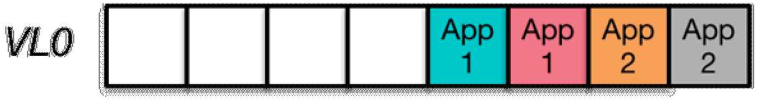


Tapered Tree with Equal Cost MultiPath (ECMP) routing

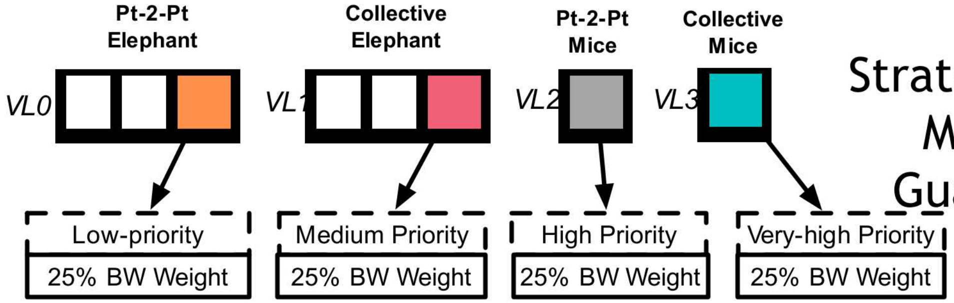


- Dfly-PAR: Adaptive routing for all traffic
- Dfly-MIN/PAR: minimal for mice, PAR for elephant traffic

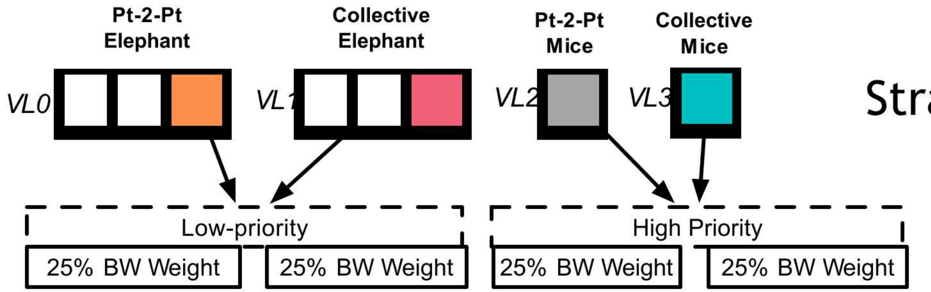
Goal: Find best combination of QoS and adaptive routing between Dragonfly and Fat Tree



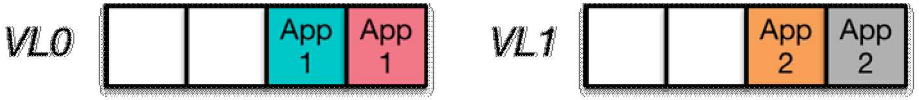
QoS



Strategy #2, ***Split-Priority QoS***: 4 priorities
Mice > Elephant, Collective > Pt-2-Pt
Guaranteed bandwidth share across VLs



Strategy #3, ***Bandwidth-Weight QoS*** with 2 priorities
Mice > Elephant, Collective = Pt-2-Pt
Guaranteed bandwidth share across VLs



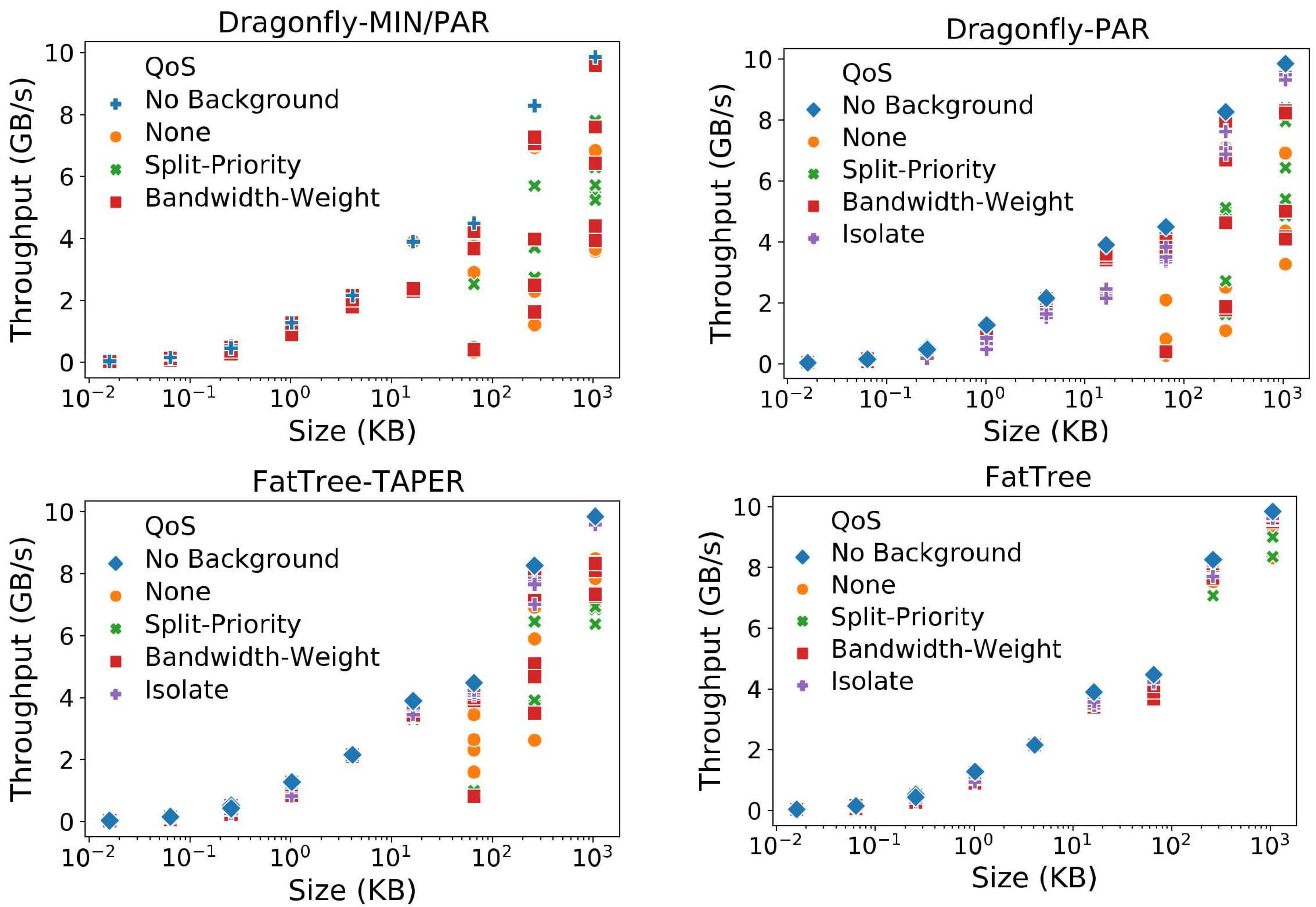
Strategy #4: ***Isolate QoS***
No priorities, but every app gets guaranteed bandwidth share through dedicate VL

Experimental Setup

- Simulations performed using Structural Simulation Toolkit
- Ran mixture of different "Foreground" and "Background" apps
  - *Halo3D:* bandwidth-intensive pt-2-pt nearest neighbor exchange
  - **FFT:** bandwidth intensive collectives across row/column subcommunicators
  - **Sweep3D:** sweep propagation across 3D space, many small pt-2-pt messages
- Filled system of size 8192 nodes with 1K-4K foreground, 4K-7K background processes

Topology	# Switches	Port Configuration
Dragonfly-PAR	512	16 injection, 32 intra-group, 16-inter-group
Dragonfly-MIN/PAR	512	16 injection, 32 intra-group, 16-inter-group
FatTree	640	32 up, 32 down
FatTree-TAPER	480	27 up, 36 down

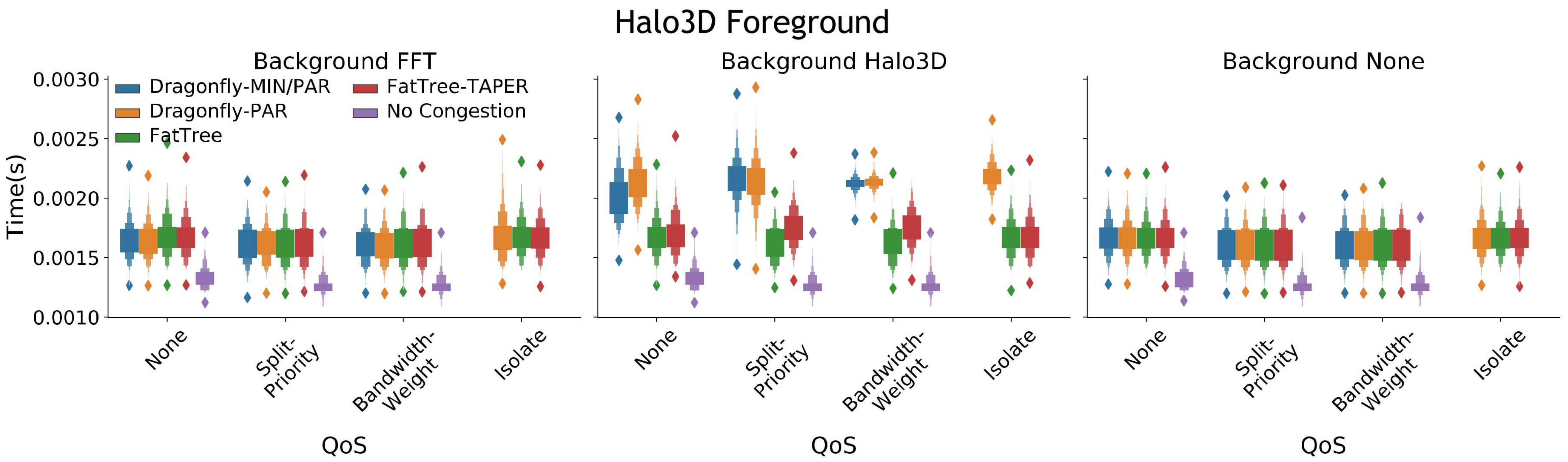
PingPong benchmark with Halo3D background illustrates effect of interference on flows of different sizes



- High degree of scatter, except with QoS = Isolate
- Fat Tree sees near-optimal throughput for all cases
- Minimal routing of mice traffic does not degrade performance on Dragonfly
- Tapered Fat Tree competitive with Dragonfly

Halo3D is bandwidth-intensive app that offers few opportunities for QoS

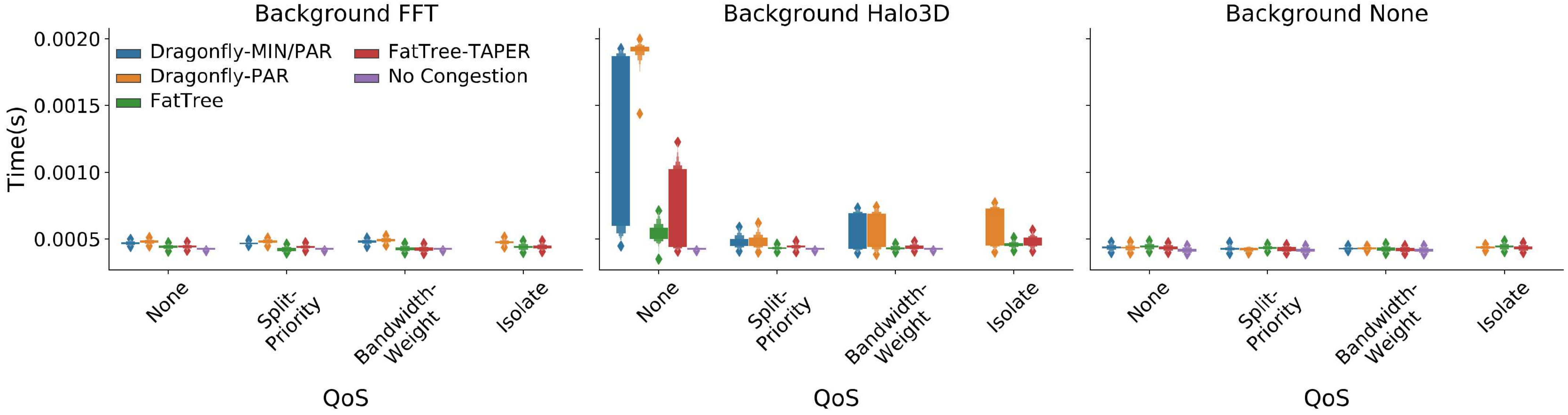
Box Plot: Show distribution of iteration times across all MPI ranks



- Tapered Fat Tree outperforming Dragonfly
- MIN/PAR matches or exceeds PAR routing
- Only Halo3D as background degrades performance
- All cases much higher than no congestion baseline
- QoS=Isolate + Fat Tree provides good performance

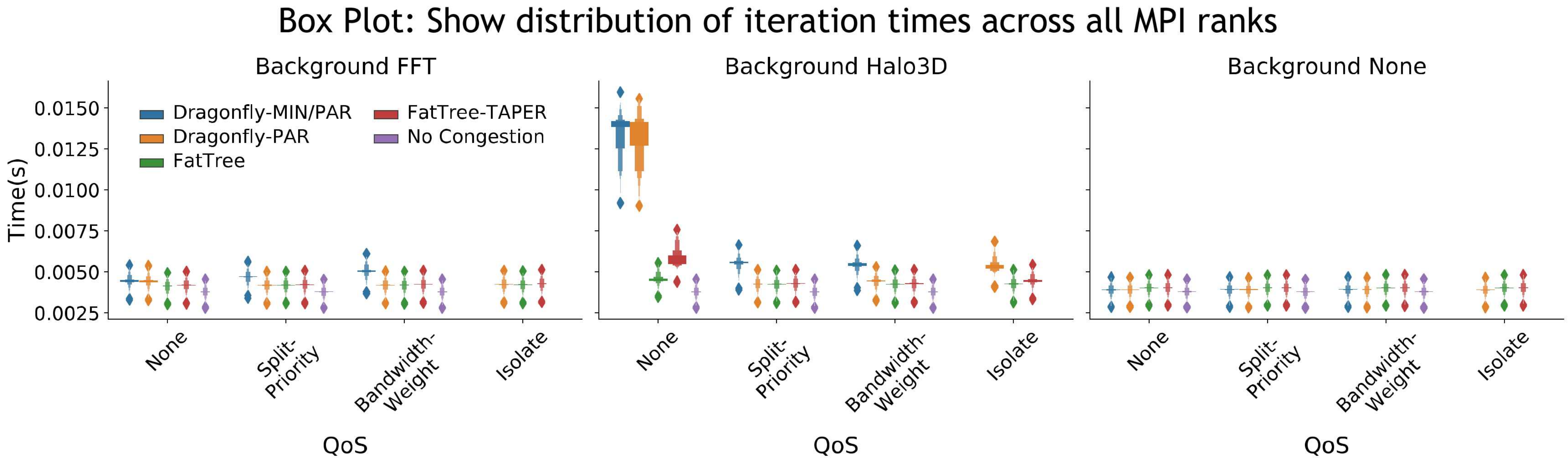
FFT contains collectives that are more latency-sensitive, hotspots delays “propagate” across whole collective

Box Plot: Show distribution of iteration times across all MPI ranks



- MIN/PAR matches or exceeds PAR on Dragonfly
- All QoS strategies bring performance back near baseline
- Halo3D background degrades FFT performance without QoS
- Tapered Fat Tree outperforms Dragonfly
- QoS=Isolate + Fat Tree provides good performance

Sweep3D is latency-sensitive, sends many smaller messages



- Tapered Fat Tree outperforms Dragonfly
- MIN/PAR performs slightly worse than PAR in certain cases
- All QoS strategies bring performance back near baseline

Conclusions

- Equal cost multipath (ECMP) is simple, but effective on tapered Fat Tree
- Mixed minimal/adaptive MIN/PAR strategy effective for Dragonfly to reduce VLs
- QoS=Isolate is most robust strategy, but only possible on Fat Tree with simpler deadlock-free routing

Tapered Fat Tree best compromise?
Reduced cost, simple but performant routing, great QoS flexibility

Acknowledgments

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA-0003525.

National Nuclear Security Administration

Sandia National Laboratories

EXASCALE COMPUTING PROJECT