Hewlett Packard Enterprise

INTERCONNECTS ATPESC 2021

Eric Borch & Igor Gorodetsky

August 2, 2021

US DOE SYSTEM ARCHITECTURE TARGETS

System attributes	2010	2018-2019		2021-2022	
System peak	2 Peta	150-200 Petaflop/sec		1 Exaflop/sec	
System memory	0.3 PB	5 PB		32-64 PB	
Node performance	125 GF	3 TF	30 TF	10 TF	100 TF
Node memory BW	25 GB/s	0.1TB/sec	1 TB/sec	0.4TB/sec	4 TB/sec
Node concurrency	12	O(100)	O(1,000)	O(1,000)	O(10,000)
System size (nodes)	18,700	50,000	5,000	100,000	10,000
Total Node Interconnect BW	1.5 GB/s	20 GB/sec		200GB/sec	
MTTI	days	O(1day)		O(1 day)	
	Past production	***Current generation (e.g., CORAL)***		***Exascale Goals***	

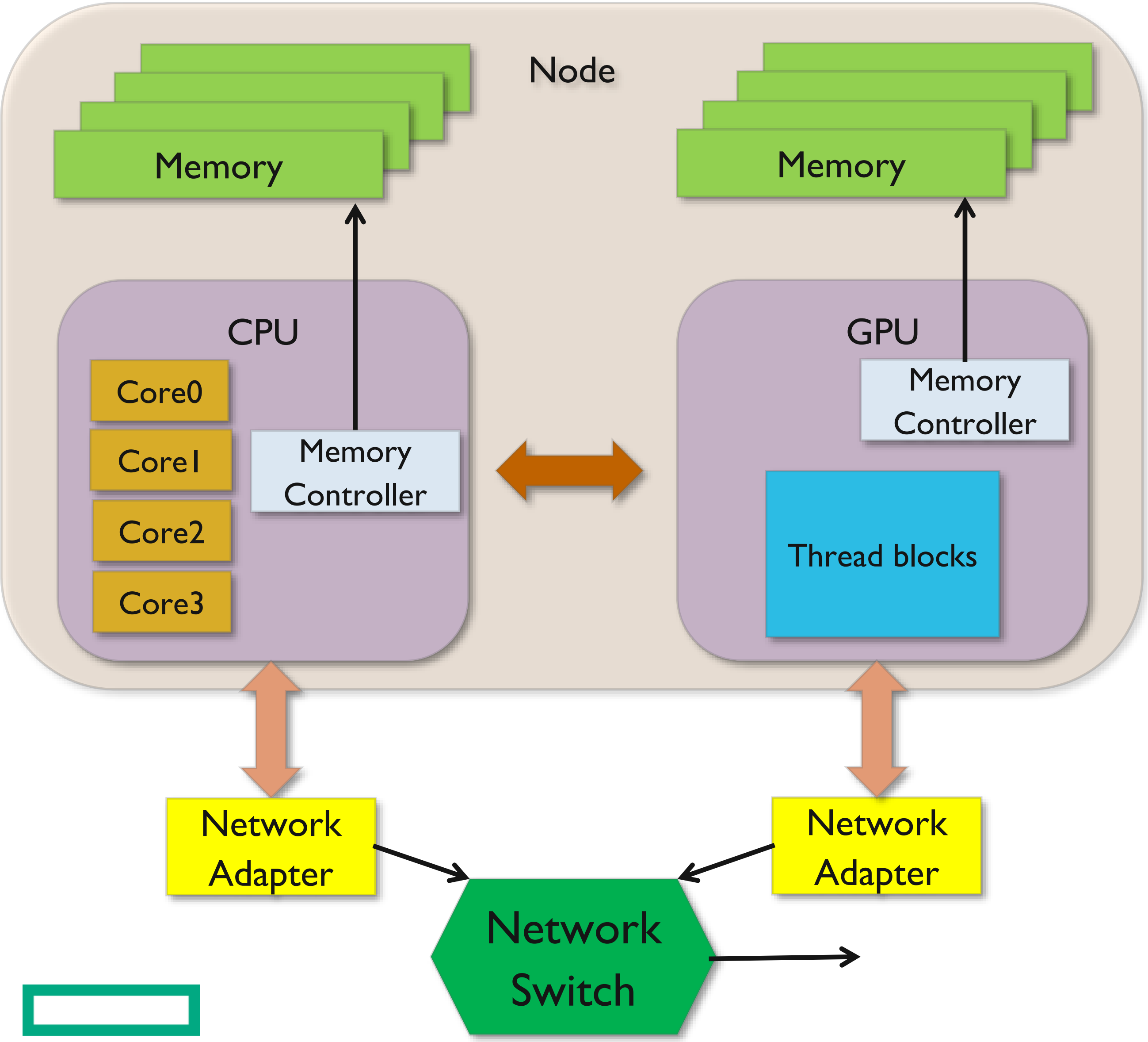
[From DOE Exascale report]

GENERAL TRENDS IN SYSTEM ARCHITECTURE

- Core clock frequency is not increasing
- Number of threads on a core is increasing
- Number of cores on a node is increasing
- Number of nodes is increasing
- Accelerators gain prominence
  - Lead to hybrid nodes

- What does this mean for networks?
  - More sharing of the network interconnect
  - The aggregate amount of communication from each node will increase moderately
    - More smaller messages
  - A single CPU core may not be able fully saturate the NIC
  - Accelerators must be able to participate in communication

SIMPLIFIED NETWORK ARCHITECTURE



- Complex nodes combining CPUs and GPUs
- Network communication requires coordination between them
- Important to use ”close” network adapter, if possible
- Several I/O technologies exist
  - PCIe, NVLink, xGMI
  - Expected to provide higher bandwidth than what network links will have

NETWORK ADAPTERS

NETWORK ADAPTER TRENDS

- Increasing complexity is motivated by diverse workloads
  - AI/ML, Realtime data streaming, traditional HPC application
- CXL (Compute Express Link) is getting traction among CPU/GPU vendors
  - CXL.io is equal to PCIe
  - CXL.cache good for offloading atomics
  - CXL.mem great potential for network-attached storage
- Data encryption is driven by increased demand for handling sensitive data
- Resource virtualization helps to improve system utilization while protecting users
- Increased degree of hardware programmability helps to optimize for diverse offloads
  - Datatypes aware transactions
  - Access control list (ACL)
  - Tunnelling RDMA and IP protocols

OFFLOADING TECHNIQUES AND CHALLENGES

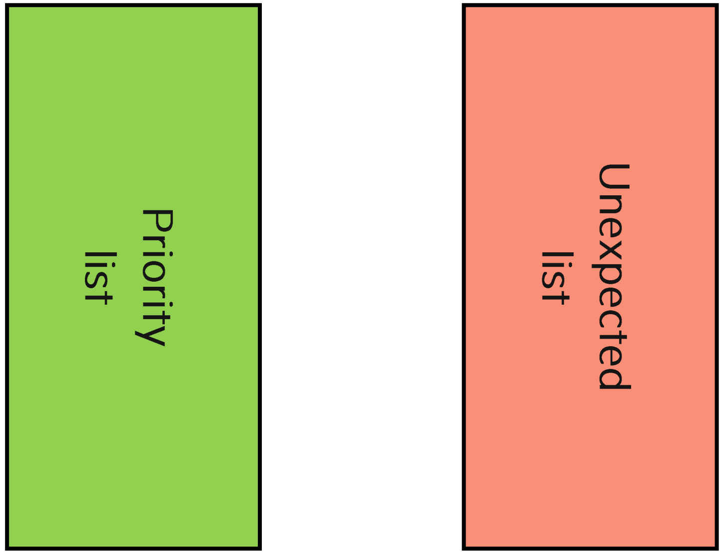
- **MPI tag matching** offload helps to achieve a high message rate
  - Large number of posted receives or unexpected messages lead to performance degradation
- **Triggered operations** can be used by GPU to issue transactions staged by CPU
  - Works well for static communication patters as CPU needs to pre-program transactions
- **Counting events** speeds up completion notifications and simplifies their processing
  - Best for batch message processing with a few distinct tags
- **Memory address translation and on-demand paging (ODP)** reduces the memory footprint of an application
  - Not pinning pages is great, but the high volume of ODP may lead to performance degradation
- **Datatypes handling** by the network adapter avoids data copying in software
  - Usually, requires pre-programming. A high number of distinct types may increase latency.

MPI TAG MATCHING

- Pool of list entries can be very large, but the number of entries held withing the adapter is always limited. Fetching additional entries might be expensive.
- Pre-posting too many receive buffers increases network latency as it takes longer to match incoming messages.
- Long unexpected list drains pool of list entries and slows down append

MPI_Recv

1. Match on Unexpected
2. Append to Priority

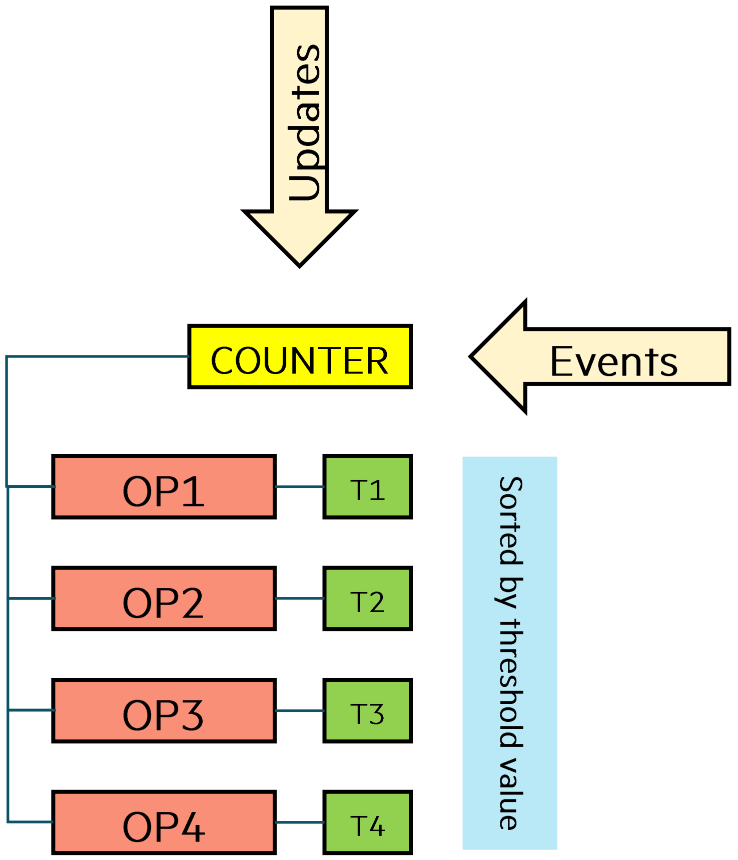


MPI_Send from peer

1. Match on Priority
2. Append to Unexpected

TRIGGERED OPERATIONS

- Transactions attached to a counter
- Each transaction configured with a threshold value
- Counter is incremented by a completion event or explicit updates by software
- Transaction is triggered when the counter is equal to its threshold value



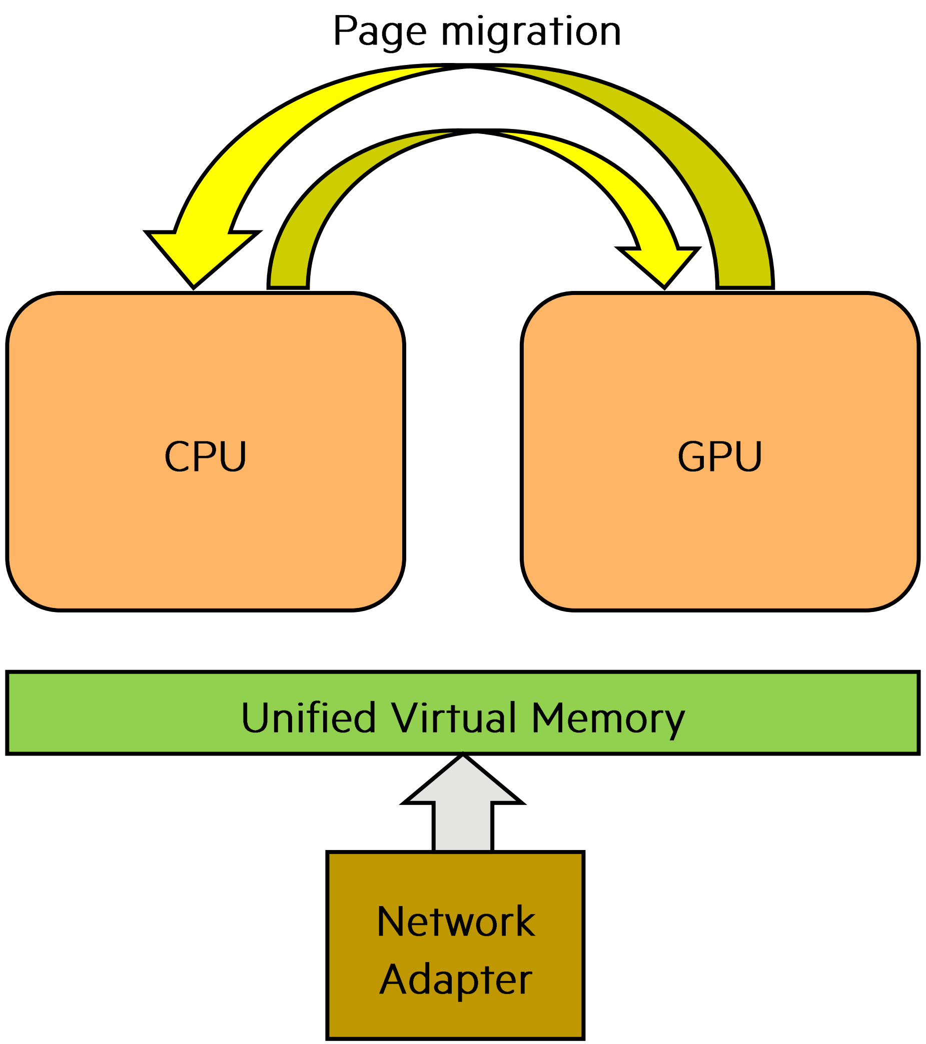
COUNTING vs. FULL EVENTS

- Counter gets incremented when the desired event occurs.
- Each instance includes Success and Error counters. If an error occurs, full event is delivered to assist with the error handling.
- Counter updates can be pushed into a host memory by the adapter. Updating the host memory on every increment or at thresholds.
- Works well when process is waiting for N messages to complete before proceeding to the next step.

- Full event is delivered at the end of every transaction and to notify of an error.
- Provides information about initiator and target. Maps better to MPI Send/Recv semantics.
- Allows handling truncated transactions as the event provides both requested and delivered length.

ADDRESS TRANSLATION AND ON-DEMAND PAGING

- A virtual address in the Unified Virtual Memory space is used in transactions.
- Network Adapter deploys PCIe Address Translation Service (ATS) to obtain and cache translated addresses.
- If physical memory is not present, the adapter can use Page Request Interface (PRI) to trigger On-Demand Paging (ODP).
- ODP may bring page from another device (e.g., GPU) or allocate a new page.

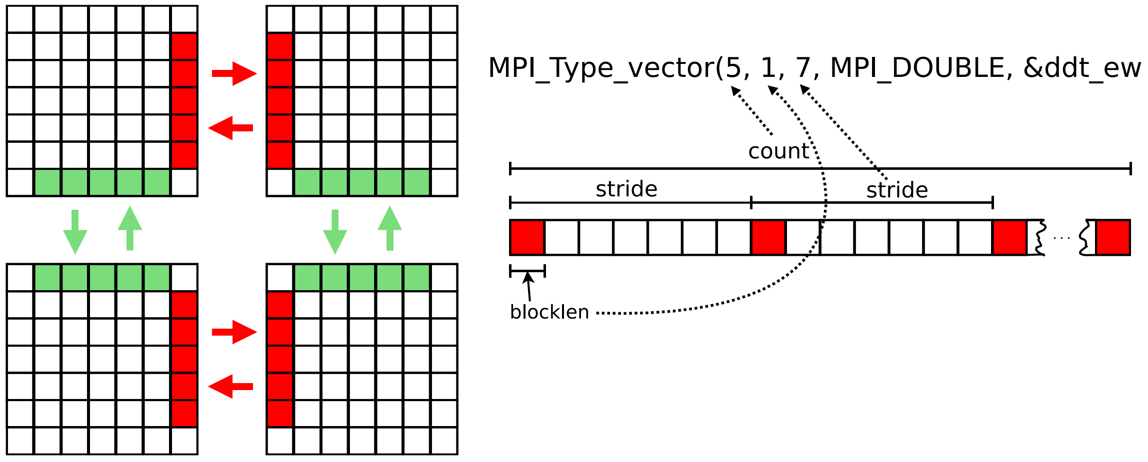


DATATYPES

- IOVEC is the best-known datatype, when data is described by a series of memory regions.

- MPI can express datatypes providing the software a way to execute transactions without copying data to a contiguous buffer.

- Offloading handling of derived datatypes to hardware allows the user to specify precisely which memory locations are involved in a Send or Receive and have that data be packed into a single, more efficient message.

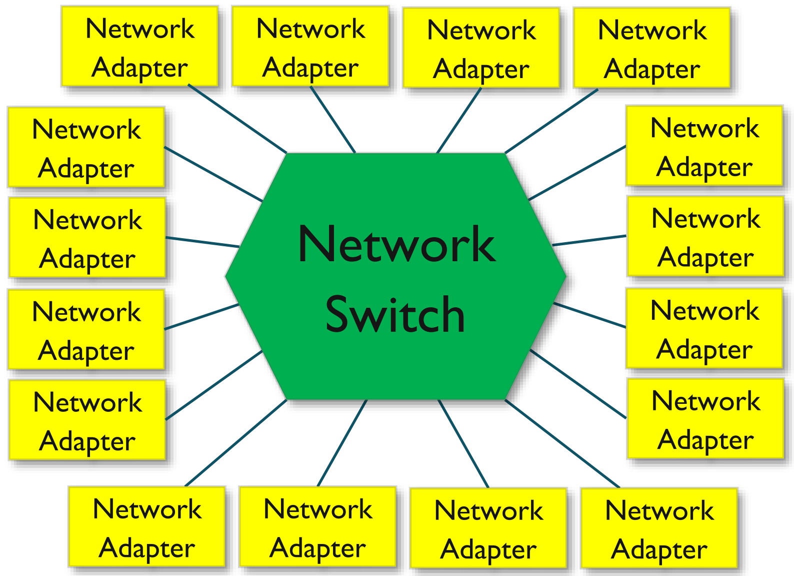


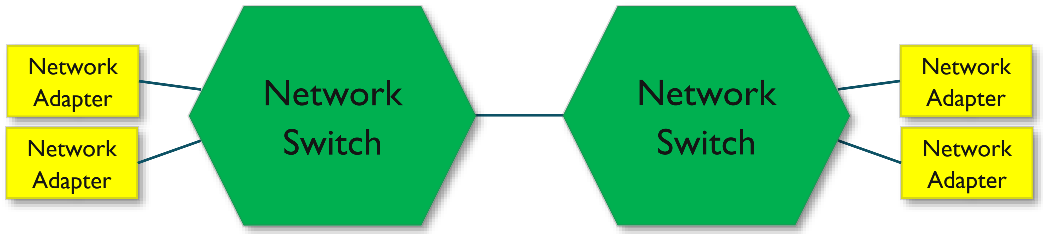
Reference: spcl.inf.ethz.ch

INTERCONNECT / TOPOLOGIES

NETWORK TOPOLOGIES

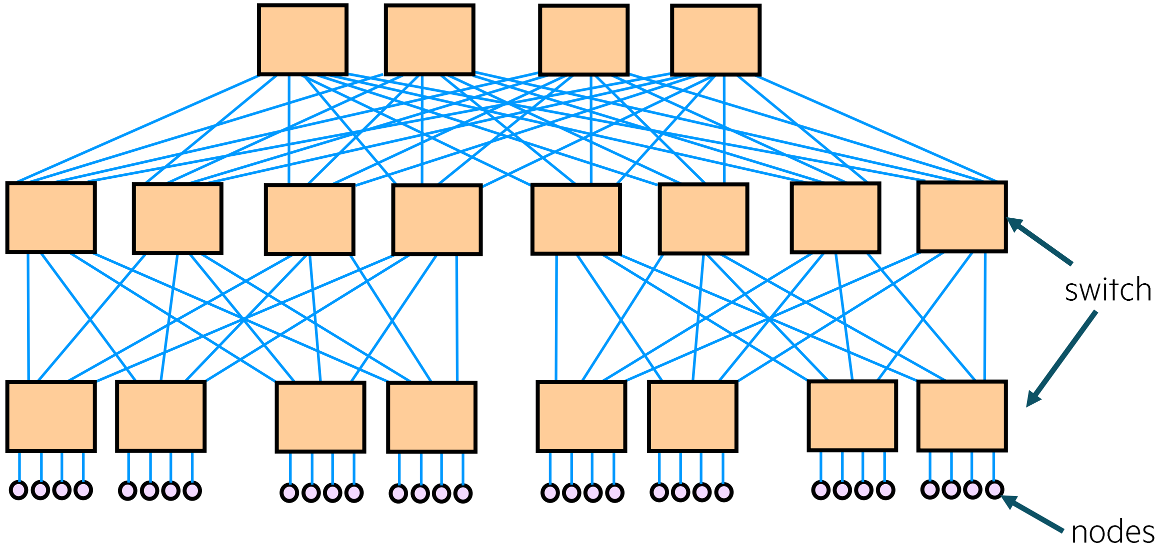
- The topology describes how switches and endpoints in a network connect to each other
- Ideal topology would be all-to-all connectivity
  - Single switch
    – Reality limits switch to ~64 ports
- Fabric hop count is the number of switch-to-switch links traversed by a packet
  - Distance between nodes
  - As node count grows, fabric hop count grows
- We need topologies able to connect hundreds, thousands, tens of thousands, hundreds of thousands of nodes
- Minimize hop count (while maintaining performance)
  - Every time a packet takes a hop
    – It consumes fabric resources (bandwidth, buffering, power)
    – It increases the probability of interference or congestion
  - Fewer hops means lower fabric cost
    – Less switches and cables required
    – More money to spend on compute

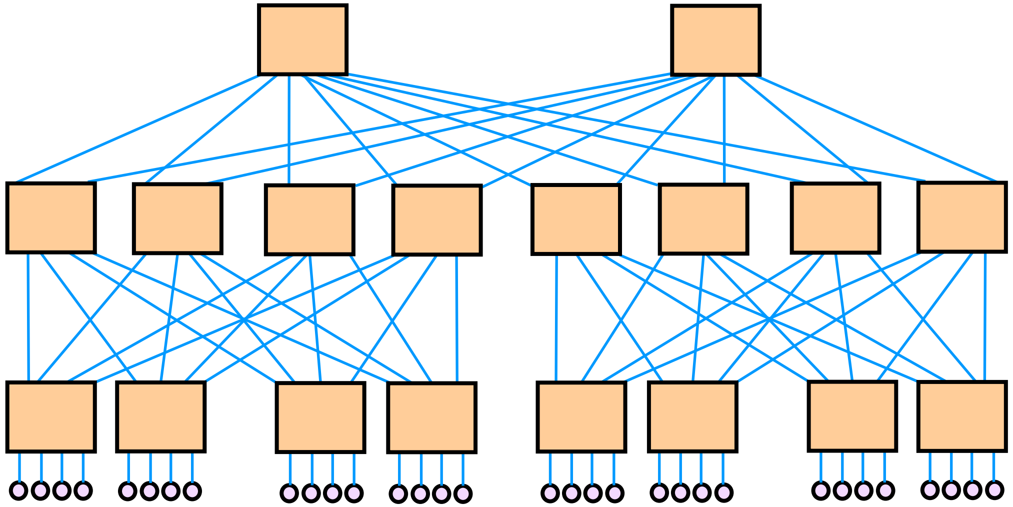




FAT-TREE TOPOLOGY

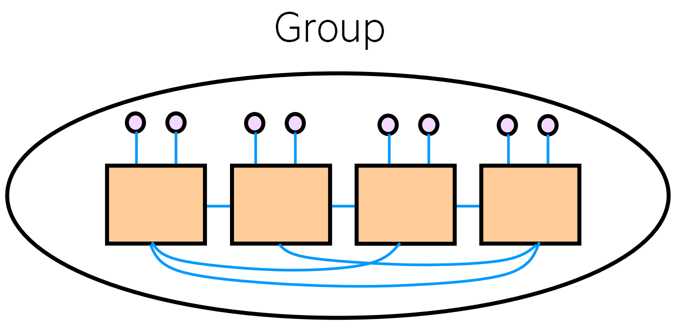
- Common topology for current systems (Sierra & Summit)
  - 3 Levels, maximum fabric hop count of 4
- Fully configured fat-tree (Summit)
  - Tree with equal bandwidth at each level
  - Non blocking between all pairs of nodes
    - Doesn't always happen in practice
  - Over provisioned bandwidth
    - Performance is quite good
    - Large percentage of fabric cables are optical
    - Can get expensive at scale
- Bandwidth tapered fat-tree
  - Unequal bandwidth at each level
  - No longer non-blocking between all pairs of nodes
  - Reduces number of cables (optical) and routers
- The number of cables and routers increases super-linearly with node count
- Scheduling a job across nodes to minimize fabric hops between hosts maximizes performance



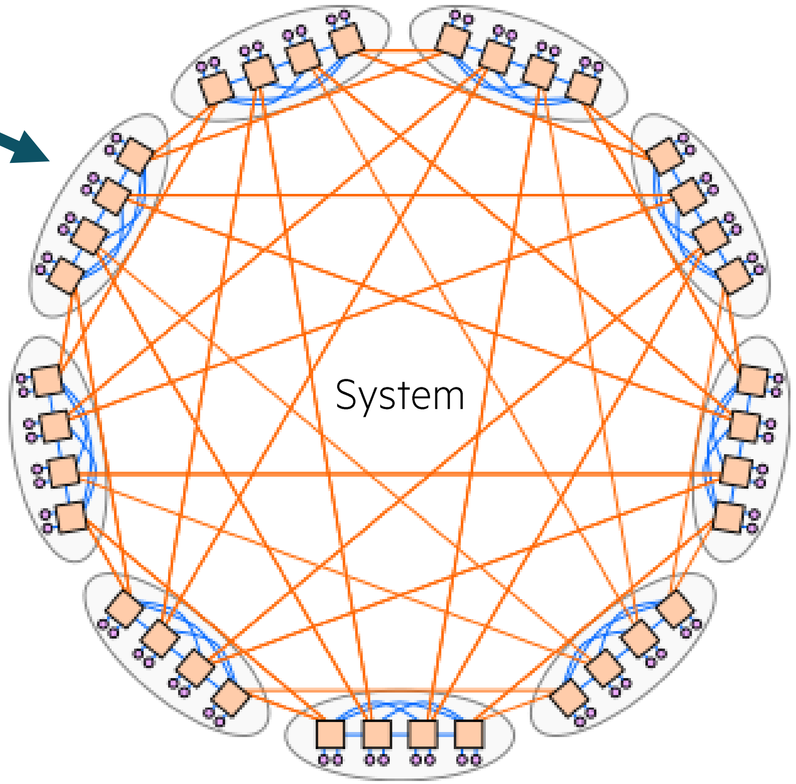


DRAGONFLY TOPOLOGY

- Every router has nodes connected
- A group contains routers that are all to all connected (1-D Dragonfly)
- All groups in the system are all to all connected



- Aurora, Frontier, El Capitan will use a 1-D Dragonfly
- Primarily driven by network cost as system scale grows
  - Linear increase in the number of cables and routers with system size
  - Less than 33% of fabric cables are optical
  - Scales to 4x number of nodes as 3 level fat-tree
  - Maximum hop count of 3
- Requires sophisticated adaptive routing
- Job scheduling
  - Intra-group if job can fit in a single group (256-512 nodes/group)
  - Randomly across system if job is larger than a single group
    - – Bandwidth between individual group pairs is low compared to node injection bandwidth into group (and total bandwidth out of group)



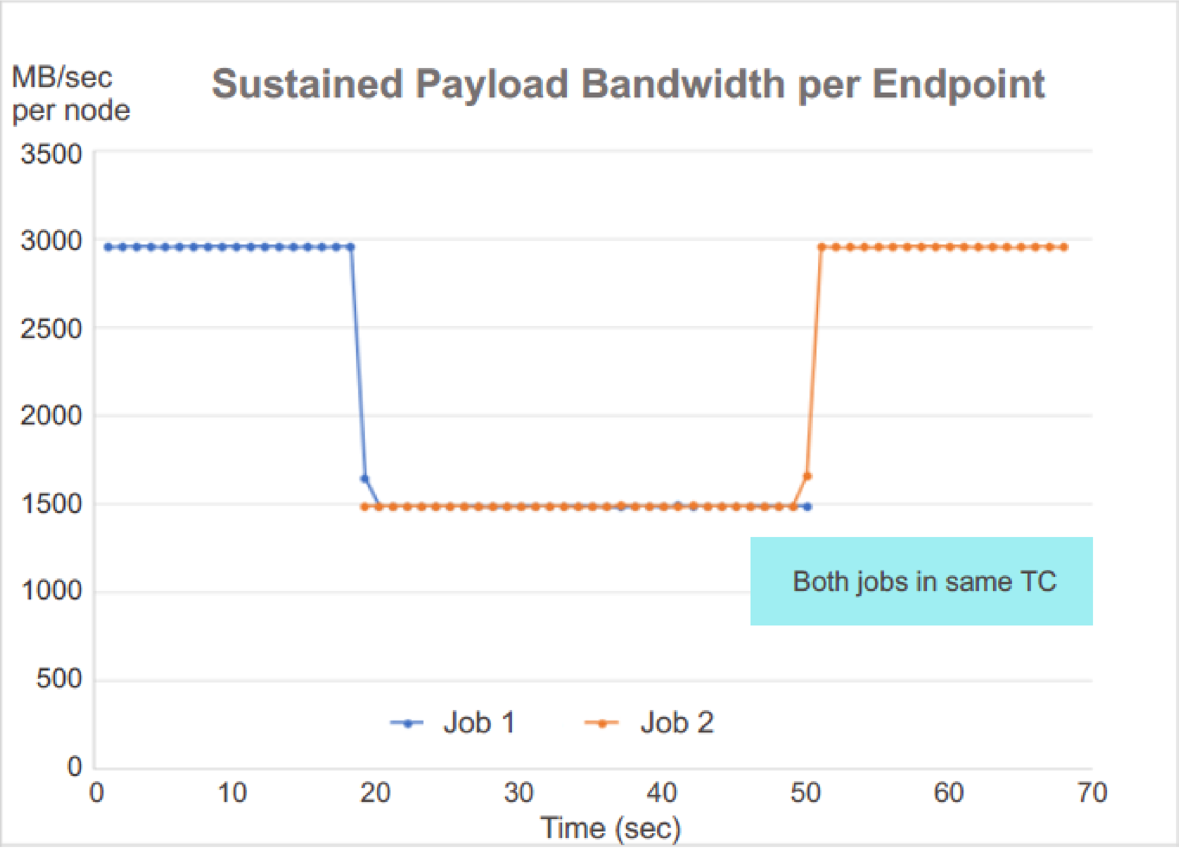
INTERCONNECT FEATURES THAT IMPROVE PERFORMANCE*

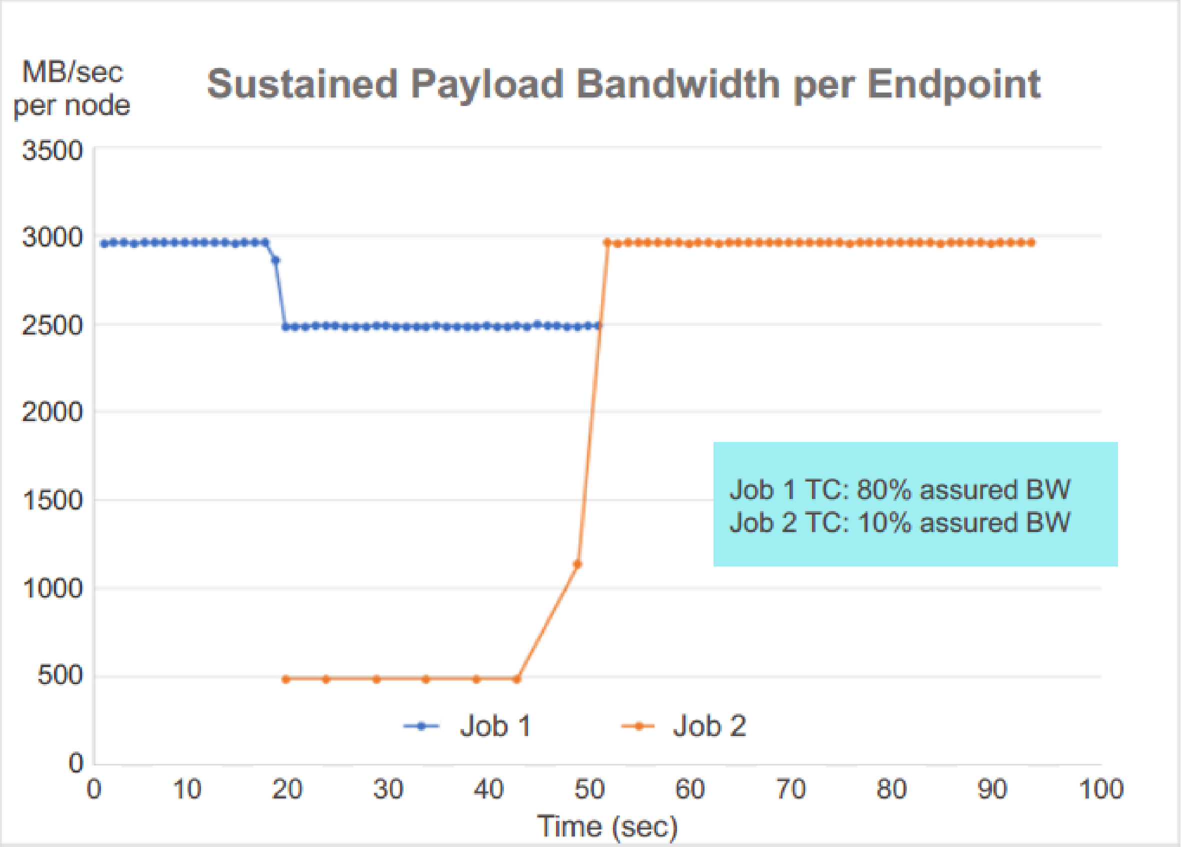
- Adaptive Routing
  - Per packet - used to choose where a packet goes
  - Targets topological based congestion (unrelated flows crossing in the network)
  - Used to route around temporal hot spots in the network
    - Used sparingly, routing via a longer path can reduce latency
    - Used excessively, routing via a longer path can increase latency and decrease bandwidth
- Quality of Service classes
  - Part of arbitration - used to choose which packet to advance
  - Tunable classes may use priority, min & max bandwidth allocation, routing biases, etc
    - Example classes: low latency, standard compute, bulk data, scavenger
  - Job can use multiple classes
  - Provides performance isolation for different classes of traffic
- Congestion management
  - Targets workload-based congestion (incast, many to few)
  - Identifies and controls causes of congestion
    - Throttles sources to prevent excess traffic from entering the network
    - Prevents highly filled buffers, congestion, contention
  - Applications much less vulnerable to other traffic on the network
  - Predictable runtimes
  - Lower mean and tail latency – a big benefit in applications with global synchronization

* Slingshot techniques described

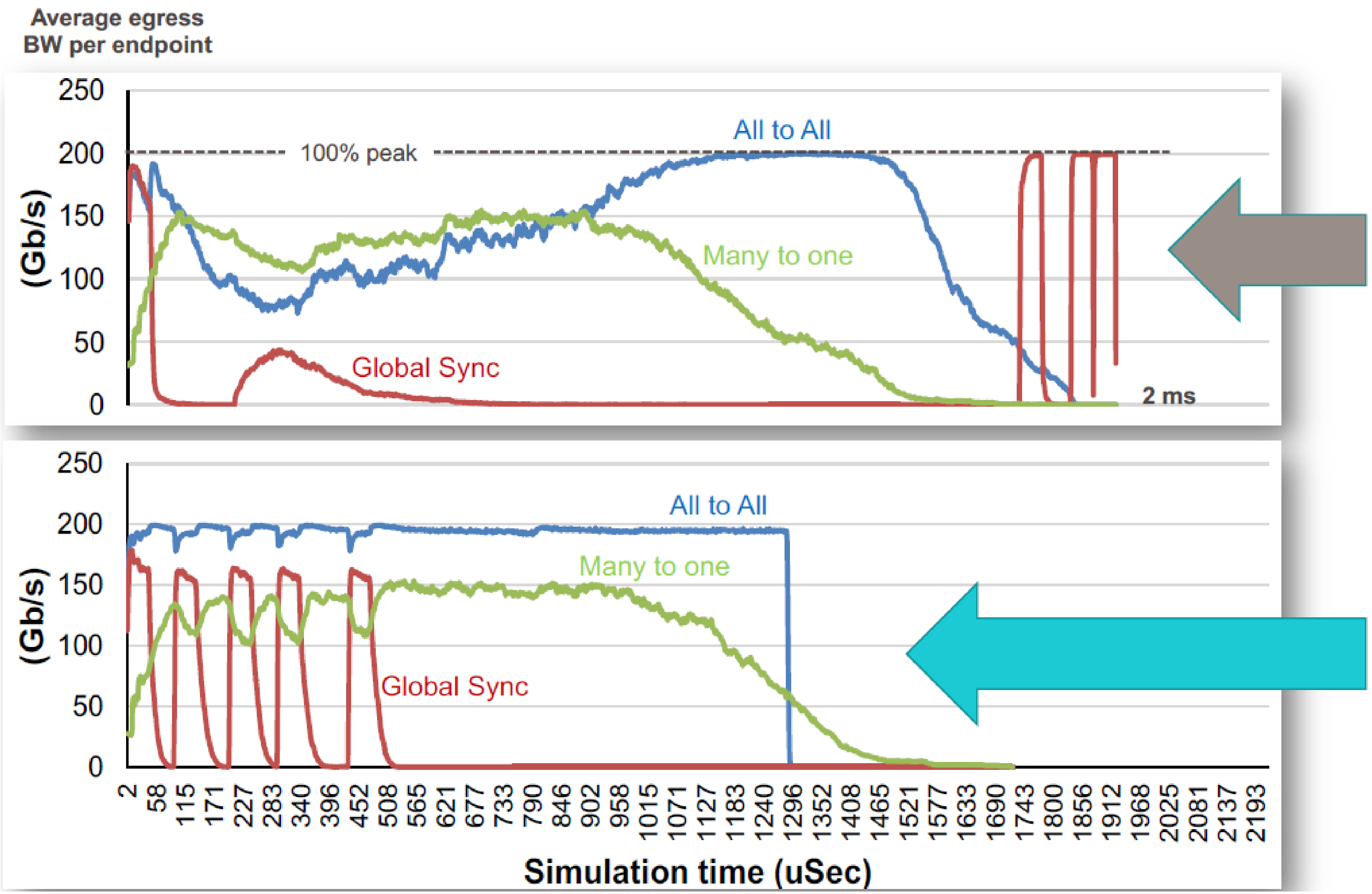
SIMPLE QUALITY OF SERVICE CLASS DEMO

- Two interleaved 64-node bisection bandwidth jobs
- 128-node Slingshot/Rosetta system, tapered to 3.125 GB/s/node peak bisection bandwidth





CONGESTION MANAGEMENT PROVIDES PERFORMANCE ISOLATION



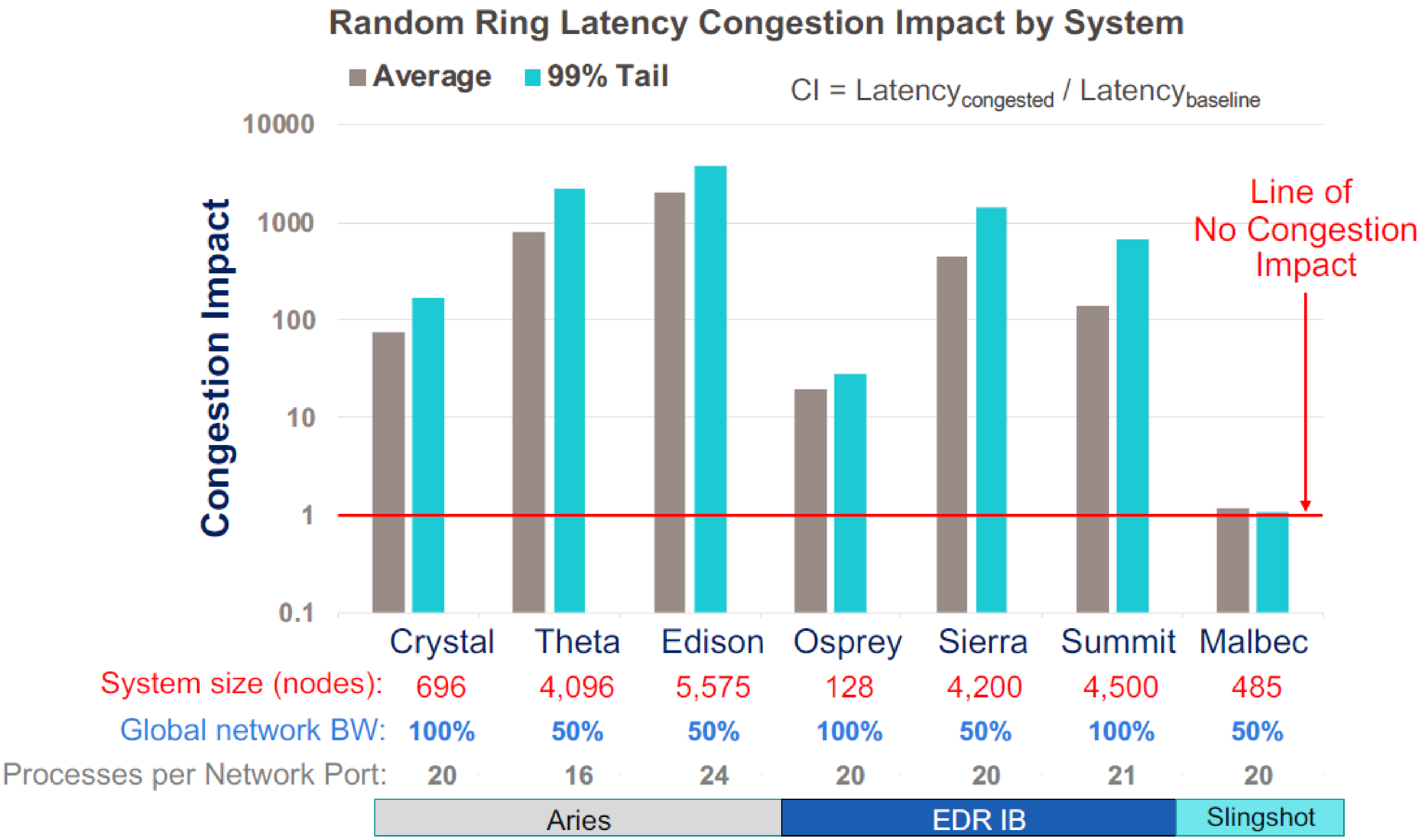
Job Interference in today's networks

Congesting (green) traffic hurts well-behaved (blue) traffic, and really hurts latency sensitive, synchronized (red) traffic.

With Slingshot Congestion Management

CONGESTION IMPACT IN REAL SYSTEMS

- Impact worsens with scale and taper
- Infiniband does somewhat better than Aries
- Slingshot does *really* well



THANK YOU

Eric.Borch@hpe.com
Igor.Gorodetsky@hpe.com