Exascale Topologies: The Good, the Bad, and the Not-so-Pretty

ExaComm Workshop @ ISC, 16 July 2015

Cyriel Minkenberg, Bogdan Prisacari, German Rodriguez Herrera, Wolfgang Denzel (IBM Research – Zurich)
Philip Heidelberger, Dong Chen, Craig Stunkel, Yutaka Sugawara (IBM TJ Watson Research Center)

Acknowledgment

- This work was supported and partially funded by Lawrence Berkeley National Laboratory (LBNL), on behalf of the US Department of Energy, under LBNL subcontract number 7078416.

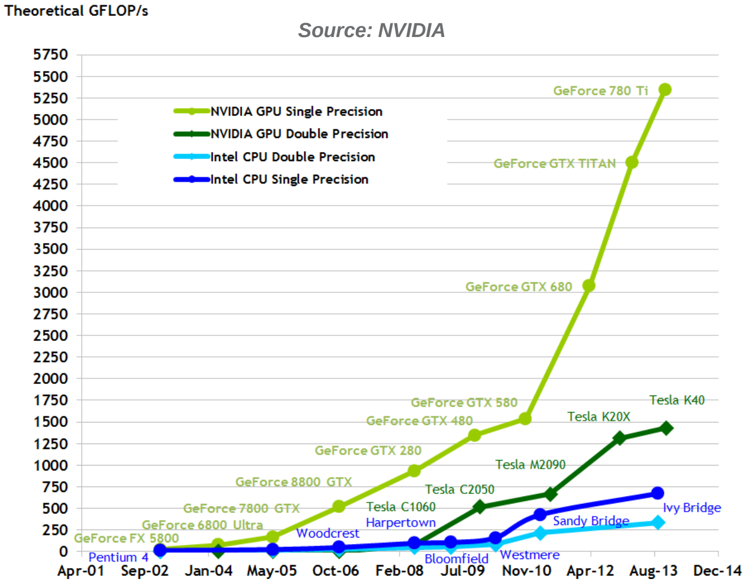
Agenda

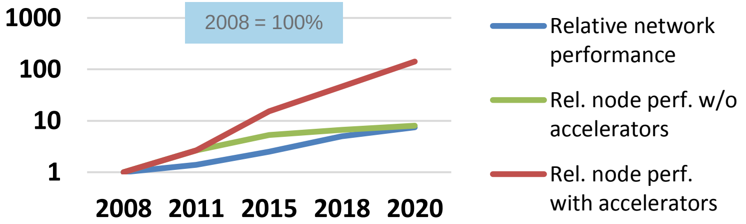
1. **Network challenges**
   – Cost, scale, energy, reliability, performance at scale, *balance*
2. **Topologies**
   – Low-diameter networks, including some new options
3. **Routing algorithms**
   – Direct, Valiant, Adaptive
4. **Performance evaluation**
   – Traffic: Uniform, adversarial, exchange patterns
   – Topologies: 1 old, 2 new
5. **Conclusions**

Network challenges

Compute nodes are getting “fat”

- On Nov. 2014 Top 500 list, 75 systems use accelerators, mostly NVIDIA GPUs or Intel MIC (Xeon Phi)
- Five of the Top 10 systems, incl. #1 & #2
- Two classes of ~20 PF/s systems
  - “Thin” nodes: 100K nodes @ 0.2 TFLOP/s/node; CPU-only
  - “Fat” nodes: 10 K nodes @ 2 TFLOP/s/node; CPU+accelerators
- “Fat” nodes imply that per-node FLOP rate is growing much faster than per-node network bandwidth!





Fat vs thin in the Top 10

#	System	Manuf. & type	Rmax [PFLOP/s]	#cores	Accel.	Nodes	TFLOPs/ node	Network & Topology	BW/node [GB/s]	B/FLOP
1	Tianhe-2	NUDT	54.9	3.12 M	XeonPhi (2+3)	16,000	3.4	Custom Fat tree	16	**0.0047**
2	Titan	Cray XK7	27.1	560 K	GPU (1+1)	18,688	1.45	Custom 3D Torus	9.6	**0.0066**
3	Sequoia	IBM BG/Q	20.1	1.57 M	-	98,304	0.2	Custom 5D Torus	20	**0.1**
4	K	Fujitsu	11.3	705 K	-	88,128	0.13	Custom 6D Torus	20	**0.15**
5	Mira	IBM BG/Q	10.1	786 K	-	49,152	0.2	Custom 5D Torus	20	**0.1**
6	Piz Daint	Cray XC30	7.8	116 K	GPU	5,272	1.48	Custom Dragonfly	64	**0.043**
7	Stampede	Dell PowerEdge	8.5	462 K	XeonPhi (2+1)	6,400	1.5	InfiniBand Fat tree	7+7	**0.009**
8	JUQUEEN	IBM BG/Q	5.9	459 K	-	28,672	0.2	Custom 3D Torus	20	**0.1**
9	Vulcan	IBM BG/Q	5.0	393 K	-	24,576	0.2	Custom 3D Torus	20	**0.1**
10		Cray CS-Storm	6.1	73 K	GPU (x+y)	?	>10?	InfiniBand Fat tree	7+7	~0.001?

Towards exascale: degrading system balance

Source: Nvidia

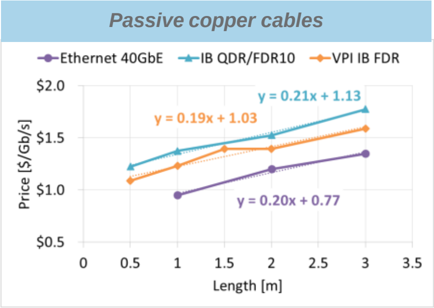
- Pre-exascale (~2017)
  - > 40 TFLOP/s per node
  - Dual-rail InfiniBand 4xEDR (2x 12.5 GB/s) per node
  - **Bytes/FLOP < 0.000625**
  - Bytes/FLOP = 0.1 would require >320 IB 4xEDR links per compute node
- Exascale balance can be expected to be similarly poor
  - E.g., node performance x2, IB links x2 (HDR)

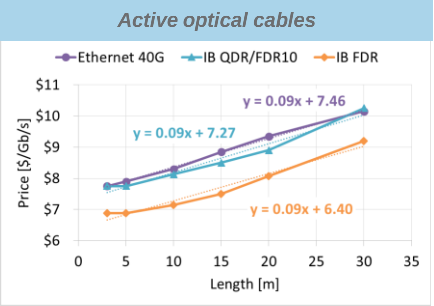
Anticipated design point for exascale systems has moved

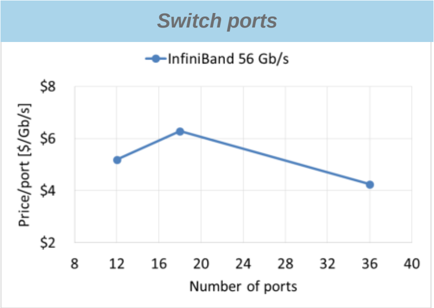
from >100,000 nodes of <10 TFLOP/s to 10,000-25,000 nodes of 40-100 TFLOP/s

Price-performance

- InfiniBand QDR/FDR cable list price data
  - Normalized w.r.t. data rate: $/Gbps
  - Passive copper (top)
  - Active optical (bottom)
  - Roughly linear with cable length
- Optical has ~6x higher offset (integrated transceivers) and ~2x lower slope
  - Large fraction of total cost in optical cables
- InfiniBand FDR switch ports
  - Normalized w.r.t. data rate: $/Gbps







Data source: colfaxdirect.com

(Very) Rough exascale network cost estimate

$$C_{\text{network}} = 8 \cdot \Gamma \cdot \beta \cdot R_{\text{max}}$$

aggregate price-performance

≈ 10 $/Gbps

peak compute rate

≈ 10^{18} FLOP/s

communication-to-computation ratio

≈ 0.1 byte/FLOP

$\times\, 267$

$$\Rightarrow C_{\text{network}} \approx 8\ \text{G\$} \gg 30\ \text{M\$} = 200 \times 15\%$$

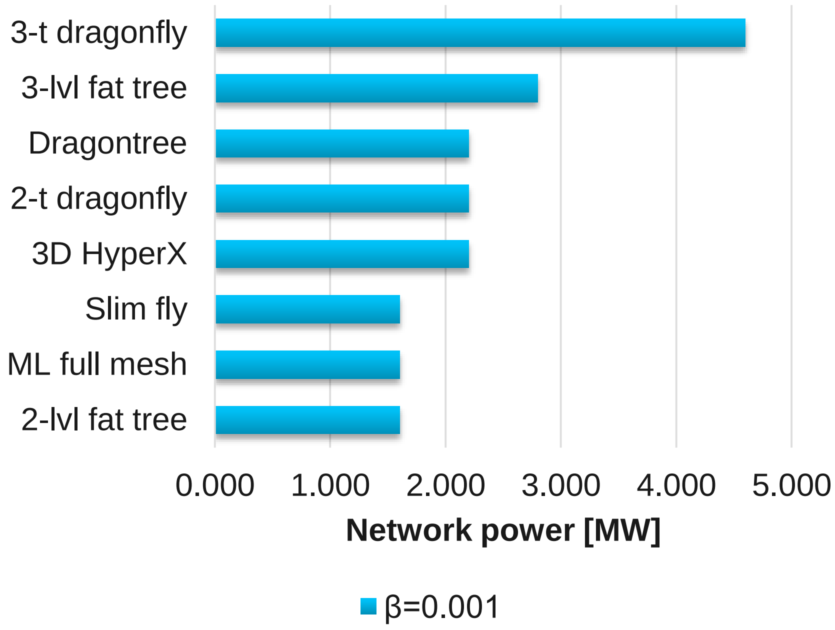
Something's gotta give…

System balance is worsening significantly

- Byte/FLOP ratios are going to have to drop by up to two orders of magnitude (< 0.001 B/F)
- Need **cost-effective** topologies with as few links and ports port endpoint as possible to achieve desired number of endpoints
- Need **optimized packaging** to maximize fraction of electrical links (backplane traces, TwinAx, coax) and minimize number of active optical links
- Major potential cost savings by integrating optical links with the switches and endpoints
  - Eliminate pluggable transceivers
  - Lead role for silicon photonics?

Network power

- Network power
  - Electrical links: integrated electrical IO; proportional to number of switch ports
  - Optical links: integrated electrical IO plus discrete optical transceiver; proportional to 2x number of optical links
  - Switching power; proportional to diameter
- $P_{\text{network}} = 8 \cdot (2L_{\text{opt}}\varepsilon_{\text{opt}} + (M+1)\varepsilon_{\text{ele}} +$

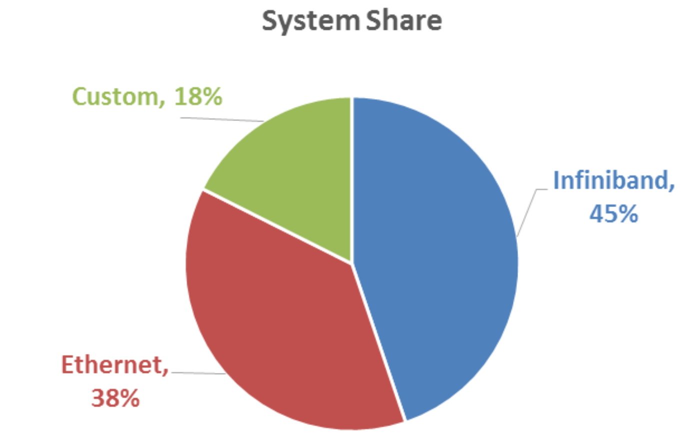


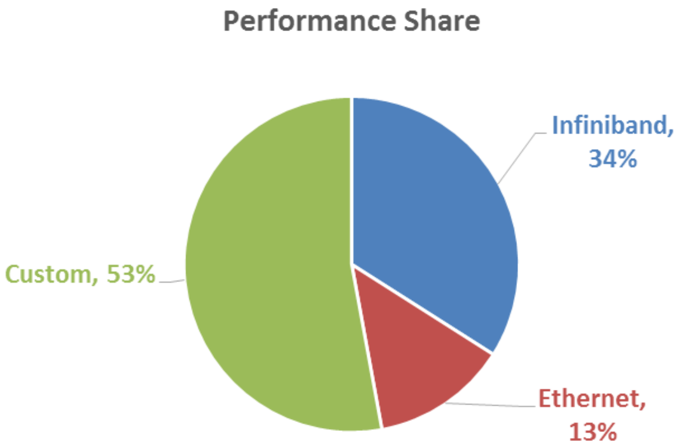
Cost is currently a stronger constraint than power

Topologies

Present network options

- Ethernet
  - Suitable for smaller commodity clusters
  - Topology options basically limited to trees
  - Lacks virtual channels & proper flow control
- Infiniband
  - Suitable for high-end systems in terms of scale, performance, features
  - Better price/performance than Ethernet at high data rates
  - Limited choice of vendors
- Custom/Proprietary
  - Aries, p775 hub, Tianhe, BG/Q torus, Tofu
  - Highest performance, densest integration
  - Substantial cost of design and implementation
  - Custom solution could integrate network on CPU, eliminating NICs and/or switches





Source: Nov. '14 Top 500

Topologies

- Network topology plays a critical w.r.t. overall cost
  - Each endpoint requires multiple links and switch ports depending on topology
  - Packaging considerations

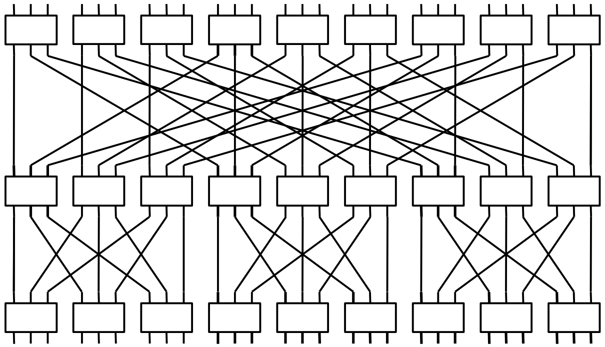
- We consider high-radix, low(ish)-diameter topologies only
  - Low diameter means lower cost, because fewer links and switch ports per end point
  - Fewer hops means lower latency
  - Discrete, high-radix switches

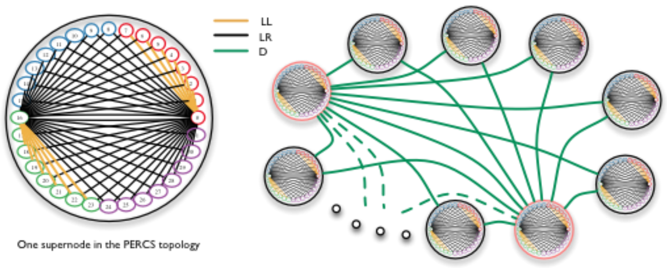
- Topologies
  - Fat tree: two-level and three-level
  - Dragonfly: two-tier and three-tier
  - Multi-layer full mesh (aka stacked all-to-all)
  - “Dragontree”
  - Slim fly
  - 3D HyperX

- Metrics
  - Scale S: number of endpoints
  - Diameter D: max. number of links across all shortest paths
  - Number of links per endpoint L
  - Number of switch ports per endpoint M

Topologies (1)

Fat tree	*Dragonfly*





Tier-1 group: full mesh of switches

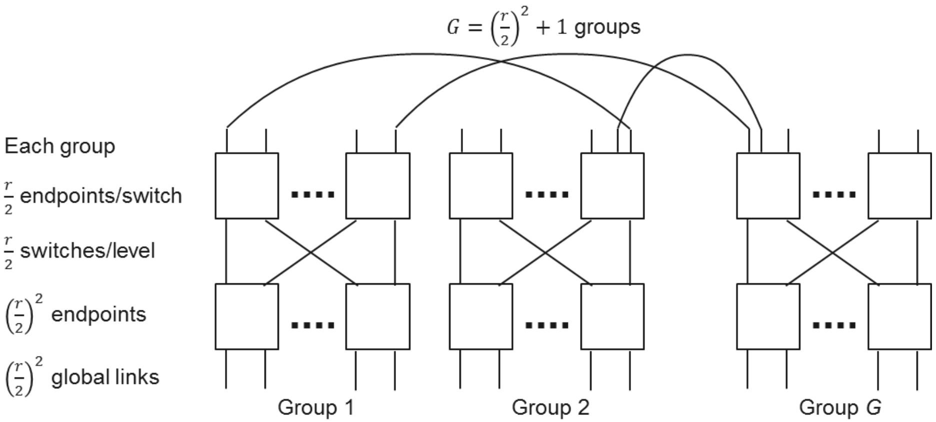
Tier-2: full mesh of tier-1 groups

- *k*-ary *n*-tree
- Max scale $S = N\left(\frac{r}{2}\right)^{n-1}$, where n is the number of levels
- Two-level: $D = 2,\ L = 2,\ M = 3$
- Three-level: $D = 4,\ L = 3,\ M = 5$

- Recursive structure: at each tier, sub-groups form a full mesh
- Max scale $S_{2t} \approx \frac{1}{64}r^4$; $S_{3t} \approx \frac{1}{16{,}384}r^8$
- Two-tier: $D = 3,\ L = 2.5,\ M = 4$
- Three-tier: $D = 7,\ L = 4.5,\ M = 8$

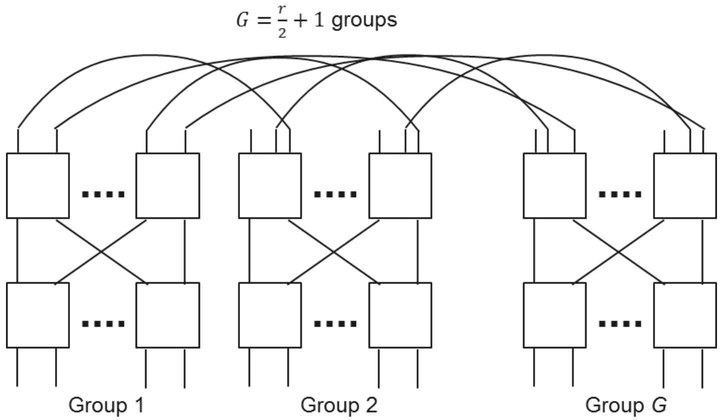
Topologies (2)

Dragontree



- Two-tier dragonfly where intra-group topology is a two-level fat tree instead of a full mesh
- $S \approx \left(\frac{r}{2}\right)^4$
- $D = 3,\ L = 2.5,\ M = 4$

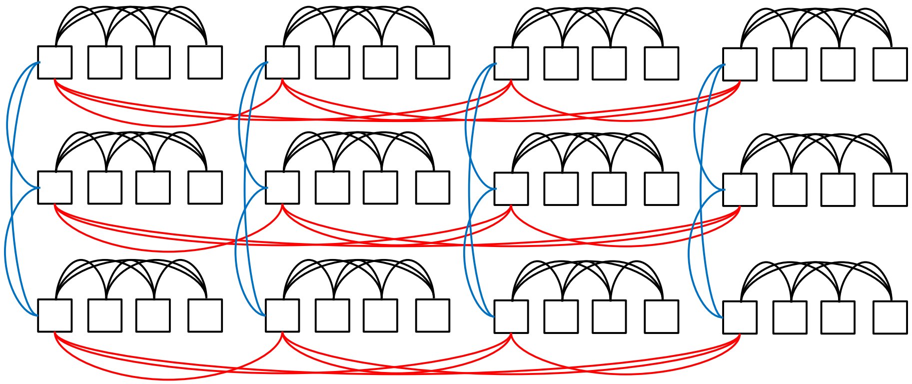
Dragontree* (with bundling)

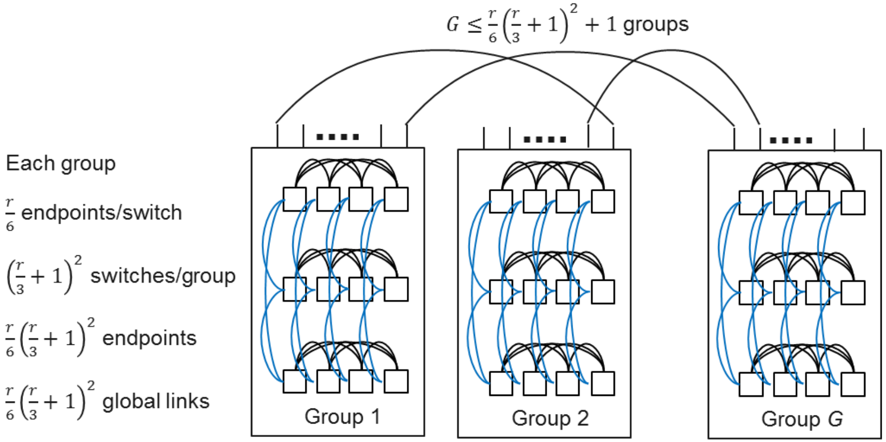


- Same, but using multiple $\left(\frac{r}{2}\right)$ links in between each pair of groups
- $S \approx \left(\frac{r}{2}\right)^3$
- $D = 3,\ L = 2.5,\ M = 4$

Topologies (3)

3D HyperX	*DragonFB*



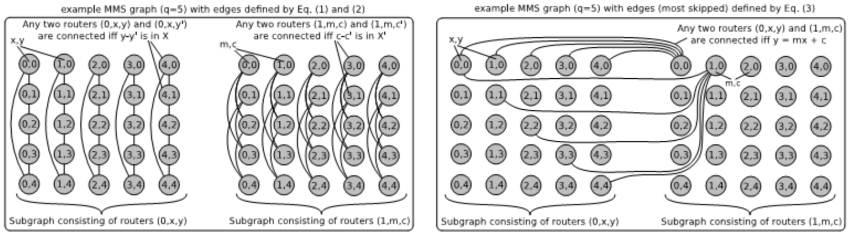


- Three-dimensional generalized hypercube aka flattened butterfly aka HyperX
- $S \approx \frac{1}{256} N^4$
- $D = 3,\ L = 2.5,\ M = 4$

- Two-tier dragonfly where intra-group topology is a 2D Generalized Hypercube instead of a full mesh
- $S \approx \left(\frac{r}{6}\right)^2 \left(\frac{r}{3}+1\right)^4 \approx \frac{r^6}{2916}$
- $D = 5,\ L = 3.5,\ M = 6$

Topologies (4)

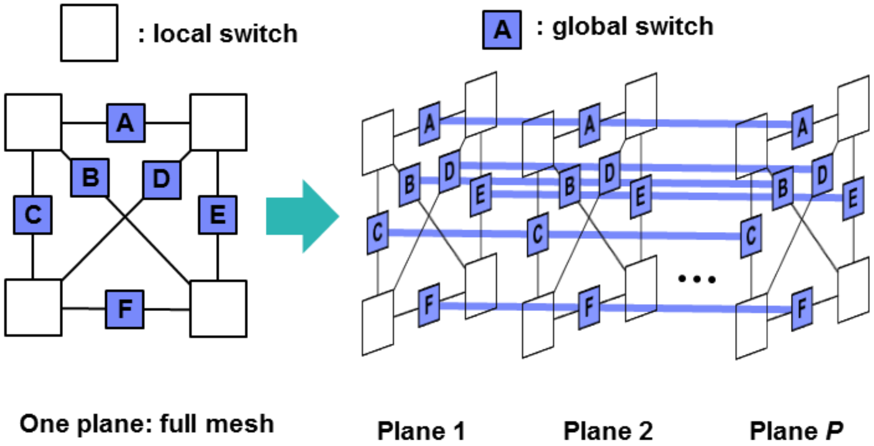
Slim fly



Source: M. Besta & T. Hoefler, "Slim Fly: A cost-effective low-diameter network topology," SC 2014

- Based on McKay-Miller-Širán (MMS) graphs
- $S \approx \left(\frac{N}{2}\right)^3$
- $D = 2,\ L = 2,\ M = 3$

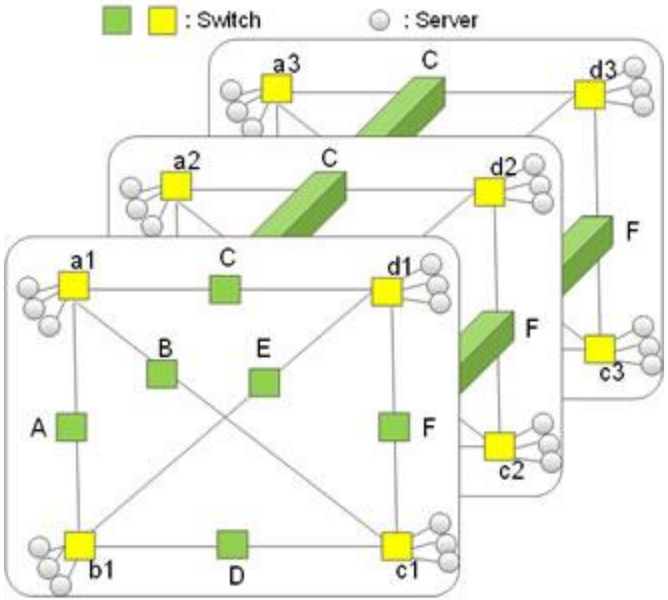
Stacked all-to-all aka multi-level full mesh



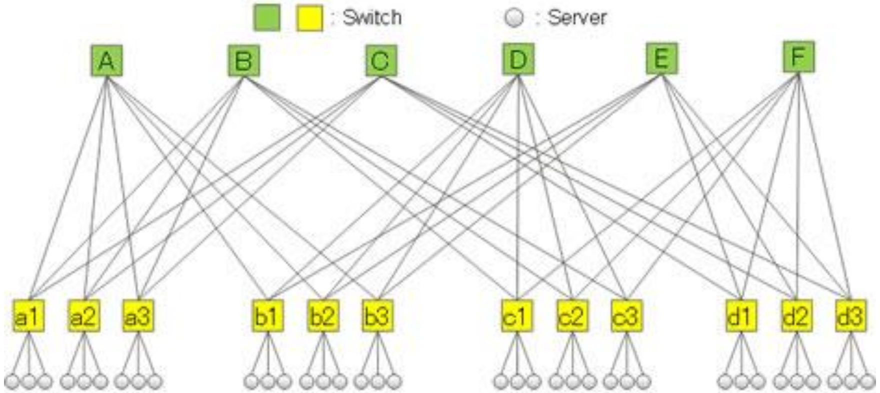
- Start from a full mesh; insert a global switch in each link of the mesh; stack multiple planes connected via the global switches
- $S \approx \left(\frac{N}{2}\right)^3$
- $D = 2,\ L = 2,\ M = 3$

Stacked All-to-all

“Stacked” representation

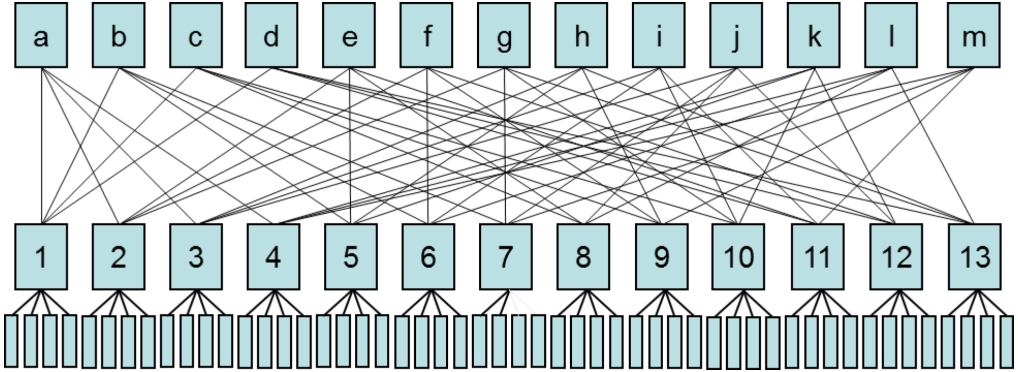


Tree representation



Source: Fujitsu, http://www.fujitsu.com/global/about/resources/news/press-releases/2014/0715-02.html

Orthogonal fat tree



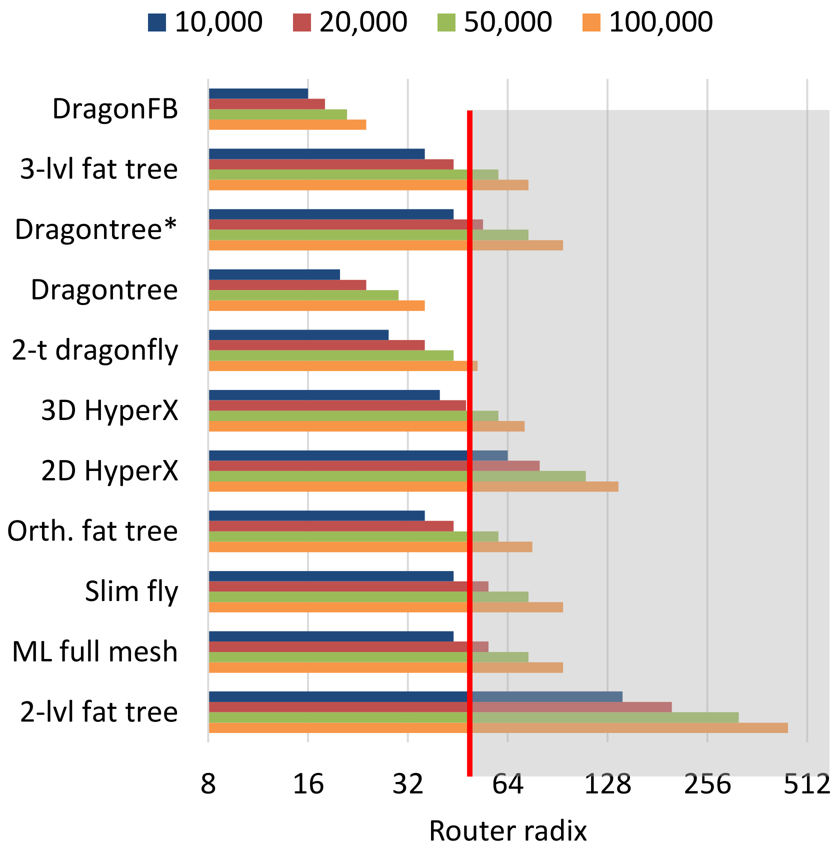
- M. Valerio, L. E. Moser and P. M. Melliar-Smith, “Recursively Scalable Fat-Trees as Interconnection Networks,” *IEEE 13th Annual Int’l Phoenix Conf. on Computers and Communications,* pp.40, 12-15 April 1994
- Trade (more) scale for (less) path diversity; construction is related to Latin Squares
- Indirect topology – diameter 2 among endpoints; diameter 3 among switches!
- $S = 2(k^3 - k^2 + k),\ D = 2,\ L = 2,\ M = 3$: twice the scale of MLFM/SF at same cost/endpoint

High-level topology comparison

Topology	Diameter		Maximum scale N				#links /endpoint	#ports/ endpoint
	dir	in	r	r = 36	r = 48	r = 64	L	
2-level Fat Tree	2	-	$\frac{r^2}{2}$	648	1152	2,048	2	3
Multi-layer Full Mesh	2	4	$\approx \frac{r^3}{8}$	6,156	14,400	33,792	2	3
Slim Fly	2	4	$\approx \frac{r^3}{8}$	6,144	14,112	32,928	2	3
Orthogonal fat tree	2	4	$\approx \frac{r^3}{4}$	11,052	26,544	63,552	2	3
3D HyperX	3	6	$\approx \frac{r^4}{256}$	9,000	26,364	78,608	2.5	4
2-tier Dragonfly	3	5 (6)	$\approx \frac{r^4}{64}$	29,412	90,300	279,312	2.5	4
Dragontree	3	6	$\approx \frac{r^4}{16}$	105,300	332,352	1 M	2.5	4
Dragontree*	3	4	$\approx \frac{r^3}{16}$	6,156	14,400	33,792	2.5	4
3-level Fat Tree	4	-	$\frac{r^3}{4}$	11,664	27,648	65,536	3	5
DragonFB (Aries)	5	8 (10)	$\approx \frac{r^6}{2,916}$	1M	$\gg$ 1M	$\gg$ 1M	3.5	6
3-tier Dragonfly	7	11 (14)	$\approx \frac{r^8}{16,384}$	$\gg$ 1M	$\gg$ 1M	$\gg$ 1M	4.5	8

Scalability

- Number of switch ports to scale to a given number of endpoints
  - Balanced network configuration: full uniform all-to-all bandwidth
- Commercially available switches are expected to have 36-48 ports
- 10,000-15,000 endpoint network provides significantly more freedom of choice w.r.t. topology
- Larger switch radix is generally better, but only if it enables smaller diameter!



Router radix required to scale to 10K, 20K, 50K, 100K endpoints

Partitionability

- Ability to divide a topology into non-interfering parts
- Main benefit is performance isolation
- Topologies that can naturally provide this: Fat trees, Multi-layer Full Mesh
- Topologies that could provide this by using slow Optical Circuit Switching: Dragonflies, HyperX, Dragontree*, DragonFB

- Not all customers care about this, YMMV

Routing algorithms

Generic routing algorithms

- **Direct:** Shortest path; adaptive load-balancing based on local queue lengths across multiple shortest paths

- **Valiant:** Indirect routing with topology-aware selection of intermediate destination to avoid unproductive hops; direct routing is applied on both segments of the Valiant path
  - Not applicable to Fat Tree
  - Never route indirectly when source and destination attached to same switch, or are within same group in Dragontree*
  - "Optimized" Dragontree* : Second-level switch can be selected as intermediate destination, eliminating down-up hops in intermediate group
  - Multi-layer full mesh: Only endpoint switches are eligible as intermediate destination

- **Adaptive:** Universal Global Adaptive Load-balanced routing: Decides whether to take Direct or Valiant path based on local queue lengths
  - Not applicable to Fat Tree (load-balance adaptively across direct paths)
  - "Optimized" Dragontree* : Decision taken at second-level switch
  - Multi-layer full mesh: Decision taken at local switch (first hop)

Adaptive routing parameters

- Number of direct paths *D*
  - Compute average output queue length *Ld* across *D* direct-path output queues
  - *D* = 1 or *D* = all
- Threshold *T*
  - If *Ld* < *T* then route to lowest cost direct path
- Number of indirect paths *I*
  - Randomly select up to *I* intermediate destinations and determine the corresponding ports to go there (eliminate already selected ports and direct ports)
  - Compute average output queue length *Li* of *I* indirect-path output queues
- Weight *W*
  - If T ≤ *Ld* ≤ *W**Li* then route to lowest cost direct path, otherwise to intermediate destination with lowest cost

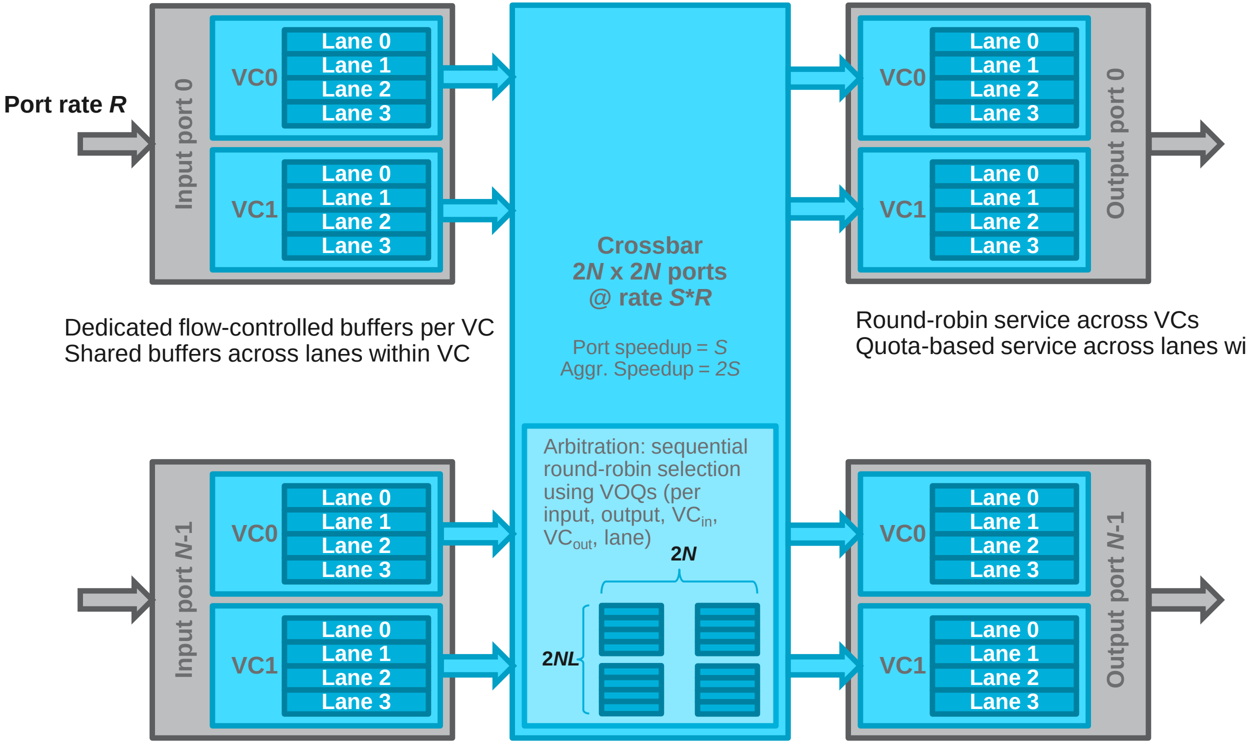
- Number of direct paths *D*
  - *D* = all
  - We consider ALL direct paths, because we need to evaluate them for direct path load-balancing anyway
- Threshold *T*
  - *T* = 10 KB
  - Prevent indirect routing when backlog is very small
- Number of indirect paths *I*
  - *I* = 1
  - We consider ONE direct path to reduce complexity
- Weight *W*
  - W = 2
  - Higher weight to indirect paths to avoid unnecessary detours (latency)
- Settings selected based on sensitivity analysis
  - To be included in final report

Performance evaluation

Topologies

- Fat tree
  - 24-ary 3-three using radix-48 switches
  - 24 level-2 switches x 24 level-1 switches x 24 endpoints = 13,824 endpoints
  - Serves as performance benchmark
- Dragontree*
  - Radix-48 switches
  - 24 groups x 24 level-1 switches x 24 endpoints = 13,824 endpoints
  - One group unpopulated: slight imbalance for direct routing (indirect can use links to unpopulated group)
- Multi-layer full mesh
  - Radix-47 local switches; radix-48 global switches
  - 24 planes x 24 switches x 24 endpoints = 13,824 endpoints
  - Slight imbalance (23/24) within plane

Combined input-output-queued switch model



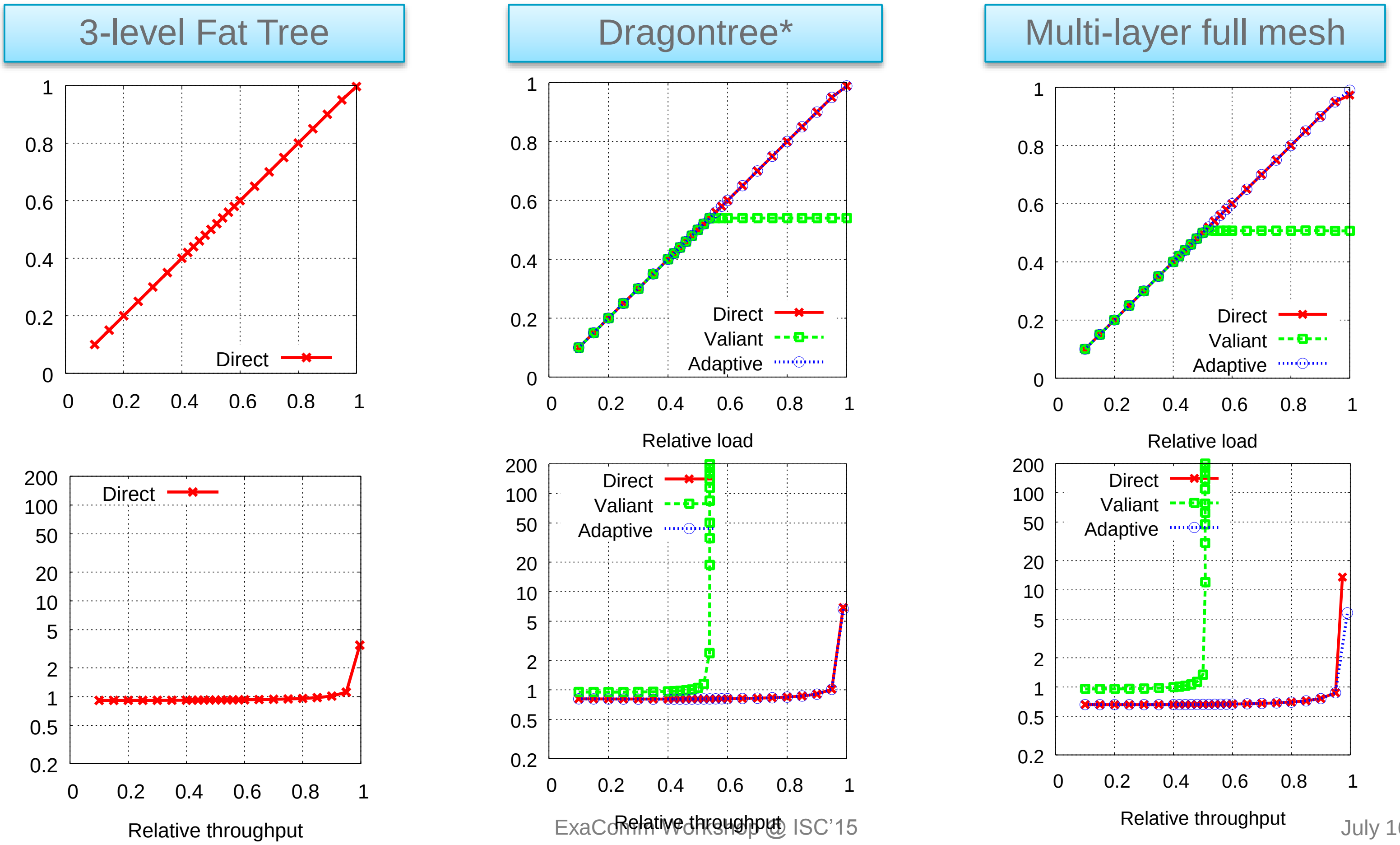
Simulation parameters

- Max. simulated time (uniform traffic) = 1 ms
- Statistics collection interval = 10 us
- Uniform traffic
  - Message size = 512 B
  - Interarrival time @ 100% load = 10.24 ns
- Switch
  - Packet size = 512 B; packet duration = 10.24 ns
  - Per-port buffer size = 50 KB input + 50 KB output
  - Ports per buffer = 2
  - Internal speedup = 1.5x
  - Number of virtual channels = 2
- Adapter buffer size (uniform traffic): 200 KB input + 200 KB output
  - Packet size = 512 B; packet duration = 10.24 ns
  - Interleaving threshold = 512 B
- Latencies
  - Switch traversal = 100 ns
  - Adapter traversal = 100 ns
  - NIC to switch = 10 ns
  - Switch to switch = 50 ns
- Reordering
  - Disabled for random uniform/shift patterns
  - Enabled for exchange patterns
- Routing
  - Direct
  - Valiant
  - Adaptive

Uniform and adversarial traffic

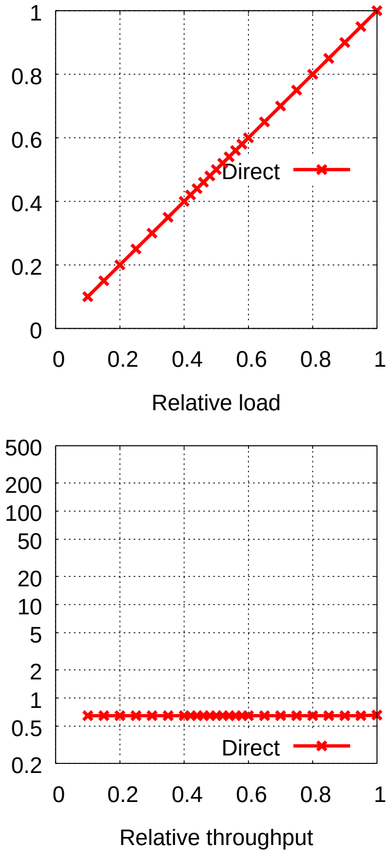
Fat Tree, Dragontree* and multi-layer full mesh

Uniform random traffic for 6,156 endpoints

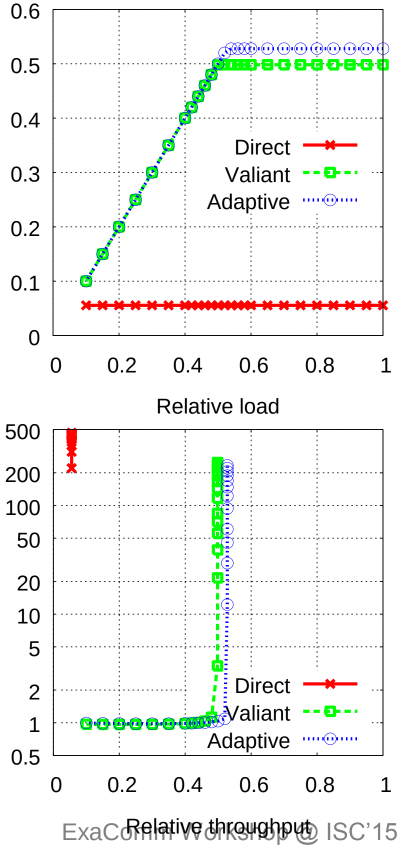


Adversarial traffic for 6,156 endpoints

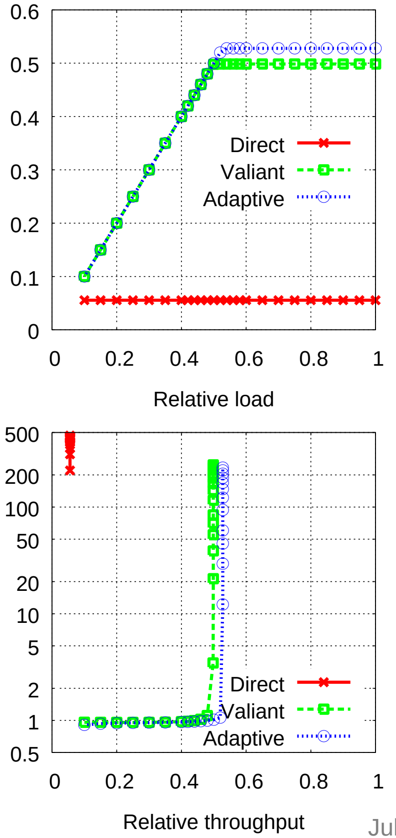
3-level Fat Tree



Dragontree*



Multi-layer full mesh



Exchange patterns

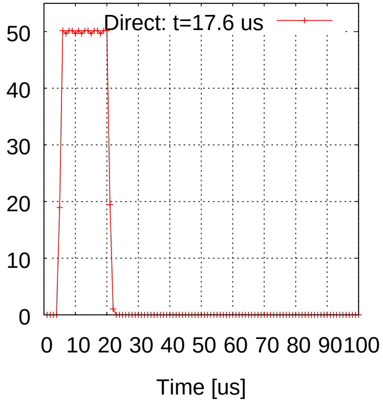
Nearest neighbor and dimension-wise all-to-all

Exchange patterns for 13,824 endpoints

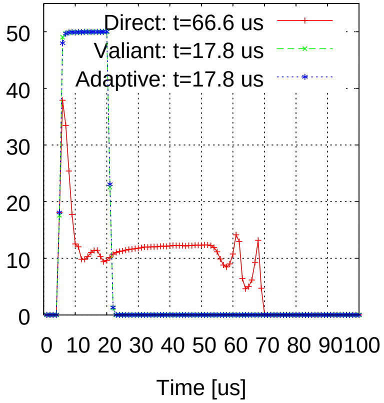
- Nearest neighbor exchange
  - Simulated tasks form a 3D torus topology
  - Each task sends one message to both neighbors along each dimension
  - Total number of message per task = 6
  - 1 task per network endpoint
- Dimension-wise all-to-all along X, Y, or Z
  - Simulated tasks from a 3D torus topology
  - X: Each task sends one message to each other task with the same Y and Z coordinates
  - Y: Each task sends one message to each other task with the same X and Z coordinates
  - Z: Each task sends one message to each other task with the same X and Y coordinates
  - Total number of message per task = #X+#Y+#Z-3
  - 1 task per network endpoint
- Torus geometry is selected to match network topology hierarchy
  - X within switch
  - Y within subtree, group or plane
  - Z across subtrees, groups, or planes

Nearest neighbor, 128 KB

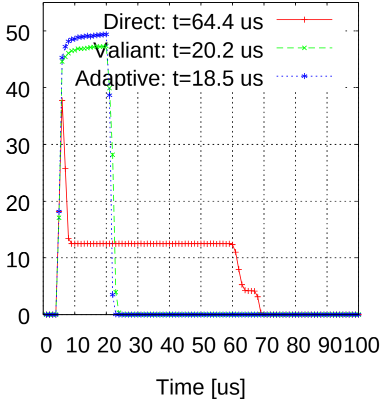
3-level fat tree



Dragontree*

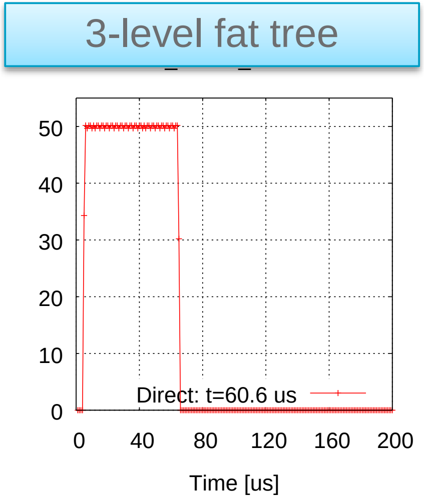


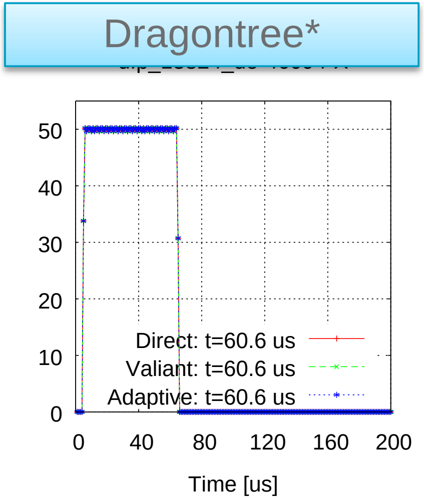
Multi-layer full mesh

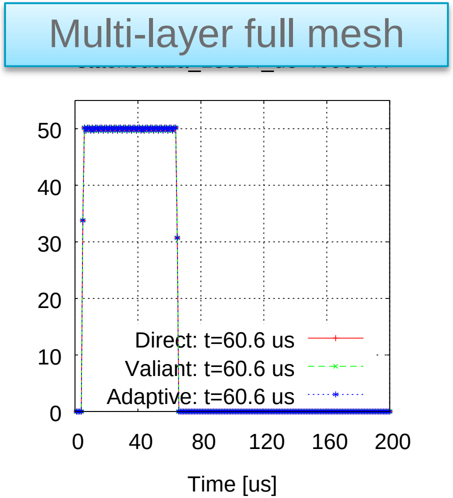


- Fat tree behaves ideal
- Dragontree*: direct routing suffers contention along Z axis; valiant and adaptive close to ideal
- MLFM: direct routing suffers contention along Y axis; adaptive best

Dimension-wise exchange along X, 128 KB



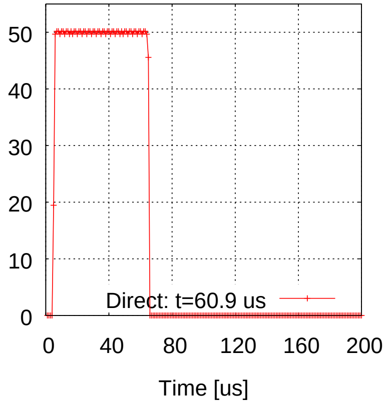




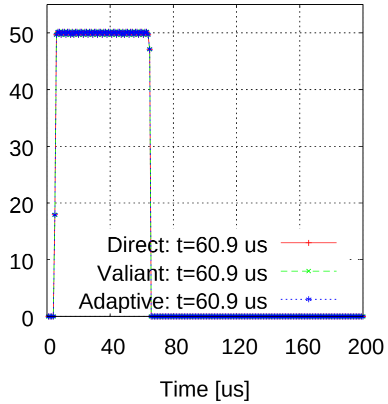
- All messages stay within the local switch, hence ideal throughput in all cases

Dimension-wise exchange along Y; 128 KB

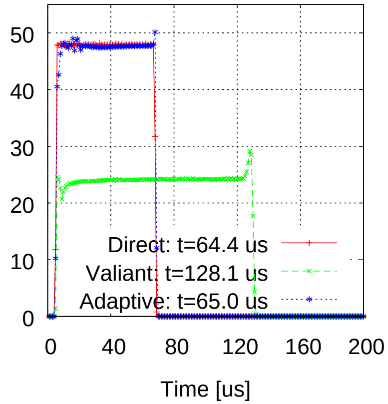
3-level fat tree



Dragontree*



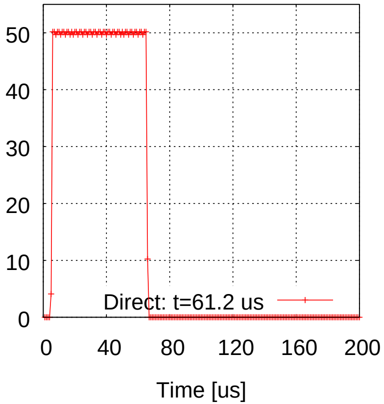
Multi-layer full mesh



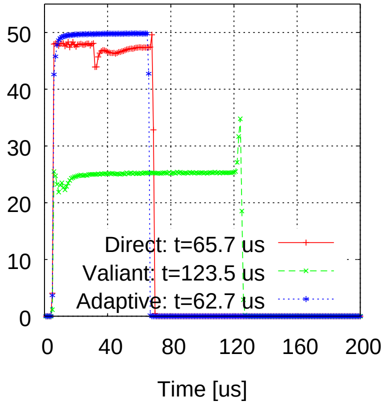
- Fat tree ideal
- Dragontree* ideal with any routing: all messages stay within group, hence full bandwidth
- MLFM: all messages within plane; Direct and adaptive almost but not quite ideal because per switch there are only 23 local links but 24 endpoints; valiant halves bandwidth

Dimension-wise exchange along Z; 128 KB

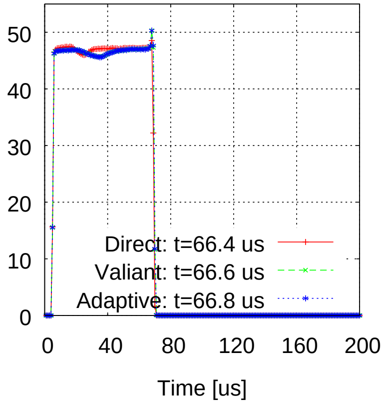
3-level fat tree



Dragontree*



Multi-layer full mesh



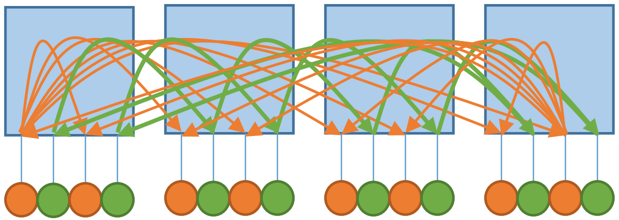
- Fat tree ideal
- Dragontree*: direct slightly less than ideal (only 23 links to every other groups but 24 endpoints); valiant halves bandwidth; adaptive close to ideal
- MLFM: all routings perform similarly; not quite full throughput (why?)

Mixed pattern

Interleaved uniform random + permutation traffic

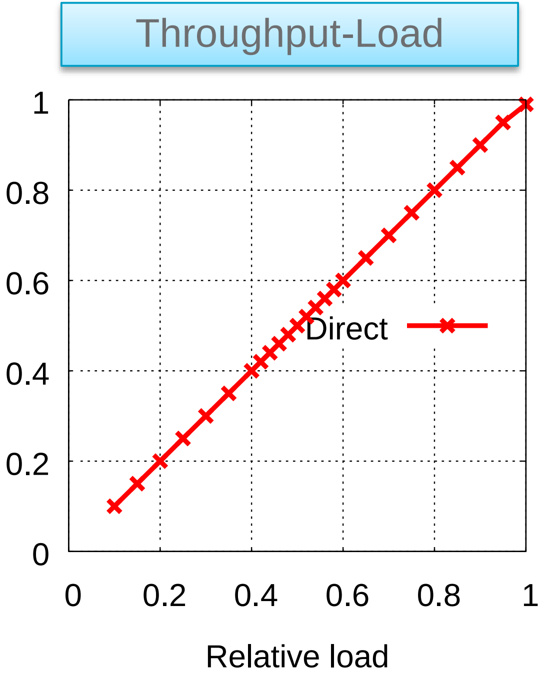
Mixed uniform random + permutation traffic

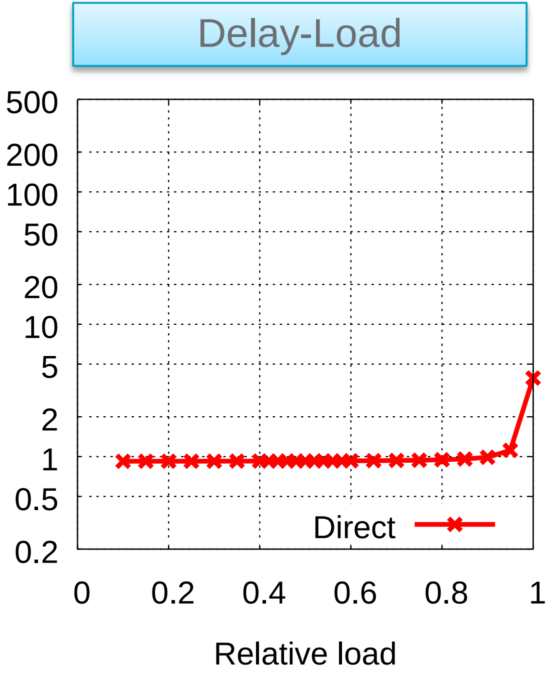
- *N* endpoints total, two workloads of *N*/2 ranks each, 1 rank per endpoint
  - Random uniform across *N*/2 ranks
  - Shift permutation across *N*/2 ranks
  - Workload ranks interleaved one by one across endpoints

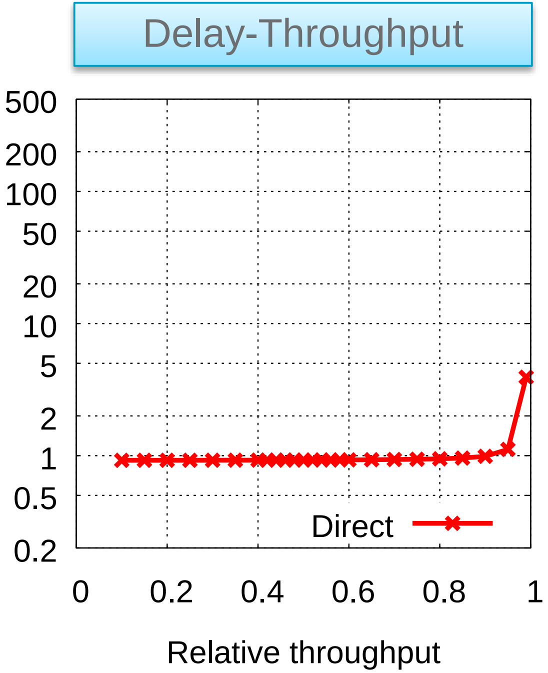


Uniform random   Shift permutation

Mixed Traffic Fat Tree: 6,156 endpoints

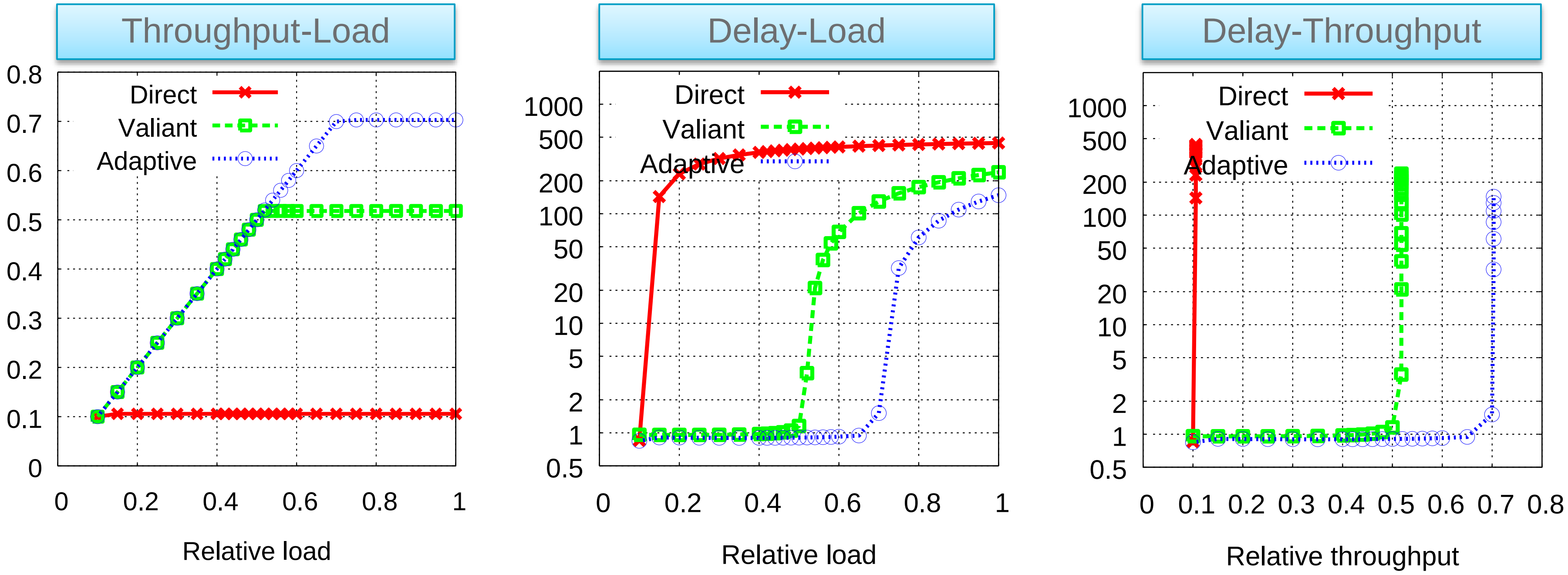






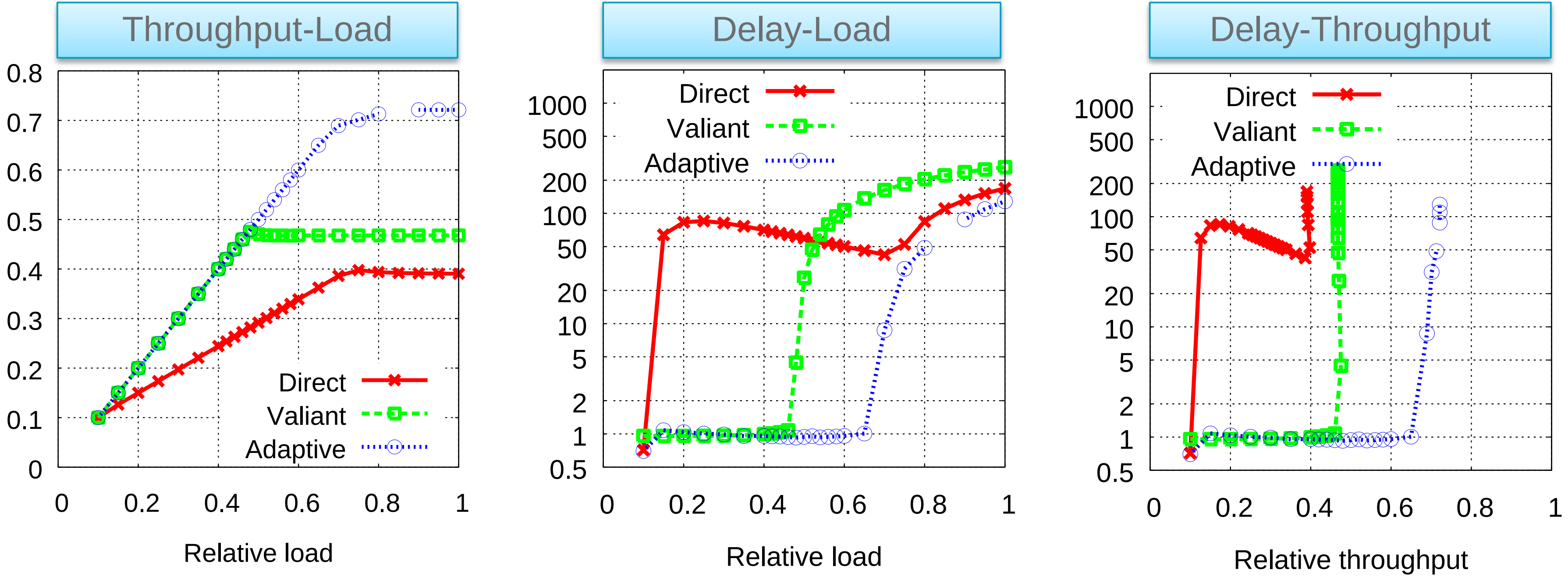
- perm_shift_size=162, perm_grp_size = 0

Mixed Traffic Dragontree*: 6,156 endpoints



- perm_shift_size=162, perm_grp_size = 0

Mixed Traffic Multi-layer Full Mesh: 6,156 endpoints



- perm_shift_size=9, perm_grp_size = 171

Conclusions

- Cost is major constraint on the system balance
- Byte per FLOP ratios can be expected to drop significantly for exascale systems
- Increasing node fatness implies that scale is less of an issue
- Diameter-2 or -3 topologies with 2 or 2.5 links and 3 or 4 ports per endpoint are a viable option given radix-48 switches
- Fat tree is the gold standard performance standard
- Performance-wise, these networks can be on par with the more expensive and higher-diameter 3-level fat tree
  - Indirect and adaptive routing is a **must**
  - Half the performance of fat tree for adversarial patterns
- Next step: Apply more realistic workload patterns via traces (extrae/paraver) and mini-apps (Ember motifs).

Thank you!

Exascale network challenges

1. **Cost**
2. **Balance: Dealing with bandwidth-challenged systems**
3. **Bandwidth density: Packaging**
4. **Energy**
5. **Reliability**