Parallel Programming of High-Performance Systems

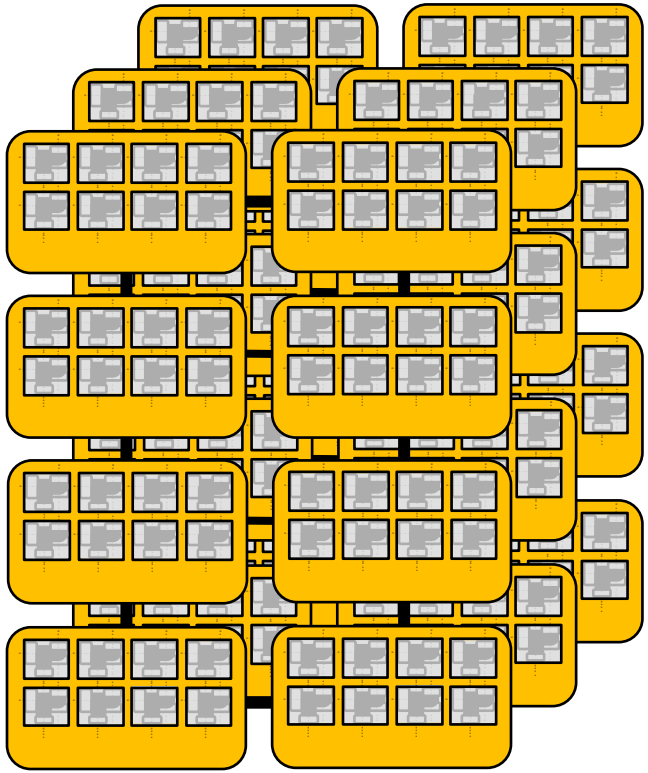
A collaborative course of NHR@FAU and LRZ Garching

Georg Hager, Volker Weinberg, Ayesha Afzal, Markus Wittmann

Distributed-Memory Computer Architecture

High Performance Computing

Distributed memory: no cache-coherent single address space



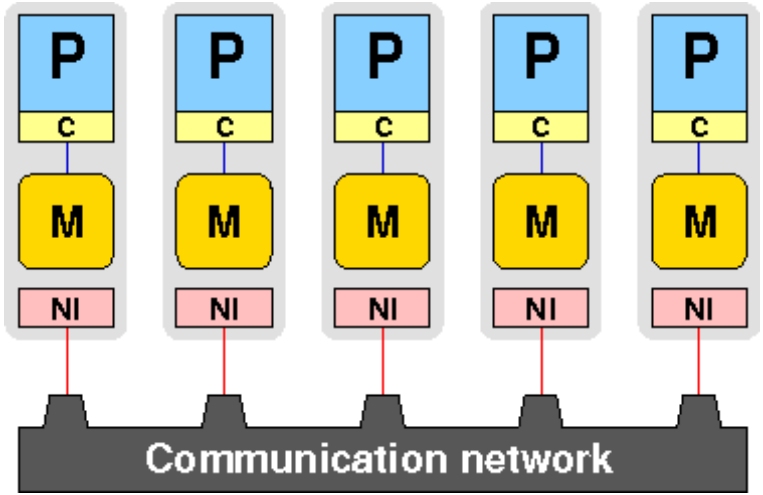
Cluster/
supercomputer

Modern supercomputers are
shared-/distributed-memory hybrids

Distributed-memory systems “back in the day”

“Pure” distributed-memory system:

- Individual processors with exclusive local memory (M) and a network interface (NI) → one “node” == one processor core
- Dedicated communication network
- Parallel program == one process per node
- Data exchange via “message passing” over the network

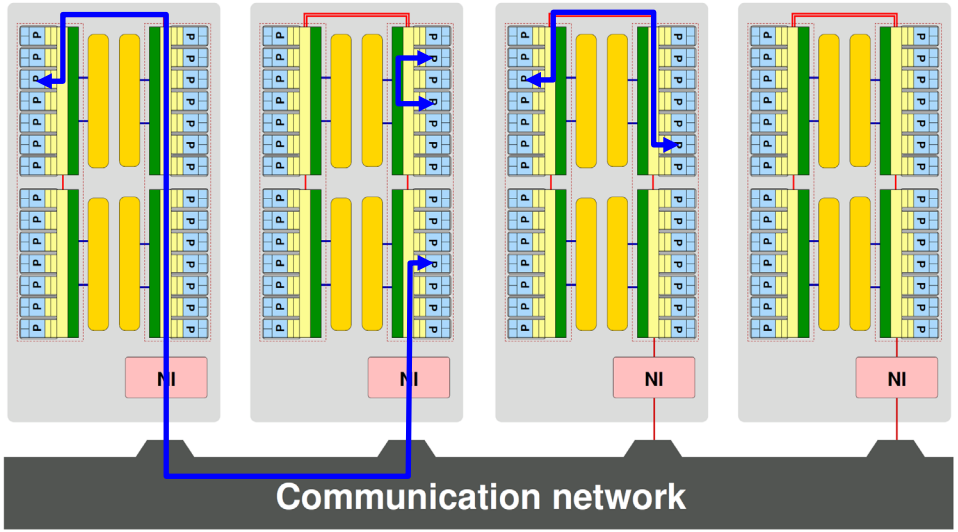


- This was a thing not so long ago…

Distributed-memory systems today

"Hybrid" distributed-/shared-memory systems

- Cluster of networked shared-memory nodes
- ccNUMA architecture per node
- Multiple cores per ccNUMA domain



- Expect strong topology effects in communication performance
  - Intra-socket, inter-socket, inter-node, all have different λ and b
  - On top: Effects from network structure

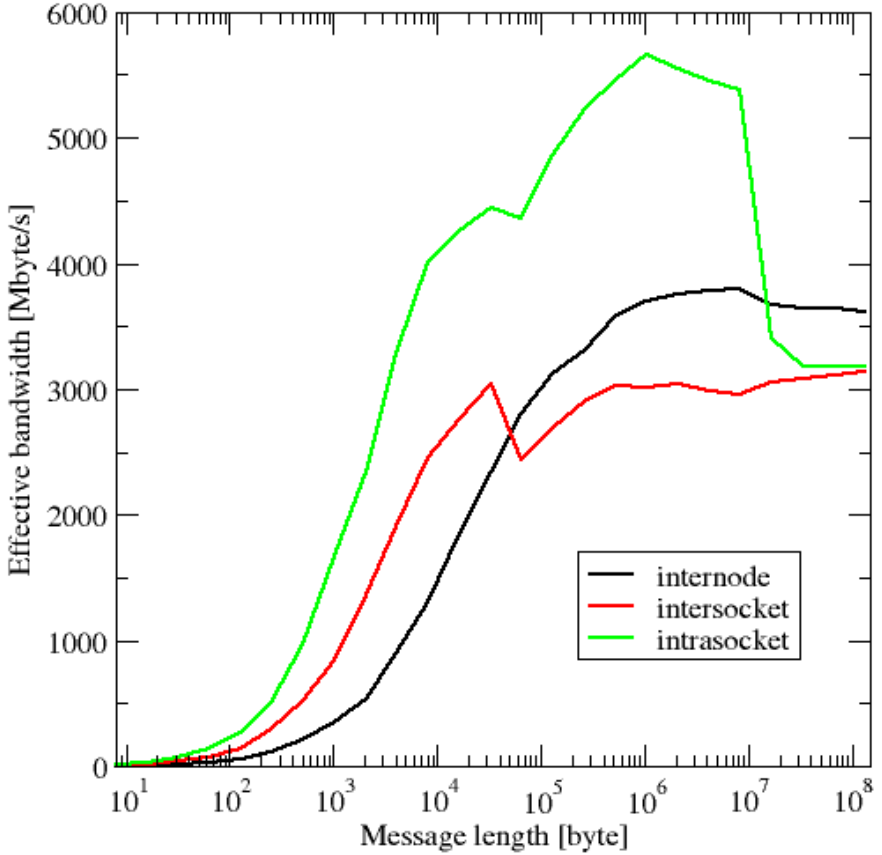
Point-to-point data transmission performance

- Simple "Hockney model" for data transfer time

$$T_{comm} = \lambda + \frac{V}{b}, \quad B_{\text{eff}} = \frac{V}{T_{comm}}$$

λ: latency, b: asymptotic BW

- Reality is more complicated
  - System topology
  - Caching effects
  - Contention effects
  - Protocol switches



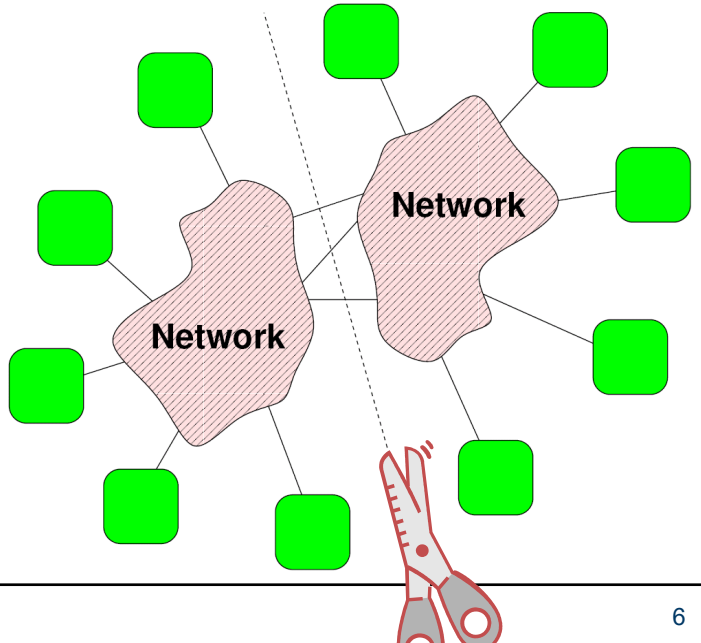
Characterizing communication networks

- Network bisection bandwidth B_b is a general metric for the data transfer "capability" of a system:

  Minimum sum of the bandwidths of all connections cut when splitting the system into two equal parts

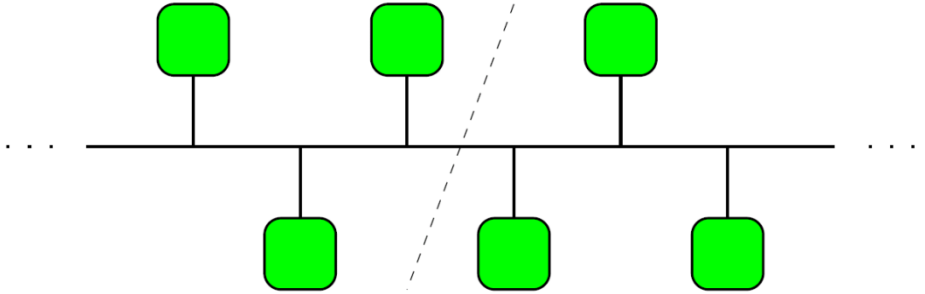
- More meaningful metric for system scalability: bisection BW per node: B_b/N_{nodes}

- Bisection BW depends on
  - Bandwidth per link
  - Network topology



Network topologies: bus

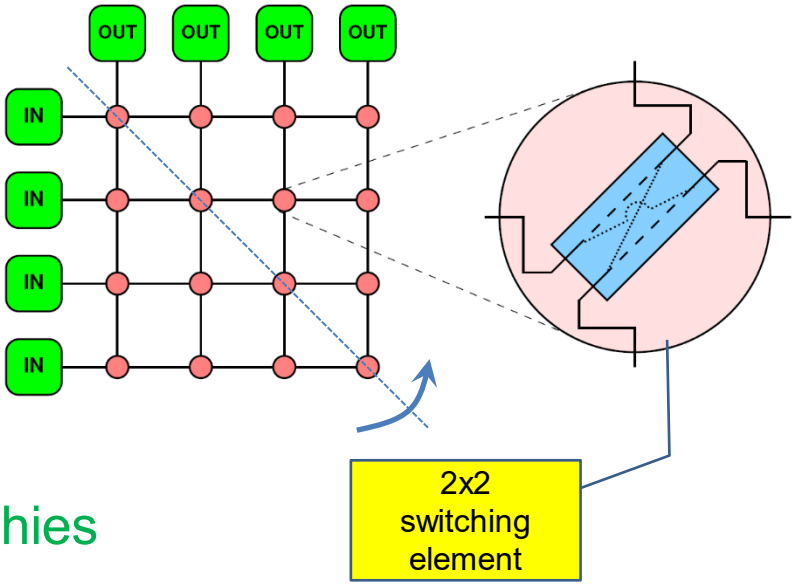
- Bus can be used by one connection at a time
- Bandwidth is shared among all devices
- Bisection BW is constant → $B_b/N_{nodes} \sim 1/N_{nodes}$
- Examples: diagnostic buses, old Ethernet network with hubs, Wi-Fi channel



- Advantages
  - Low latency
  - Easy to implement
- Disadvantages
  - Shared bandwidth, not scalable
  - Problems with failure resiliency (one defective agent may block bus)
  - Large signal power per agent

Network topologies: non-blocking crossbar

- Non-blocking crossbar can mediate a number of connections among groups of input and output elements
- This can be used as a n-port non-blocking switch (fold at the secondary diagonal)



- Switches can be cascaded to form hierarchies (common case)
  - Allows scalable communication at high hardware/energy costs
  - Crossbars are rarely used as interconnects for large scale computers
    - NEC SX9 vector system (“IXS”)

Network topologies: switches and fat trees

- Standard clusters are built with switched networks
- Compute nodes (“devices”) are split up in groups – each group is connected to single (non-blocking crossbar-)switch (“leaf switches”)
- Leaf switches are connected with each other using an additional switch hierarchy (“spine switches”) or directly (for small configurations)
- Switched networks: “Distance” between any two devices is heterogeneous (number of “hops” in switch hierarchy)
- Diameter of network: The maximum number of hops required to connect two arbitrary devices (e.g., diameter of bus=1)

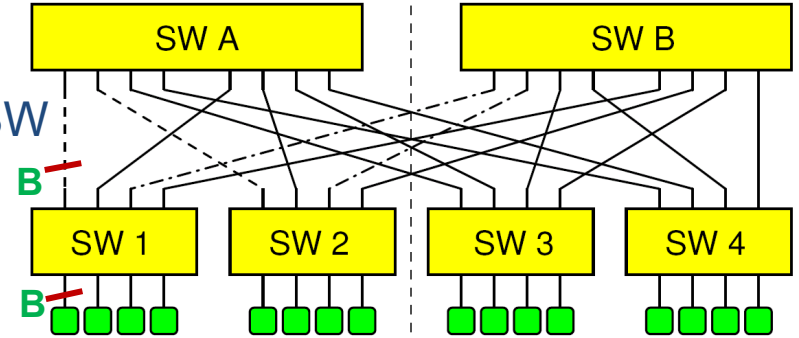
- “Perfect” world: “Fully non-blocking”, i.e. any choice of $N_{nodes}/2$ disjoint node (device) pairs can communicate at full speed

Fat tree switch hierarchies

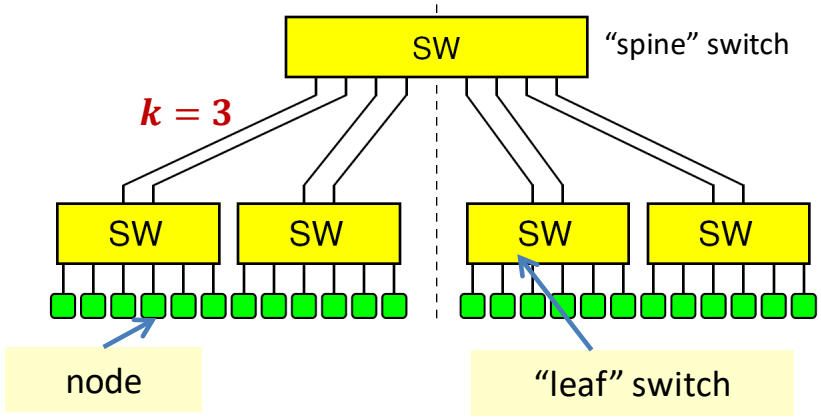
- "Fully non-blocking"
  - $N_{nodes}/2$ end-to-end con-nections with full BW

    → $B_b = B \times N_{nodes}/2$, $B_b/N_{nodes} = B/2$

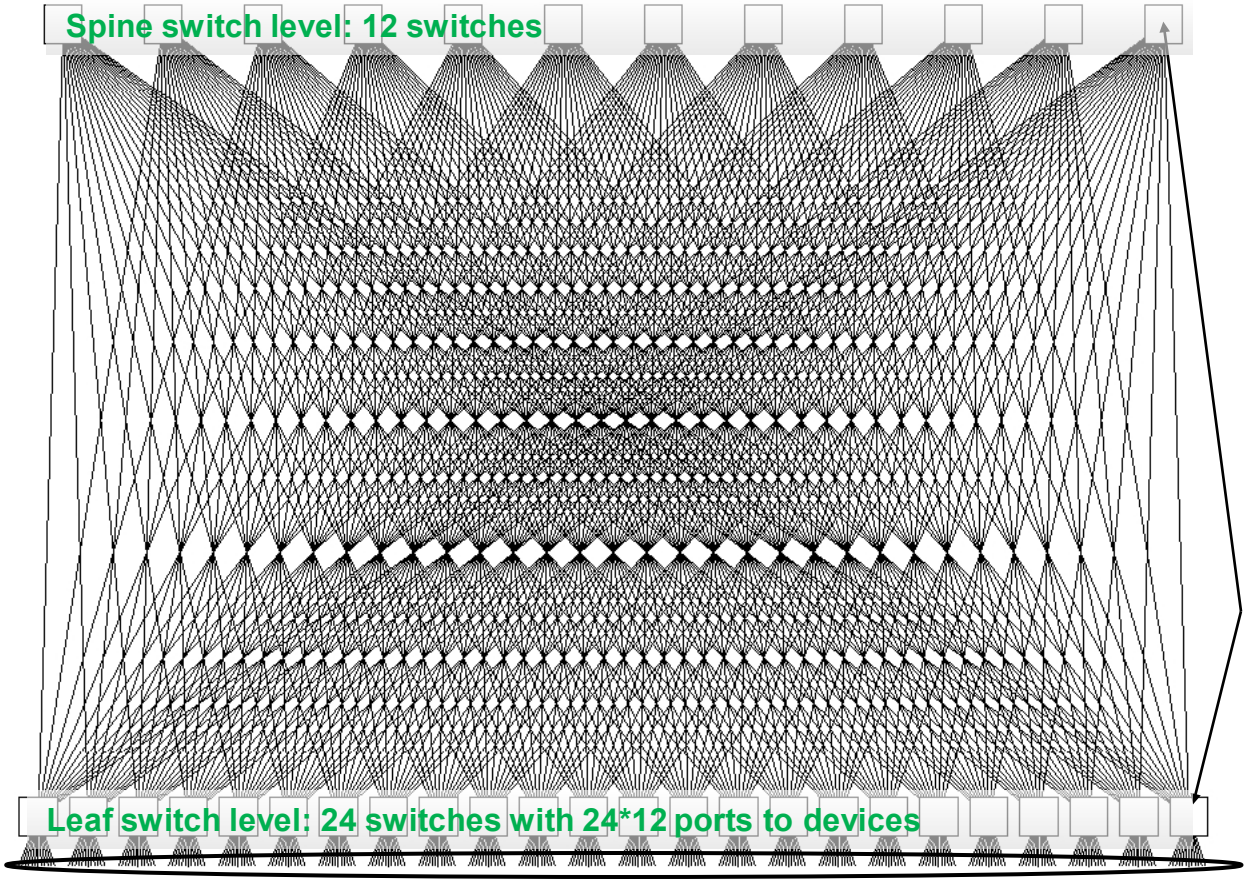
  - Sounds good, but see next slide



- "Oversubscribed"
  - Spine does not support $N_{nodes}/2$ full BW end-to-end connections
  - $B_b/N_{nodes} = const. = B/(2k)$, with k = oversubscription factor
  - Resource management (job placement) is crucial



A “single” 288-port InfiniBand DDR switch

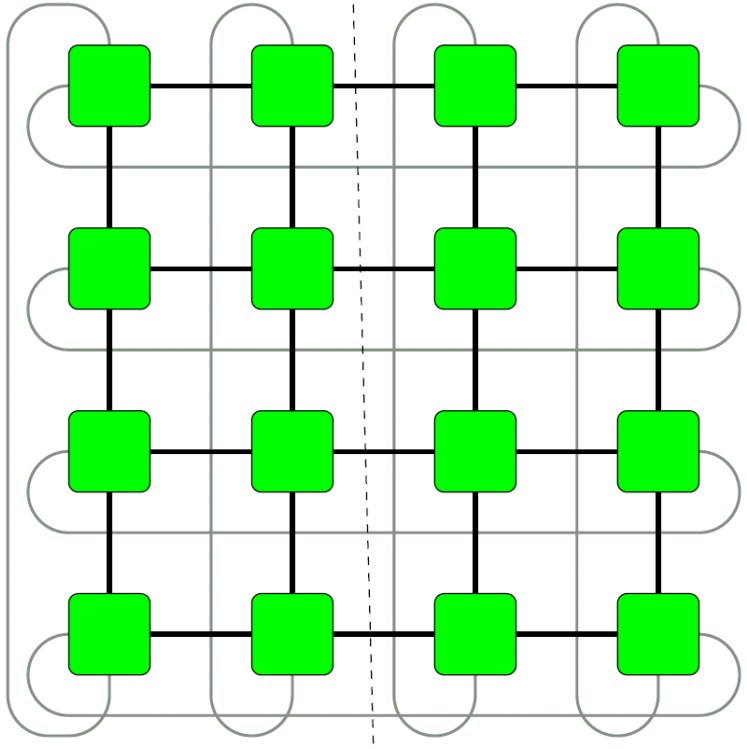


Examples for fat tree networks in HPC

- Ethernet
  - 1, 10, 25, and 100 Gbit/s variants
- InfiniBand: Dominant high-performance “commodity” interconnect
  - DDR: 20 Gbit/s per link and direction (Building blocks: 24-port switches)
  - QDR: 40 Gbit/s per link and direction, building blocks: 36-port switches
    → “Large” 36x18=648-port switches
  - FDR-10 / FDR: 40/56 Gbit/s per link and direction
  - EDR: 100 Gbit/s per link and direction, HDR: 200 Gbit/s
- Expensive & complex to scale to very high node counts

Mesh networks

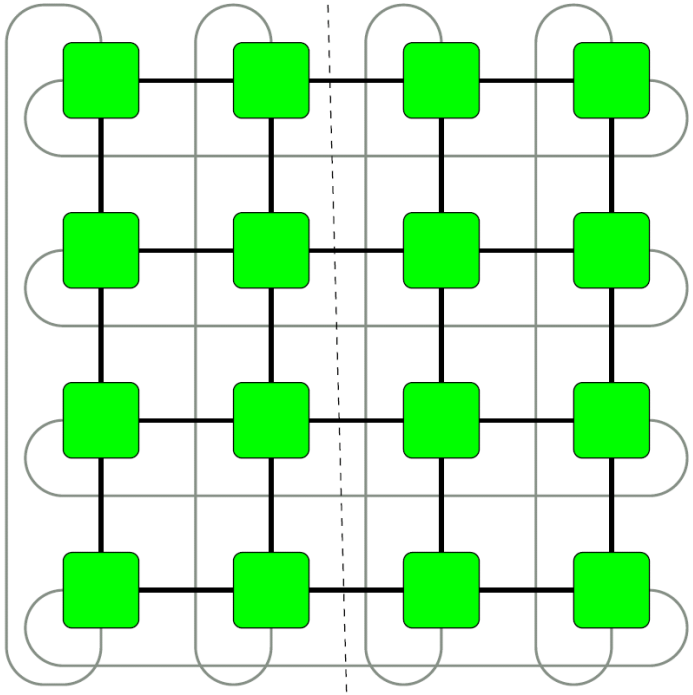
- Fat trees can become prohibitively expensive in large systems
- Compromise: Meshes
  - n-dimensional Hypercubes
  - Toruses (2D / 3D)
  - Dragonfly
  - Many others (including hybrids)



Example: 2D torus mesh

2D torus mesh

- This is not a non-blocking corossbar!
  - Intelligent resource management and routing algorithms are essential
- Direct connections only between direct neighbors
  - Each node is/has a router
- Toruses in very large systems: Cray XE/XK series, IBM Blue Gene
  - $B_b \sim N_{nodes}^{(d-1)/d}$ → $B_b/N_{nodes} \to 0$ for large N_{nodes}
  - Sounds bad, but those machines show good scaling for many codes
  - Well-defined and predictable bandwidth behavior!



HPE Slingshot (Dragonfly topology)

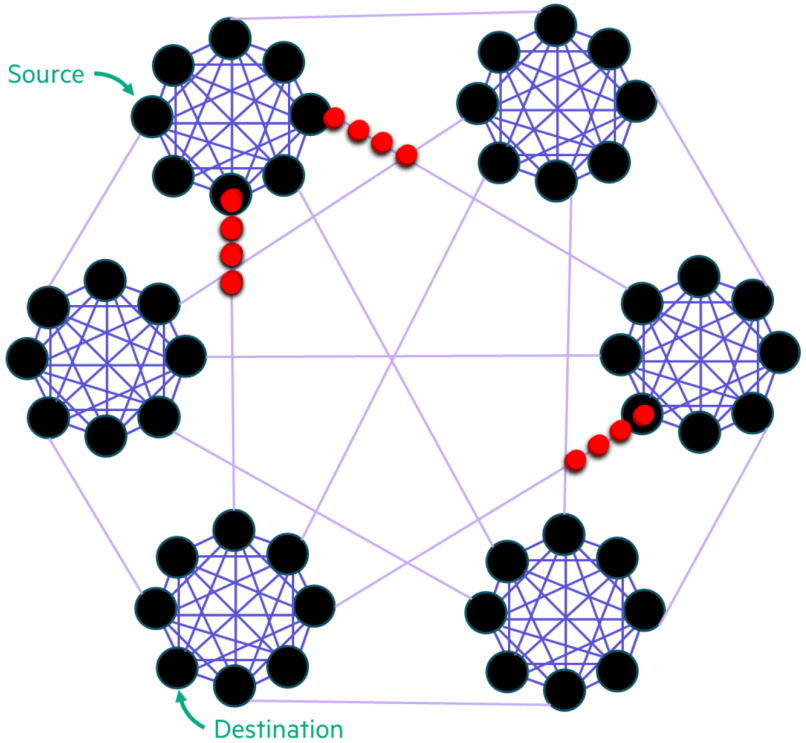
HPE SLINGSHOT

Dragonfly Network Architecture
- Packet-by-packet routing of unordered traffic (e.g. MPI/Lustre bulk data) optimally routed at each hop
- Adaptive routing of ordered traffic (e.g. Ethernet) Each new flow can take an optimal new path

Rosetta Switch

64 port switch, 200 Gb/s
- Advanced adaptive routing
- Congestion control, QoS

Cassini NIC
- MPI hardware tag matching
- MPI progress engine
- Hardware support for one-sided operations
- Hardware support for collective operations
- 200 Gb/s



Slide by C. Simmendinger, HPE

Summary of distributed-memory architecture

- "Pure" distributed-memory parallel systems are rare
  - Hierarchical parallelism rules
- Simple latency/bandwidth model good for insights, but unrealistic
  - Protocol switches, contention
- Wide variety of network topologies available
  - Nonblocking crossbar
  - Fat tree
  - Meshes (torus, hypercube, Dragonfly, hybrids)
  - Adds more layers of topology on top of node level

- For advanced programming of hybrid hierarchical systems, see "Hybrid Programming in HPC – MPI+X" tutorial by HLRS, NHR@FAU, VSC