HPC Interconnects

- Interconnects allow communication between nodes
- Important attributes of interconnects
  - Bandwidth
    - How much data can be moved per second
  - Latency
    - Time to move one byte between nodes
  - Cost
    - How many ports and network links?
  - Power
    - Can be significant in terms of overall system power
    - Won’t discuss in this slide deck

Diameter

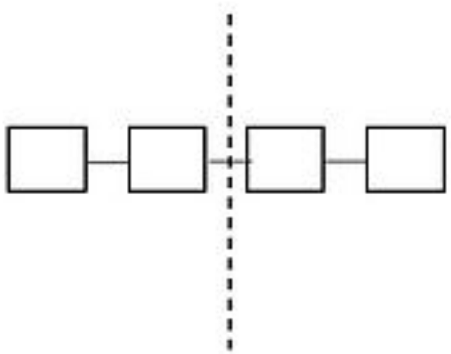
- Motivation
  - Would prefer that all nodes are fully connected
    - Obviously, this is cost prohibitive and is impractical
    - Settle for "few hops", but would also prefer that the number of hops is a constant
      - Or at least a slow-growing function of the number of nodes
- To compute diameter:
  - Just compute the maximum hop distance between two nodes

Bisection Bandwidth

- Motivation
  - All-to-all communication stresses the network, and can depend on moving data “far across” the network
  - Takes into account “bottleneck bandwidth”
- To compute bisection bandwidth:
  - 1. Cut network such that the number of nodes is equal, and
  - 2. Ensure the cut minimizes the bandwidth between the partitions
  - Can do this easily by simply counting the number of links it takes to bisect the network into two partitions
    - Means that maximum bisection bandwidth given N nodes is $(N/2)^2$

Linear Array

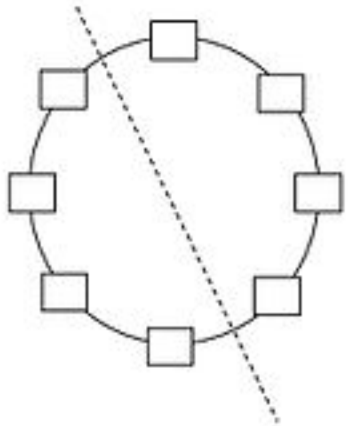
- Not realistic (just for illustration)
- If N nodes, then:
  - Diameter is N-1 (remember, worst case)
  - Bisection bandwidth is 1 (cut shown)
  - Cost:
    - N-1 network links



Source for image: Wikipedia

Ring

- Also not realistic (and just for illustration)
- If N nodes, then:
  - Diameter is N/2
  - Bisection bandwidth is 2 (two paths through the cut)
  - Cost:
    - 3 I/O ports per switch, N switches
      - (one I/O port is to the node)
    - N network links



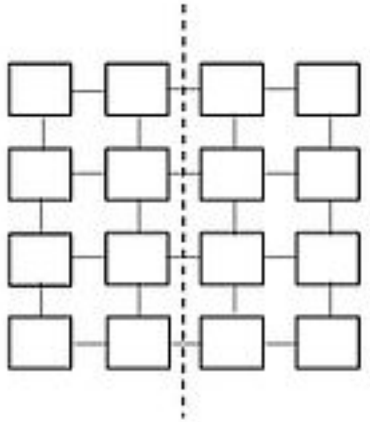
Source for image: Wikipedia

Fully Connected

- Also not realistic (and just for comparison)
- If N nodes, then:
  - Diameter is 1
  - Bisection bandwidth is $(N/2)^2$
    - Have to cut half of the links from a node to disconnect
  - Cost (yikes):
    - N ports/switch, N switches
      - Completely unrealistic; no switch has, say, thousands of ports
    - N*(N-1)/2 network links

2D Mesh

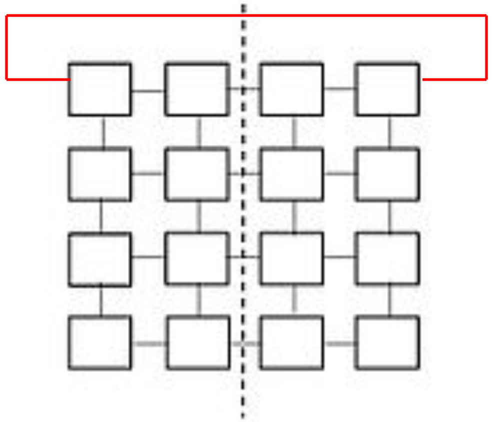
- People have actually used meshes in HPC systems
  - A decade or two ago
- If N nodes, then:
  - Diameter is 2 * ($\sqrt{N}$-1)
  - Bisection bandwidth is $\sqrt{N}$
  - Cost:
    - 5 ports per switch, N switches
    - $\sqrt{N}$ * ($\sqrt{N}$-1) * 2 network links



Source for image: Wikipedia

2D Torus

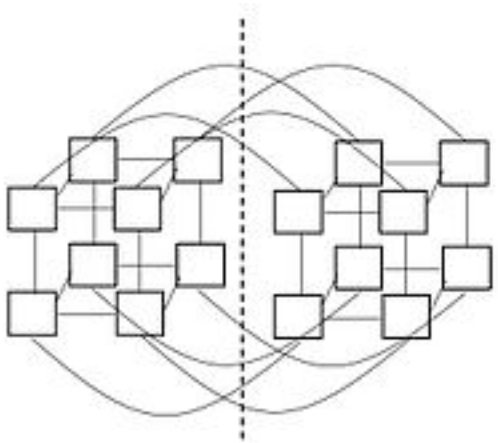
- People have actually used 2D torii in HPC systems
  - Not that long ago, though now they're 3D
- If N nodes, then:
  - Diameter is $\sqrt{N}$
  - Bisection bandwidth is 2 * $\sqrt{N}$
    - Have to cut the wraparound links also
      - (In addition to the links as for the 2D Mesh)
  - Cost:
    - Same as 2D mesh, except an additional 2 * $\sqrt{N}$ wrap links
- The wraparound links really do help



Source for image: Wikipedia

Hypercube

- People have actually used hypercubes in HPC systems
  - Couple decades ago
- If N nodes ($N = 2^k$), then:
  - Diameter is $\log(N) = k$
  - Bisection bandwidth is $N/2 = 2^{k-1}$
    - Must cut all links between cubes
  - Cost:
    - k+1 ports/switch (k-dimensional cube), N switches
    - O(N) network links

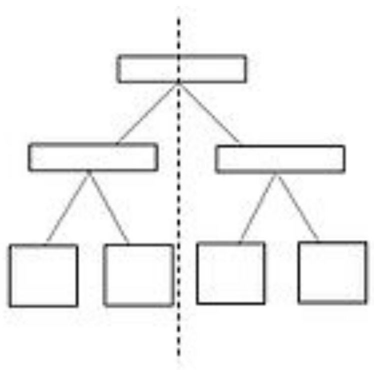


Source for image: Wikipedia

16 nodes → 32 links; 32 nodes → 80 links; 256 nodes → 1024 links

Tree

- People have actually used trees in HPC systems
  - Today!
- If N nodes and direct connection from node to leaf switch, then:
  - Diameter is log(N)
  - Bisection bandwidth is 1
  - Cost:
    - 3 ports per switch, N switches
    - O(N) network links

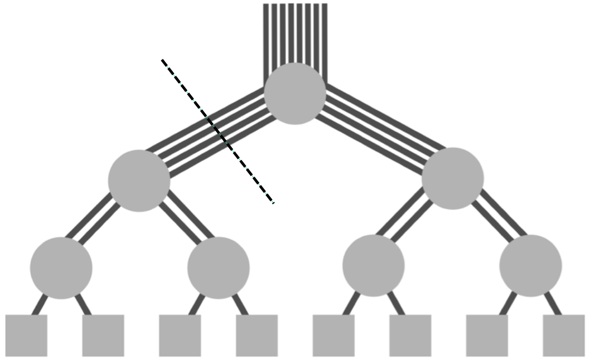


Source for image: Wikipedia

- How can this possibly be a good idea, given that the bisection bandwidth is the same as a bus?

Fat Tree

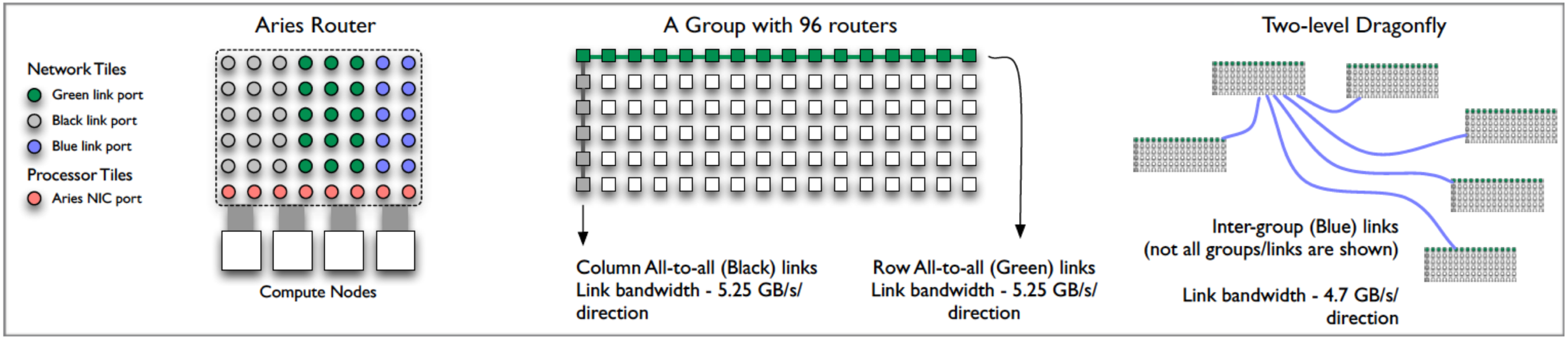
- Same as a tree, except extra links going up tree:
  - Bandwidth increases (usually doubles) as you move up
  - Still have a single logical link across the bisection cut:
    - But given that the bandwidth doubles, this isn't a problem
  - Diameter is still O(log(N))
    - Constant in big-O is dependent on number of levels of tree
  - Bisection bandwidth depends on number of links
    - N nodes and bandwidth doubling results in bisection bandwidth of N/2



Source for image: cluster-design.org

Dragonfly

- Dragonfly exists in current HPC systems
- If N nodes, then:
  - Diameter is 5 (note: independent of N)
    - This assumes static routing
  - Bisection bandwidth is complicated
    - Depends on the number of nodes in a group, number of groups, number of links between groups
    - Definitely lower than fat tree



Routing

- Two extremes
  - Shortest path
  - Fully adaptive
    - Take traffic into account and "route around it", like the Internet
- Many points in between the two extremes
  - Example: Dragonfly has adaptive routing with "minimal path bias" levels.
    - As message gets closer to destination, increase bias for taking shortest path by some amount
- Must worry about deadlock
  - Fully adaptive routing could fail to deliver message