The System Area Network Has Evolved

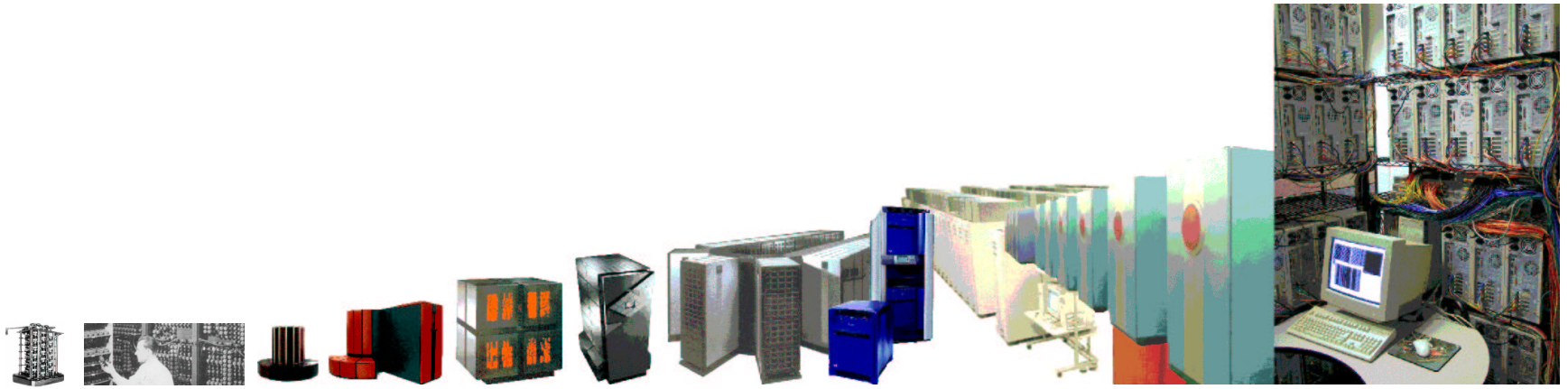
University of Cincinnati

Friday, February 28, 2003

Hank Dietz
Electrical and Computer Engineering Department
University of Kentucky
Lexington, KY 40506-0046
http://aggregate.org/

The Evolution Of Supercomputers

- Most fit survives, even if it looks funny....
- Evolution works.



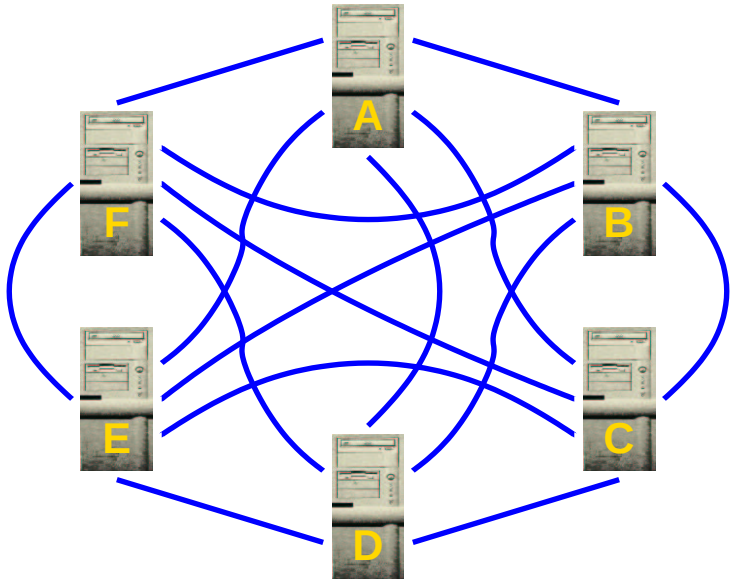
Pre–Siliconian Era | **Vectorian Period** | **Early Massivian Period** | **Sharedmemian Period** | **Late Massivian Period** | **Clusterian Period**

An Overview Of SAN Design

- SAN = "System Area Network"
- Assumptions
  - Links are bidirectional
  - Fixed maximum number of network interfaces per node
- Topologies
- Hardware
- Software

No Network

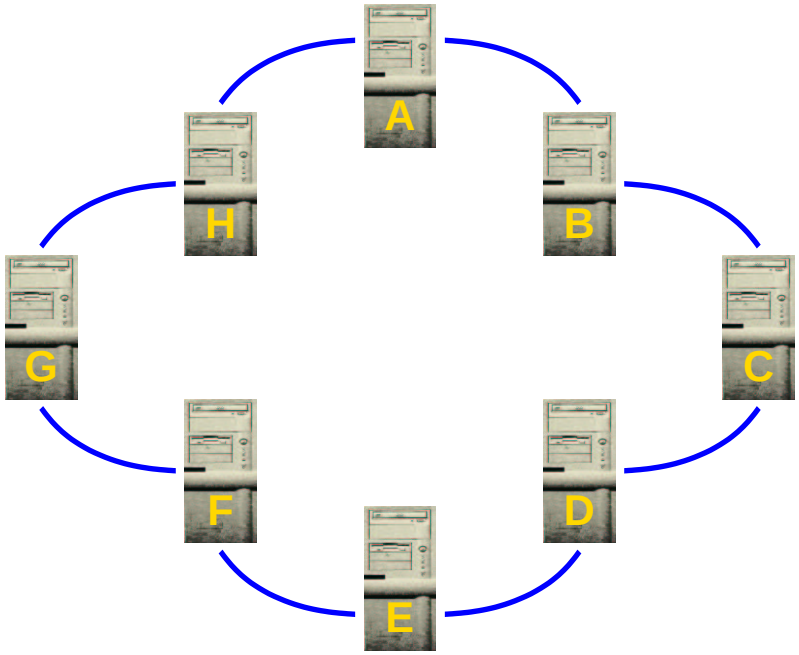
Direct Connections (Fully Connected)



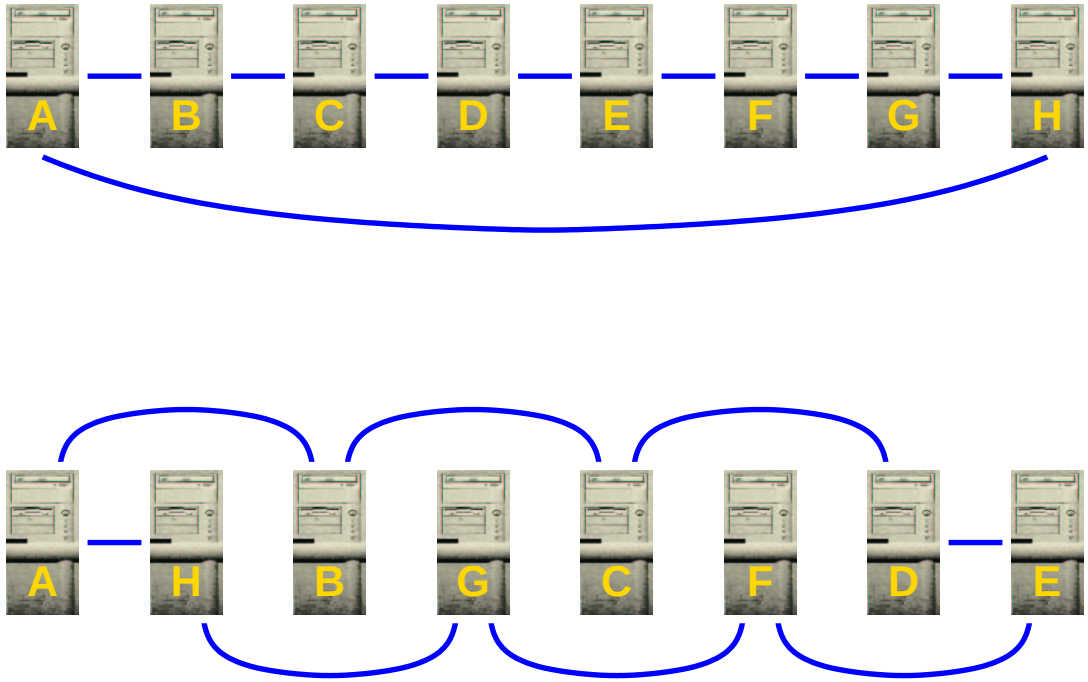
Toroidal Hyper-Meshes

- A generic term for the most common network topologies that use nodes as through-routing elements
- Includes rings, meshes, and hypercubes
- Can have any dimensionality: 1D, 2D, 3D, 4D, etc.
- Optionally have toroidal "wrap around" links
- Through-routing tends to imply high latency, and some processor overhead

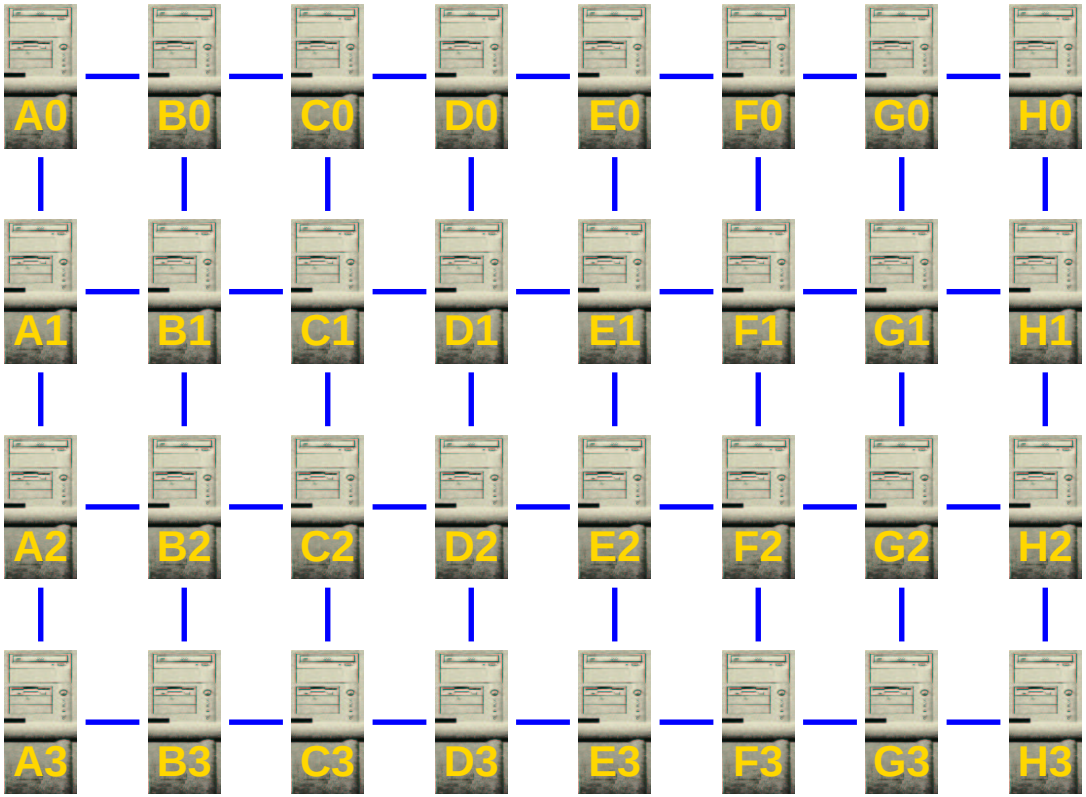
Toroidal 1D Mesh, AKA Ring



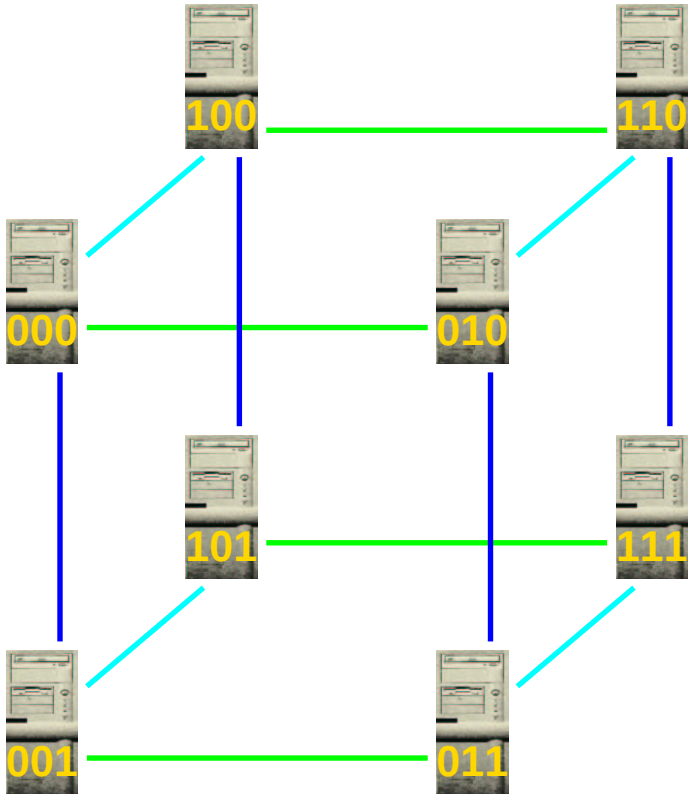
Physical Layout of a Ring



Non-Toroidal 2D Mesh



Non-Toroidal 3D Mesh or 3-Cube



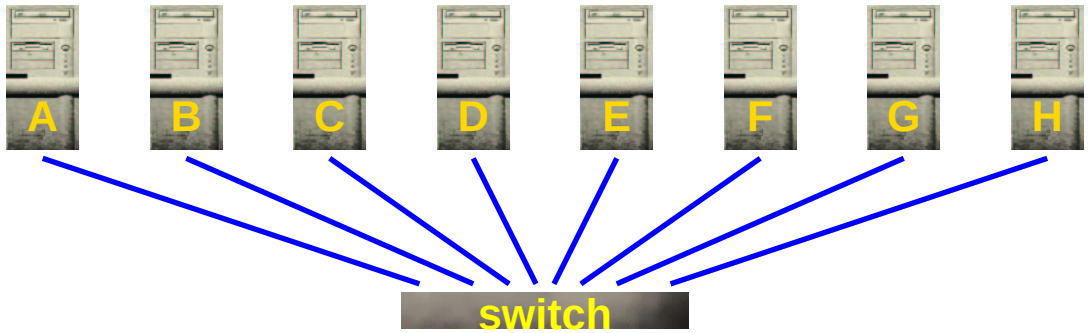
Switch Networks: What's a Switch?

- Hub: a shared bus in a box
- Full Duplex Repeater (FDR): two buffered busses
- Switch: interconnected switching elements
  - KxK switch chips interconnected (usually in a ring)
  - Bandwidth depends on interconnect;
    want *wire speed* and *non-blocking*
  - Latency a little higher than a hub or FDR
- Router: an expensive switch with smart routing abilities
  - Bandwidth can be helped by *trunking*
  - Latency often higher than dumb switch
    (due to routing processor)

The Ideal Switched Network

- No pair of nodes are separated by more than the latency of a single switch
- *Bisection Bandwidth* of each switch contributes to the total bisection bandwidth

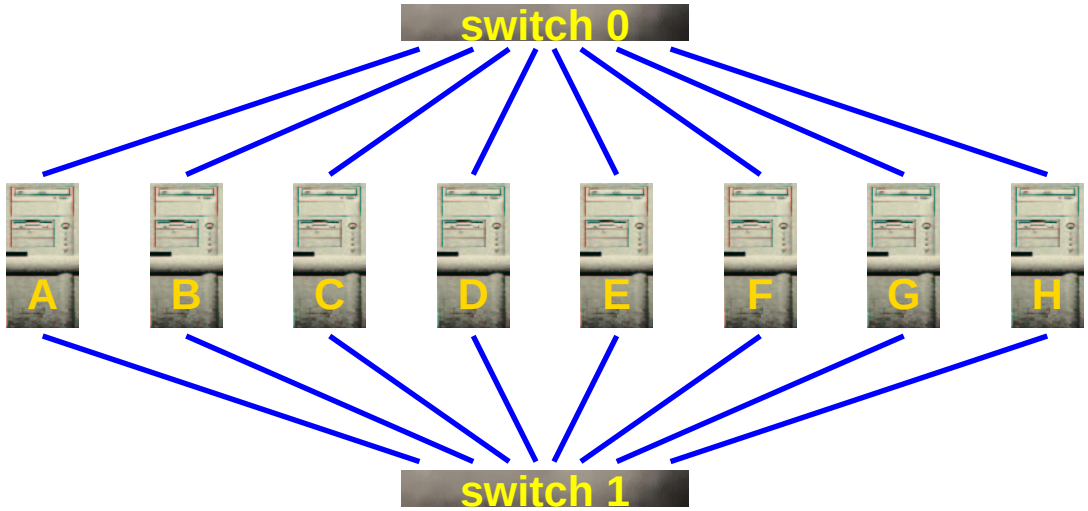
Simple Switch (8-port)



Channel Bonding

- The original "Beowulf" technology
- Use multiple, parallel, ethernet NICs
  - Dynamically load share between paths;
    Often, 2-5 100Mb/s NICs can match one Gb/s
  - Latency is not seriously degraded
- Beowulf implementation clones MAC addresses
  - Confuses switches if two NICs reachable
  - We are building a version that doesn’t
- Bonding performance limited by:
  NIC overhead, PCI bus, interrupt traffic

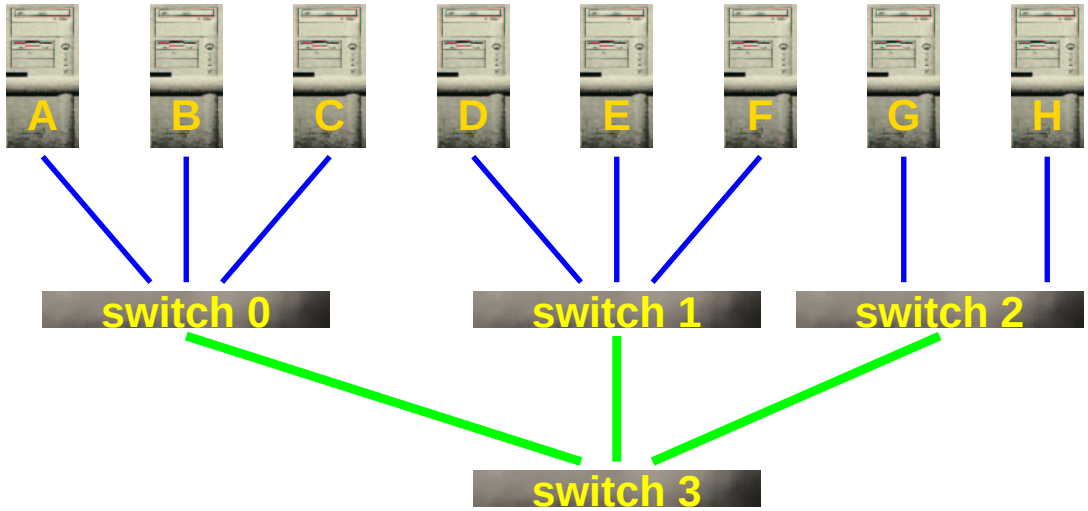
Channel Bonding (2-way, 8-port switches)



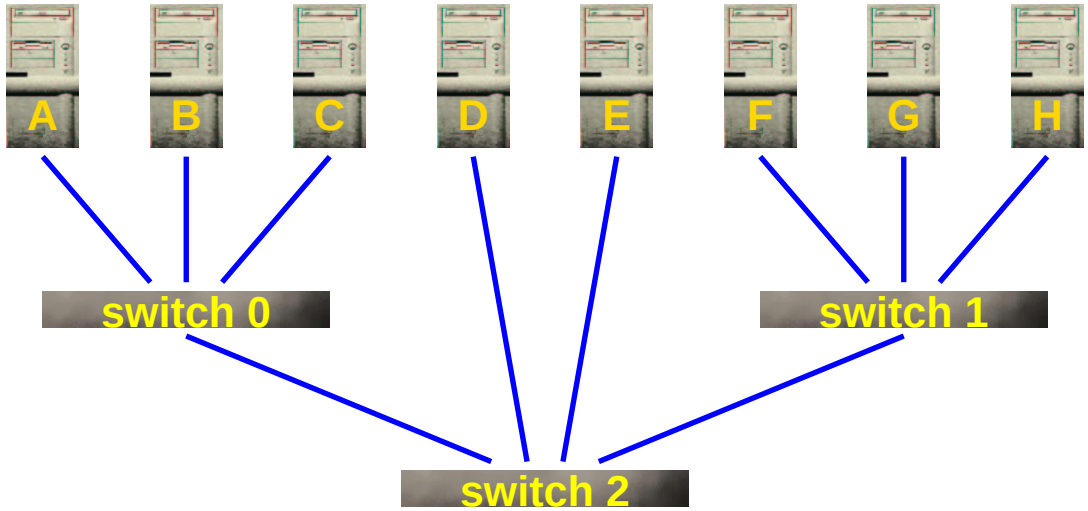
Switch Fabrics

- Number of nodes exceeds ports per switch...
  use multiple switches for connectivity
- Big switches tend to use internal ring-of-rings
  (often with mediocre bandwidth & latency)
- Most supercomputers, especially large clusters, use *hierarchical* switch fabrics
  - Most common are *trees* and *fat trees*
  - Want bisection bandwidth preserved at all tree levels

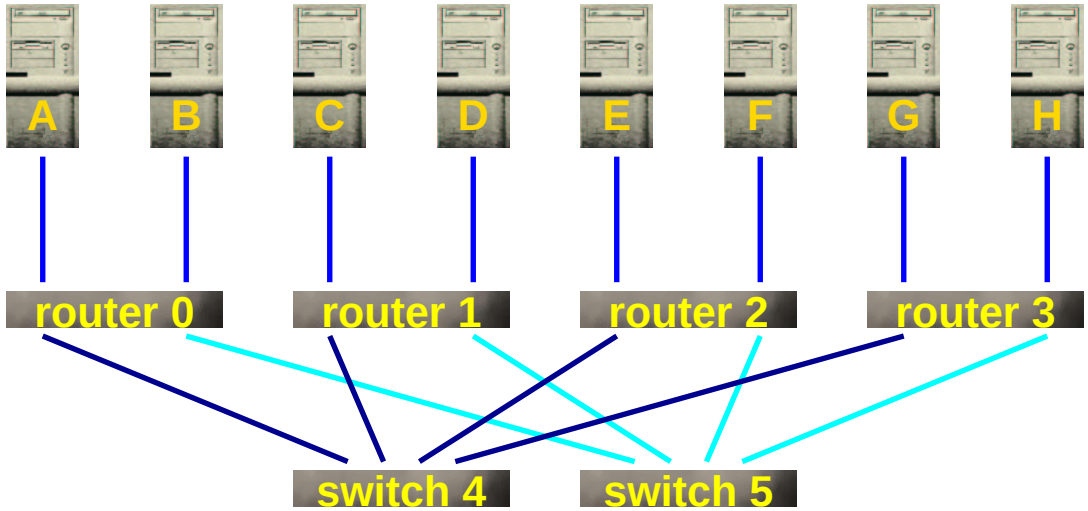
Tree (4-port switches)



A Better Tree (4-port switches)



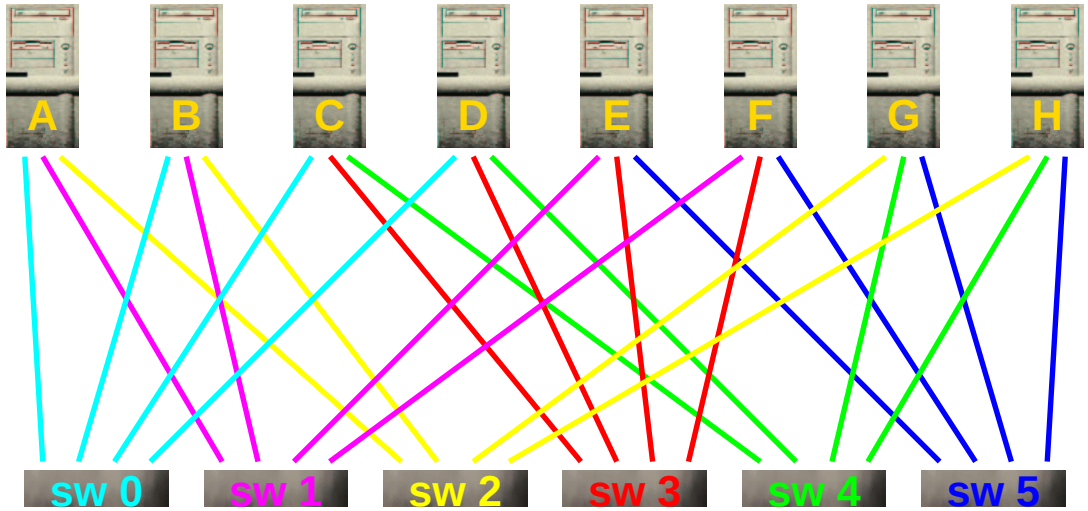
Fat Tree (4-port switches)



The Flat Neighborhood Network (FNN) Insight

- The ideal switch network uses one wide switch, but all PCs don’t have to share the *same* switch
- In a FNN:
  - Every pair of PCs shares at least one switch (each switch is a network "neighborhood")
  - Multiple NICs/PC connect to multiple neighborhoods
  - Topology is flat (no switch is connected to another switch)

Universal FNN (4-port switches)



Flat Neighborhood vs. Fat Tree

- Typically, FNN uses no more switches than Fat Tree, but uses dumb switches and more NICs
- Flat vs. Fat Latency
  - 8 PCs, 4 port: 1.0 vs. (1.0+3.0*6)/7 = 2.7 switch delays
  - 64 PCs, 32 port: 1.0 vs. 2.5 switch delays
- Flat vs. Fat Pairwise Bandwidth
  - 8 PCs, 4 port: 1.29 vs. 1.0 NIC bandwidth units
  - 64 PCs, 32 port: 1.48 vs. 1.0 NIC bandwidth units
- Incremental improvement using FNNs: E.g., a 4th NIC/PC yields 1.86 NIC bandwidth units

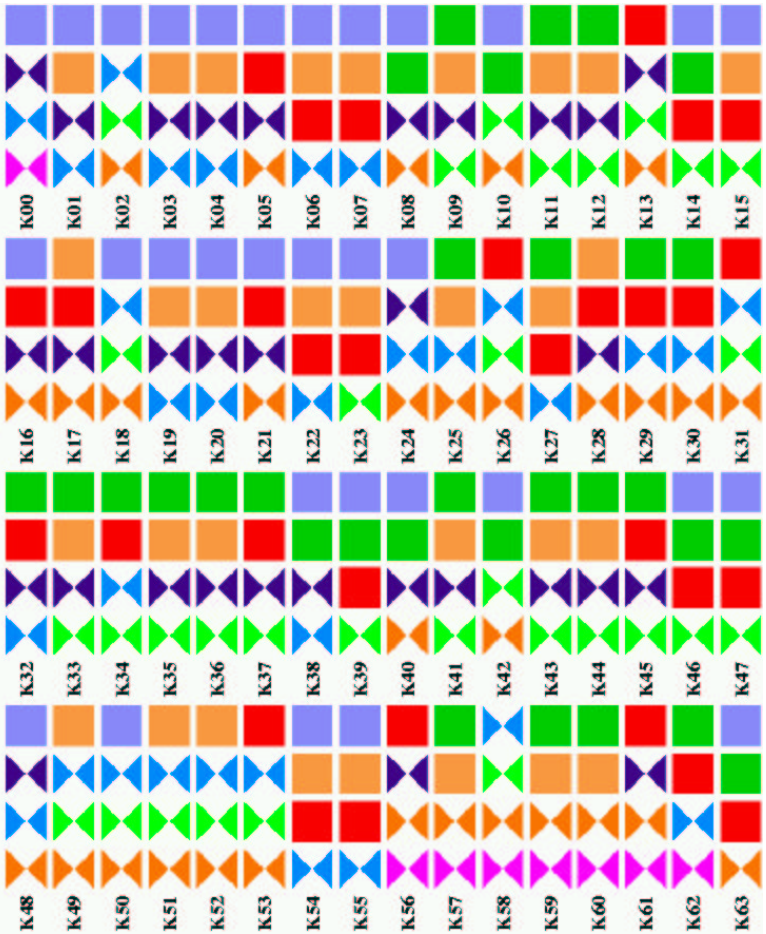
Flat Neighborhood Network Design

- FNN design is a hard problem
- Our Genetic Algorithm (GA) design tool:
  1. Attempt to scale-down the problem
  2. Start with population of random wiring patterns, evolve FNN over multiple generations of *crossover*, *mutation*, & *merit-based selection*
  3. Scale-up solution to original problem
  4. Evolve with complex merit function
- Best design is usually **asymmetric**!

FNN Evolution

- Simple merit counts FNN property only
- Complex merit also checks:
  - Given a set of communication patterns,
    prefer extra bandwidth for those node pairs
  - Given a sequence of communication patterns,
    prefer delay of NIC reuse
    (facilitate pipelining of communications)
- The GA code is optimized and parallel
  - SWAR parallelism within GA operations
  - Population(s) spread across a cluster

The First FNN: KLAT2's Original Wiring

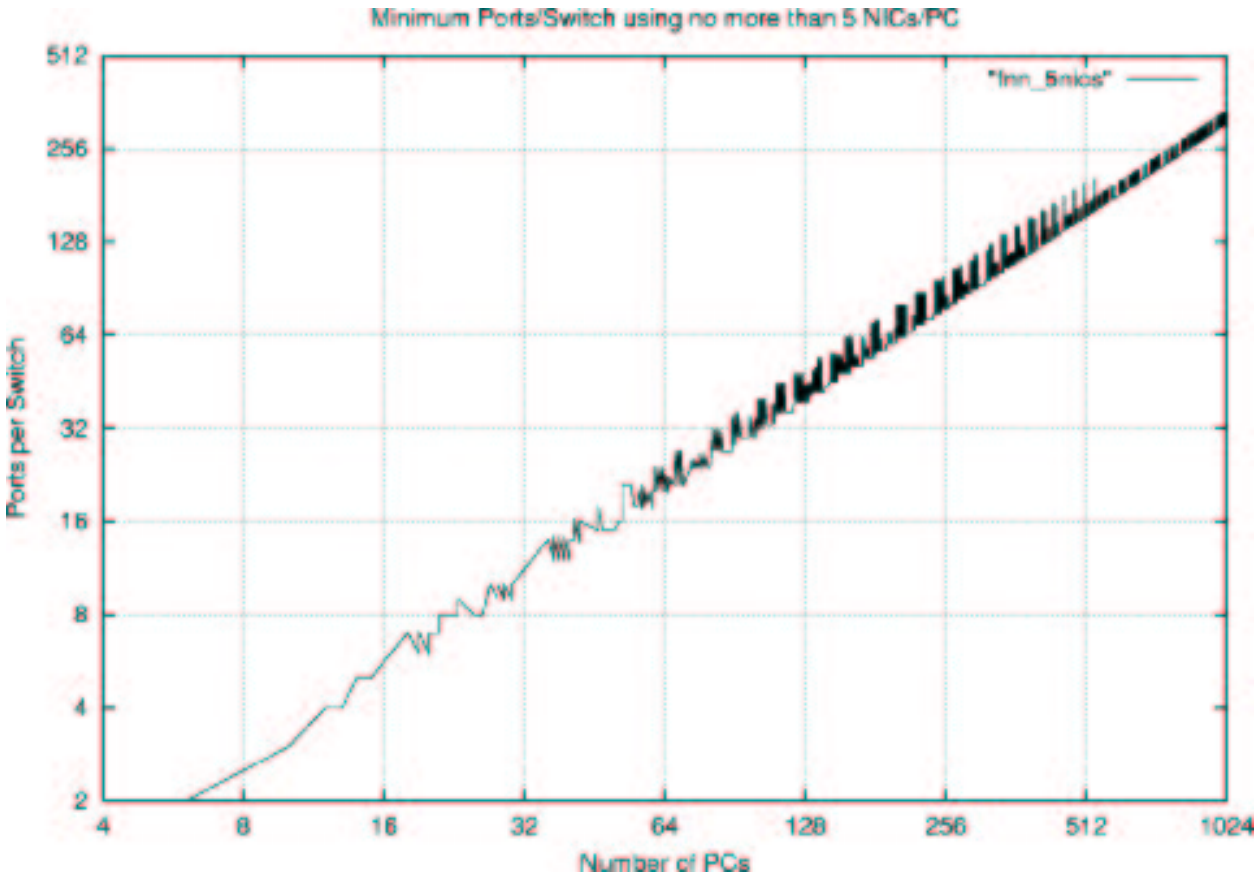


KLAT2's flat neighborhood network
Above: physical wiring
Right: neighborhood pattern

Immediate Results

- KLAT2's network vs. best conventional alternative:
  - Bisection bandwidth: 13Gb/s-25Gb/s vs. 9Gb/s
  - Latency: 30us (using IP!) vs. 40us
  - Cost: $8,100 vs. $250,000
- KLAT2 1st supercomputer to break $1000/GFLOPS
- 160+ news stories & articles about KLAT2
- 2000 Gordon Bell Award,
  Honorable Mention for CFD Price/Performance
- 2000 HPC Games, Most Innovative Architecture
- 2001 Computerworld Smithsonian awards,
  among the 6 ITs most advancing science

But How Scalable Are FNNs?



One More Insight (in 2002)

- FNN originally gave all pairs shared neighborhoods; why not only require **neighbors in communication patterns?**
- How does the number of required neighbors scale?
  - Any particular pattern adds at most 1 neighbor
  - Most pattern families scale by a log factor
  - The union of many (symmetric) patterns is asymmetric, neighbors often reached by multiple patterns
- Two FNN flavors: Universal & Application-Specific
- Needed a much "smarter" GA, etc.

Do Application-Specific FNNs Work?

- We haven’t built one... yet
- We have a new GA that has designed some
  - Largest design thus far covers many patterns... for 10K nodes using 48-port switches!
  - Designs for 1K nodes covering most known patterns cost between $100K and $150K!
  - Patterns not covered have higher latency, not necessarily lower bisection bandwidth
- Given design & software support, they will work; biggest open theory issue is fault tolerance...

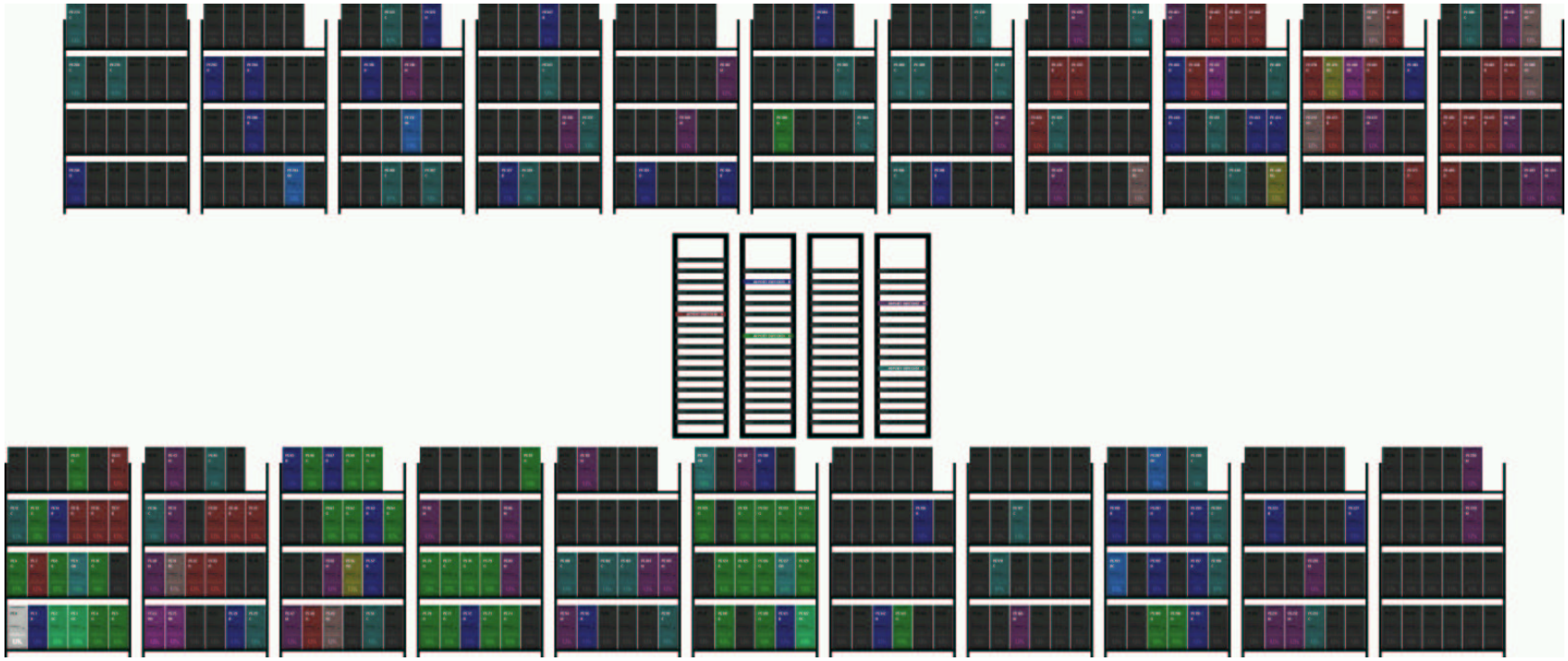
An Application-Specific FNN Design

- Proposed KASY1 (Kentucky ASYmmetric testbed 1)
  - 512+32 nodes, 48-port Fast Ethernet switches
  - Design will be improved as GA is improved
- 51Gb/s-292Gb/s bisection bandwidth
- 30us latency
- Each node has 4-6 NICs
- Just 42-49 neighbors/node, 45.45 average
  (64+2 node KLAT2 has 65 neighbors/node!)
- Total cost of network hardware well under $50K!

Communication Patterns Required for KASY1

- 1D torus (512 nodes) +/- power-of-2 offsets
- 2D torus (32x16 nodes) +/- power-of-2 X/Y offsets
- 2D torus (32x16 nodes) diagonals (i.e., X+/-1 Y+/-1)
- 3D torus (8x8x8 nodes) +/- power-of-2 X/Y/Z offsets
- 3D torus (8x8x8 nodes) edges
  (i.e., X+/-1 Y+/-1, X+/-1 Z+/-1, or Y+/-1 Z+/-1)
- 3D torus (8x8x8 nodes) corners (i.e., X+/-1 Y+/-1 Z+/-1)
- 9D hypercube (2x2x2x2x2x2x2x2x2)
- perfect shuffle (and its inverse)
- bit-reversal

A Little Simulation... If There's Time....



Conclusions

- Requiring symmetry destroys price/performance
- Asymmetric network design is beyond human ability; however, the process can be automated
- Customized GA technology is a viable design method; there is lots of room for improvement
- Scaling of node neighbor sets is sublinear; scaling SANs is much easier than previously thought
- Various supporting technologies need to be developed. from Automatic Node Configuration to Fault Tolerance

Any Questions?

Everything we do gets posted at **http://aggregate.org/**