

# Supercomputing Systems

*CPE380, Spring 2022*

**Hank Dietz**

<http://aggregate.org/hankd/>

# What Is A **Supercomputer**?

- One of the most expensive computers?
- A very fast computer?
- Really two key characteristics:
  - Computer that **solves big problems**...  
stuff that wouldn't fit on a PC  
stuff that would take too long to run
  - Performance can **scale**...  
more money buys a faster machine
- A supercomputer can be cheap!

# The Key Is Parallel Processing

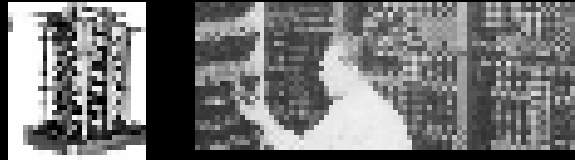
- Process  $N$  “pieces” simultaneously, get up to factor of  $N$  **speedup**
- Modular hardware designs:
  - Relatively easy to scale – add modules
  - Higher **availability** (if not **reliability**)

# The Evolution Of Supercomputers

- Most fit survives, even if it's ugly
- Rodents outlast dinosaurs...  
and bugs will outlast us all!

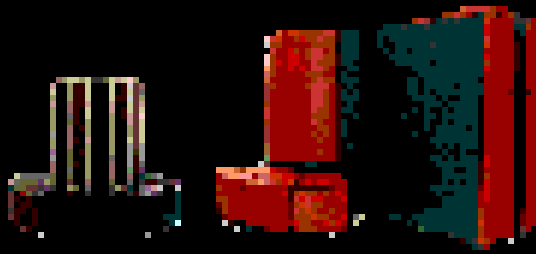






## Pre- Siliconian Era

- Before cheap transistors (– 1960s)
- E.g.: ENIAC
- Faster because machines are faster than humans
- Does one operation at a time because circuitry is too expensive to do more

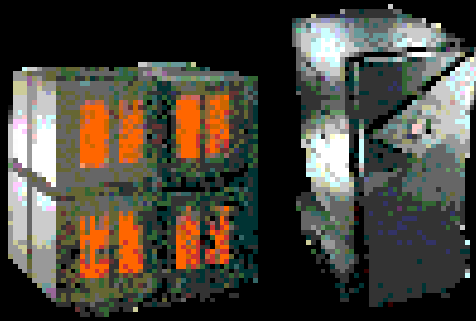


## Vectorian Period

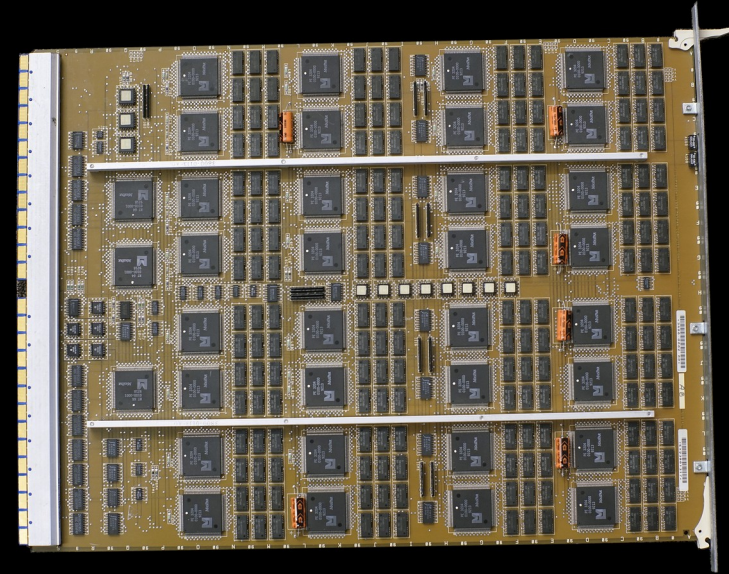
- Cheap transistors (1970s)
- E.g.: Cray 1
- Faster by applying the same operation to a vector of data at a time:

$$A[0..N] = B[0..N] + C[0..N]$$

- Can't change parallelism width without changing all register sizes and datapaths



## Early Massivian Period



- Cheap chips with many transistors (1980s)
- E.g.: Thinking Machines Connection Machine
- Faster by applying the **same operation** to a **vector** of data at a time
- **Modularly scalable SIMD** (Single Instruction, Multiple Data) – can change parallelism simply by adding modules



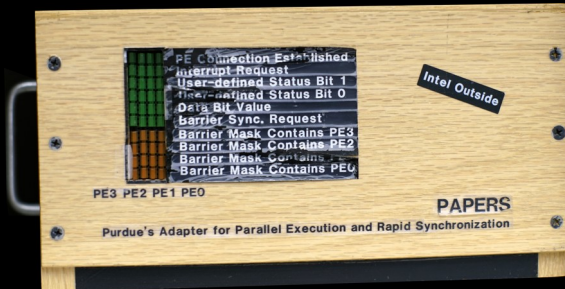
## Sharedmemian Period

- Cheap processors with custom memory
- E.g.: SGI Origin
- Faster by executing different code in parallel, using a custom shared memory to interact
- Modularly scalable MIMD (Multiple Instruction, Multiple Data) – can change parallelism simply by adding modules



### Late Massivian Period

- Cheap processors with custom network (1990s)
- E.g.: **ASCI Red**
- Faster by **executing different code in parallel**, using a **custom messaging network** to interact
- **Modularly scalable MIMD** (Multiple Instruction, Multiple Data) – **can change parallelism simply by adding modules**



# Clusters!



Clusterian  
Period

- Commodity parts and interfaces (1994 – )
- E.g.: Spareparticus, Beowulf, KLAT2, NAK
- Faster by executing different code in parallel, leverages commodity processors and network and also standard interfaces for custom parts
- Modularly scalable MIMD (Multiple Instruction, Multiple Data) – can change parallelism simply by adding modules



- GPUs (Graphics Processing Units) as cheap, fast, add-in cards in cluster nodes (2010 – )
- E.g.: NAK (and *most of the Top500* )
- The thing(s) on video cards... SIMDish, but:
  - Lots of little SIMDs (low fanout)
  - Multithreading to hide memory latency
  - Various restrictions to simplify HW
- NVIDIA CUDA and OpenCL...
- Intel's Xeon Phi is *sort-of* a GPU, but MIMDish
- GPU(s) will be on chip with SMP cores

# Quantum?



- Uses **QuBits** rather than Bits
- **Superposition** allows 0, 1, or *indeterminate*
- **Entangled** superposed QuBits can hold all possible values simultaneously
  - E.g., 6 QuBits can hold {0, 1, 2, ... 63}
  - **Parallel processing without parallel hardware**
- Reading a QuBit's value collapses superposition; **you get only a single answer**
- Could be the next big thing... or not



# Clusters And Bigger

- Mostly from interchangeable (PC) parts... and mostly running some form of Linux
- **Cluster** or **Beowulf** is a *parallel supercomputer* with tightly coupled, homogeneous, nodes
- **Farm** is homogeneous, colocated, machines with a common purpose (e.g., a render farm)
- **Warehouse Scale Computer** is a warehouse full of racked clusters used for *throughput*
- **Grid** is many internet-connected machines
- **Cloud** is *virtualized* grid/WSCs providing *services*

# Types Of Hardware Parallelism

- Pipeline
- Superscalar, VLIW, EPIC
- SWAR (SIMD Within A Register)
- SMP (Symmetric MultiProcessor; multi-core)
- GPU (Graphics Processing Unit)
- Cluster
- Farm / Warehouse Scale Computer
- Grid / Cloud

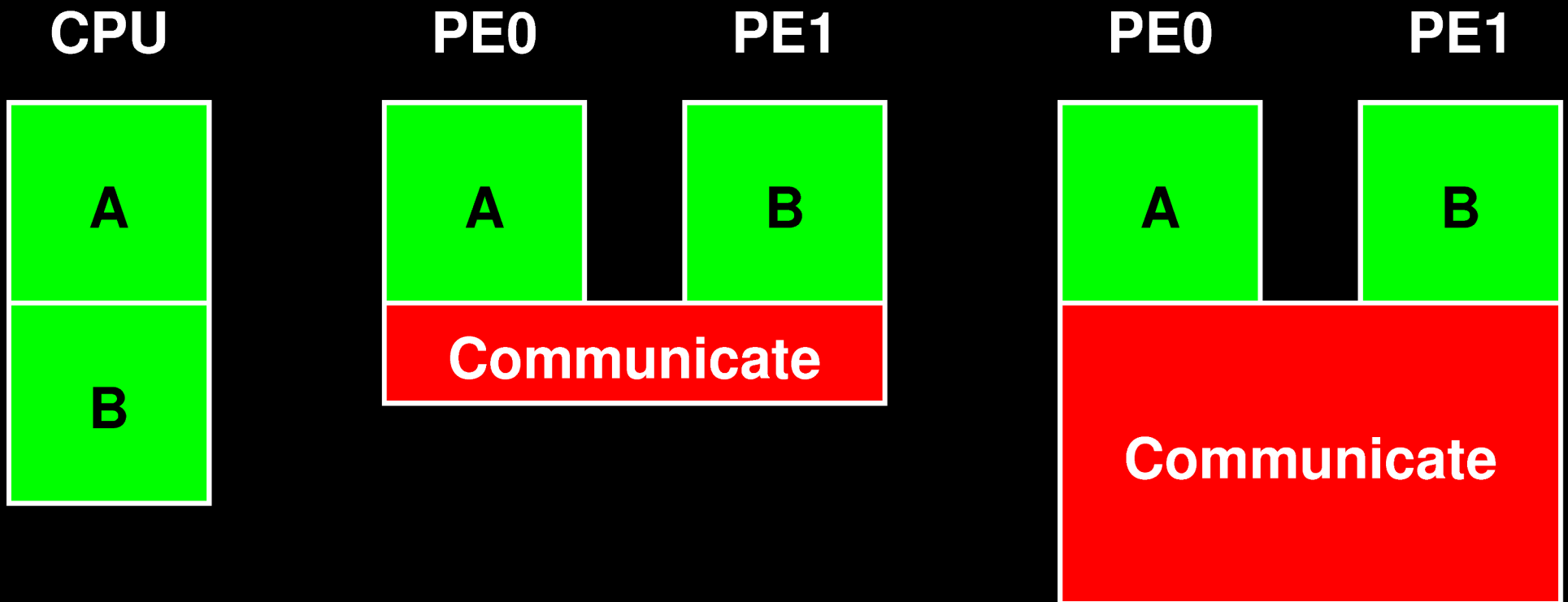
# Engineering A Cluster

- This is a *systems* problem
- Optimize *integrated effects* of:
  - Computer architecture
  - Compiler optimization/parallelization
  - Operating system
  - Application program
- Payoff for good engineering **can be HUGE!**  
(penalty for bad engineering **is HUGE!**)

# One Aspect: Interconnection Network

- Parallel supercomputer **nodes** interact
- **Bandwidth**
  - Bits transmitted per second
  - **Bisection Bandwidth** most important
- **Latency**
  - Time to send something here to there
  - Harder to improve than bandwidth....

# Latency Determines Smallest Useful Parallel **Grain Size**



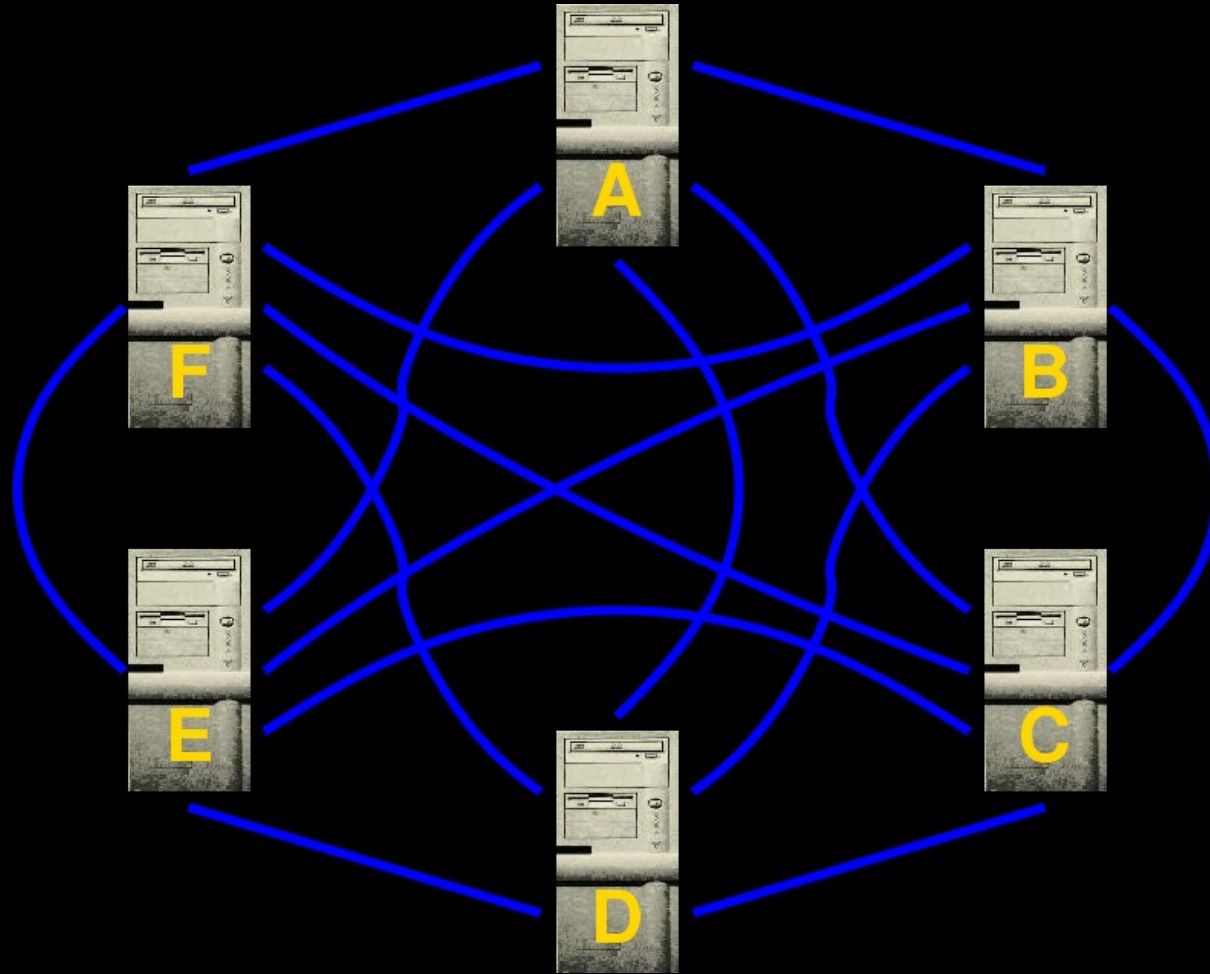
# Network Design

- Assumptions
  - Links are bidirectional
  - Bounded # of network interfaces/node
  - Point-to-point communications
- **Topology**
- Hardware
- Software

# No Network

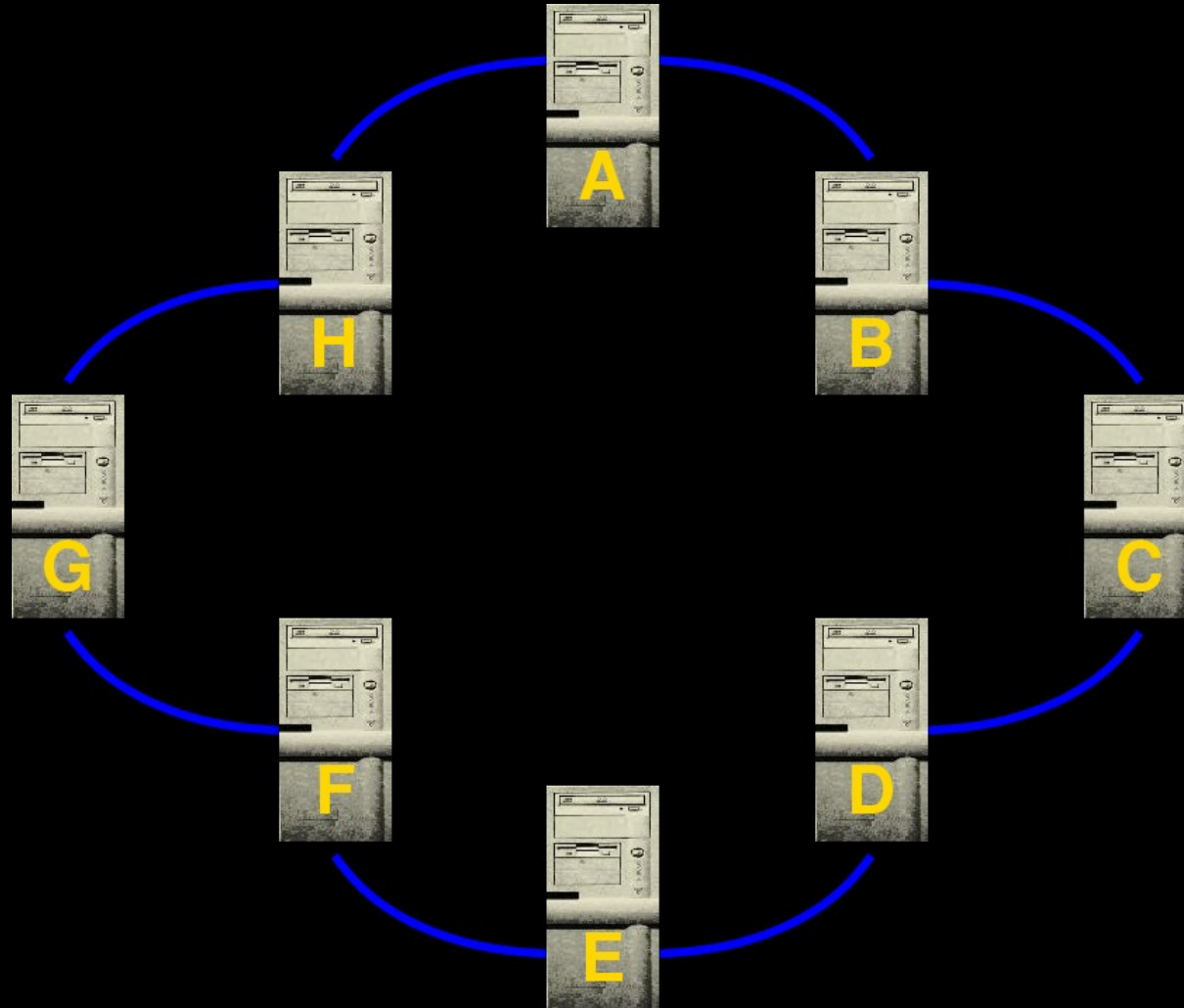


# Direct Fully Connected

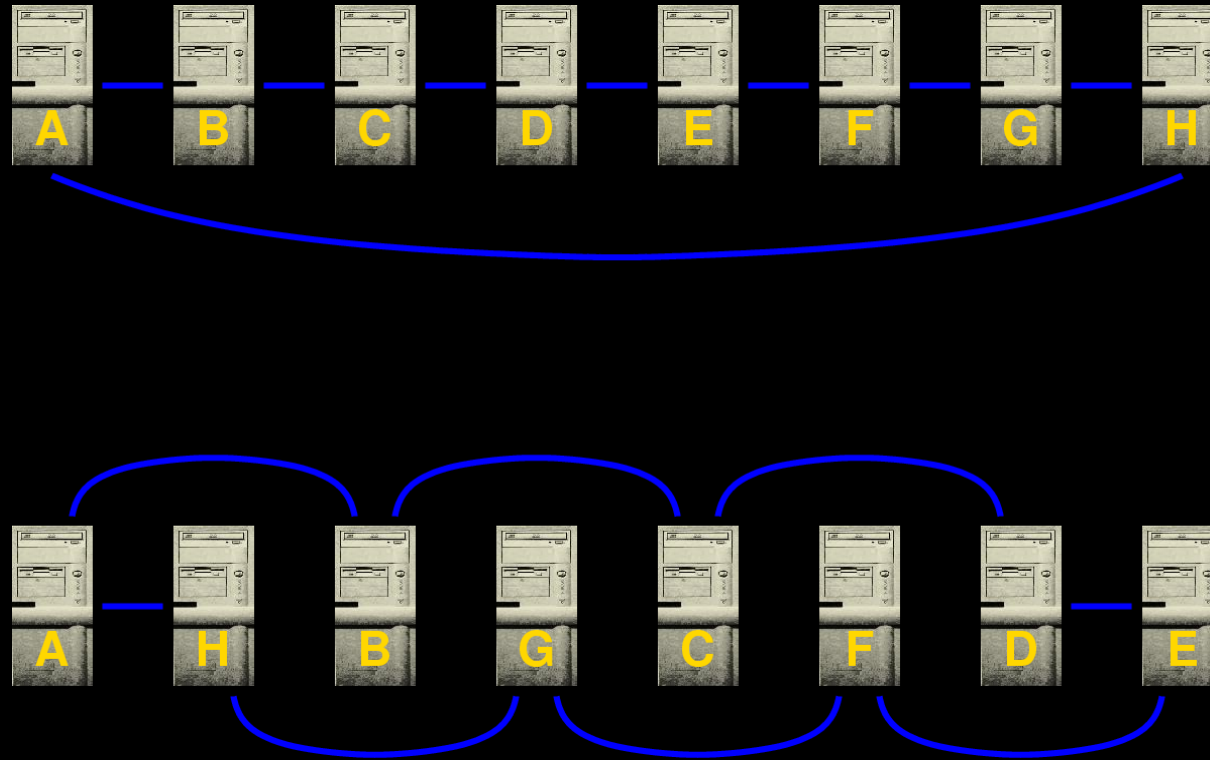




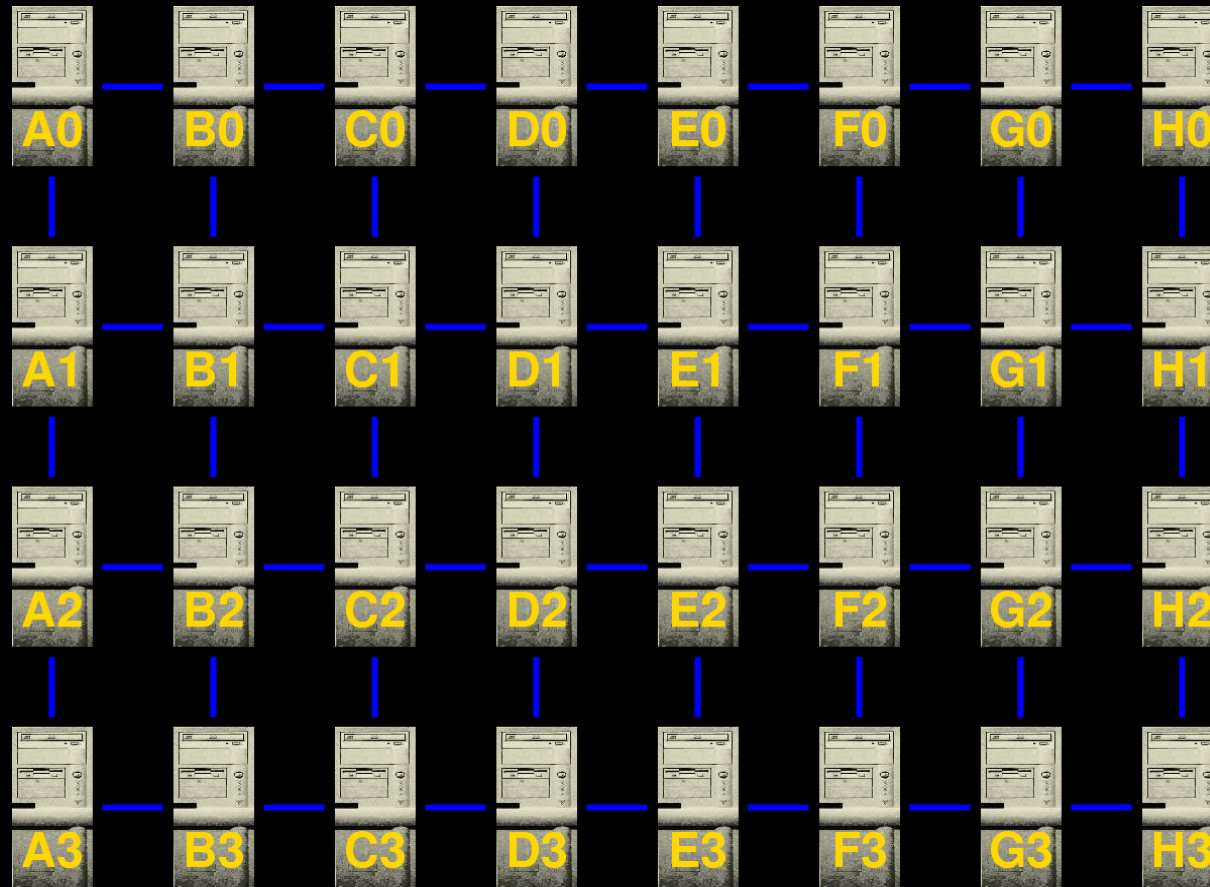
# Toroidal 1D Mesh (Ring)



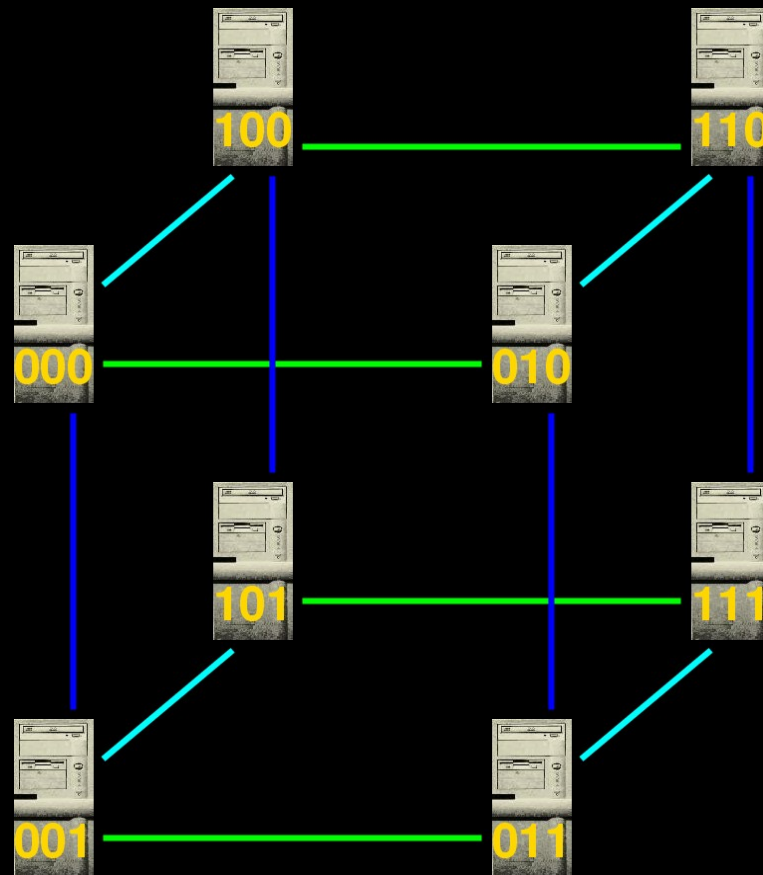
# Physical Layout Of Ring



# Non-Toroidal 2D Mesh



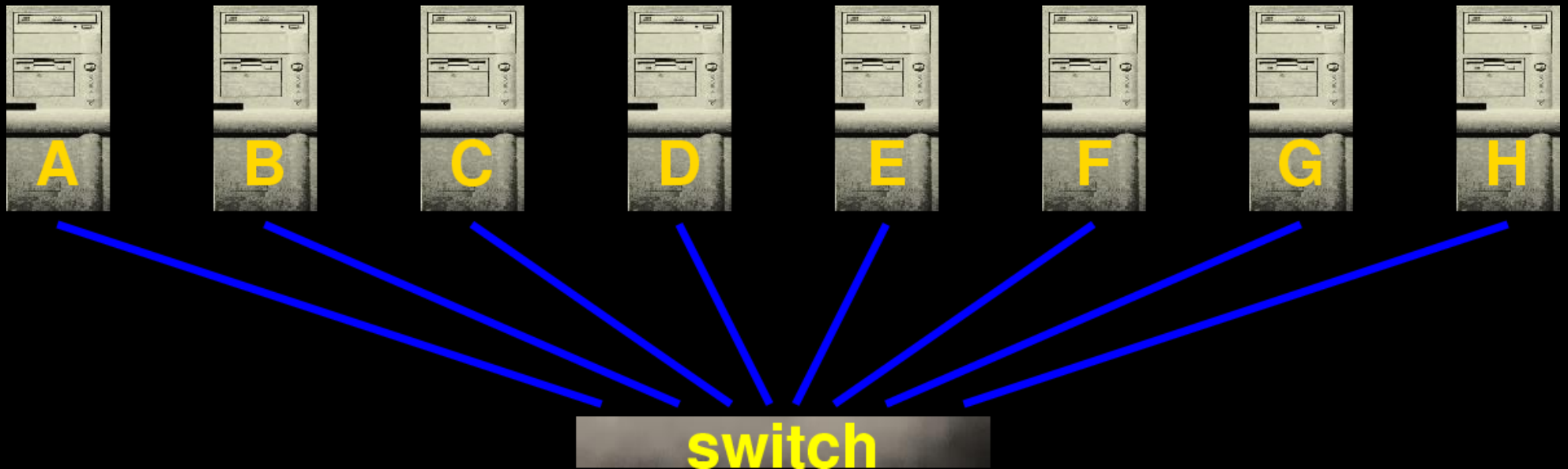
# 3-Cube (AKA 3D Mesh)



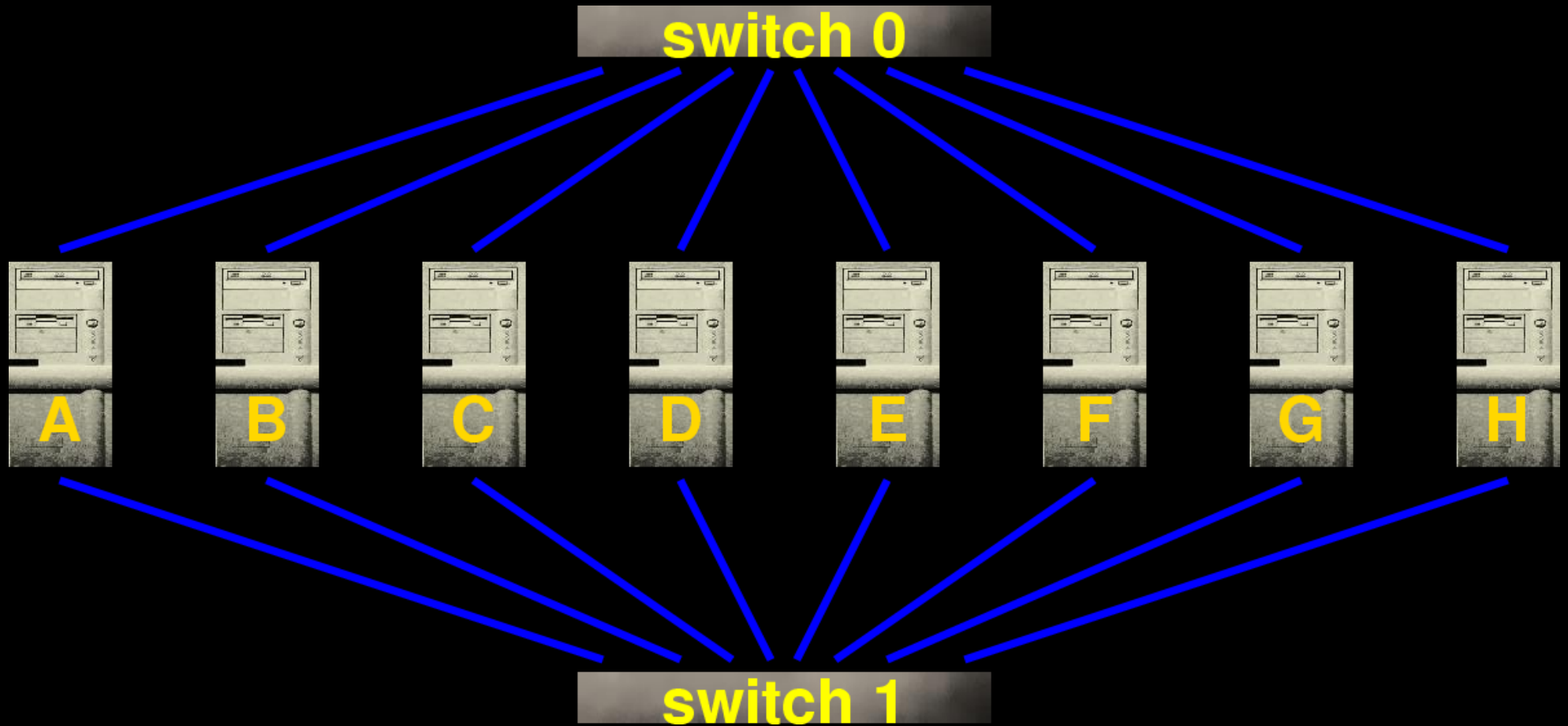
# Switch Networks

- Ideal **switch** connects  $N$  things such that:
  - Bisection bandwidth = # ports
  - Latency is low (~30us for Ethernet)
- Other switch-like units:
  - **Hubs, FDRs** (Full Duplex Repeaters)
  - **Managed Switches, Routers**
- Not enough ports, build a **Switch Fabric**

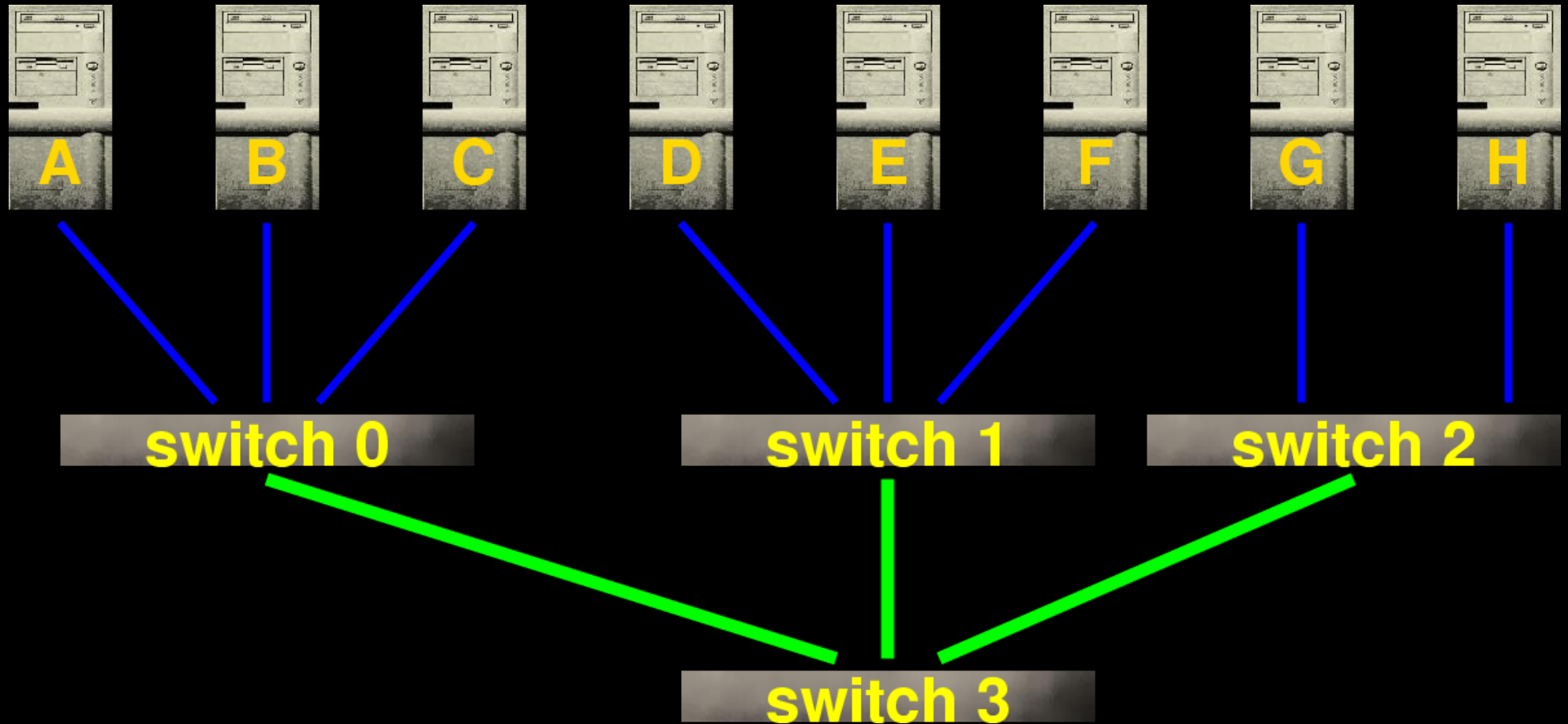
# Simple Switch (8-Port)



# 2-Way Channel Bonding

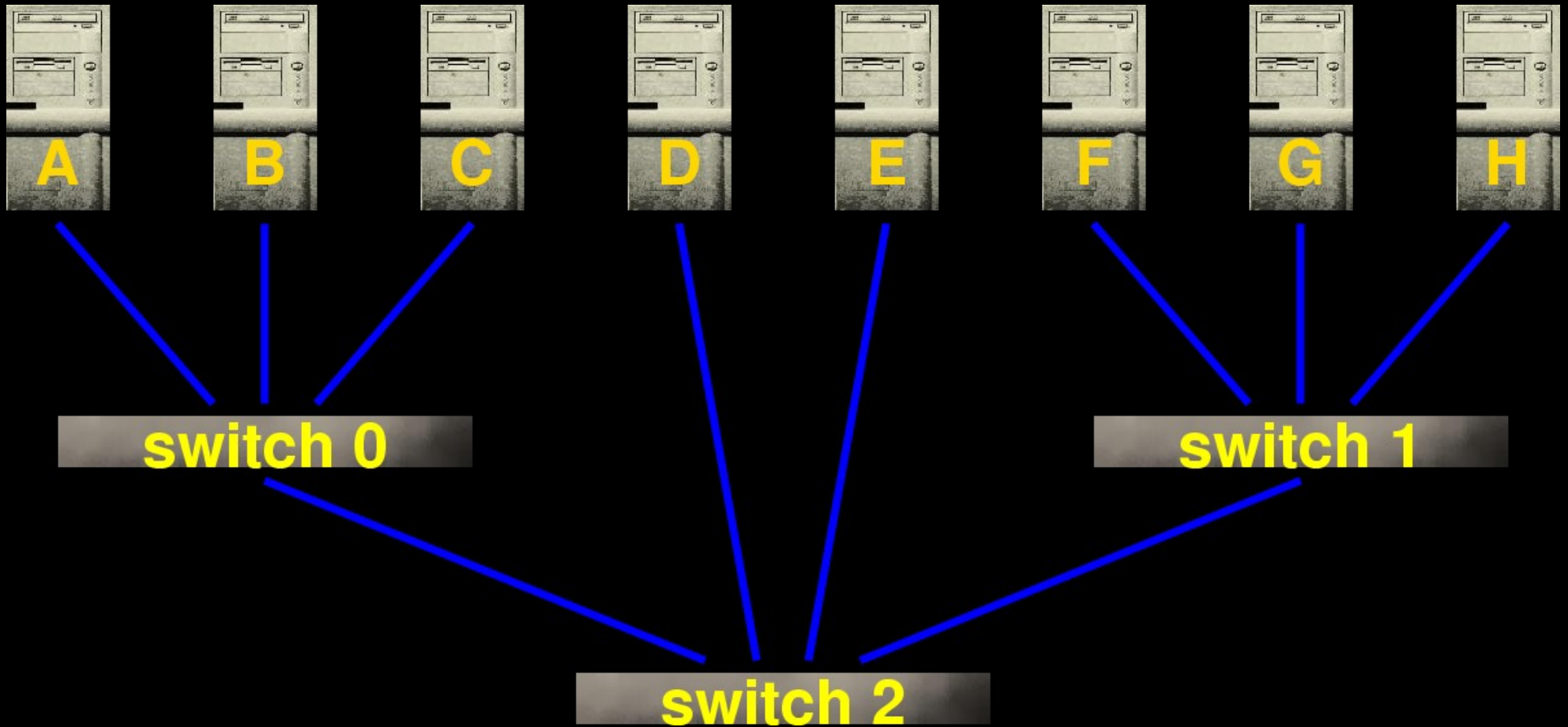


# Tree (4-Port Switches)

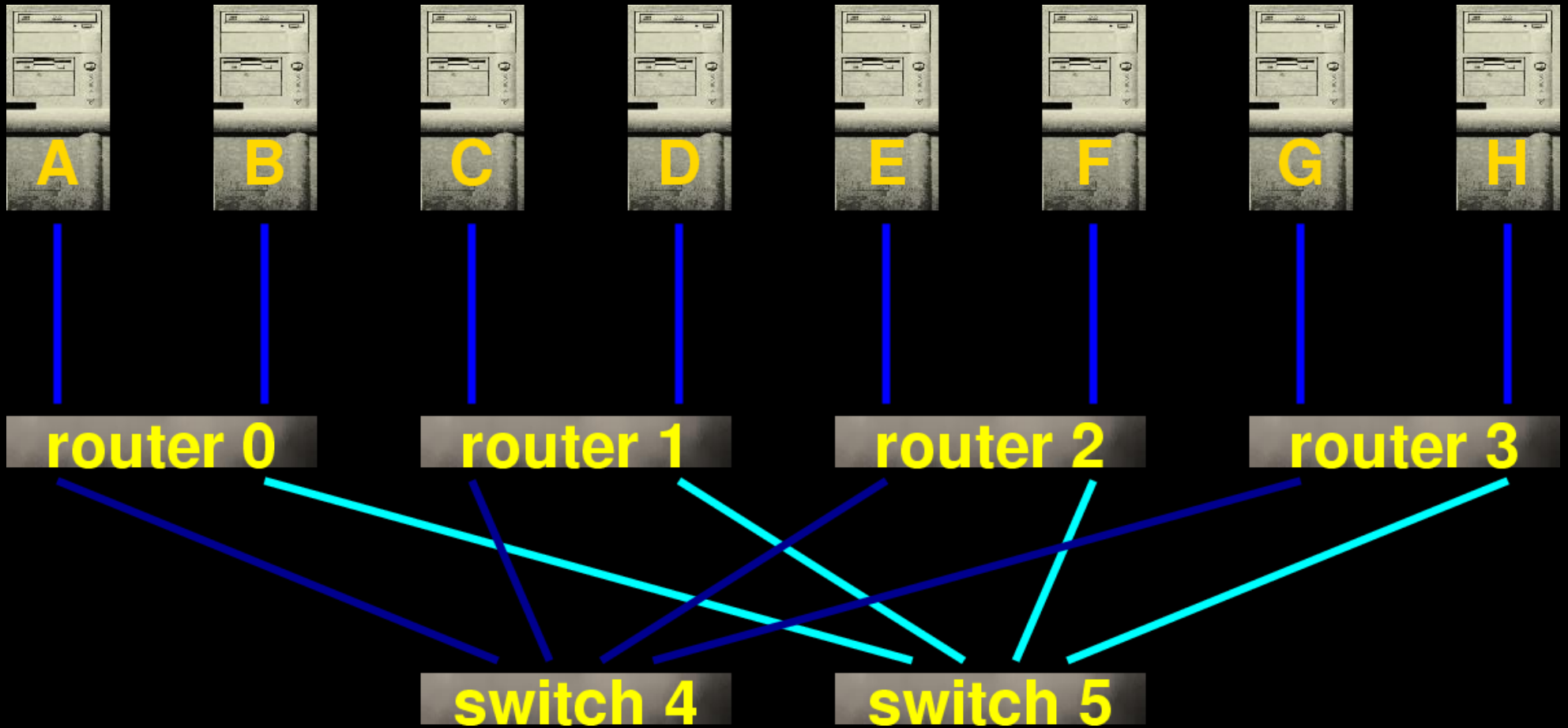




# A Better Tree

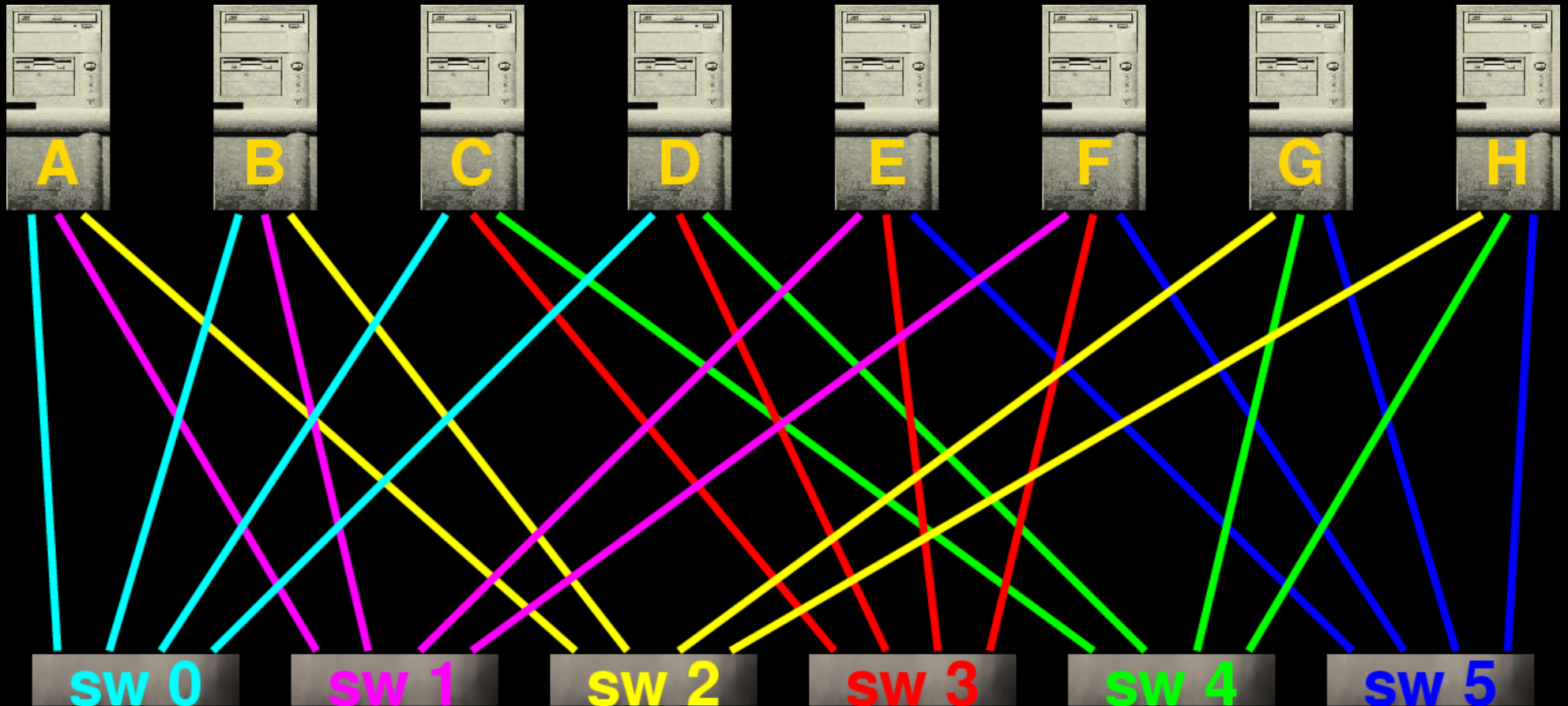


# Fat Tree



# Flat Neighborhood Network...

## from UK!



# Flat Vs. Fat

- Latency:
  - 8 node, 4 port: 1.0 vs. 2.7 switch delays
  - 64 node, 32 port: 1.0 vs. 2.5
- Pairwise bisection bandwidth:
  - 8 node, 4port: 1.29 vs. 1.0 units
  - 64 node, 32 port: 1.48 vs. 1.0
- Cost: more interfaces vs. smart routers
- Summary: Flat Neighborhood wins!

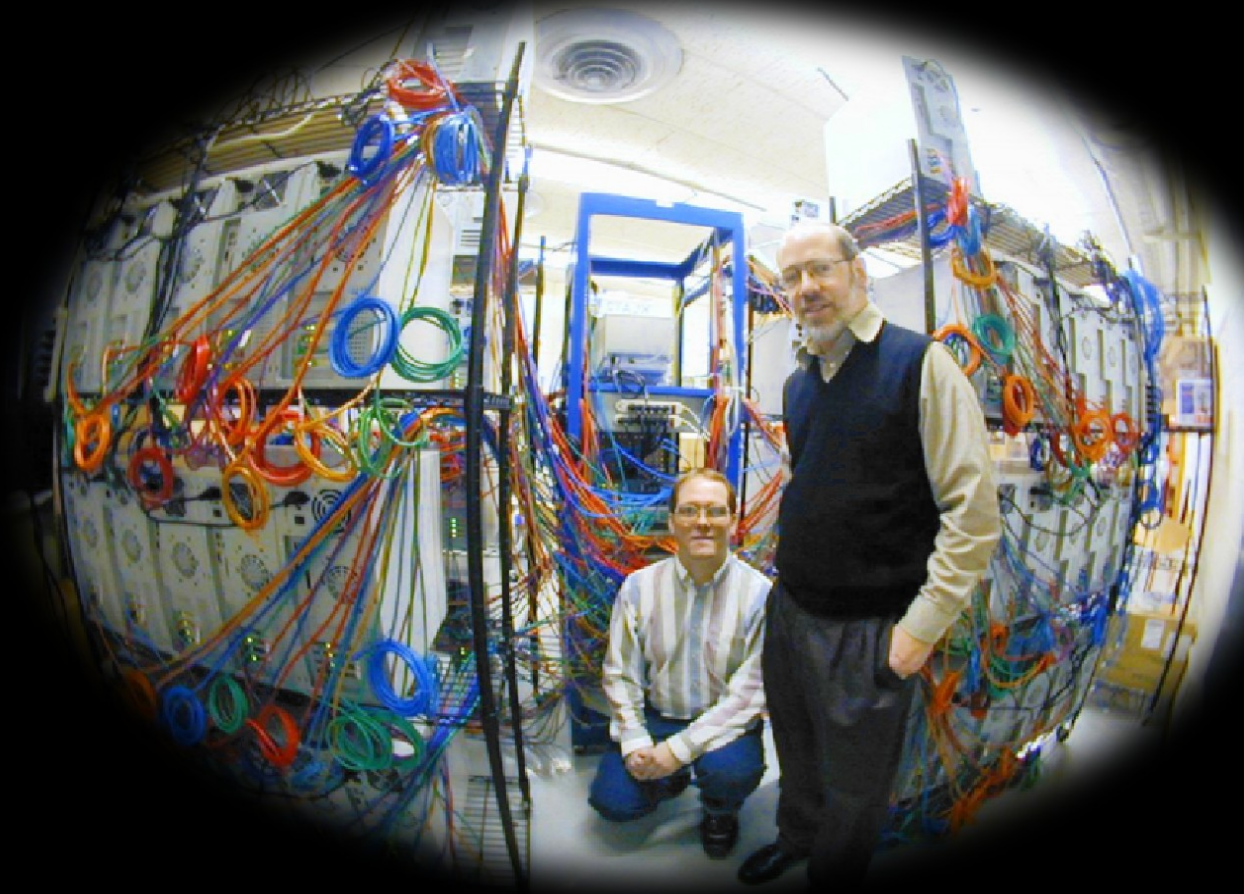
# KLAT2, Gort, & Klaatu



# KLAT2 Changed Everything

- KLAT2 (Kentucky Linux Athlon Testbed 2):
  - 1<sup>st</sup> network designed by computer
  - 1<sup>st</sup> network deliberately asymmetric
  - 1<sup>st</sup> supercomputer under \$1K/GFLOPS
- 160+ news stories about KLAT2
- Various awards:
  - 2000 Gordon Bell (price/performance)
  - 2001 Computerworld Smithsonian, among 6 its most advancing science

# 2000, KLAT2



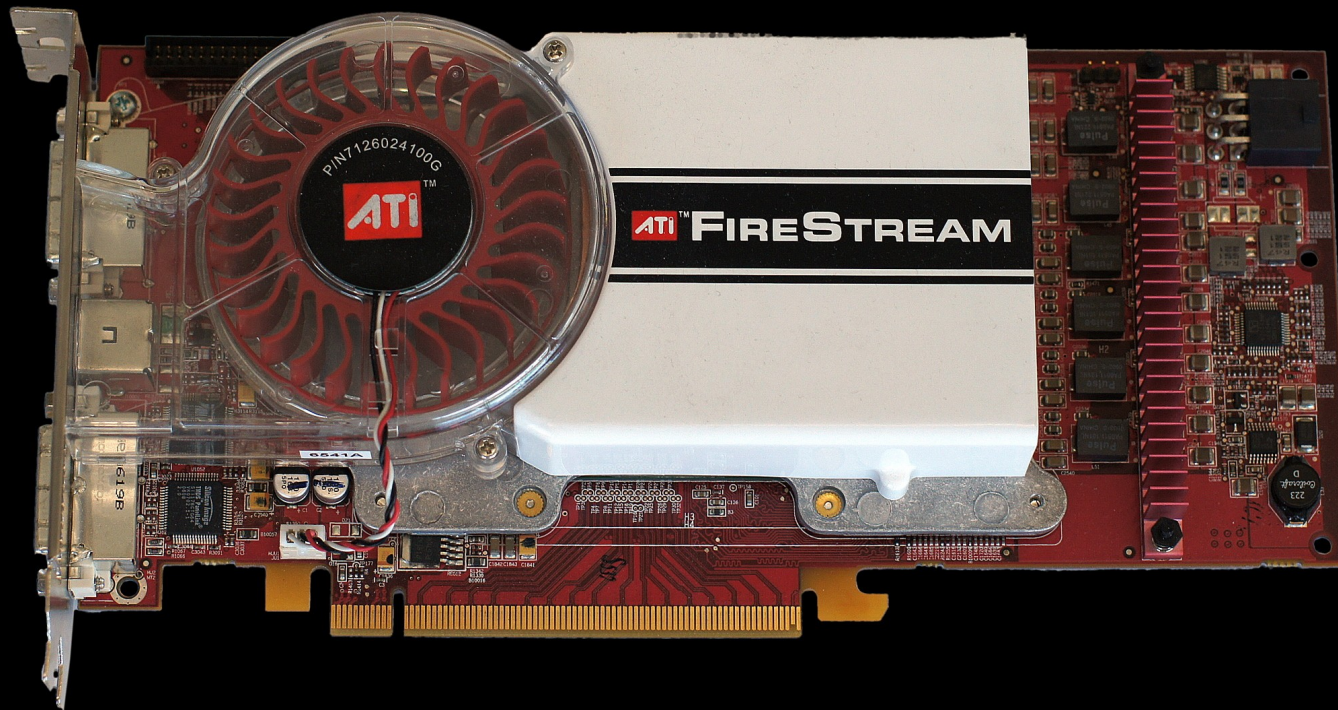


# 2003, KASYO

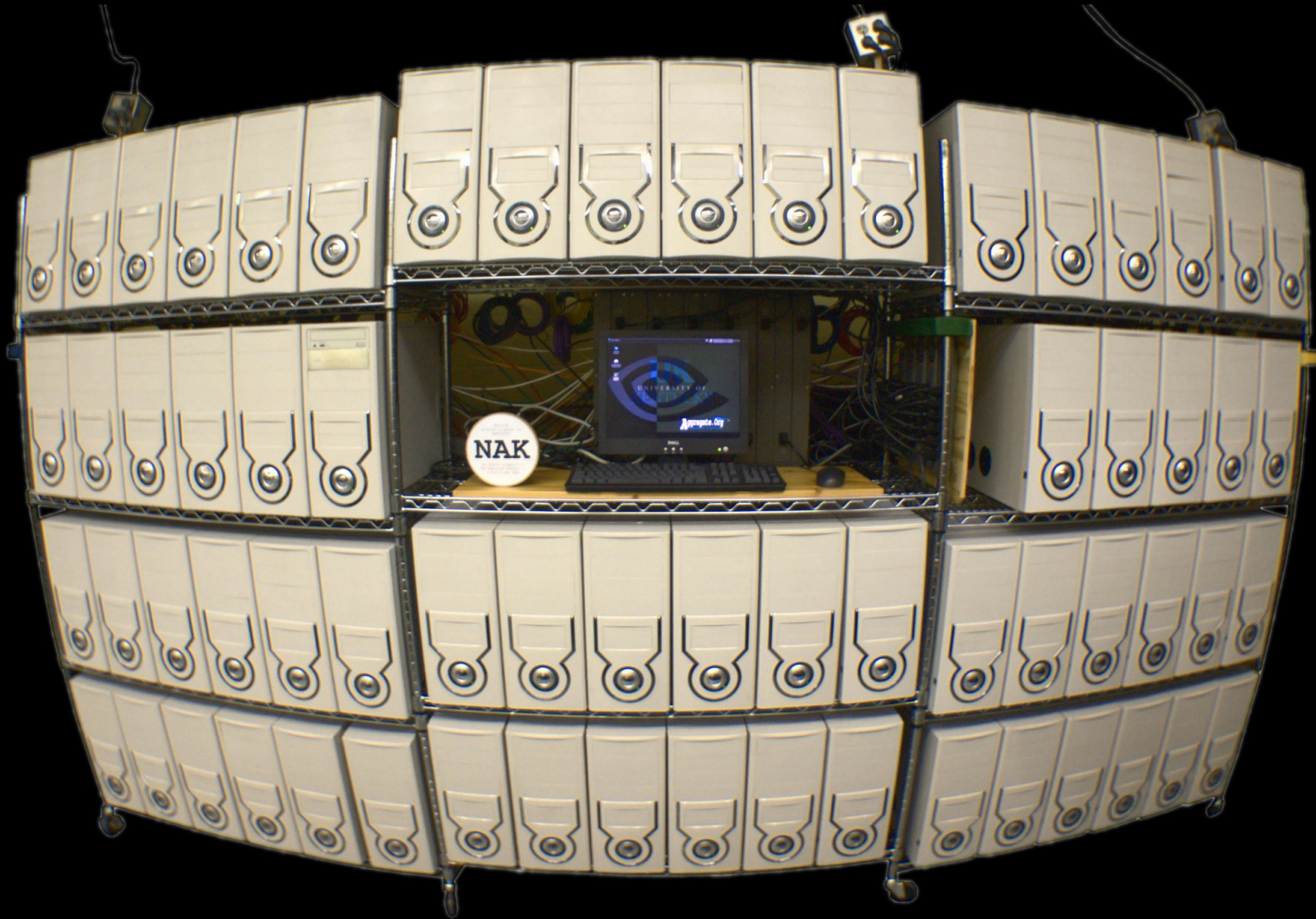




# 2006, ATI GPU



# 2010, NAK





# 2012, GPU Clusters



# *A Little Progress...*

A GFLOPS is 1 Billion {+,\*} per second

1992     MasPar MP1     \$1,000,000 / GFLOPS

# *A Little Progress...*

A GFLOPS is 1 Billion {+,\*} per second

1992	MasPar MP1	\$1,000,000 / GFLOPS
2000	KLAT2	\$650 / GFLOPS

# *A Little Progress...*

A GFLOPS is 1 Billion {+,\*} per second

1992	MasPar MP1	\$1,000,000 / GFLOPS
2000	KLAT2	\$650 / GFLOPS
2003	KASY0	\$84 / GFLOPS

# *A Little Progress...*

A GFLOPS is 1 Billion {+,\*} per second

1992	MasPar MP1	\$1,000,000 / GFLOPS
2000	KLAT2	\$650 / GFLOPS
2003	KASY0	\$84 / GFLOPS
2010	NAK	\$0.65 / GFLOPS

2022 GeForce RTX3090Ti peak is 40 TFLOPS  
@ \$2K (not counting host)... \$0.05 / GFLOPS

# The Future

- Everything is moving down...  
Your cell phone outruns the 1992 MasPar MP1; supercomputer today, in your cell phone soon
- More parallelism (and maybe **quantum** too?)
- More heterogeneous (helped by **dark silicon**)
- Everything contains a connected computer (e.g., **IoT: Internet of Things**)... a good thing?



# A Quick Virtual Tour of the KAOS Lab in Marksbury





# Dietz's ECE Lab, 108





# Machine Room, 108A



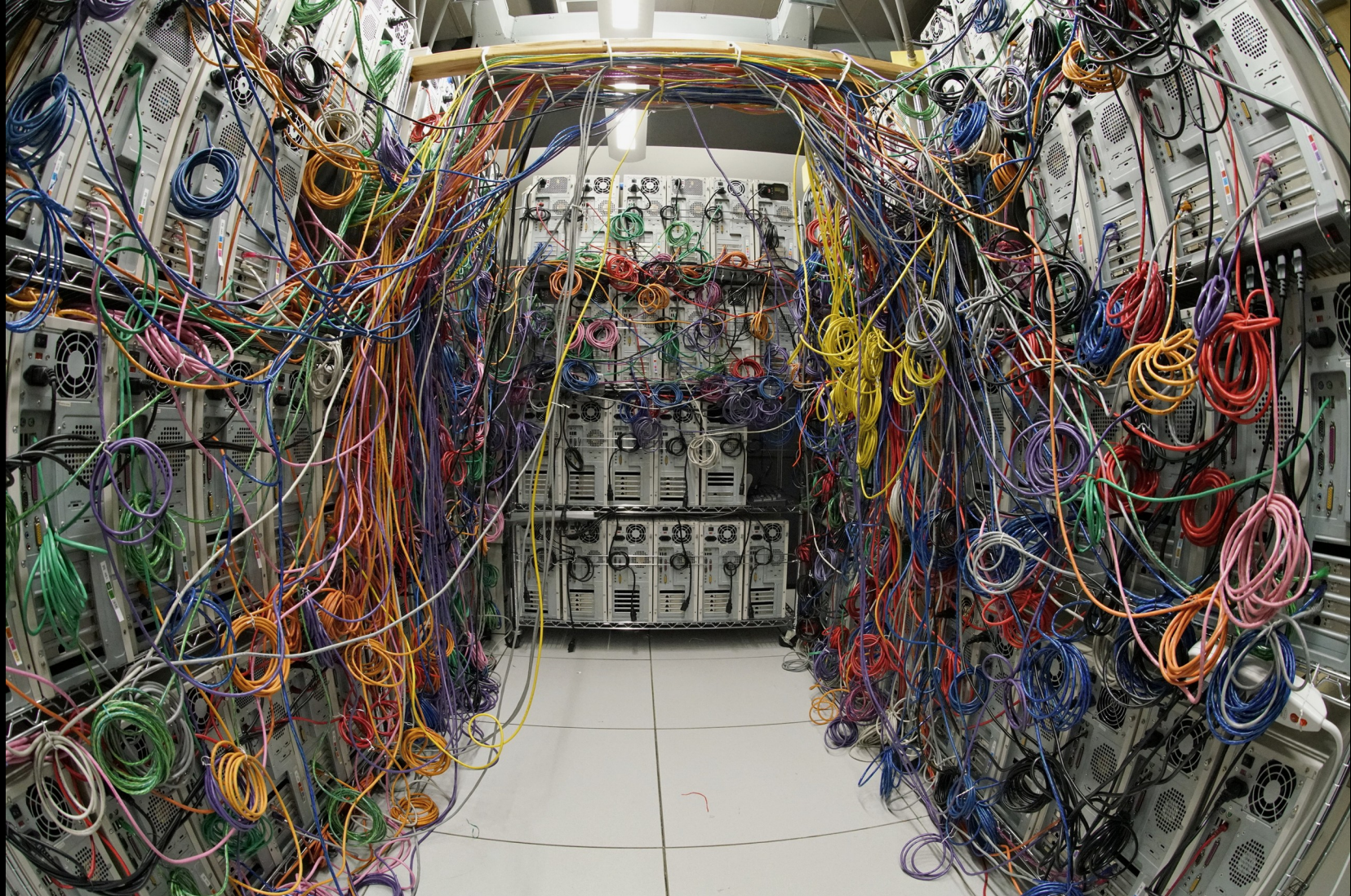


# RAID, Heads, & Rack Cluster



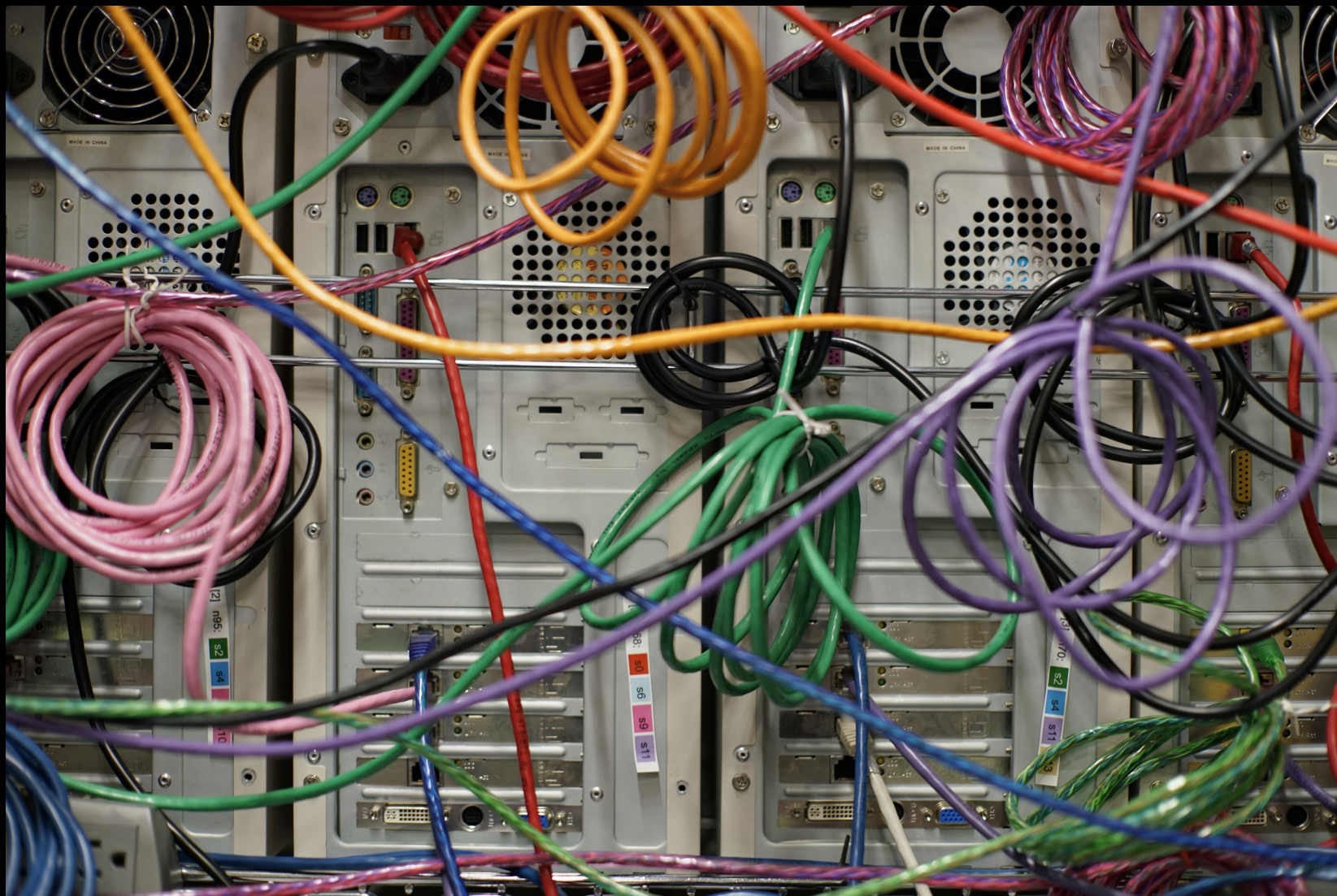


# Our “Big” FNN GPU Cluster





# How to Wire an FNN





# Our Pseudo-Quantum Stuff: Parallel Bit Pattern Computing

