

CSc 522: Parallel and Distributed Computing

- Instructor: David Lowenthal

Parallel Architecture

Why parallelism?

1. Finish applications sooner
 - Search engine
 - High-res graphics
 - Weather prediction
 - Nuclear reactions
 - Bioinformatics
2. Because CPUs aren't getting faster
3. Obtain more resources
 - E.g., More memory, disk

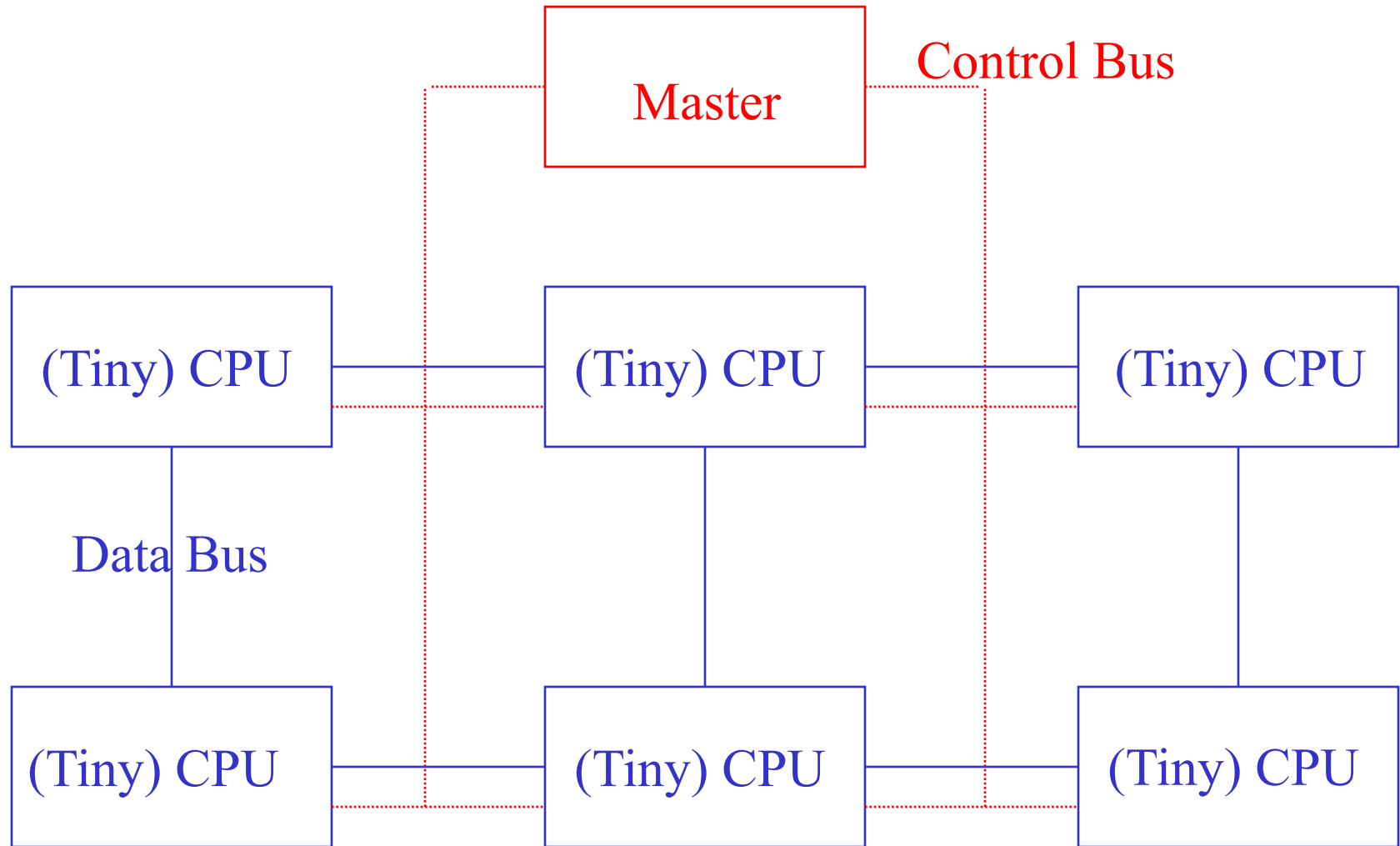
Why distributed computing?

- Reliability
- Load sharing
- Availability

Parallelization issues

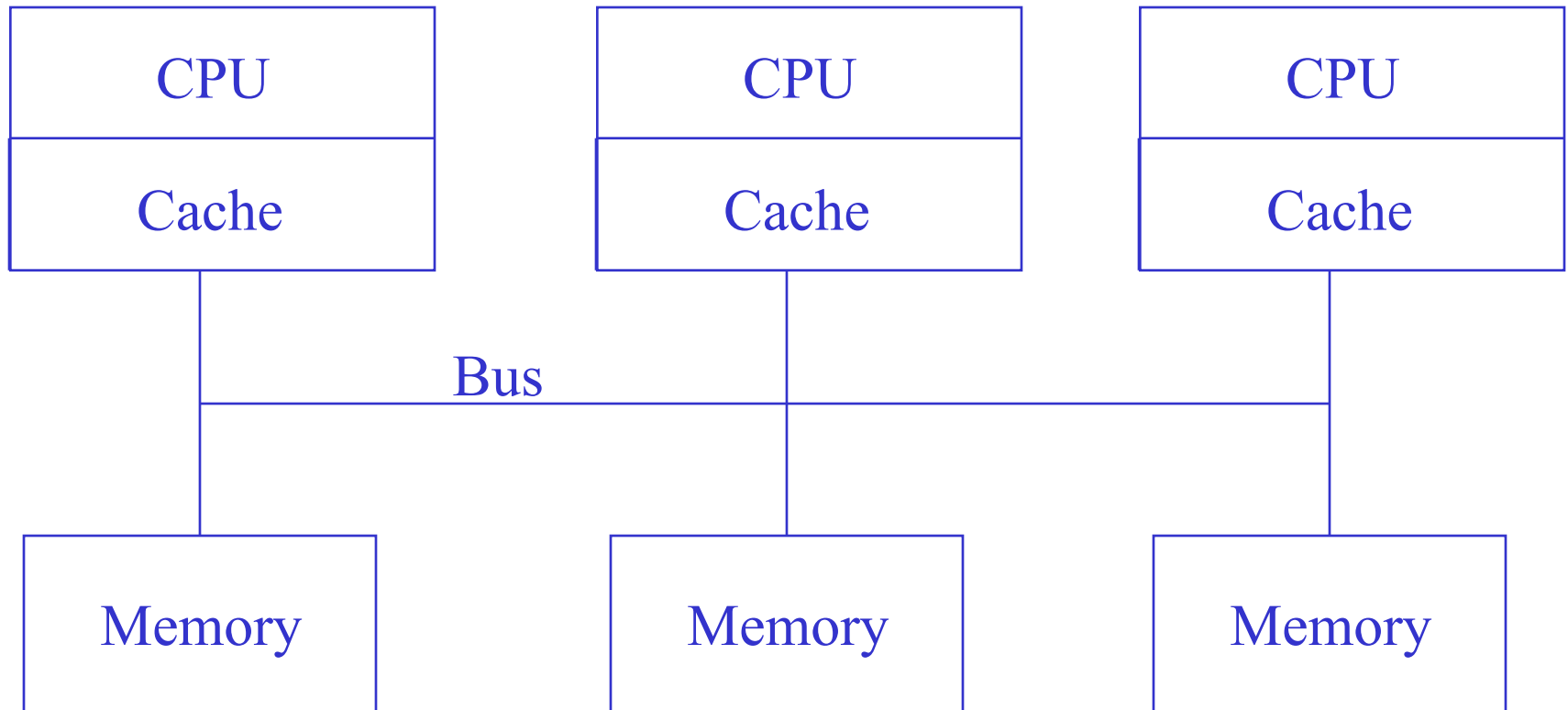
- How many CPUs?
- How to synchronize?
- How to communicate?
- How to determine granularity?
- General purpose vs special purpose?
- What is the programmer's view of the machine?

SIMD machine (e.g., Connection Machine)



Instructions broadcast to all; implicit synchronization betw. instructions

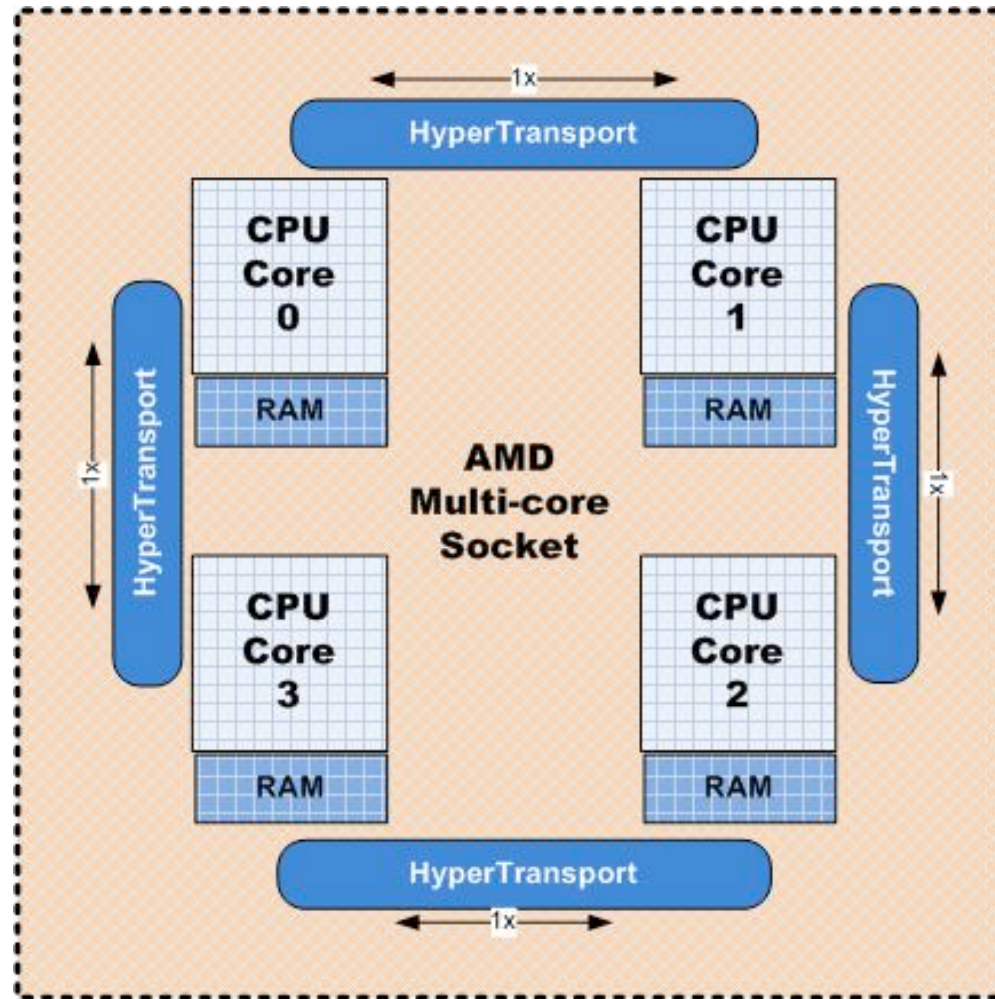
Shared-Memory Multiprocessor ("Multicore")



Memory is shared; Cache coherence is an issue

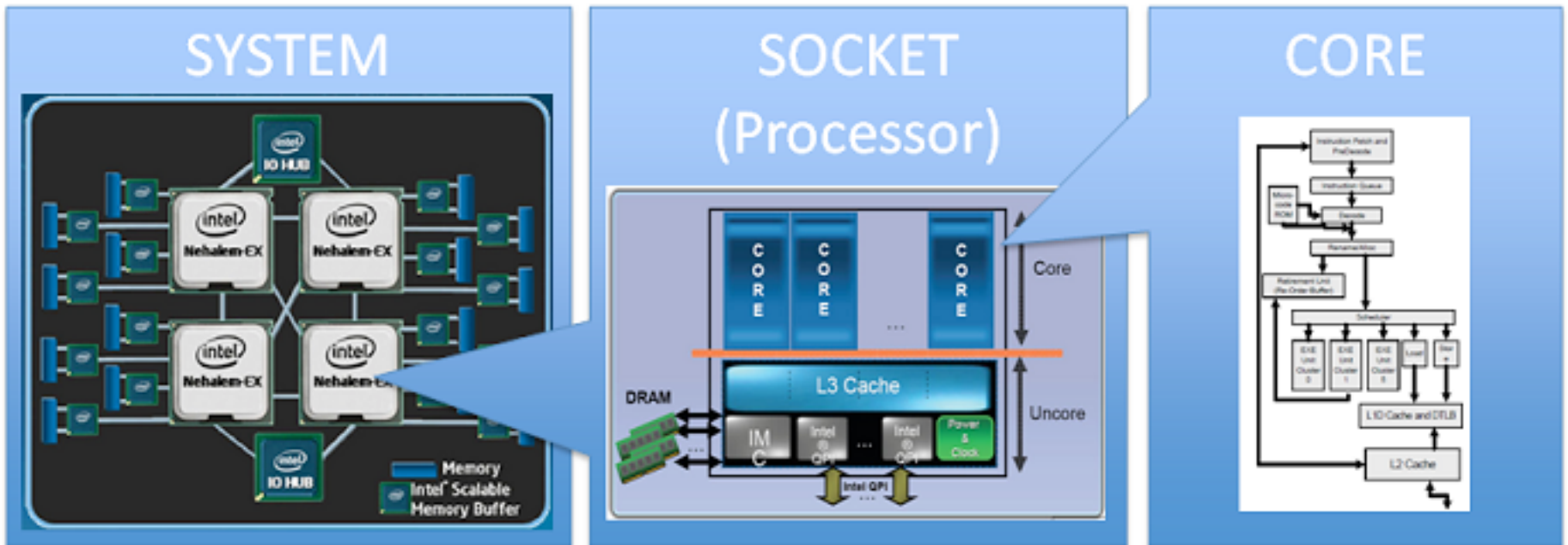
MIMD machine; each core can execute independent instruction stream

Typical Layout of a Socket

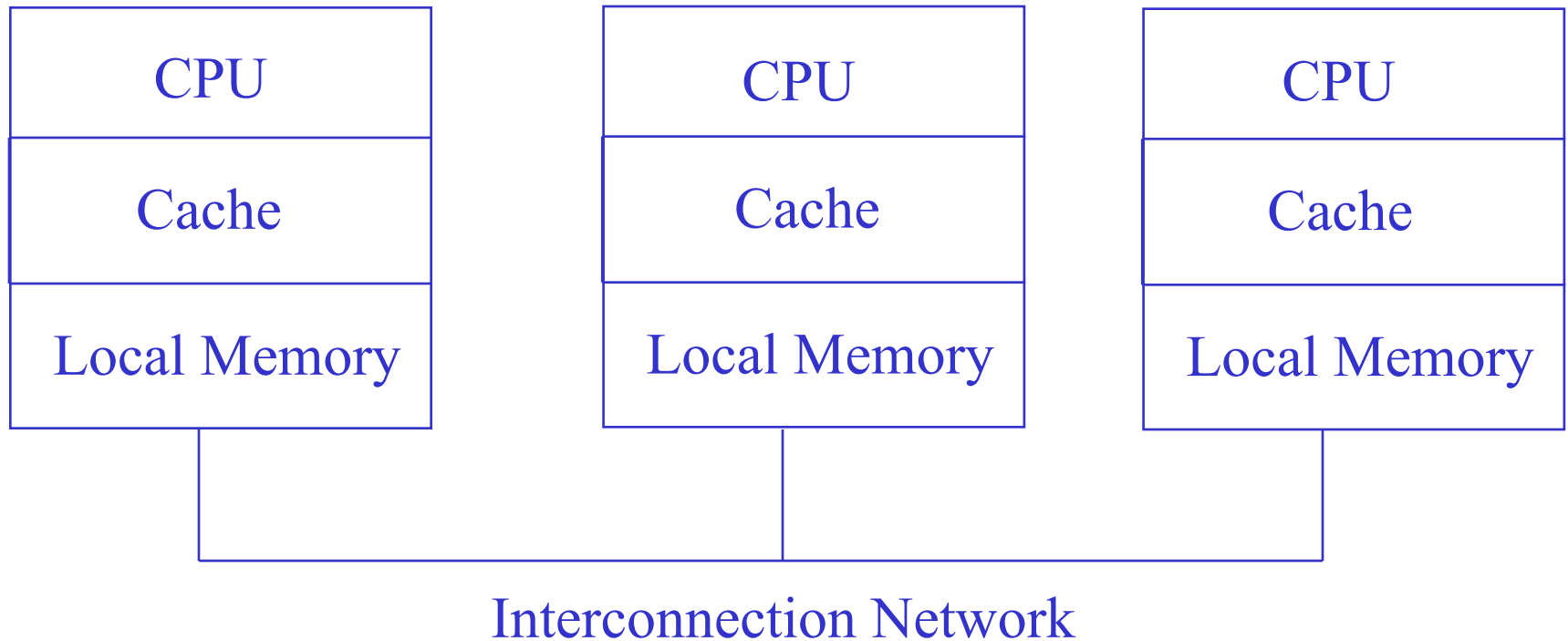


Multiple Sockets on a Chip

(picture courtesy of Intel)

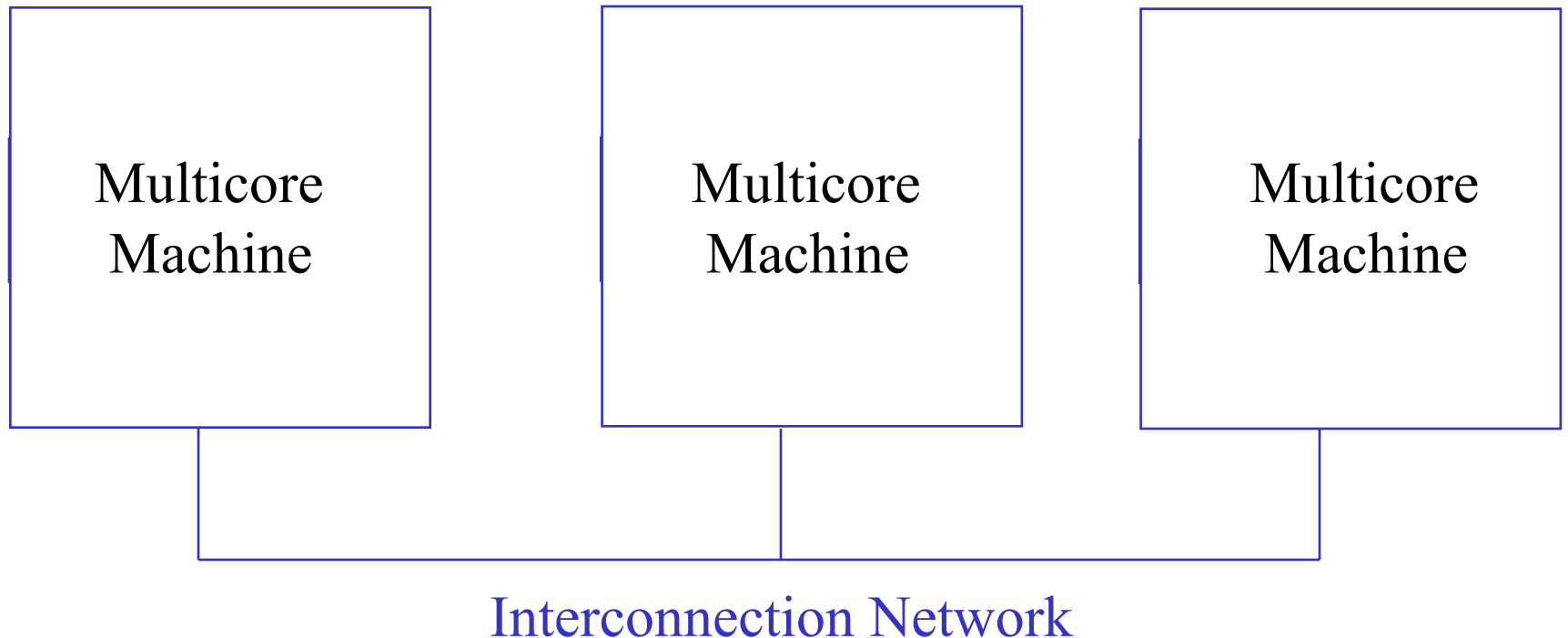


Distributed Memory Multicomputer



Memory is not shared
Also a MIMD machine

All Machines are Multicore (this is still a multicomputer)



Memory is not shared between machines

Key Advantage/Disadvantage: Shared-Memory Multiprocessors

- Advantage:
 - Can write sequential program, profile it, and then parallelize the expensive part(s)
 - No other modification necessary
- Disadvantage:
 - Does not scale to large core counts
 - Bus saturation, hardware complexity

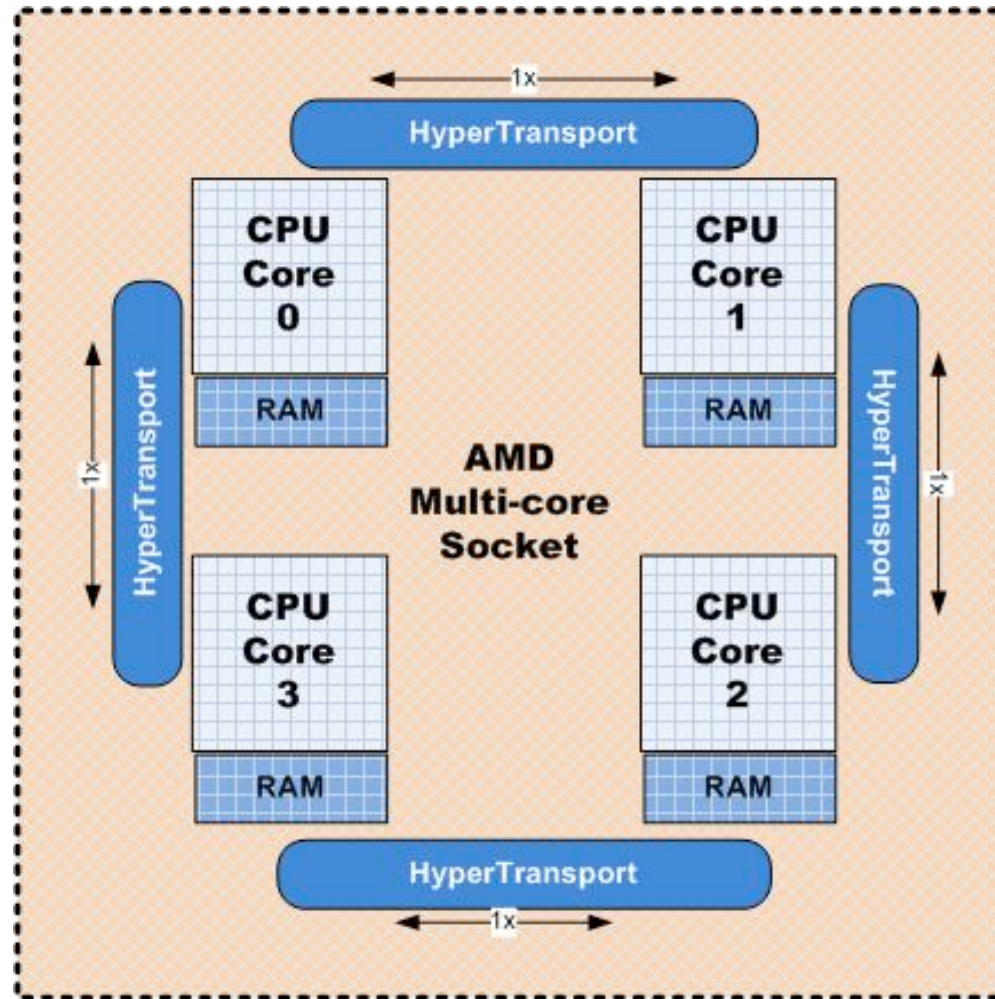
Key Advantage/Disadvantage: Distributed-Memory Multicomputers

- Advantage:
 - Can scale to large numbers of nodes
- Disadvantage:
 - Harder to program
 - Must modify *entire* program even if only a small part needs to be parallelized

Hybrid machines

- NUMA shared memory machines
 - NUMA: Non-Uniform Memory Access time
 - Physically distributed memory in the hardware, but user sees a shared-memory model
 - Hardware satisfies any remote memory request
 - Most multicore machines are actually NUMA (e.g., AMD Opterons)
 - Even if programmer can use shared-memory programming model, must pay attention to locality for maximum performance

Typical Layout of a Socket



Significant NUMA effects

The Cloud

- Cloud computing is generally thought to be aimed at distributed computing, but this is not really true any more
 - Example: Amazon EC2 rents HPC cluster nodes.
 - Recently fast networking has become available
 - Used to be 10Gb Ethernet was fastest
 - EC2 now has a 100Gb instance, and Microsoft Azure has Infiniband

High-End Architectures

- BlueGene/L (Lawrence Livermore National Lab)
 - #1 in world from 2004--2007
 - Up to over 100K cores
 - Disruptive design
 - In a sense, was similar to the rise of multicore machines--- instead of a smaller number of fast machines, a (much) larger number of slow machines

High-End Architectures

- Jaguar (Oak Ridge National Lab)
 - Petaflop machine; #1 in world in 2009
 - 224,000 Opteron cores total
 - 18,688 compute nodes; each is a dual-socket six-core node
 - Infiniband network
 - Provides low latency (can be < 1 microsecond) and high bandwidth (think several GB/s)
 - Consumes 7 MW of power
 - A lot of power for 1.75 petaflops (why is this relevant?)
 - Flop is a floating point operation per second

High-End Architectures

- Tianhe-1A (China)
 - Overtook Jaguar in 2010 (4 petaflops peak)
 - 14K Xeons plus 7K GPUs
 - Custom network; twice as good as Infiniband
 - Consumes only 4 MW of power
 - Xeons more power efficient (also a later chip); plus, GPUs are extremely power efficient
 - However, how easy is it to reach peak performance?

High-End Architectures

- K computer (Japan)
 - 8 petaflops (took #1 ranking in 2011)
 - 88K Sparcs at 8 cores each
 - Custom network called *Tofu* (3-d torus interconnect)
 - Consumes 10-13MW of power

High-End Architectures

- Sequoia [BG/Q] (IBM/Lawrence Livermore)
 - 16 petaflops (took #1 ranking in 2012)
 - 98K Power nodes at 16 cores each
 - Consumes 8 MW of power

High-End Architectures: Tianhe-2

- 54.9 Petaflops
- 32,000 Ivy Bridge Xeon sockets
 - Each has 12 cores
- 48,000 Xeon Phi accelerators
 - Each has 57 cores
- Total: 3.1M cores
- Custom interconnect (fat tree); low latency (9 microseconds) and high bandwidth (6 GB/s)
- Consumes 17.6 MW of power

Tianhe-2 Compute Node

- 2 Ivy Bridge sockets; 3 Xeon Phi boards
- 64 GB RAM
- Xeon Phi acts as coprocessor
 - Each of the 57 cores has 4 hardware threads (“hyperthreads”) and runs at 1.1 GHz (low clock speed, but many cores)

Tianhe-2 Power Consumption

- 17.6 MW peak power
- Additionally, 7 MW for cooling using chilled water
- Well over DOE's "limit", assuming that limit is total power
- Performance 2x that of Titan (ORNL), but power consumption also 2x

Tianhe-2 Software

- Uses variant of Linux
- Provides common libraries for high-performance computing
 - Plus a mechanism for expressing codes for the Phi

High-End Architectures

- Summit (IBM/Oak Ridge)
 - 148 Petaflops
 - 4,608 nodes
 - 44 cores/node (22 cores/socket, 2 sockets/node)
 - 4 hyperthreads/core
 - 27,648 GPUs (six/node)
 - Consumes 10 MW of power
 - Can we program it to get near peak performance?

Summit and Sierra

- Summit (ORNL) and Sierra (LLNL) are the two fastest supercomputers in the world
- Both use “fat nodes”
 - Dual sockets
 - Many GPUs
 - More memory
- With fat nodes, there are fewer nodes
 - Better for reducing number of messages
 - Lower runtime variability
- Adaptive routing

Power Issues

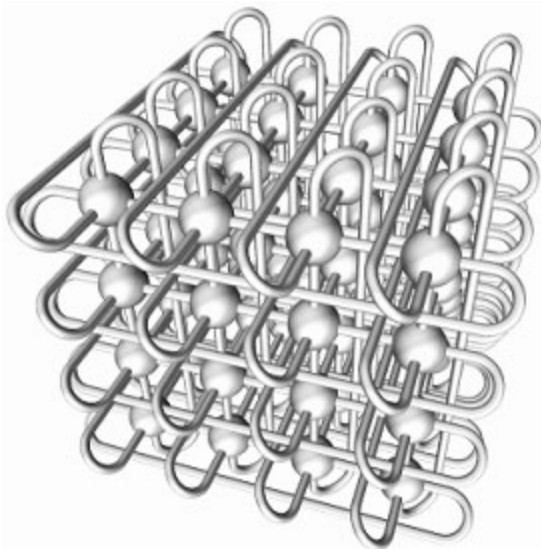
- Current HPC goal is to hit an exaflop
 - 1 exaflop is 1000 petaflops
- DOE (i.e., the government/customer) originally allocated 20 MW of power to hit an exaflop
 - Current target is 40 MW for the first exaflop machine
 - Will we be able to get near an exaflop for anything other than the “race car” applications?

Interconnects

- How are the nodes of a system connected?
- Critical question for efficiency, as HPC applications communicate
 - Potentially a lot of data sent/received, and frequently
- Will cover this in detail later in the semester

BlueGene/L Torus Network

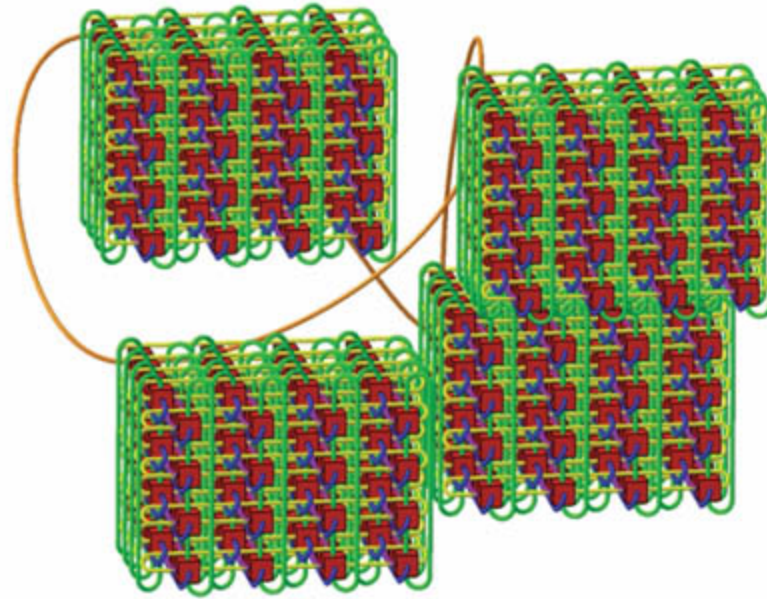
(picture courtesy of cluster-design.org)



- Each node has six neighbors
- If dimensions are $N \times N \times N$, worst case number of hops is $3N/2$.
 - This is because in each direction, the worst case is hopping half the size of that dimension

BlueGene/Q 5-d Torus Network

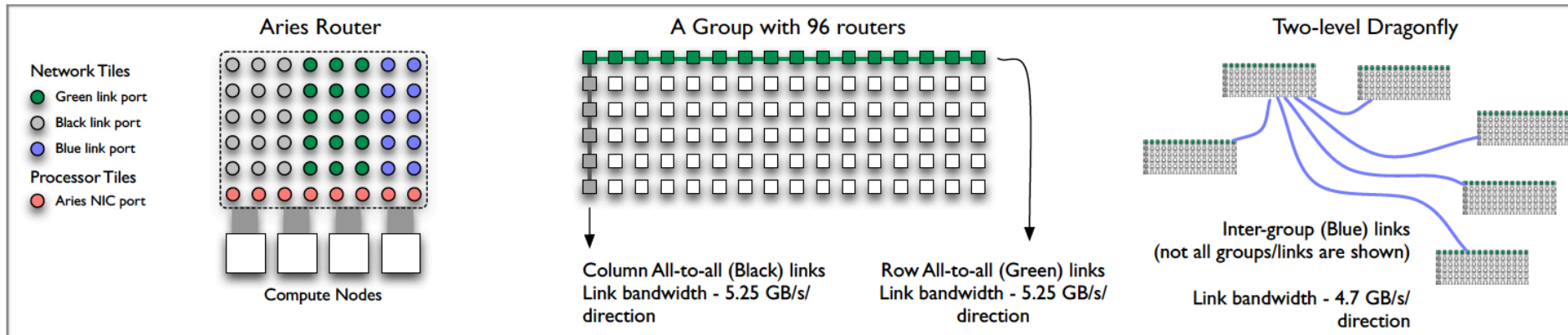
(picture courtesy of LLNL)



- Each node has ten neighbors
- If dimensions are $N \times N \times N \times N \times N$, worst case number of hops is $5N/2$.
 - N will decrease in size as dimensionality of Torus increases, assuming a fixed node count

Dragonfly Network

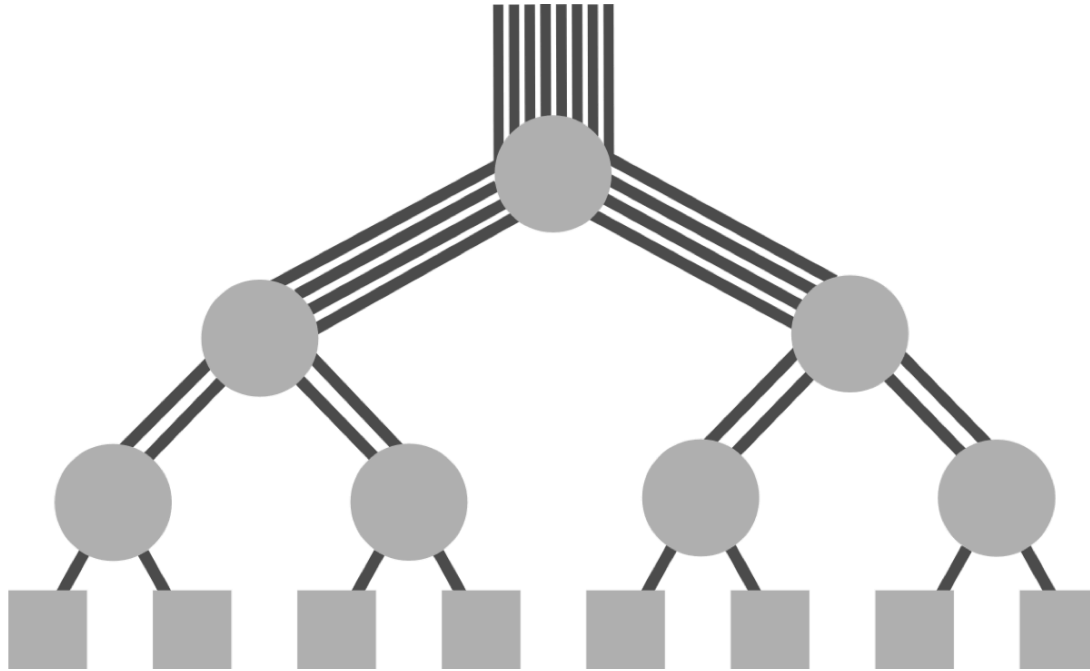
(picture courtesy of paper by Bhatele et al.)



- All-to-all connectivity in row and column of each group
 - Can get to any node in group in two hops
 - Implies ability to get to any node in any group in no more than five hops
 - Will do adaptive routing if there is congestion

Fat Tree Interconnects

(picture courtesy of cluster-design.org)



- Nodes are at bottom of tree; switches at interior nodes
 - Bandwidth increases higher in the tree
 - Handles collective communication