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

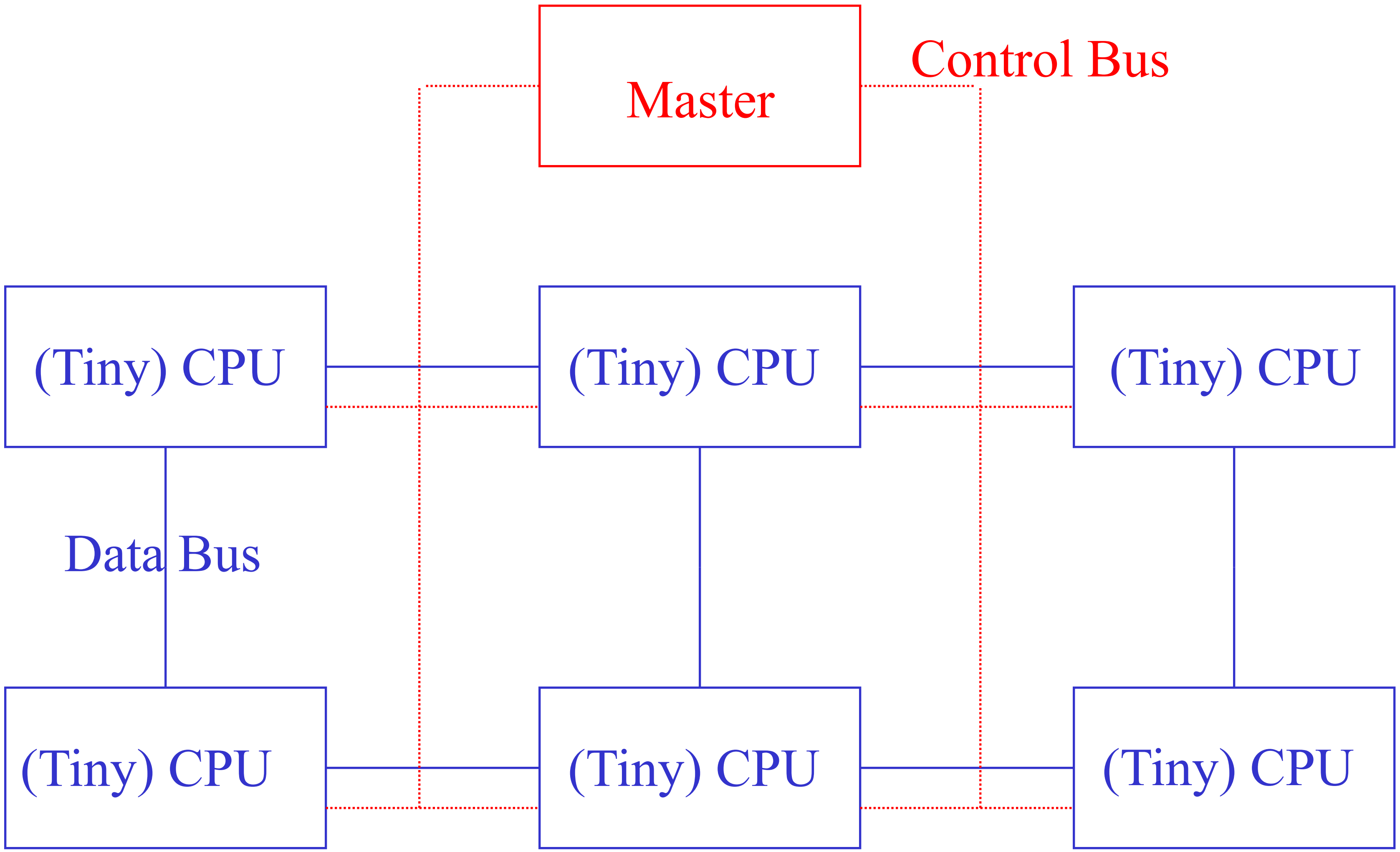
Why distributed computing?

- Reliability
- Load sharing
- Availability

Parallelization issues

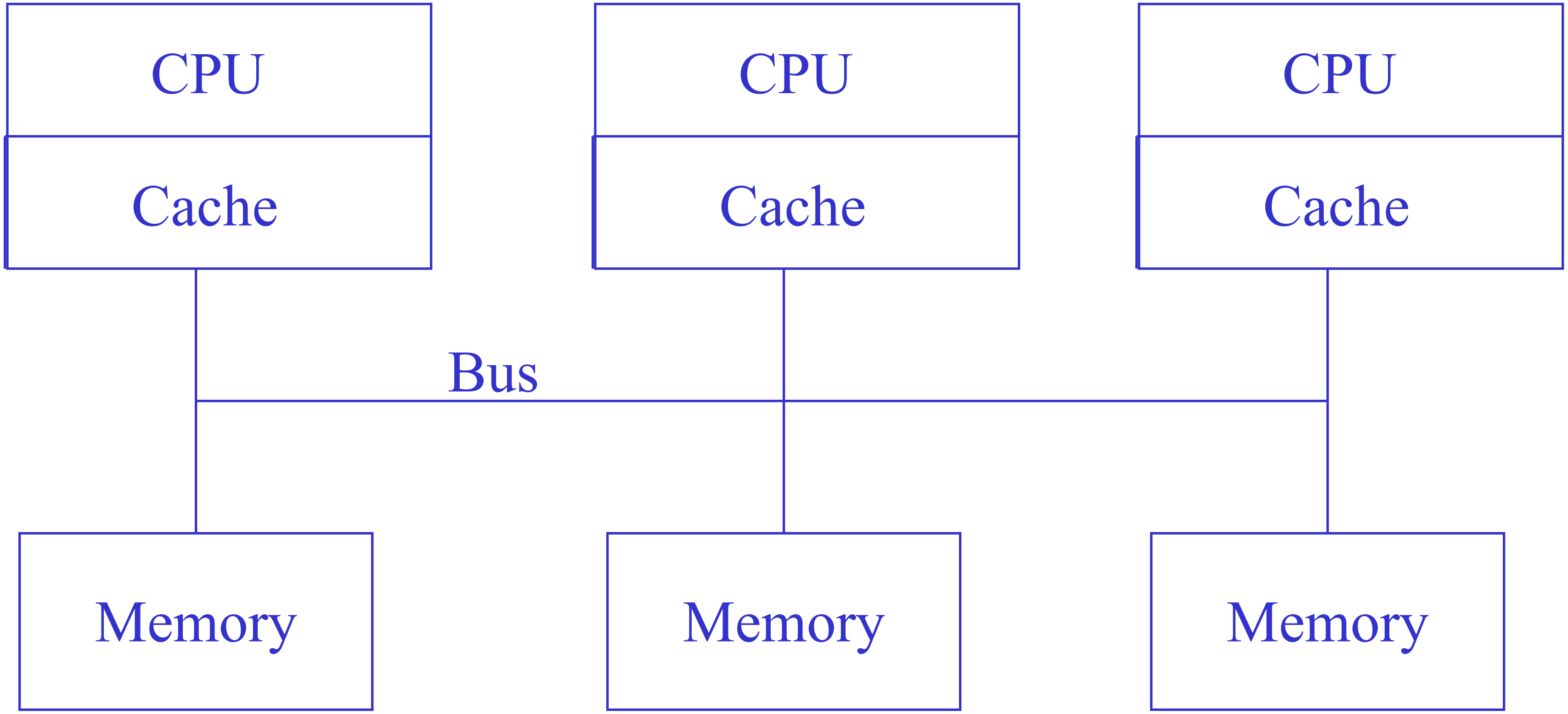
- How many CPUs?
- How to synchronize?
- How to communicate?
- How to determine granularity?
- General purpose vs special purpose?
- What is the programmer’s view of the machine?

SIMD machine (e.g., Connection Machine)



Instructions broadcast to all; implicit synchronization betw. instructions

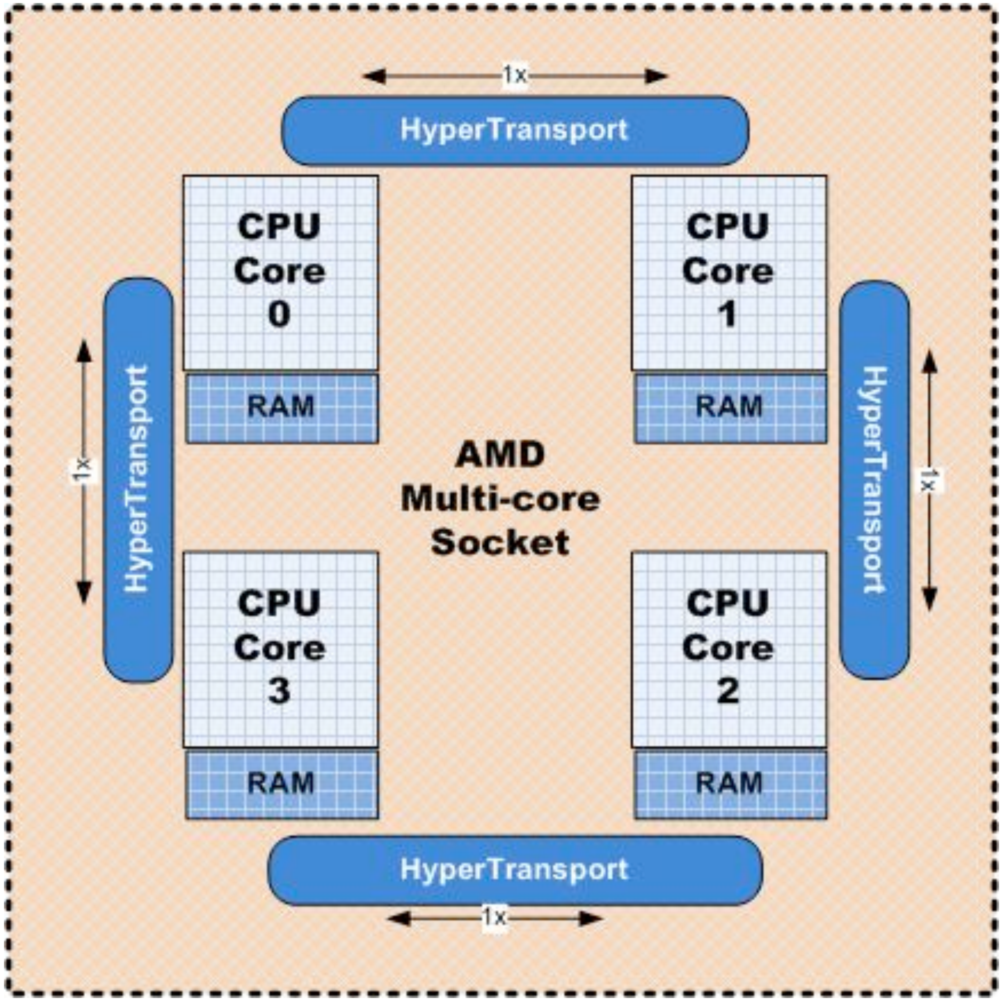
Shared-Memory Multiprocessor (“Multicore”)



Memory is shared; Cache coherence is an issue

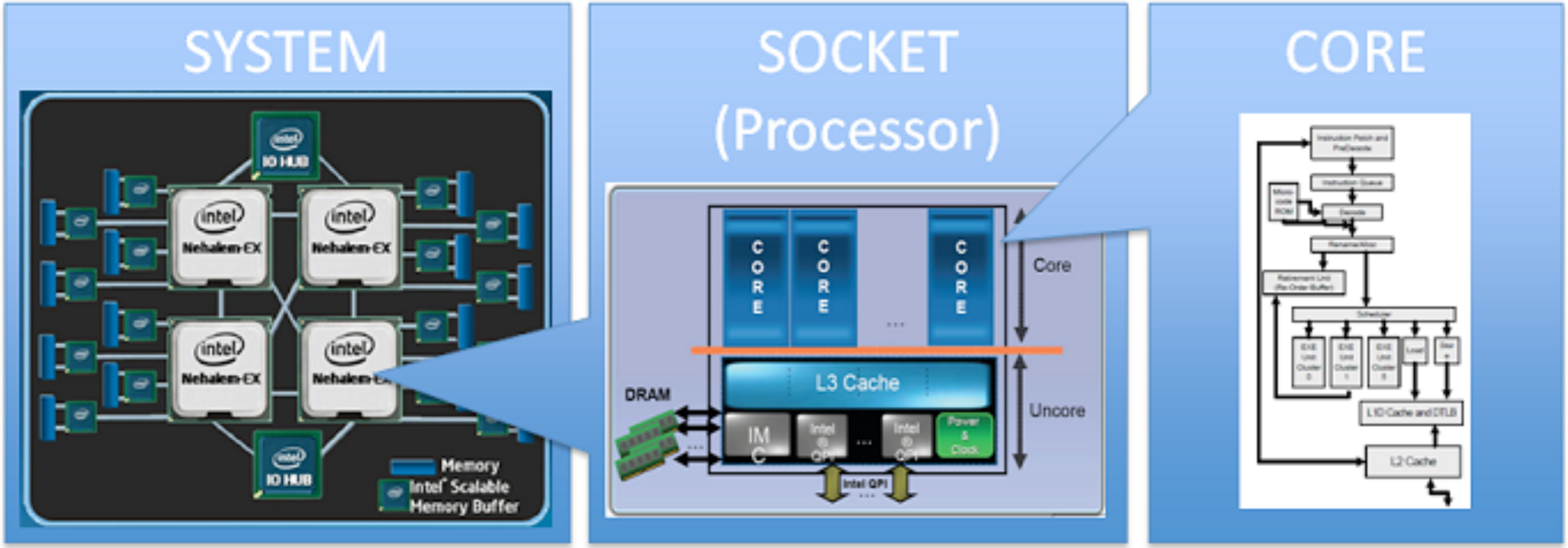
MIMD machine; each core can execute independent instruction stream

Typical Layout of a Socket

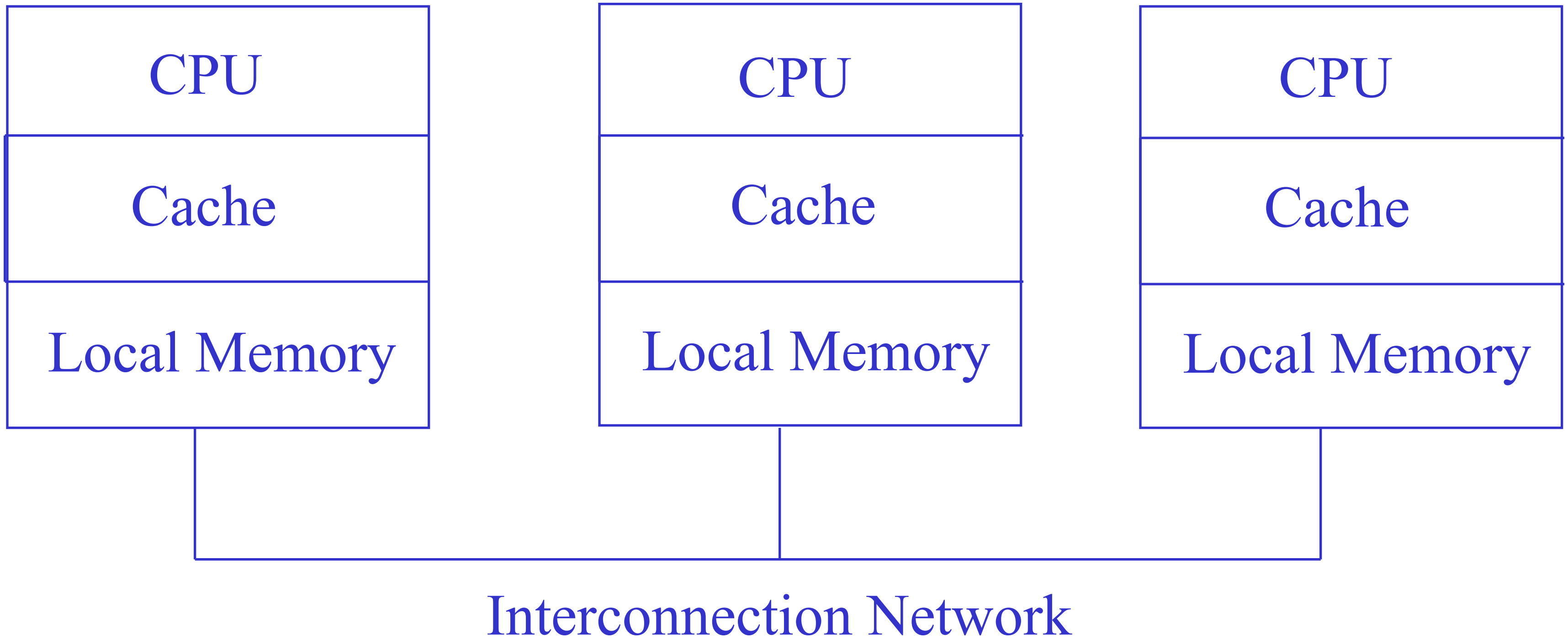


Multiple Sockets on a Chip

(picture courtesy of Intel)

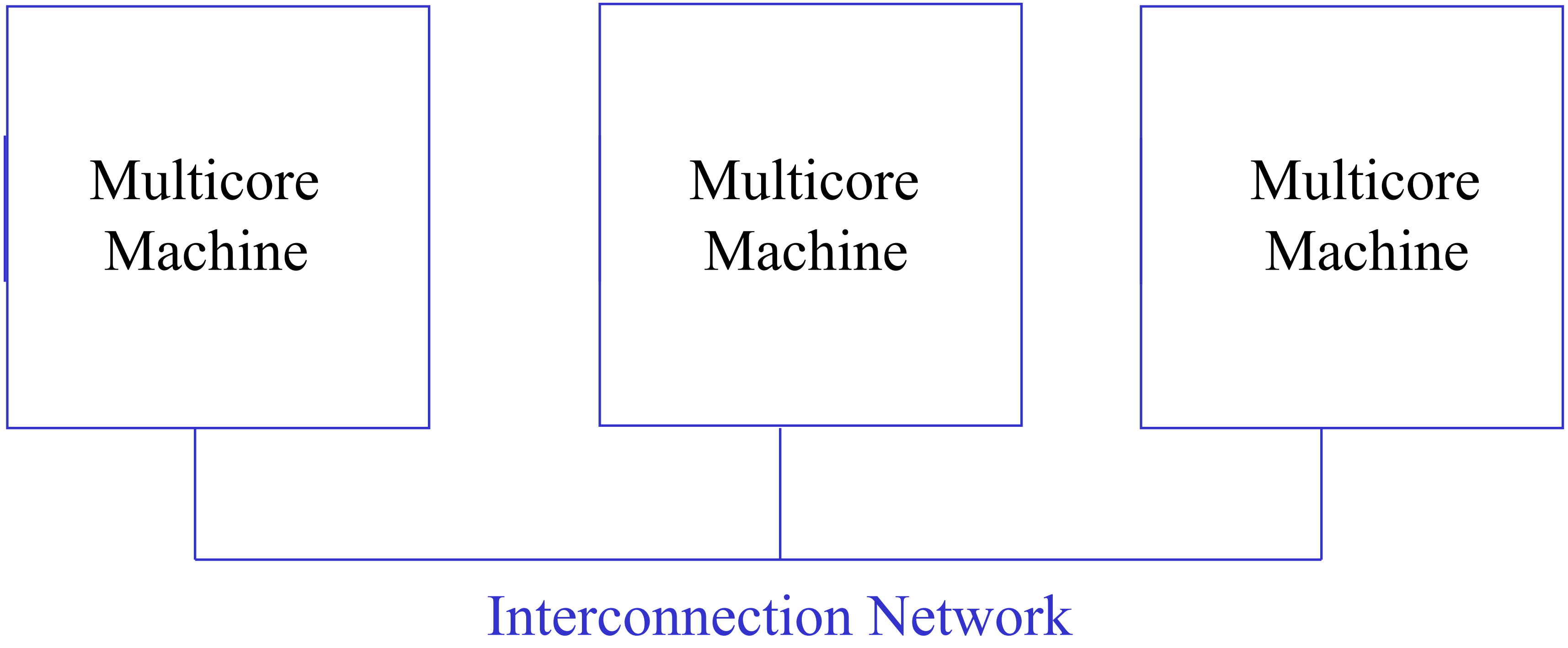


Distributed Memory Multicomputer



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

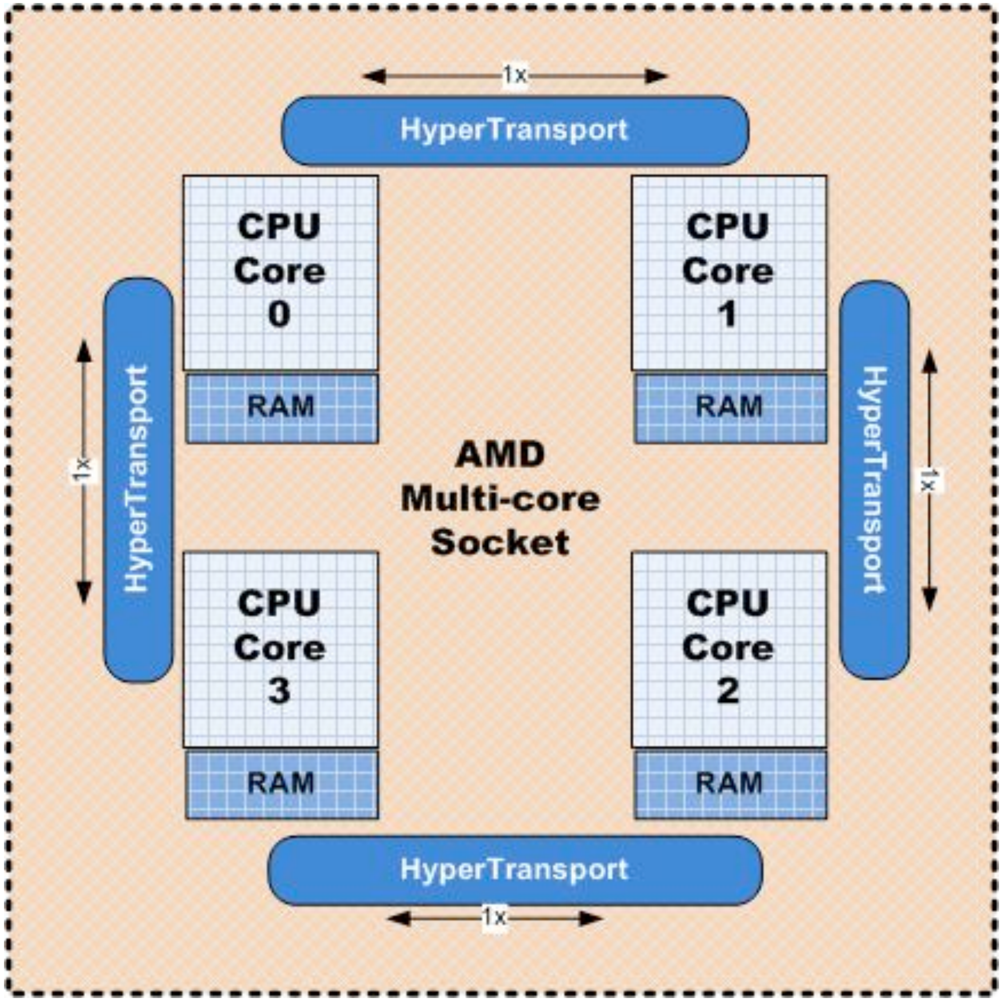
Key Advantage/Disadvantage: Distributed-Memory Multicomputers

- 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.
  - Communication performance, however, is stuck at 10G Ethernet
    - This is still slow compared to typical Infiniband-based supercomputers, but many applications do not require Infiniband
    - Infiniband options exist (e.g., Profitbricks)

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?)

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 (China)
  - 54.9 Petaflops
  - 32,000 Ivy Bridge Xeon sockets
  - 48,000 Xeon Phi accelerators
  - Consumes 17.6 MW of power

Power Issues

- Current HPC goal is to hit an exaflop
  - 1 exaflop is 1000 petaflops
  - Note: FLOPS are floating point ops per second
- DOE (i.e., the government/customer) has allocated 20 MW of power to hit an exaflop
  - Power much closer to 20 MW than performance is to an exaflop
  - Hardware improvements will help
    - But, there will need to be advances in every facet of supercomputing to achieve an exaflop in 20 MW

BlueGene/L

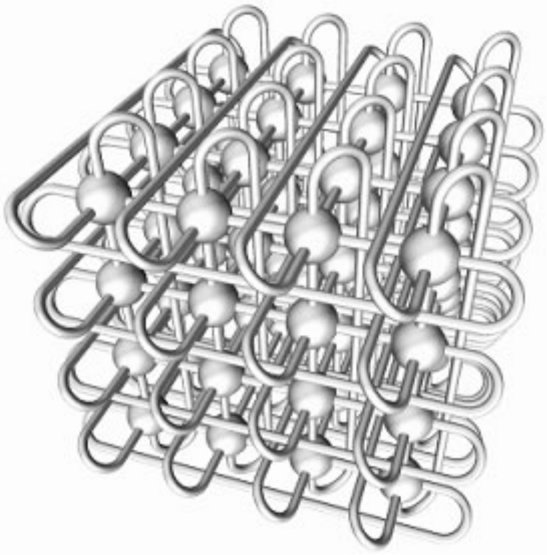
- Achieve massive parallelism with cheap, low-power processors (total power ~ 1 MW)
- Uses embedded power PC chips
  - 2 cores per chip, one for compute and one for communicating; idea is to avoid interrupting computation for communication
  - However, can utilize both cores for computation (if so, better not have much communication)
- Good for applications that scale well to large numbers of processors

BlueGene/L, continued

- Have one I/O node for many compute nodes
- Several network connections per chip (not one)
  - one to the 3-d torus (for data exchange)
  - one to a barrier network
    - idea: isolate this traffic as barriers happen quite often
  - one to a tree network (including to I/O node)
    - idea: similar to barrier network, but for fast collective operations

BlueGene/L Torus Network

(picture courtesy of cluster-design.org)



- Each node has six neighbors
- If dimensions are NxNxN, 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/L, continued

- Core cycle time slow
  - So cycles to memory are low also
    - A bit counterintuitive---the slower the processor clock is, the faster memory is in a relative sense
  - 6-10 cycles for L2 hit, 25 cycles for L2 miss, 75 cycles for L3 miss

BlueGene/L, continued

- Lightweight kernel runs on compute nodes
  - Want (ideally) no system "noise"
  - Reads/writes are shipped to I/O node, i.e., no read syscall implementation in compute node
  - No context switching or demand paging! (why?)
- Very good performance in terms of flops per watt (about 210 MF/W)

BlueGene/L, continued

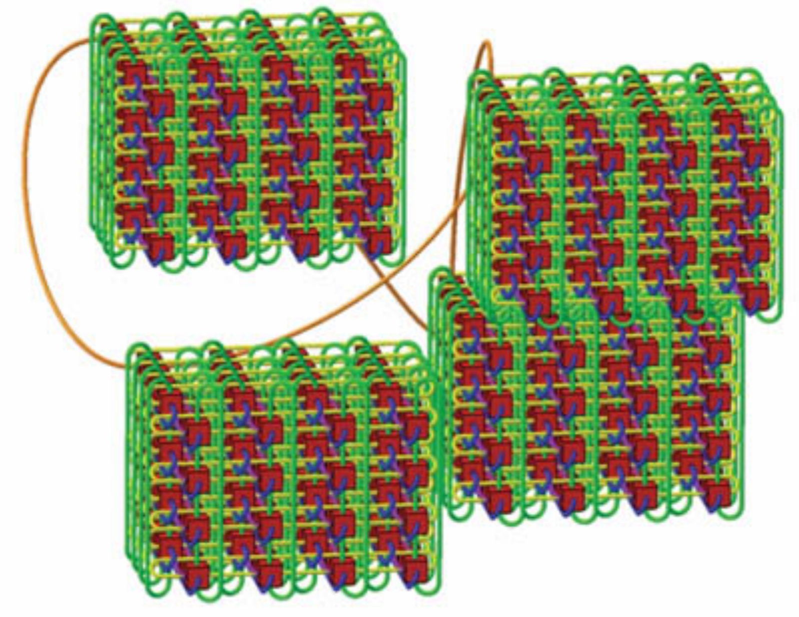
- Will not perform well for:
  - Apps that are disk bound
  - Apps that are not scalable and so need fast CPUs
  - Apps that cannot be run at a fine granularity

BlueGene/Q

- Interesting aspects are:
  - 18 1.6 GHz cores per chip
    - 1 core for communication; 1 core for failures
  - Interconnect is set up to allow groups of 512 nodes to be completely isolated
  - Hardware support for transactional memory and speculative execution
    - Former can substitute (more efficiently) for locks; latter can parallelize seemingly sequential loops
  - More general purpose than BG/L
  - Achieves 2000 MF/W
    - But no software control of power

BlueGene/Q 5-d Torus Network

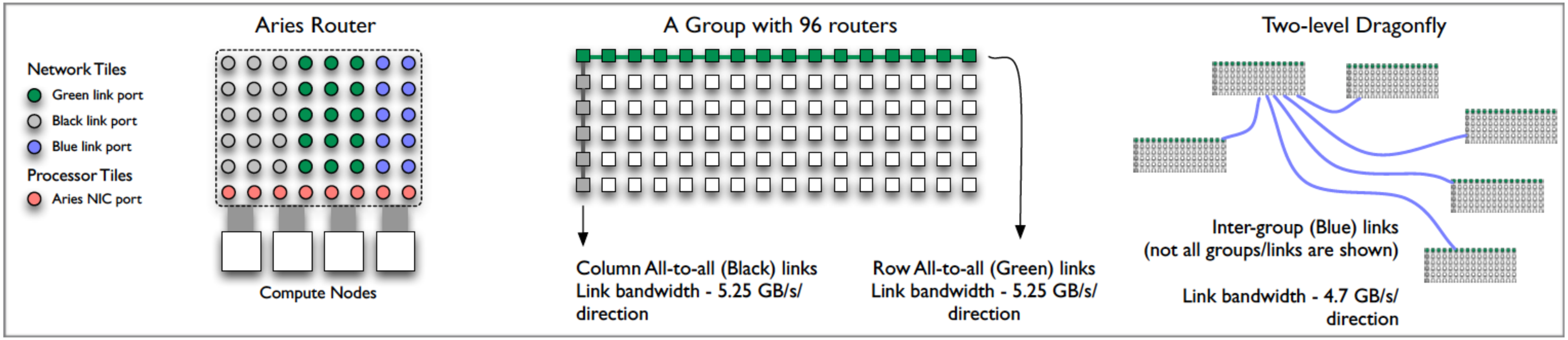
(picture courtesy of LLNL)



- Each node has ten neighbors
- If dimensions are NxNxNxNxN, 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.)



Aries Router
Network Tiles
Green link port
Black link port
Blue link port
Processor Tiles
Aries NIC port
Compute Nodes
A Group with 96 routers
Column All-to-all (Black) links
Link bandwidth - 5.25 GB/s/ direction
Row All-to-all (Green) links
Link bandwidth - 5.25 GB/s/ direction
Two-level Dragonfly
Inter-group (Blue) links
(not all groups/links are shown)
Link bandwidth - 4.7 GB/s/ direction

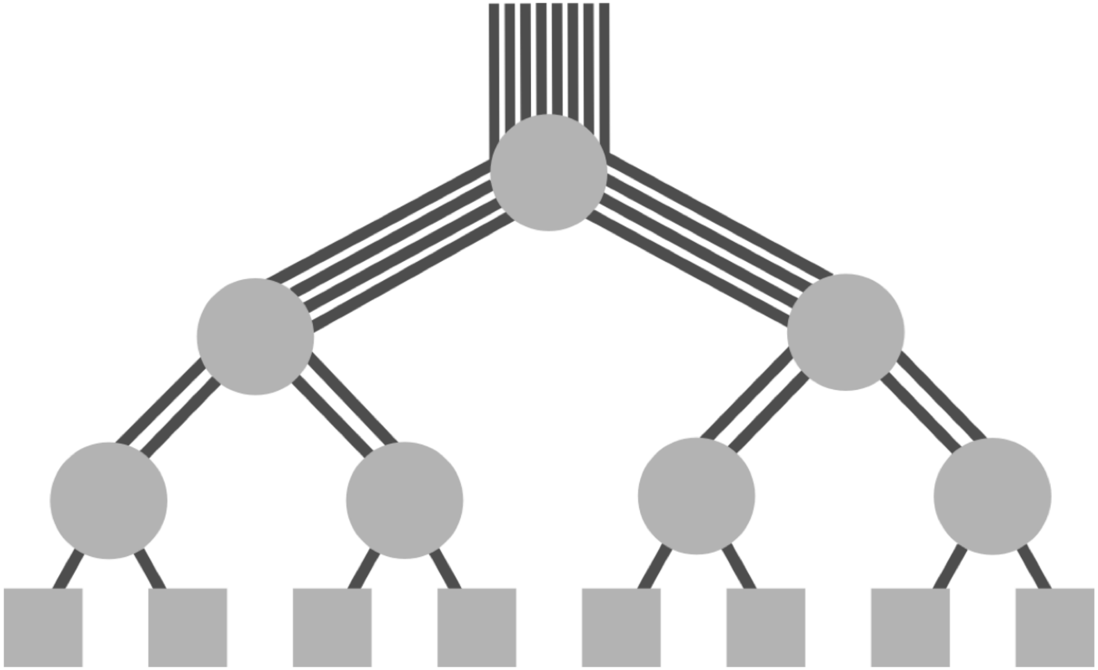
- All-to-all connectivity in row and column of each group
  - Can get to any node in group in two hops
  - Implies ability to get to any node in any group in no more than five hops
  - Will do adaptive routing if there is congestion

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 microsecs) and high bandwidth (6 GB/s)

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

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 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