

CSC 447: Parallel Programming for Multi-Core and Cluster Systems

Parallel Architectures

Instructor: Haidar M. Harmanani

Spring 2017

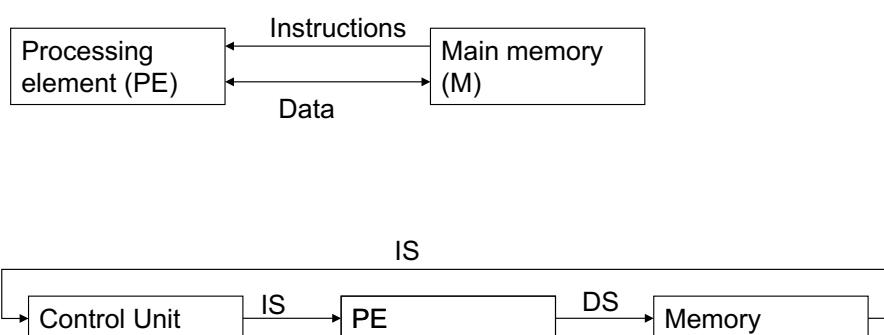
Outline

- Parallel architecture types
- Instruction-level parallelism
- Vector processing
- SIMD
- Shared memory
 - Memory organization: UMA, NUMA
 - Coherency: CC-UMA, CC-NUMA
- Interconnection networks
- Distributed memory
- Clusters
- Clusters of SMPs
- Heterogeneous clusters of SMPs

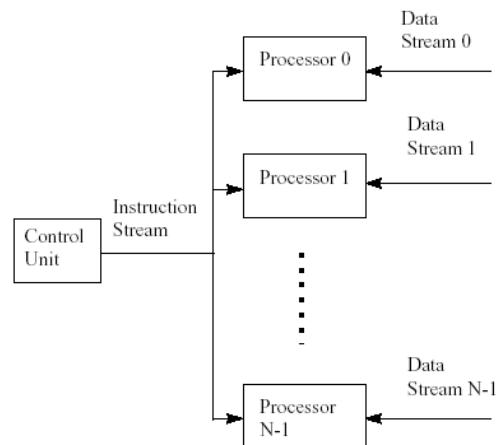
Flynn's Taxonomy

- The most universally accepted method of classifying computer systems
- Any computer can be placed in one of 4 broad categories
 - SISD: Single instruction stream, single data stream
 - SIMD: Single instruction stream, multiple data streams
 - MIMD: Multiple instruction streams, multiple data streams
 - MISD: Multiple instruction streams, single data stream

SISD

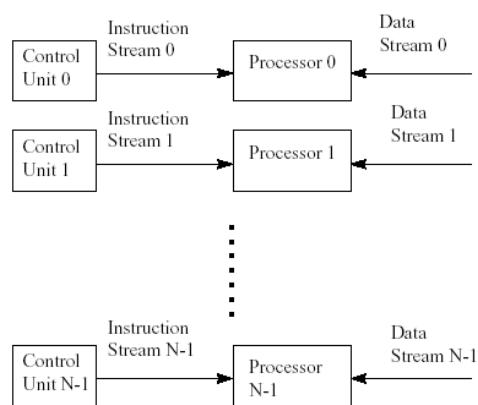


SIMD



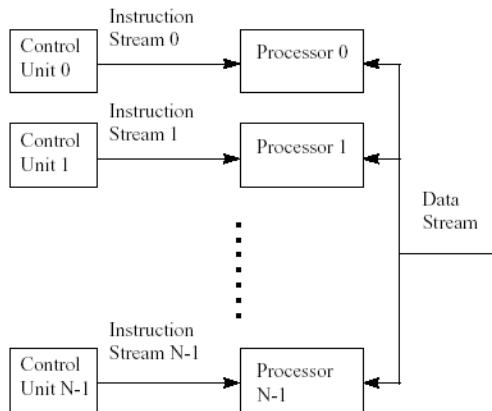
SISD system architecture of [Fly66]

MIMD



MIMD system architecture of [Fly66]

MISD

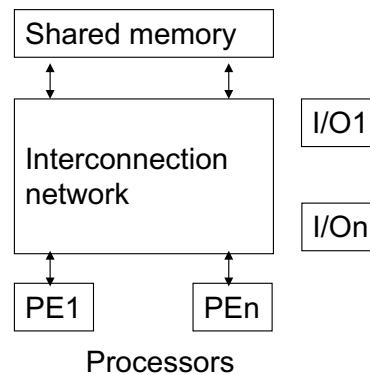


MISD system architecture of [Fly66]

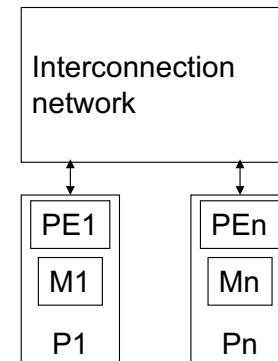
Flynn taxonomy

- Advantages
 - Universally accepted
 - Compact Notation
 - Easy to classify a system
- Disadvantages
 - Very coarse-grain differentiation among machine systems
 - Comparison of different systems is limited
 - Interconnections, I/O, memory not considered in the scheme

Classification Based on Memory



Shared memory - multiprocessors



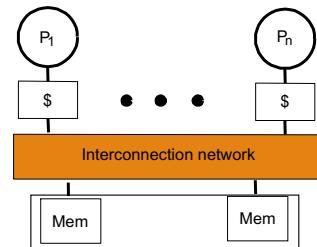
Message passing - multicompilers

Shared-memory multiprocessors

- Memory is common to all the processors.
- Processors easily communicate by means of shared variables.
- Models
 - Uniform Memory Access (UMA)
 - Non-Uniform Memory Access (NUMA)
 - Cache-only Memory Architecture (COMA)

The UMA Model

- Tightly-coupled systems (high degree of resource sharing)
- Suitable for general-purpose and time-sharing applications by multiple users.



Symmetric and asymmetric multiprocessors

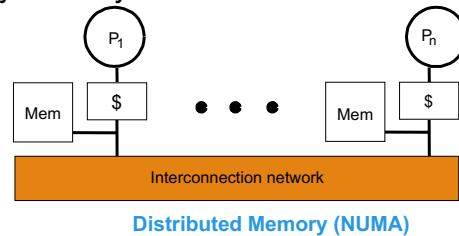
- Symmetric:
 - all processors have equal access to all peripheral devices.
 - all processors are identical.
- Asymmetric:
 - one processor (master) executes the operating system
 - other processors may be of different types and may be dedicated to special tasks.

The NUMA Model

- The access time varies with the location of the memory word.
- Shared memory is distributed to local memories.
- All local memories form a global address space accessible by all processors

Access time: Cache, Local memory, Remote memory

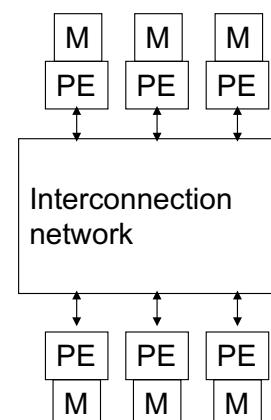
COMA - Cache-only Memory Architecture



Distributed Memory (NUMA)

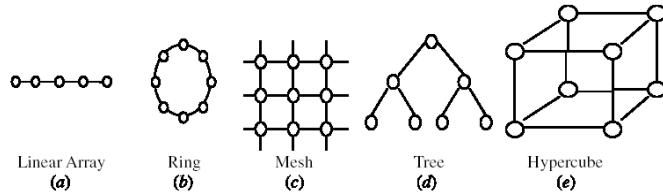
Distributed memory multicomputers

- Multiple computers- nodes
- Message-passing network
- Local memories are private with its own program and data
- No memory contention so that the number of processors is very large
- The processors are connected by communication lines, and the precise way in which the lines are connected is called the topology of the multicomputer.
- A typical program consists of subtasks residing in all the memories.

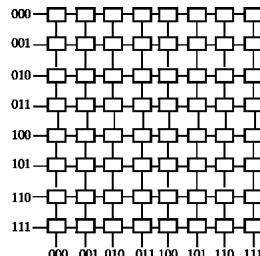


Classification based on type of interconnections

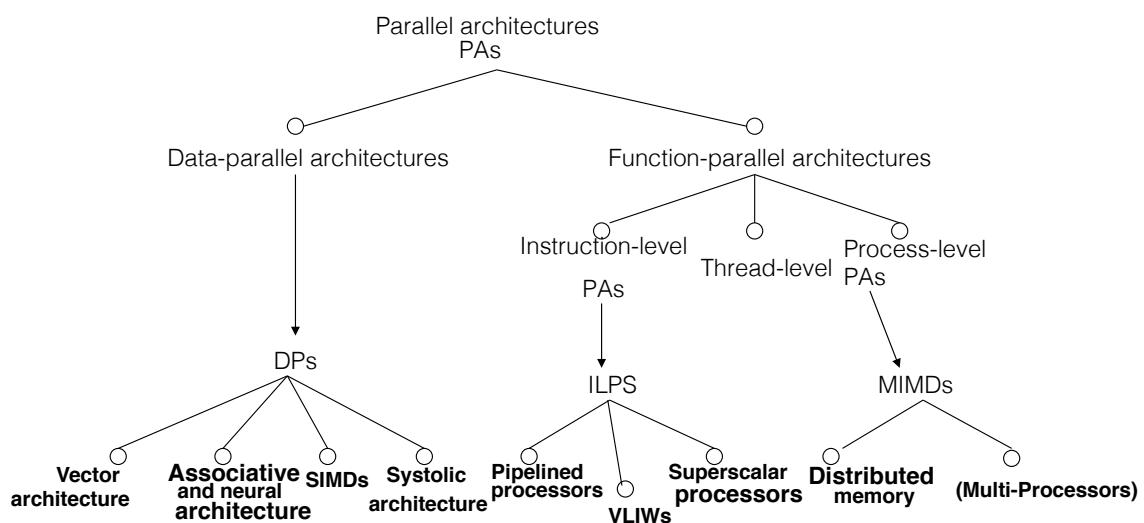
- Static networks



- Dynamic networks



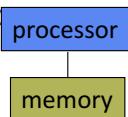
Classification based on the kind of parallelism



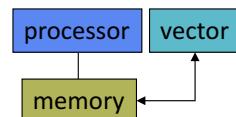
Parallel Architecture Types

Uniprocessor

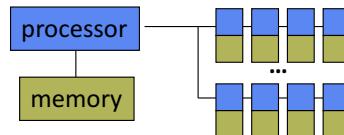
- Scalar processor



- Vector processor

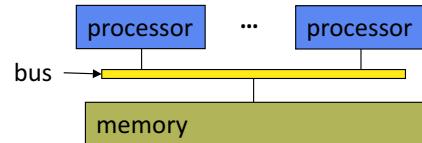


- Single Instruction Multiple Data (SIMD)

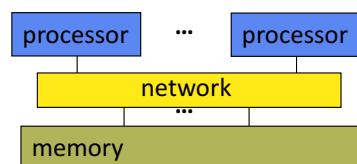


Symmetric MultiProcessing (SMP)

- Shared memory address space
- Bus-based memory system



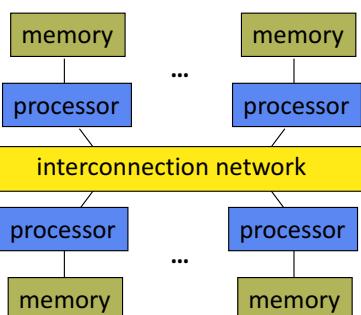
- Interconnection network



Parallel Architecture Types (2)

Distributed Memory Multiprocessor

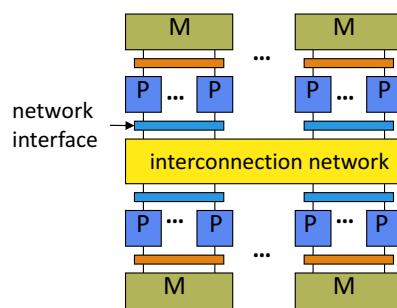
- Message passing between nodes



- Massively Parallel Processor (MPP)
 - Many, many processors

Cluster of SMPs

- Shared memory addressing within SMP node
- Message passing between SMP nodes

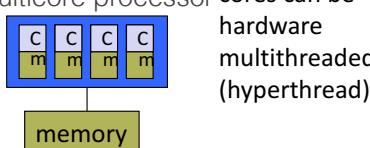


- Can also be regarded as MPP if processor number is large

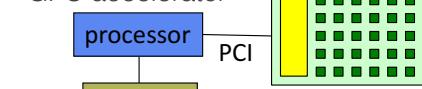
Parallel Architecture Types (3)

Multicore

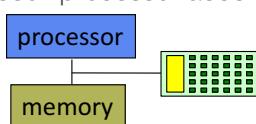
- Multicore processor cores can be hardware multithreaded (hyperthread)



- GPU accelerator

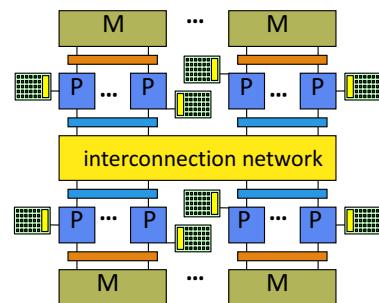


- “Fused” processor accelerator



Multicore SMP+GPU Cluster

- Shared memory addressing within SMP node
- Message passing between SMP nodes
- GPU accelerators attached



How do you get parallelism in hardware?

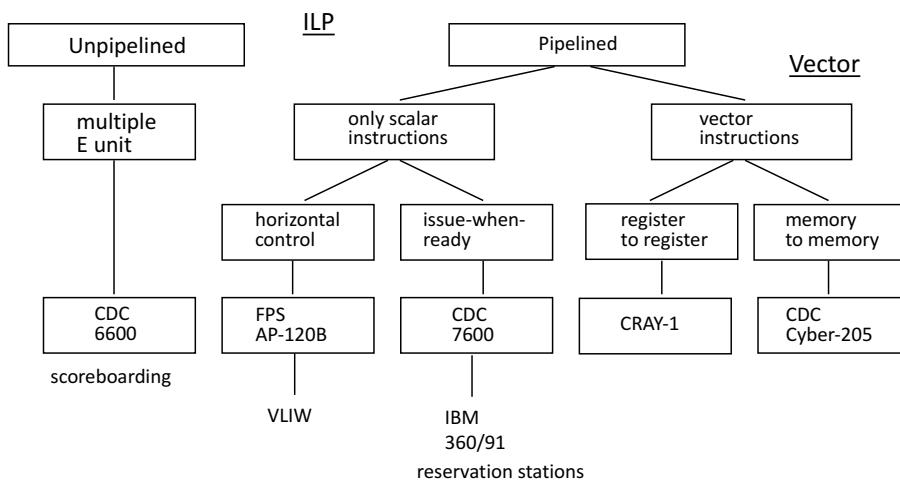
- Instruction-Level Parallelism (ILP)
- Data parallelism
 - Increase amount of data to be operated on at same time
- Processor parallelism
 - Increase number of processors
- Memory system parallelism
 - Increase number of memory units
 - Increase bandwidth to memory
- Communication parallelism
 - Increase amount of interconnection between elements
 - Increase communication bandwidth

Instruction-Level Parallelism

- Opportunities for splitting up instruction processing
- Pipelining within instruction
- Pipelining between instructions
- Overlapped execution
- Multiple functional units
- Out of order execution
- Multi-issue execution
- Superscalar processing
- Superpipelining
- Very Long Instruction Word (VLIW)
- Hardware multithreading (hyperthreading)

Parallelism in Single Processor Computers

History of processor architecture innovation



Vector Processing

- Scalar processing
 - Processor instructions operate on scalar values
 - integer registers and floating point registers
- Vectors
 - Set of scalar data
 - Vector registers
 - integer, floating point (typically)
 - Vector instructions operate on vector registers (SIMD)
- Vector unit pipelining
- Multiple vector units
- Vector chaining

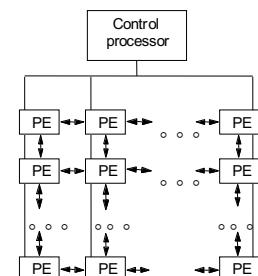
Liquid-cooled with inert fluorocarbon. (That's a waterfall fountain!!!)



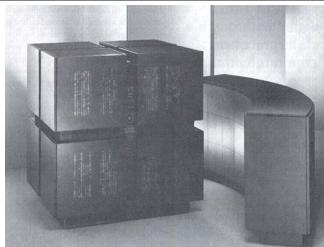
Cray 2

Data Parallel Architectures

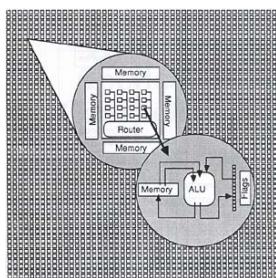
- SIMD (Single Instruction Multiple Data)
 - Logical single thread (instruction) of control
 - Processor associated with data elements
- Architecture
 - Array of simple processors with memory
 - Processors arranged in a regular topology
 - Control processor issues instructions
 - All processors execute same instruction (maybe disabled)
 - Specialized synchronization and communication
 - Specialized reduction operations
 - Array processing



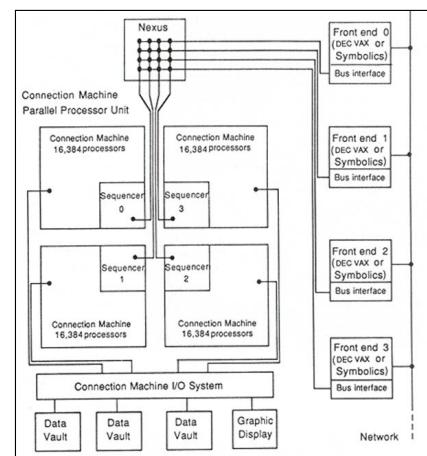
Thinking Machines Connection Machine



16,000 processors!!!

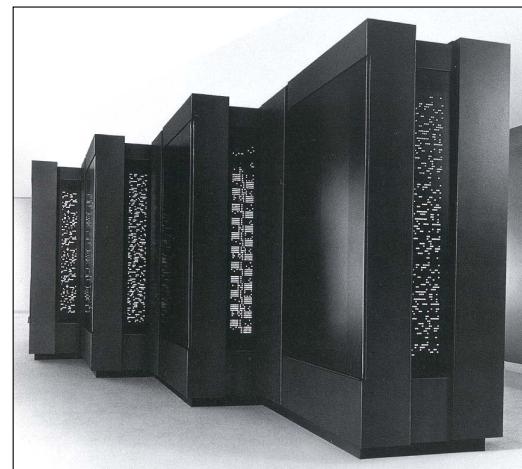


(Tucker, IEEE Computer, Aug. 1988)

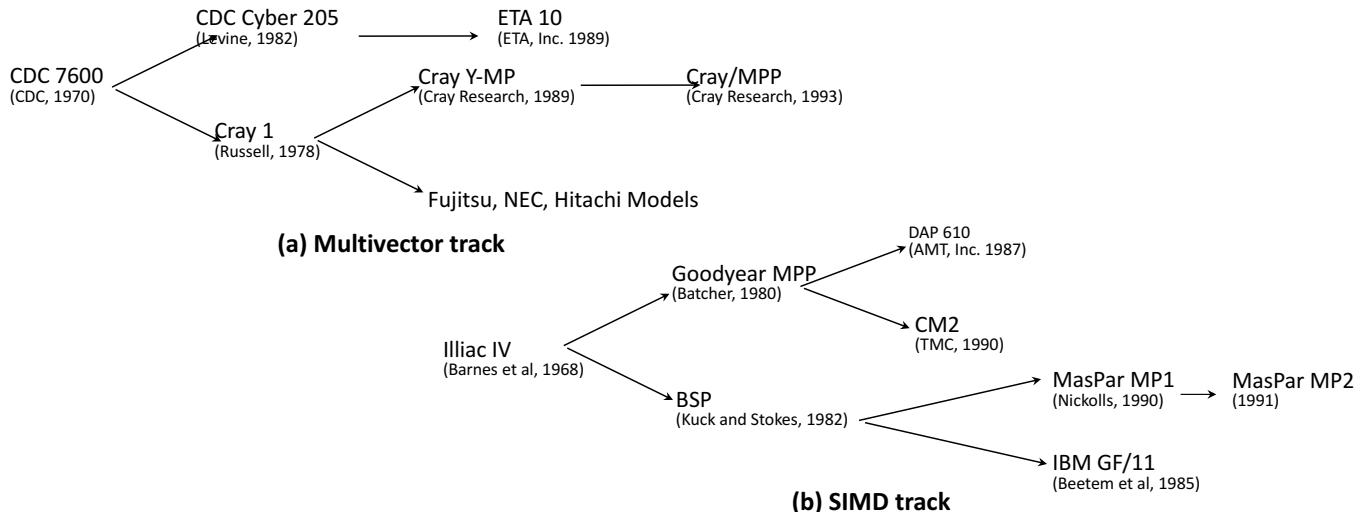


Thinking Machine CM-5

- Repackaged SparcStation
 - 4 per board
- Fat-Tree network
- Control network for global synchronization
- Suffered from hardware design and installation problems

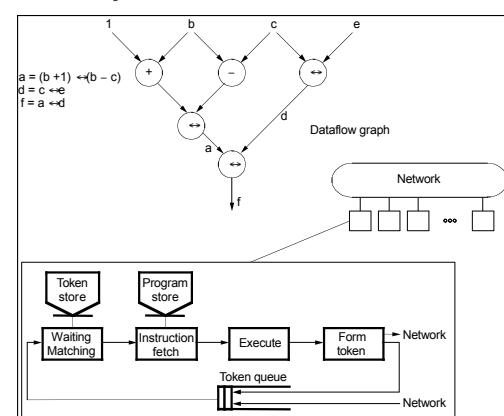


Vector and SIMD Processing Timeline



Dataflow Architectures

- Represent computation as graph of dependencies
- Operations stored in memory until operands are ready
- Operations can be dispatched to processors
- Tokens carry tags of next instruction to processor
- Tag compared in matching store
- A match fires execution
- Machine does the hard parallelization work
- Hard to build correctly

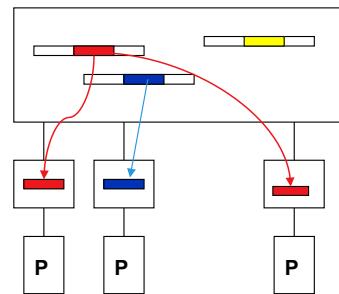


Shared Physical Memory

- Add processors to single processor computer system
- Processors share computer system resources
 - Memory, storage, ...
- Sharing physical memory
 - Any processor can reference any memory location
 - Any I/O controller can reference any memory address
 - Single physical memory address space
- Operating system runs on any processor, or all
 - OS see single memory address space
 - Uses shared memory to coordinate
- Communication occurs as a result of loads and stores

Caching in Shared Memory Systems

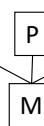
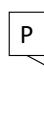
- Reduce average latency
 - automatic replication closer to processor
- Reduce average bandwidth
- Data is logically transferred from producer to consumer to memory
 - store reg → mem
 - load reg ← mem
- Processors can share data efficiently
- What happens when store and load are executed on different processors?
- Cache coherence problems



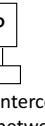
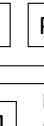
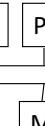
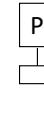
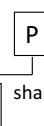
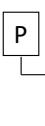
Shared Memory Multiprocessors (SMP)

- Architecture types

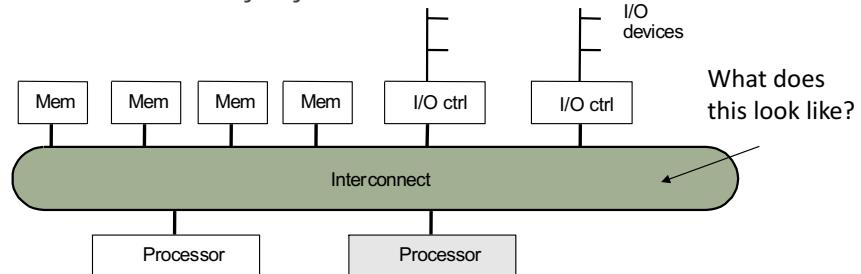
Single processor



Multiple processors

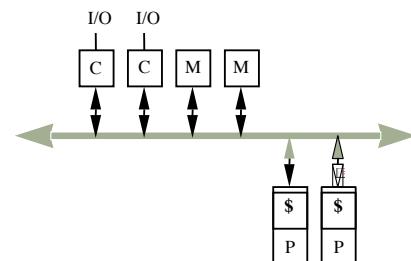


- Differences lie in memory system interconnection



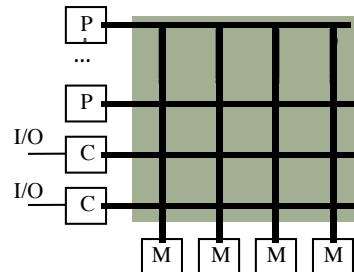
Bus-based SMP

- Memory bus handles all memory read/write traffic
- Processors share bus
- Uniform Memory Access (UMA)
 - Memory (not cache) uniformly equidistant
 - Take same amount of time (generally) to complete
- May have multiple memory modules
 - Interleaving of physical address space
- Caches introduce memory hierarchy
 - Lead to data consistency problems
 - Cache coherency hardware necessary (CC-UMA)



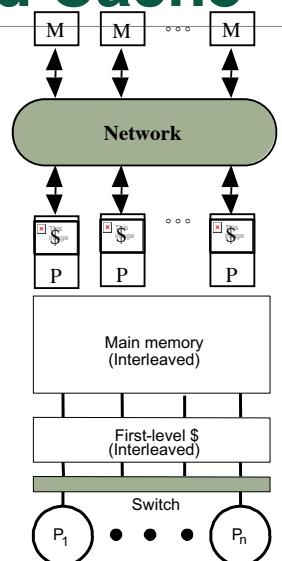
Crossbar SMP

- Replicates memory bus for every processor and I/O controller
 - Every processor has direct path
- UMA SMP architecture
- Can still have cache coherency issues
- Multi-bank memory or interleaved memory
- Advantages
 - Bandwidth scales linearly (no shared links)
- Problems
 - High incremental cost (cannot afford for many processors)
 - Use switched multi-stage interconnection network



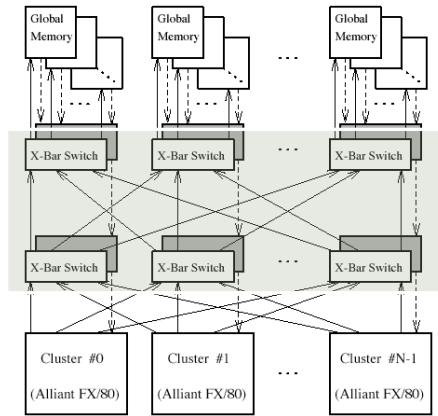
“Dance Hall” SMP and Shared Cache

- Interconnection network connects processors to memory
- Centralized memory (UMA)
- Network determines performance
 - Continuum from bus to crossbar
 - Scalable memory bandwidth
- Memory is physically separated from processors
- Could have cache coherence problems
- Shared cache reduces coherence problem and provides fine grained data sharing

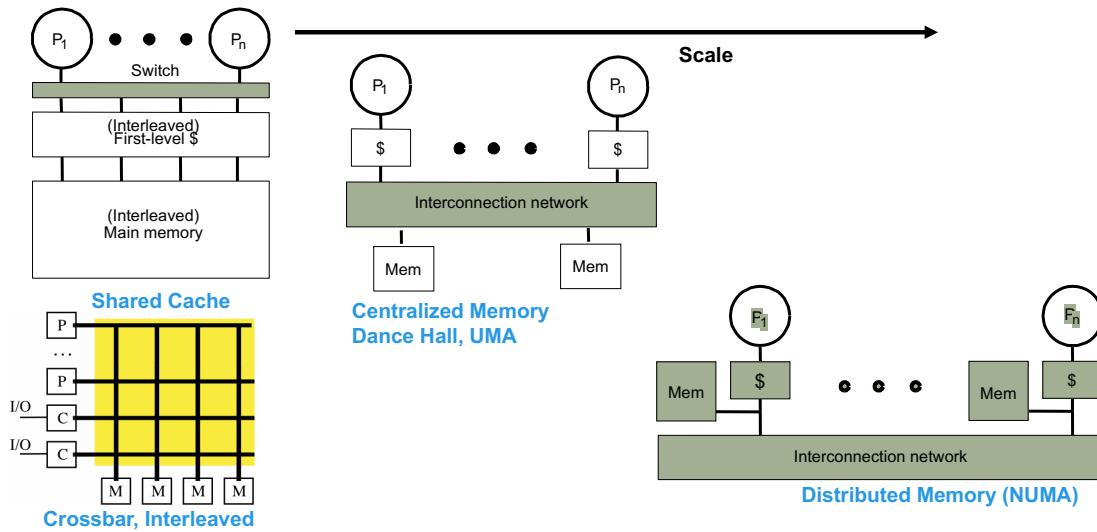


University of Illinois CSRD Cedar Machine

- Center for Supercomputing Research and Development
- Multi-cluster scalable parallel computer
- Alliant FX/80
 - 8 processors w/ vectors
 - Shared cache
 - HW synchronization
- Omega switching network
- Shared global memory
- SW-based global memory coherency

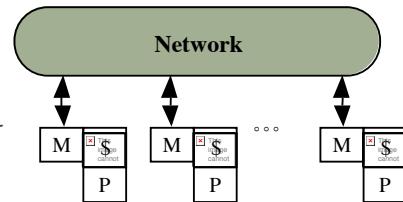


Natural Extensions of the Memory System



Non-Uniform Memory Access (NUMA) SMPs

- Distributed memory
- Memory is physically resident close to each processor
- Memory is still shared
- Non-Uniform Memory Access (NUMA)
 - Local memory and remote memory
 - Access to local memory is faster, remote memory slower
 - Access is non-uniform
 - Performance will depend on data locality
- Cache coherency is still an issue (more serious)
- Interconnection network architecture is more scalable



Cache Coherency and SMPs

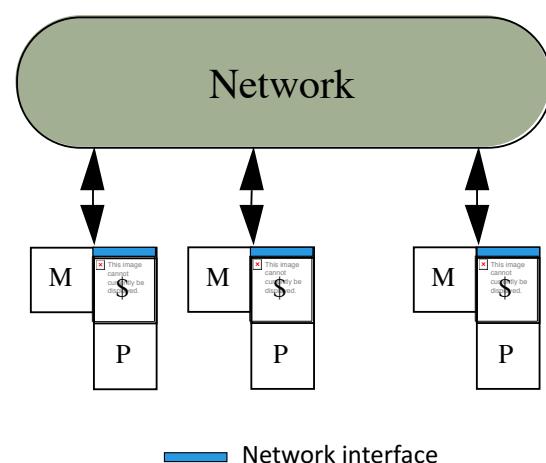
- Caches play key role in SMP performance
 - Reduce average data access time
 - Reduce bandwidth demands placed on shared interconnect
- Private processor caches create a problem
 - Copies of a variable can be present in multiple caches
 - A write by one processor may not become visible to others
 - o they'll keep accessing stale value in their caches
- ⇒ Cache coherence problem
- What do we do about it?
 - Organize the memory hierarchy to make it go away
 - Detect and take actions to eliminate the problem

Distributed Memory Multiprocessors

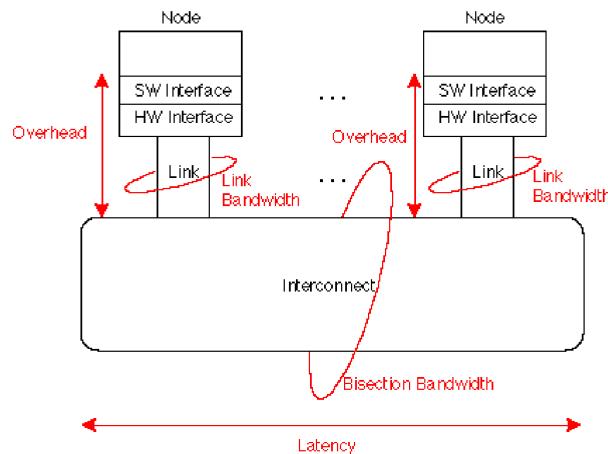
- Each processor has a local memory
 - Physically separated memory address space
- Processors must communicate to access non-local data
 - Message communication (message passing)
 - o Message passing architecture
 - Processor interconnection network
- Parallel applications must be partitioned across
 - Processors: execution units
 - Memory: data partitioning
- Scalable architecture
 - Small incremental cost to add hardware (cost of node)

Distributed Memory (MP) Architecture

- Nodes are complete computer systems
 - Including I/O
- Nodes communicate via interconnection network
 - Standard networks
 - Specialized networks
- Network interfaces
 - Communication integration
- Easier to build



Network Performance Measures



Overhead: latency of interface vs. **Latency:** network

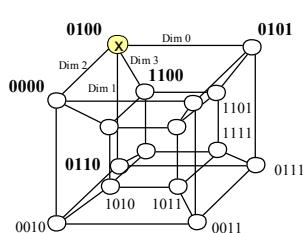
Performance Metrics: Latency and Bandwidth

- Bandwidth
 - Need high bandwidth in communication
 - Match limits in network, memory, and processor
 - Network interface speed vs. network bisection bandwidth
- Latency
 - Performance affected since processor may have to wait
 - Harder to overlap communication and computation
 - Overhead to communicate is a problem in many machines
- Latency hiding
 - Increases programming system burden
 - Examples: communication/computation overlaps, prefetch

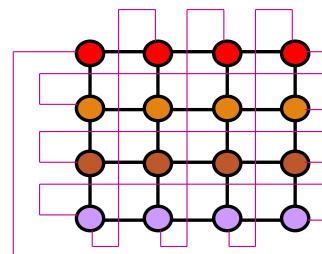
Scalable, High-Performance Interconnect

- Interconnection network is core of parallel architecture
- Requirements and tradeoffs at many levels
 - Elegant mathematical structure
 - Deep relationship to algorithm structure
 - Hardware design sophistication
- Little consensus
 - Performance metrics?
 - Cost metrics?
 - Workload?
 - ...

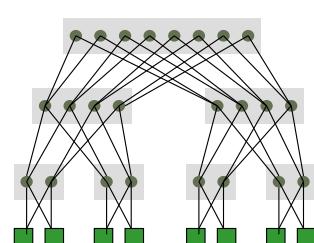
Some Example Interconnection Networks



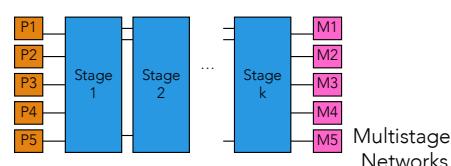
Hypercube



Mesh and Torus



Fat-tree



Multistage Networks

Communication Performance

- $\text{Time}(n)_{s-d} = \text{overhead} + \text{routing delay} + \text{channel occupancy} + \text{contention delay}$
- $\text{occupancy} = (n + n_h) / b$

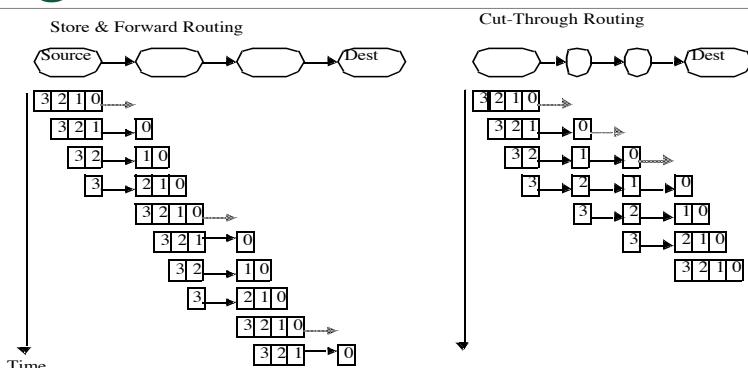
n = message #bytes

n_h = header #bytes

b = bitrate of communication link

- What is the routing delay?
- Does contention occur and what is the cost?

Store-and-Forward vs. Cut-Through Routing



- $h(n/b + \Delta)$
- $n/b + h \Delta$
- What if message is fragmented?
- Wormhole vs. Virtual cut-through

Message Passing Model

- Hardware maintains send and receive message buffers
- Send message (synchronous)
 - Build message in local message send buffer
 - Specify receive location (processor id)
 - Initiate send and wait for receive acknowledge
- Receive message (synchronous)
 - Allocate local message receive buffer
 - Receive message byte stream into buffer
 - Verify message (e.g., checksum) and send acknowledge
- Memory to memory copy with acknowledgement and pairwise synchronization

Advantages of Shared Memory Architectures

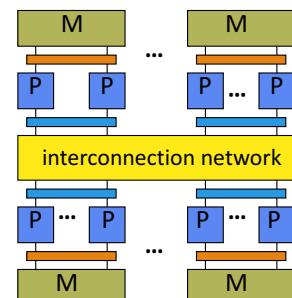
- Compatibility with SMP hardware
- Ease of programming when communication patterns are complex or vary dynamically during execution
- Ability to develop applications using familiar SMP model, attention only on performance critical accesses
- Lower communication overhead, better use of BW for small items, due to implicit communication and memory mapping to implement protection in hardware, rather than through I/O system
- HW-controlled caching to reduce remote communication by caching of all data, both shared and private

Advantages of Distributed Memory Architectures

- The hardware can be simpler (especially versus NUMA) and is more scalable
- Communication is explicit and simpler to understand
- Explicit communication focuses attention on costly aspect of parallel computation
- Synchronization is naturally associated with sending messages, reducing the possibility for errors introduced by incorrect synchronization
- Easier to use sender-initiated communication, which may have some advantages in performance

Clusters of SMPs

- Clustering
 - Integrated packaging of nodes
- Motivation
 - Ammortize node costs by sharing packaging and resources
 - Reduce network costs
 - Reduce communications bandwidth requirements
 - Reduce overall latency
 - More parallelism in a smaller space
 - Increase node performance
- Scalable parallel systems today are built as SMP clusters



Berkeley Network Of Workstations (NOW)

- 100 Sun Ultra2 workstations
- Intelligent network interface
 - proc + mem
- Myrinet network
 - 160 MB/s per link
 - 300 ns per hop



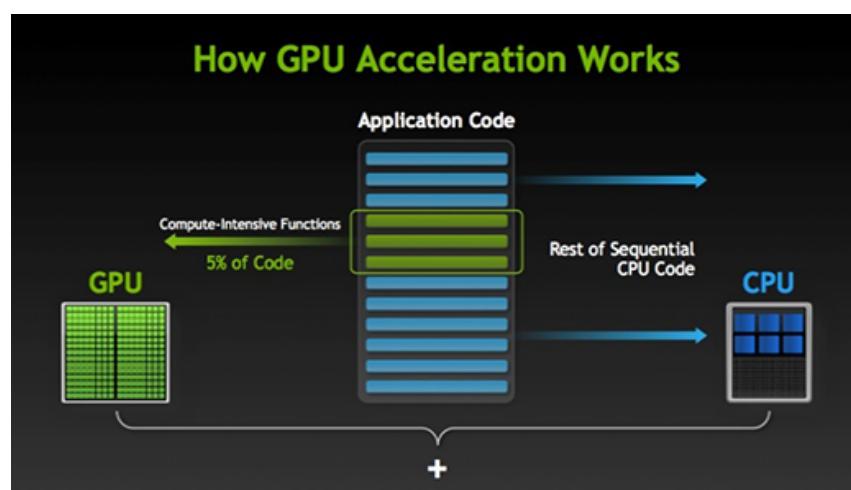
Parallelization Using Accelerators

- GPUs Accelerators
- Specialized Accelerators such as the Xeon Phi
- Applications
 - Deep learning, analytics, and engineering

GPU Accelerators

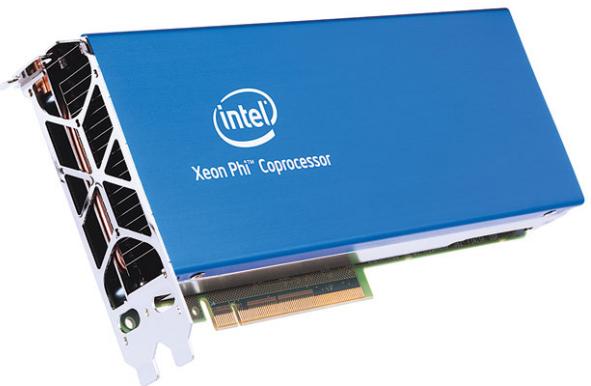
- Use of graphics processing unit (GPU) together with a CPU to accelerate applications
- Pioneered in 2007 by NVIDIA
- GPU accelerators now power energy-efficient data centers in government labs, universities, enterprises, and small-and-medium businesses around the world.

GPU Accelerators



Intel's Xeon Phi Accelerator

- Up to 61 cores, 244 threads, and 1.2 teraflops performance
- Intel says the accelerator has a few advantages over GPGPUs
 - They can operate independently of CPUs and they don't require special code to program.



Examples...

BlueGene/L

- A 64x32x32 torus = 65K 2-core processors
- Cut-through routing gives a worst-case latency of 6.4 μ s
- Processor nodes are dual PPC-440 with “double hummer” FPUs
- Collective network performs global reduce for the “usual” functions



BlueGene/P



BlueGene/Q

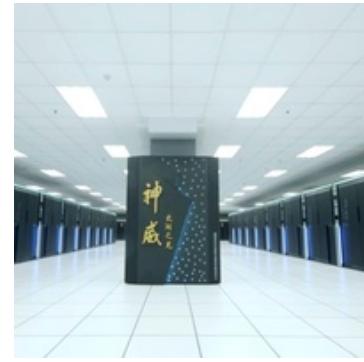


Top 500

- **Sunway TaihuLight: National Supercomputing Center in Wuxi**
– No.1 in Jun 2016
- **Tianhe-2 (MilkyWay-2): National University of Defense Technology**
– No.1 from Jun 2013 until Nov 2015
- **Titan: Oak Ridge National Laboratory**
– No.1 in Nov 2012
- **Sequoia: Lawrence Livermore National Laboratory**
– No.1 in Jun 2012
- **K Computer: RIKEN Advanced Institute for Computational Science**
– No.1 from Jun 2011 until Nov 2011
- **Tianhe-1A: National Supercomputing Center in Tianjin**
– No.1 in Nov 2010

Sunway TaihuLight

- Fastest supercomputer in the world
- LINPACK benchmark rating of 93 petaflops
 - Three times as fast as the previous holder of the record, the Tianhe-2, which ran at 34 petaflops
- 10,649,600 CPU cores across the entire system
 - 40,960 Chinese-designed SW26010 manycore 64-bit RISC processors based on the Sunway architecture
 - o Each processor chip contains 256 processing cores, and an additional four auxiliary cores for system management
- Processing cores feature 64 KB of scratchpad memory for data (and 16 KB[5] for instructions) and communicate via a network on a chip, instead of having a traditional cache hierarchy

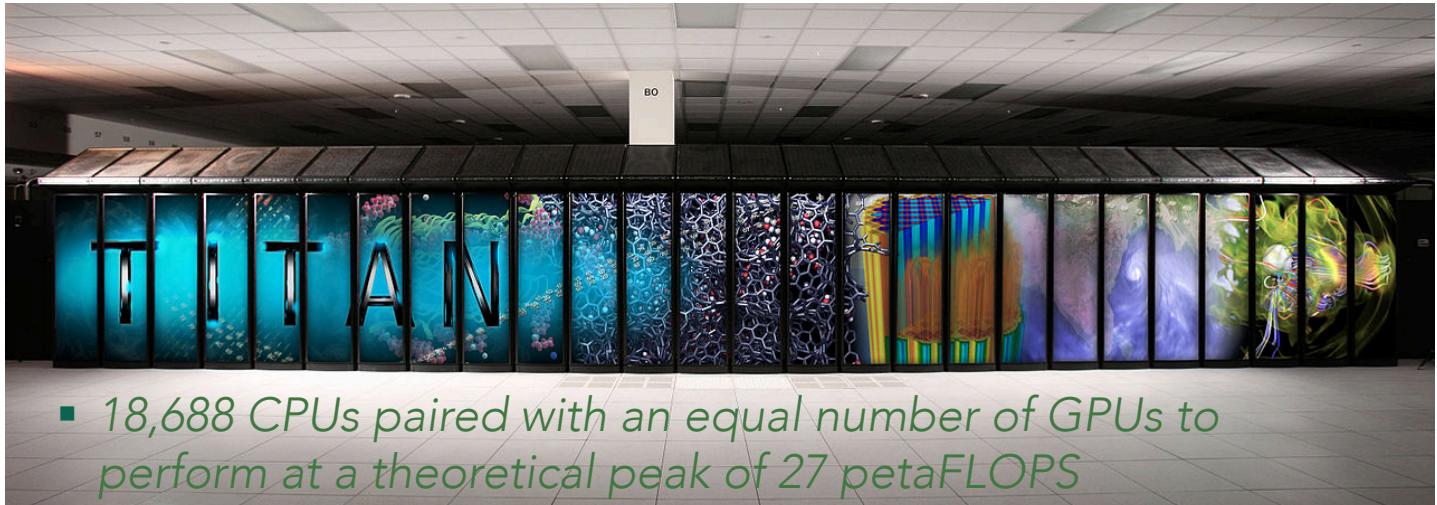


Tianhe-2 Super Computer

- 33.86 petaflops
- 33.86 thousand trillion floating point operations per second
- 16,000 nodes that each contain two *Intel Xeon IvyBridge* processors and three *Xeon Phi* processors, adding up to a total of 3.12 million computing cores.
- China's unprecedented level of investment in supercomputing is resulting in huge numbers of software engineers coming out of China



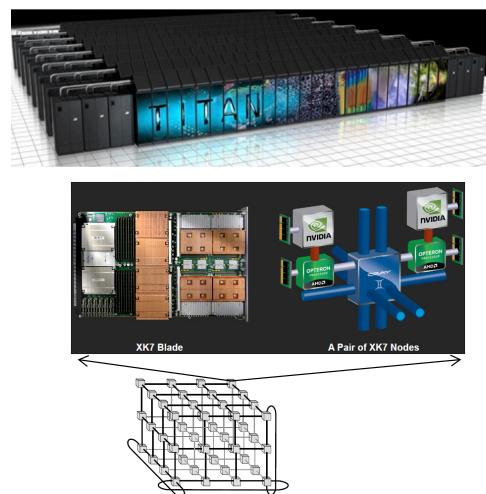
Cray Titan Supercomputer



- 18,688 CPUs paired with an equal number of GPUs to perform at a theoretical peak of 27 petaFLOPS

ORNL Titan (<http://www.olcf.ornl.gov/titan>)

- Cray XK7
 - 18,688 nodes
 - AMD Opteron
 - o 16-core Interlagos
 - o 299,008 Opteron cores
 - NVIDIA K20x
 - o 18,688 GPUs
 - o 50,233,344 GPU cores
- Gemini interconnect
 - 3D torus
- 20+ petaflops



Tianhe-1A GPU Super Computer

- 2.5 petaflops, a number that places it in the number one slot in the list of the world's top 500 supercomputers in 2010
 - <http://www.top500.org/lists/2014/11/>



Tianhe-1A GPU Super Computer

- Designed at the National University of Defense Technology in China
 - 7,000 graphics processors
 - 14,000 Intel chips
 - 20,000 clustered computers
 - Runs on Linux operating system
 - Covers 17,000 square feet
 - Consumes 4.04 megawatts of power
- Performance record at 2.507 petaflops
 - Two-and-a-half thousand trillion operations per second
 - 40% faster than the Cray XT5 Jaguar's speed of 1.75 petaflops.
 - 29 million times more powerful than the earliest supercomputers of the 1970s