SCHOOL OF
HUMAN SCIENCES
& TECHNOLOGY

OPERATING SYSTEMS & PARALLEL COMPUTING

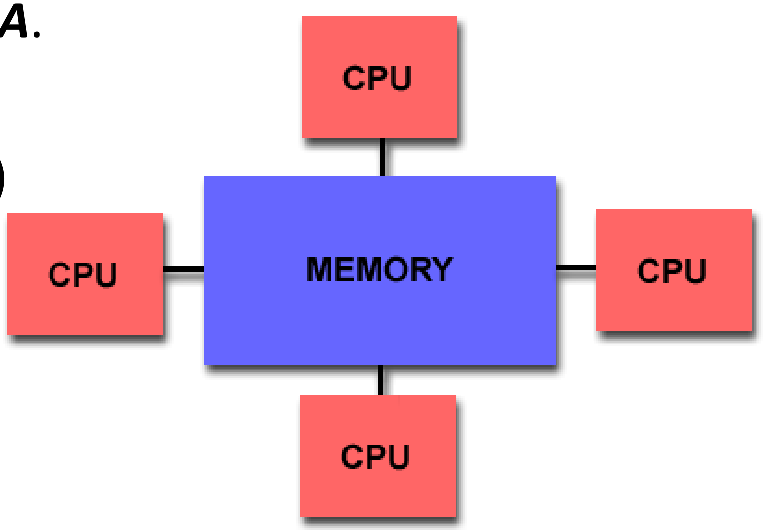
Parallel Computer Architecture

Parallel Computer Memory Architectures

- Shared Memory.
- Distributed Memory.
- Hybrid Distributed-Shared Memory.

Shared Memory

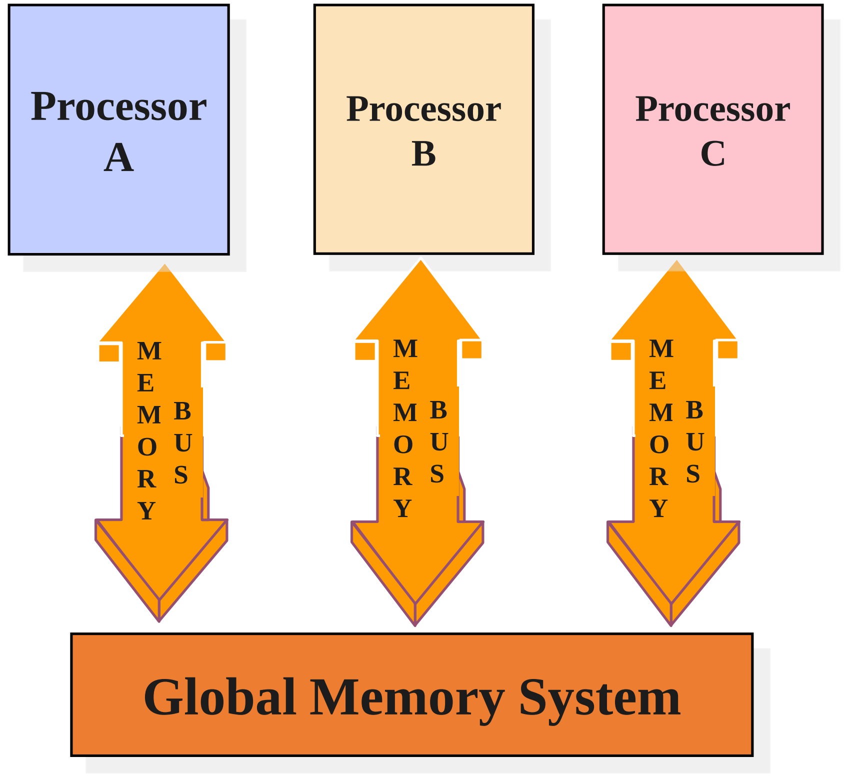
- Shared memory parallel computers vary widely, but generally have in common the ability for all processors to access all memory as global address space.
- Multiple processors can operate independently but share the same memory resources.
- Changes in a memory location effected by one processor are visible to all other processors.
- Shared memory machines can be divided into two main classes based upon memory access times: ***UMA*** and ***NUMA***.
  - Uniform Memory Access (UMA):
  - Non-Uniform Memory Access (NUMA)



Shared Memory : UMA vs. NUMA

- Uniform Memory Access (UMA):
  - Most commonly represented today by Symmetric Multiprocessor (SMP) machines
  - Identical processors
  - Equal access and access times to memory
  - Sometimes called CC-UMA - Cache Coherent UMA. Cache coherent means if one processor updates a location in shared memory, all the other processors know about the update. Cache coherency is accomplished at the hardware level.
- Non-Uniform Memory Access (NUMA):
  - Often made by physically linking two or more SMPs
  - One SMP can directly access memory of another SMP
  - Not all processors have equal access time to all memories
  - Memory access across link is slower
  - If cache coherency is maintained, then may also be called CC-NUMA - Cache Coherent NUMA

Shared Memory MIMD machine



Comm: Source PE writes data to GM & destination retrieves it

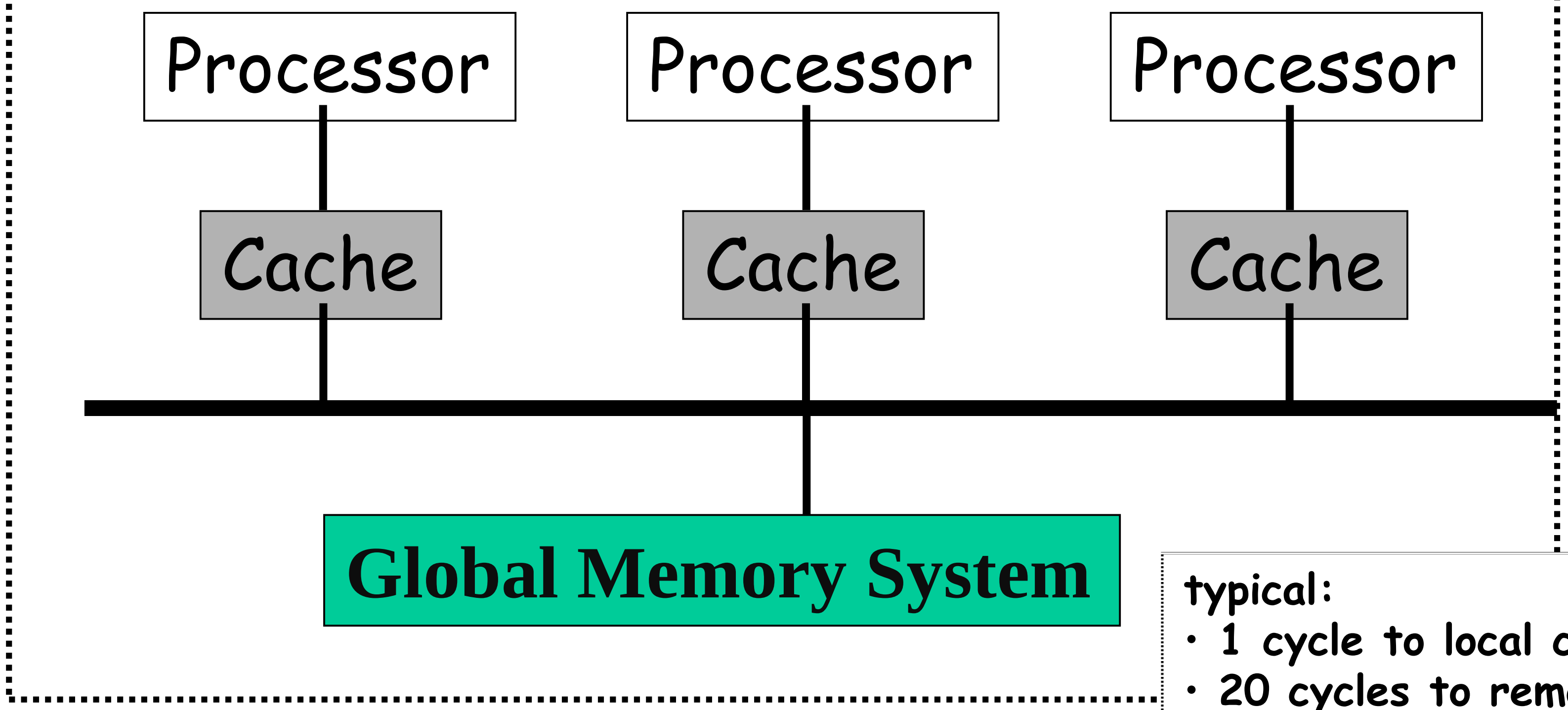
➔Easy to build, conventional OSes of SISD can be easily be ported

➔Limitation : reliability & expandibility. A memory component or any processor failure affects the whole system.

➔Increase of processors leads to memory contention.

Ex. : Silicon graphics supercomputers....

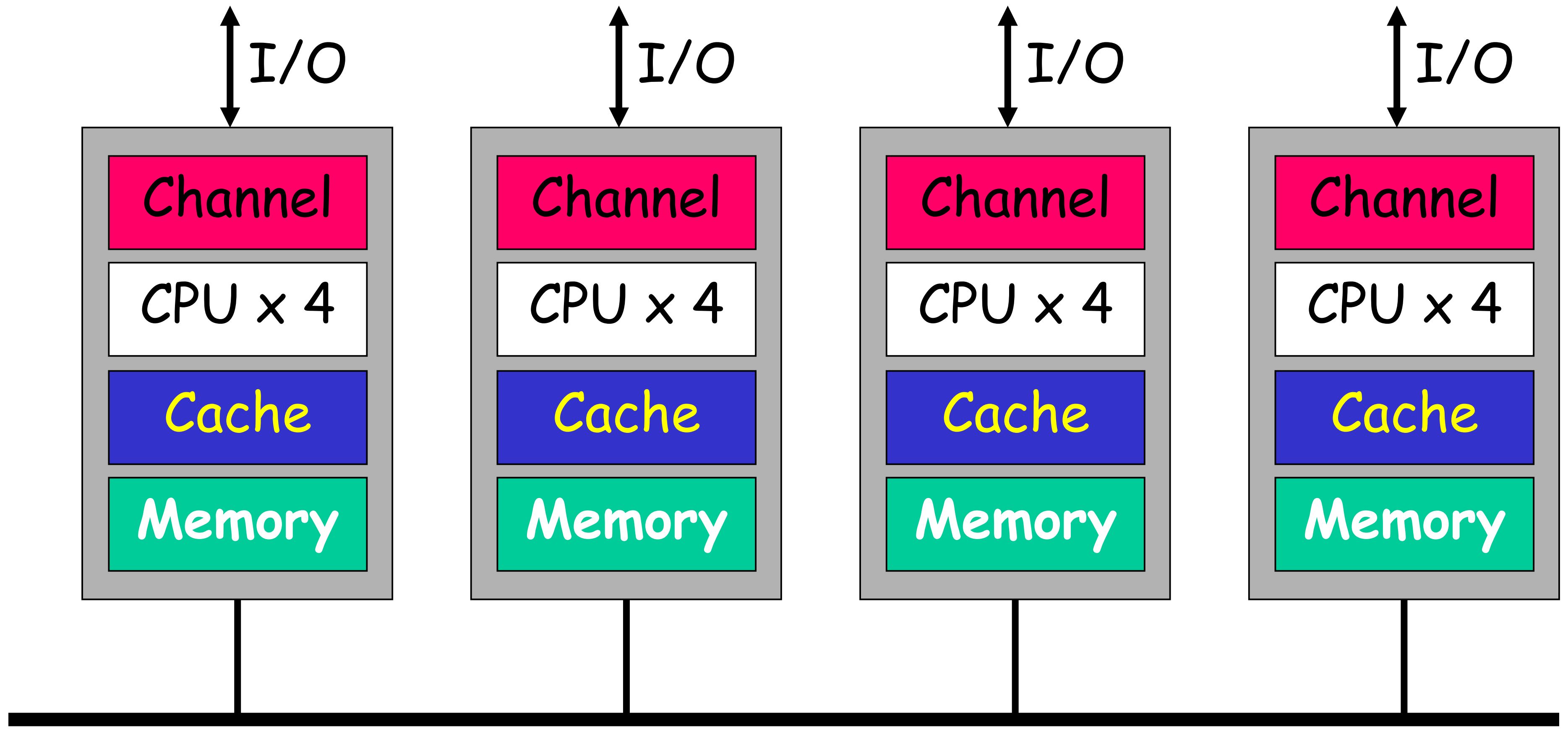
Uniform Memory Access - UMA



Symmetric Multiprocessor SMP

typical:
- 1 cycle to local cache
- 20 cycles to remote cache
- 100 cycles to memory

- **Memory: centralized with uniform access time ("UMA") and bus interconnect, I/O**
- **Examples: Dell Workstation 530, Sun Enterprise, SGI Challenge**



Non-Uniform Memory Access - NUMA

- **Examples: DASH/Alewife/FLASH (academic), SGI Origin, Compaq GS320, Sequent (IBM) NUMA-Q**

SCHOOL OF HUMAN SCIENCES & TECHNOLOGY

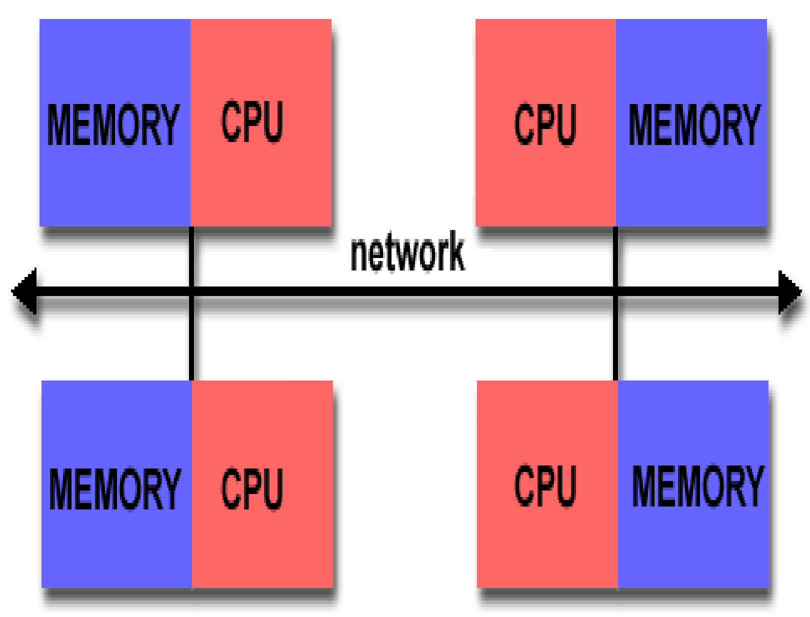
Shared Memory: Pros and Cons

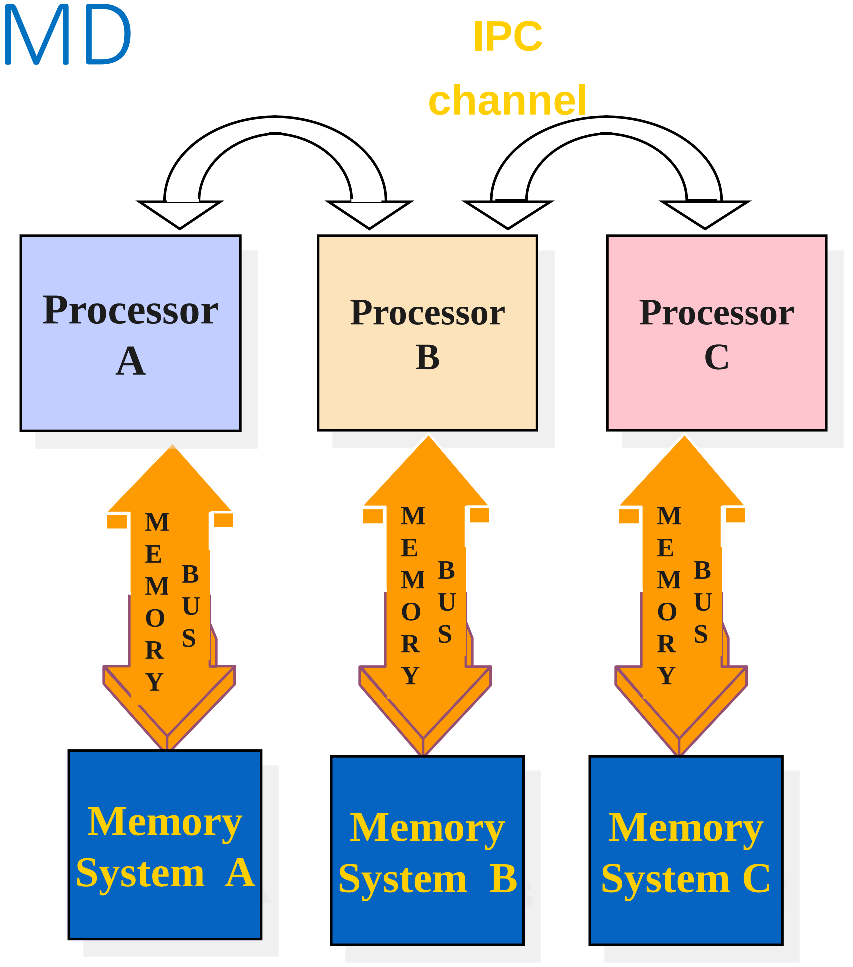
- Advantages
  - Global address space provides a user-friendly programming perspective to memory
  - Data sharing between tasks is both fast and uniform due to the proximity of memory to CPUs
- Disadvantages:
  - Primary disadvantage is the lack of scalability between memory and CPUs. Adding more CPUs can geometrically increases traffic on the shared memory-CPU path, and for cache coherent systems, geometrically increase traffic associated with cache/memory management.
  - Programmer responsibility for synchronization constructs that insure "correct" access of global memory.
  - Expense: it becomes increasingly difficult and expensive to design and produce shared memory machines with ever increasing numbers of processors.

Distributed Memory

- Like shared memory systems, distributed memory systems vary widely but share a common characteristic. Distributed memory systems require a communication network to connect inter-processor memory.
- Processors have their own local memory. Memory addresses in one processor do not map to another processor, so there is no concept of global address space across all processors.
- Because each processor has its own local memory, it operates independently. Changes it makes to its local memory have no effect on the memory of other processors. Hence, the concept of cache coherency does not apply.
- When a processor needs access to data in another processor, it is usually the task of the programmer to explicitly define how and when data is communicated. Synchronization between tasks is likewise the programmer's responsibility.
- The network "fabric" used for data transfer varies widely, though it can can be as simple as Ethernet.

Distributed Memory MIMD





- Communication : IPC on High Speed Network.
- Network can be configured to ... Tree, Mesh, Cube, etc.
- Unlike Shared MIMD
  - ➔ easily/ readily expandable
  - ➔ Highly reliable (any CPU failure does not affect the whole system)

Distributed Memory: Pro and Con

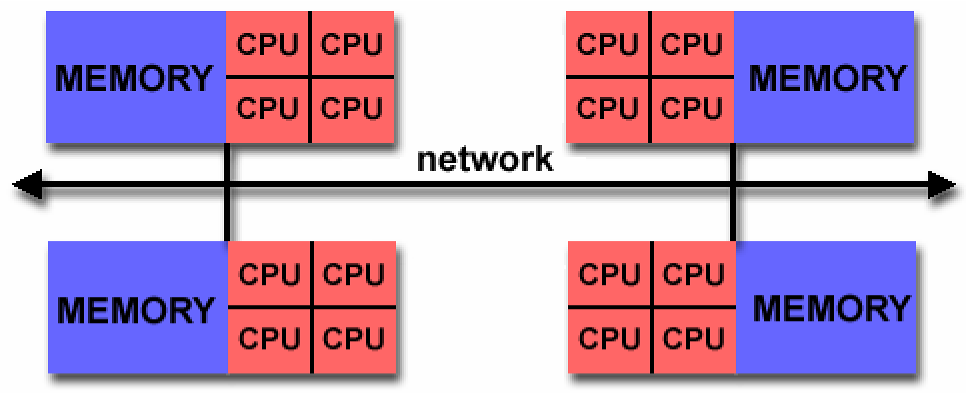
- Advantages:
  - Memory is scalable with number of processors. Increase the number of processors and the size of memory increases proportionately.
  - Each processor can rapidly access its own memory without interference and without the overhead incurred with trying to maintain cache coherency.
  - Cost effectiveness: can use commodity, off-the-shelf processors and networking.
- Disadvantages
  - The programmer is responsible for many of the details associated with data communication between processors.
  - It may be difficult to map existing data structures, based on global memory, to this memory organization.
  - Non-uniform memory access (NUMA) times

- Distributing memory among processing nodes has 2 pluses:
  - It's a great way to save some bandwidth
    - With memory distributed at nodes, most accesses are to local memory within a particular node
    - No need for bus communication
  - Reduces latency for accesses to local memory

- It also has 1 big minus!
  - Have to communicate among various processors
    - Leads to a higher latency for intra-node communication
    - Also need bandwidth to actually handle communication

Hybrid Distributed-Shared Memory

- The largest and fastest computers in the world today employ both shared and distributed memory architectures.
- The shared memory component is usually a cache coherent SMP machine. Processors on a given SMP can address that machine's memory as global.
- The distributed memory component is the networking of multiple SMPs. SMPs know only about their own memory - not the memory on another SMP. Therefore, network communications are required to move data from one SMP to another.
- Current trends seem to indicate that this type of memory architecture will continue to prevail and increase at the high end of computing for the foreseeable future.
- Advantages and Disadvantages: whatever is common to both shared and distributed memory architectures.

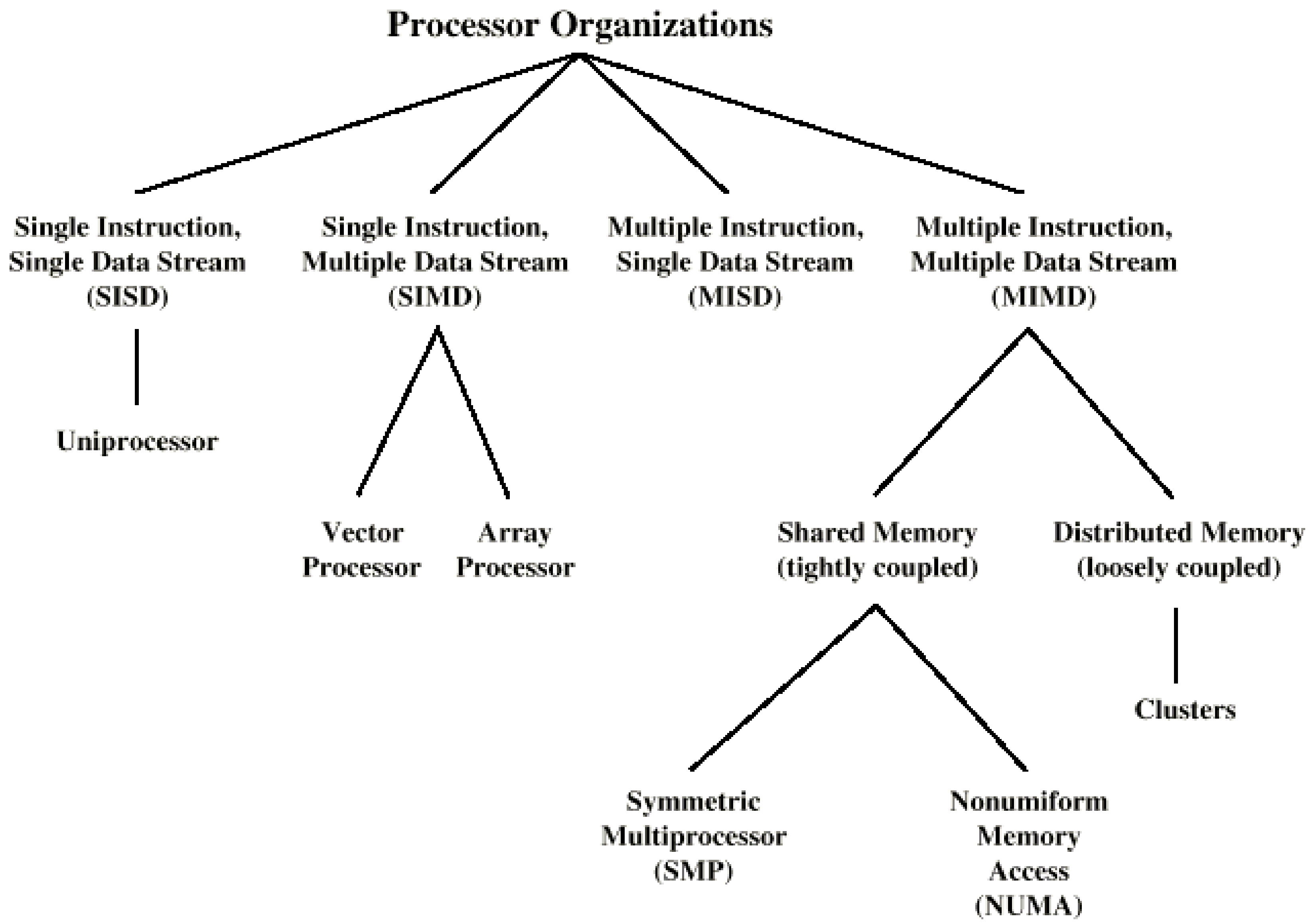


Hybrid Distributed-Shared Memory

Summarizing a few of the key characteristics of shared and distributed memory machines

Comparison of Shared and Distributed Memory Architectures			
Architecture	CC-UMA	CC-NUMA	Distributed
Examples	SMPs Sun Vexx DEC/Compaq SGI Challenge IBM POWER3	Bull NovaScale SGI Origin Sequent HP Exemplar DEC/Compaq IBM POWER4 (MCM)	Cray T3E Maspar IBM SP2 IBM BlueGene
Communications	MPI Threads OpenMP shmem	MPI Threads OpenMP shmem	MPI
Scalability	to 10s of processors	to 100s of processors	to 1000s of processors
Draw Backs	Memory-CPU bandwidth	Memory-CPU bandwidth Non-uniform access times	System administration Programming is hard to develop and maintain
Software Availability	many 1000s ISVs	many 1000s ISVs	100s ISVs

Taxonomy of Parallel Processor Architectures


Processor Organizations
Single Instruction, Single Data Stream (SISD)
Single Instruction, Multiple Data Stream (SIMD)
Multiple Instruction, Single Data Stream (MISD)
Multiple Instruction, Multiple Data Stream (MIMD)
Uniprocessor
Vector Processor
Array Processor
Shared Memory (tightly coupled)
Distributed Memory (loosely coupled)
Clusters
Symmetric Multiprocessor (SMP)
Nonumiform Memory Access (NUMA)

SCHOOL OF
HUMAN SCIENCES
& TECHNOLOGY

More detail on cache coherency protocols with some examples…

More on centralized shared memory

- Its worth studying the various ramifications of a centralized shared memory machine
  - (and there are lots of them)
  - Later we'll look at distributed shared memory…
- When studying memory hierarchies we saw…
  - …cache structures can substantially reduce memory bandwidth demands of a processor
    - Multiple processors may be able to share the same memory

More on centralized shared memory

- Centralized shared memory supports private/shared data
  - If 1 processor in a multiprocessor network operates on private data, caching, etc. are handled just as in uniprocessors
  - But if shared data is cached there can be multiple copies and multiple updates
    - Good b/c it reduces required memory bandwidth; bad because we now must worry about cache coherence

SCHOOL OF HUMAN SCIENCES & TECHNOLOGY

Cache coherence – why it's a problem

Time	Event	Cache contents for CPU A	Cache contents for CPU B	Memory contents for location X
0				1
1	CPU A reads X	1		1
2	CPU B reads X	1	1	1
3	CPU A stores 0 into X	0	1	0

- Assumes that neither cache had value/location X in it 1st
- Both a write-through cache and a write-back cache will encounter this problem
- If B reads the value of X after Time 3, it will get 1 which is the wrong value!

Coherence in shared memory programs

- Must have coherence and consistency
- Memory system coherent if:
  - Program order preserved (always true in uniprocessor)
    - Say we have a read by processor P of location X
    - Before the read processor P wrote something to location X
    - In the interim, no other processor has written to X
    - A read to X should always return the value written by P
  - A coherent view of memory is provided
    - 1st, processor A writes something to memory location X
    - Then, processor B tries to read from memory location X
    - Processor B should get the value written by processor A assuming…
      - **Enough time has past b/t the two events**
      - **No other writes to X have occurred in the interim**

Coherence in shared memory programs (continued)

- Memory system coherent if: (continued)
  - Writes to same location are serialized
    - Two writes to the same location by any two processors are seen in the same order by all processors
    - Ex. Values of A and B are written to memory location X
      - **Processors can't read the value of B and then later as A**
    - If writes not serialized…
      - **One processor might see the write of processor P2 to location X 1st**
      - **Then, it might later see a write to location X by processor P1**
      - **(P1 actually wrote X before P2)**
      - **Value of P1 could be maintained indefinitely even though it was overwritten**

Coherence/consistency

- Coherence and consistency are complementary
  - Coherence defines actions of reads and writes to same memory location
  - Consistency defines actions of reads and writes with regard to accesses of other memory locations

- Assumption for the following discussion:
  - Write does not complete until all processors have seen effect of write
  - Processor does not change order of any write with any other memory accesses
    - Not exactly the case for either one really…but more later…

Caches in coherent multiprocessors

- In multiprocessors, caches at individual nodes help w/ performance
  - Usually by providing properties of “migration” and “replication”
  - Migration:
    - Instead of going to centralized memory for each reference, data word will “migrate” to a cache at a node
    - Reduces latency
  - Replication:
    - If data simultaneously read by two different nodes, copy is made at each node
    - Reduces access latency and contention for shared item

- Supporting these require cache coherence protocols
  - Really, we need to keep track of shared blocks…

Summary

- Shared-memory machine
  - All communication is implicit, through loads and stores
  - Parallelism introduces a bunch of overheads over uniprocessor
- Cache Coherence Problem
  - Local Caches $\Rightarrow$ Copies of data $\Rightarrow$ Potential inconsistencies
- Memory Coherence:
  - Writes to a given location eventually propagated
  - Writes to a given location seen in same order by everyone
- Memory Consistency:
  - Constraints on ordering between processors and locations
- Sequential Consistency:
  - For every parallel execution, there exists a serial interleaving
- Snoopy Bus Protocols
  - Make use of broadcast to ensure coherence
  - Various tradeoffs:
    - Write Through vs Write Back
    - Invalidate vs Update

SCHOOL OF
HUMAN SCIENCES
& TECHNOLOGY