Architecture Classifications (Mostly using Multiprocessor)

(Processor= Control Unit (CU)+Processing Unit/Element(PU/PE+ Memory Shared or Local Memory(LM) or Distributed Memory)

Hwang

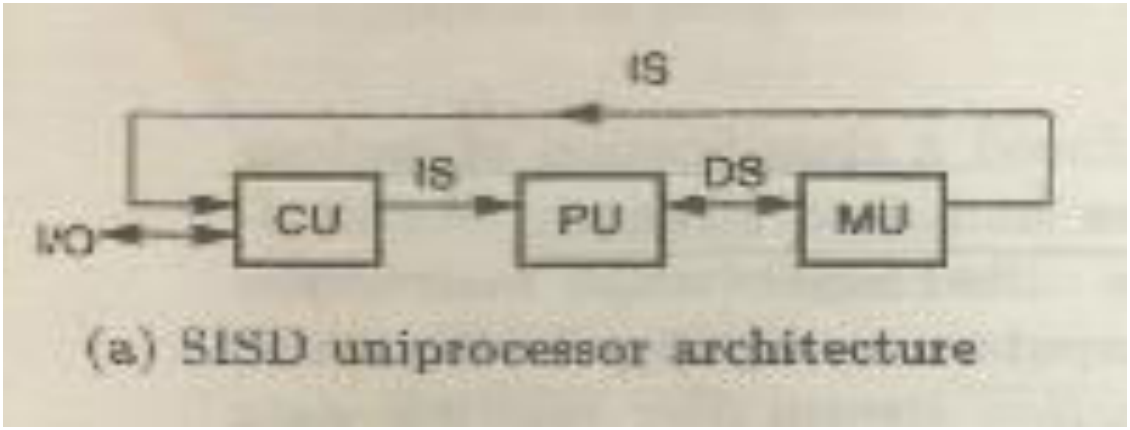
Chapter 1

Flynn's Classification(1972)

Flynn, M. J. (September 1972). "Some Computer Organizations and Their Effectiveness". IEEE Trans. Comput. ***C–21*** *(9): 948–960. doi:10.1109/TC.1972.5009071.*

(Based on instruction and data streams)

- SISD – Single instruction stream over single data stream machines
  - Sequential machines

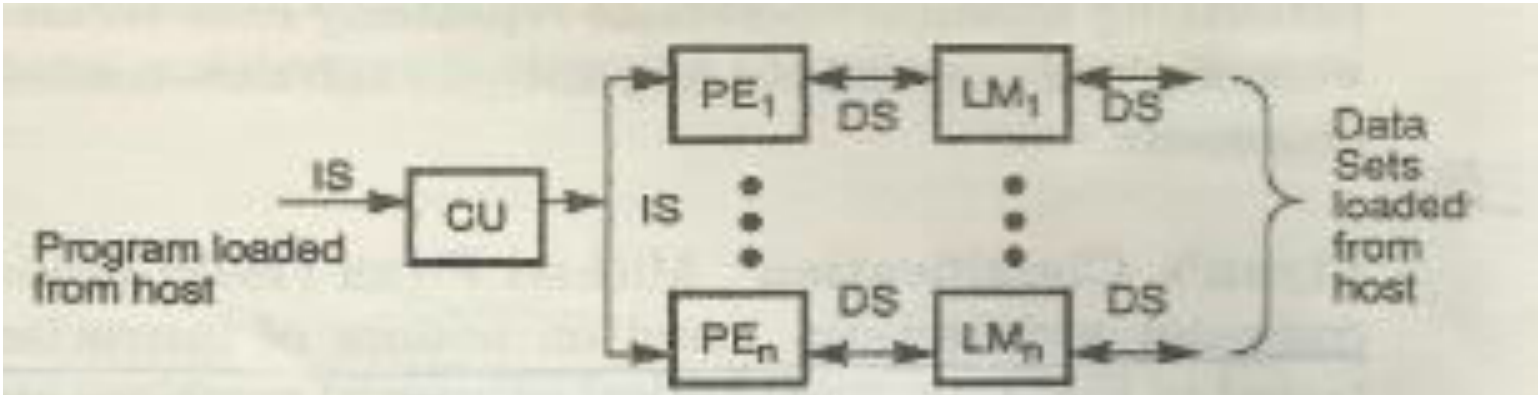


(a) SISD uniprocessor architecture

Flynn's Classification contd..

Many different ways to organize the processors and memory:

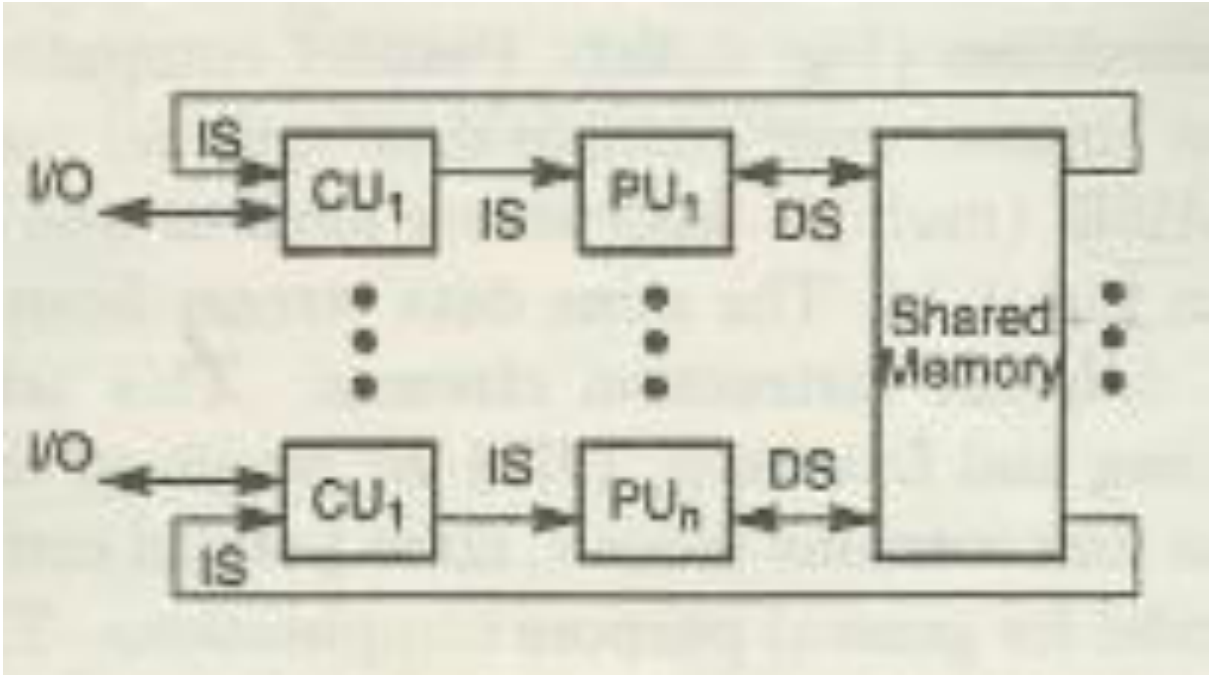
- SIMD- Single instruction stream over multiple data stream machines
  - SIMD are Vector computers equipped with scalar and vector hardware(special purpose)



(b) SIMD architecture (with distributed memory)

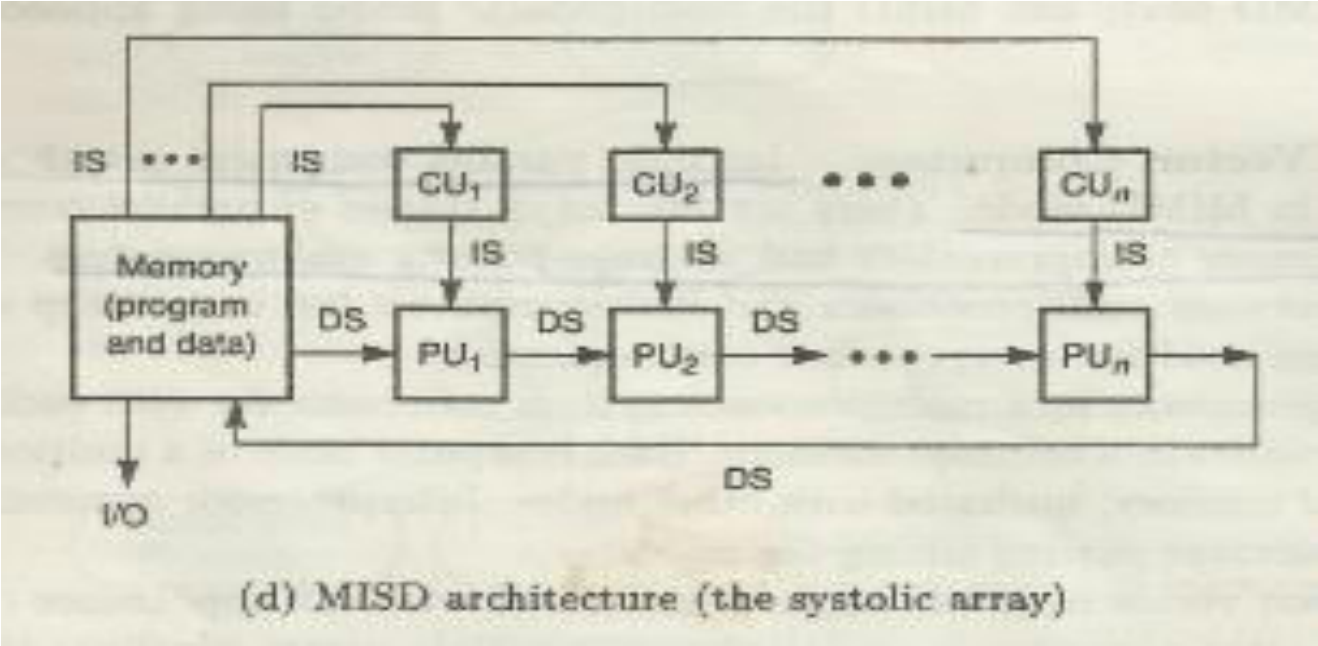
Flynn's Classification contd..

- MIMD- Multiple instruction streams over multiple data streams machines
  - **These Parallel computers have been used for general purpose computations**
  - Processors connected with memory by fast memory bus or switches



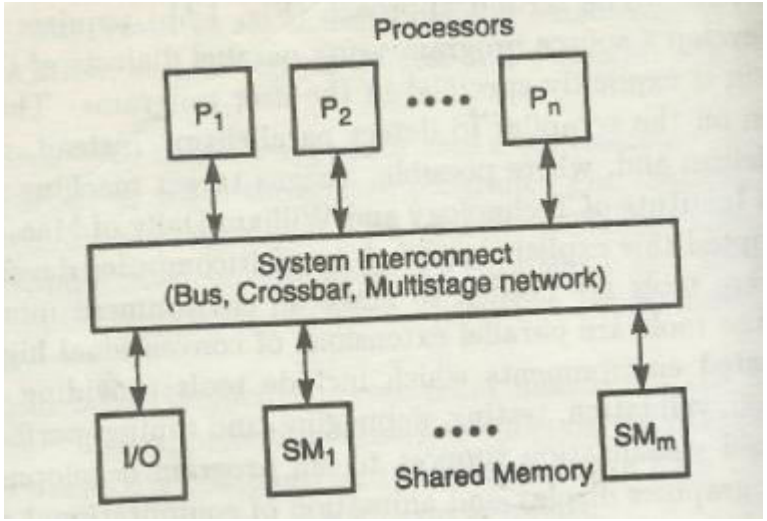
Flynn's Classification contd..

- MISD- Multiple instruction streams over single data streams machines
  - Same data stream goes through a linear array of processors each executing different instruction stream



(d) MISD architecture (the systolic array)

Multiprocessor System- 2 Types



- 1)Shared Memory Multiprocessor called TIGHTLY COUPLED system, they communicate through shared main memory
  - Performance degradation when 2 or more processors try to access same memory locations
  - Can perform with high degree of interaction between tasks, at a higher performance level

Multiprocessor System- 2 Types

- 2) Distributed Memory Multiprocessors (also called Multicomputers)
  - Called LOOSELY COUPLED system, they communicate through message transfer system(MTS)
  - Multiple inexpensive computers are connected
  - Efficient when the interaction between task is low

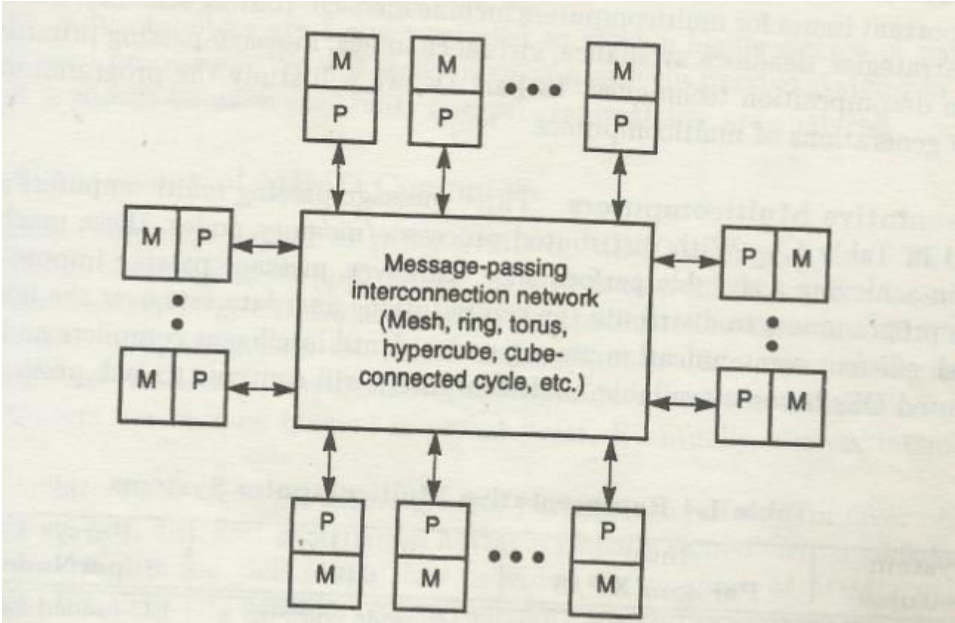


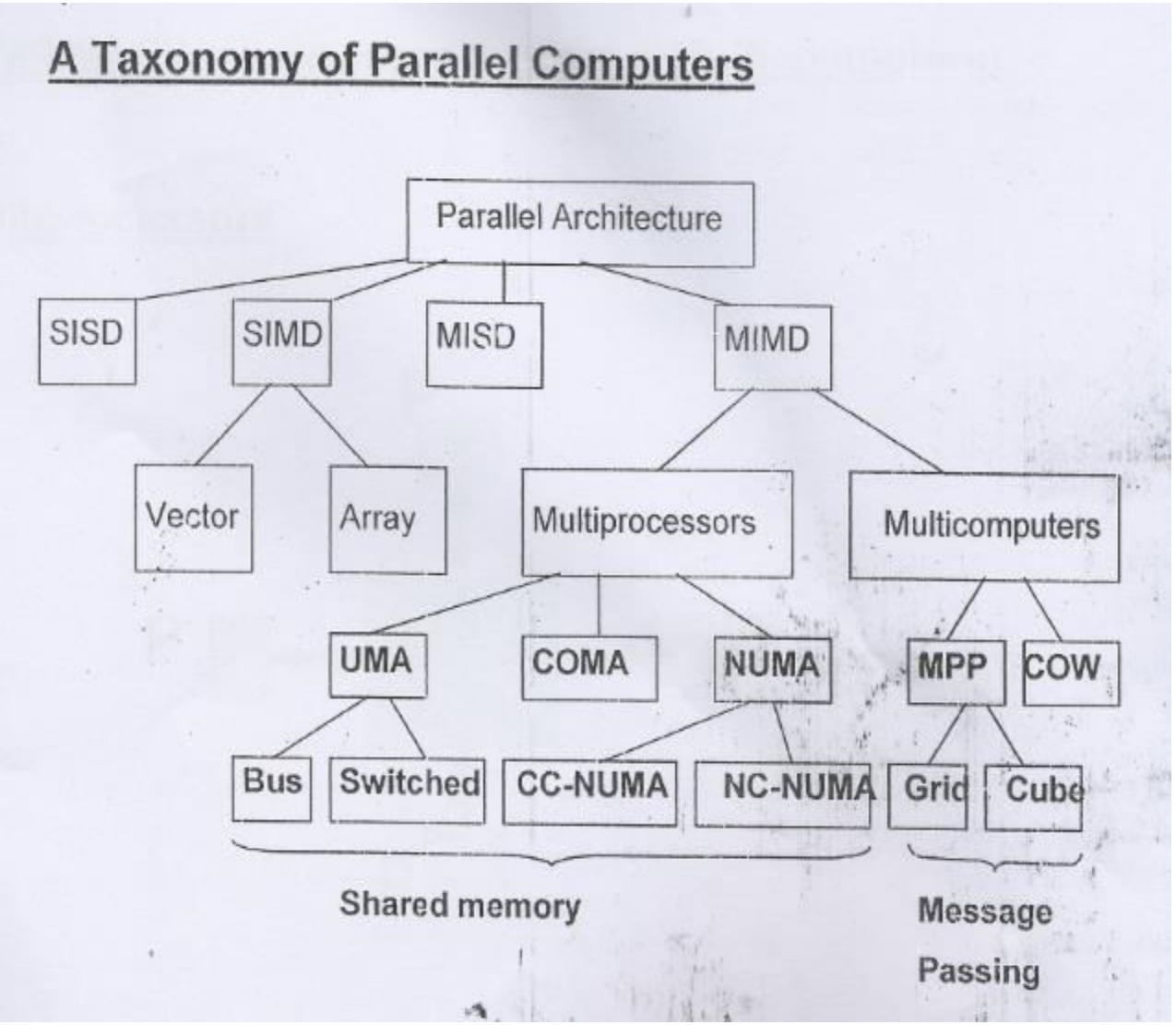
Figure 1.9 Generic model of a message-passing multicomputer.

Loosely Coupled vs Strongly Coupled

- Loosely coupled architecture means changes in one module / section / component hardly affect the other components and each module is somewhat independent of each other. Technologically independent , Build independent and may be even release independent. Loosely coupled architecture is robust (as problem doesn't propagate to related application) and easy to maintain and scale. Example are - SOA (Service Oriented Architecture), MVC (Model View Controller).

  On the other hand, tightly coupled architecture promotes inter dependent applications and code. Tighly coupled architecture is fragile as minor issue in one segment can bring the whole architecture down.

Classification at a Glance



UMA	Uniform Memory Access
NUMA	Non Uniform Memory Access
COMA	Cache Only Memory Access
MPP	Massively Parallel Processor
COW	Cluster Of Workstations
CC-NUMA	Cache Coherent NUMA
NC-NUMA	No Cache NUMA

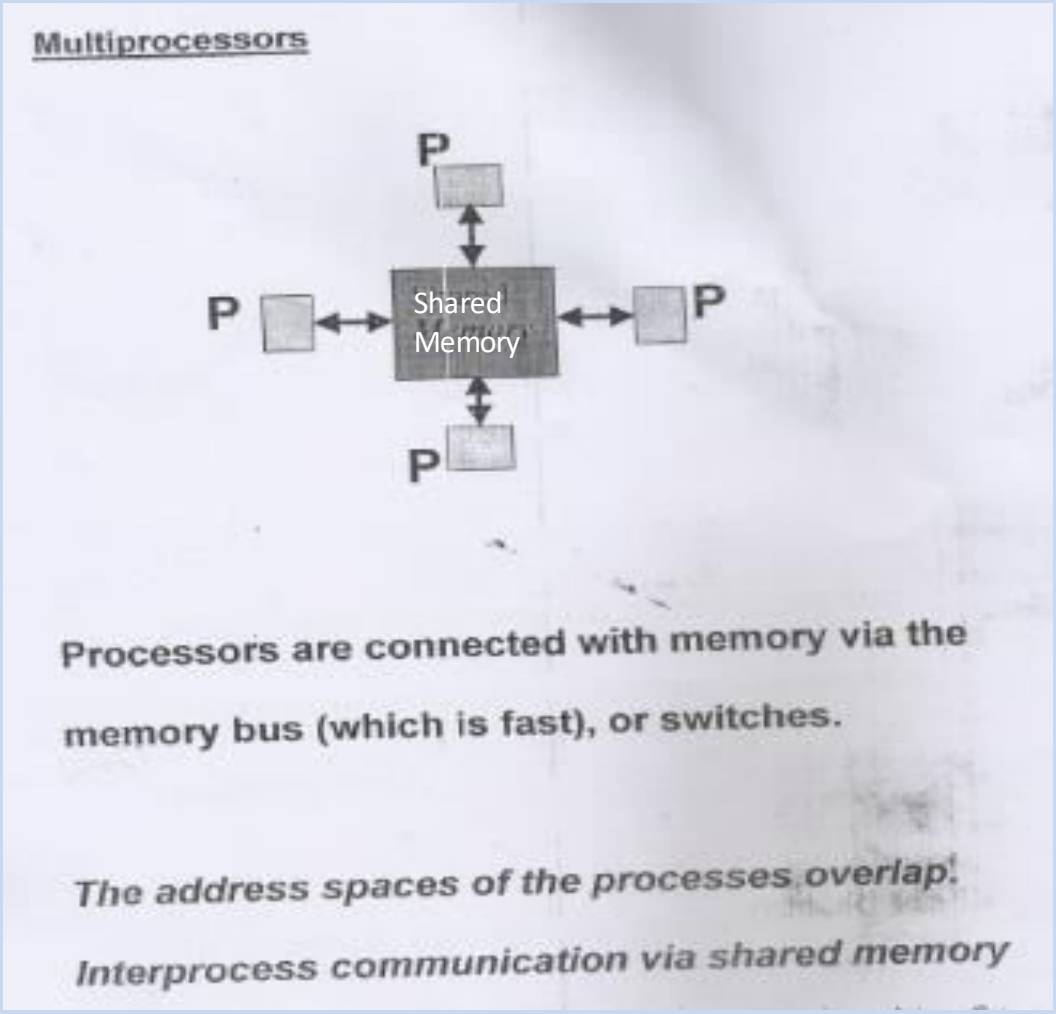
Salient Features of Each Type of MIMD-Memory architectures

Shared Memory Microprocessors

Shared Memory
General Characteristics

- Shared Memory Multiprocessor called TIGHTLY COUPLED system, they communicate through shared main memory
  - Performance degradation when 2 or more processors try to access same memory locations
  - Can perform with high degree of interaction between tasks, at a higher performance level

MIMD-Shared Memory Microprocessors-UMA(Uniform Memory Access)

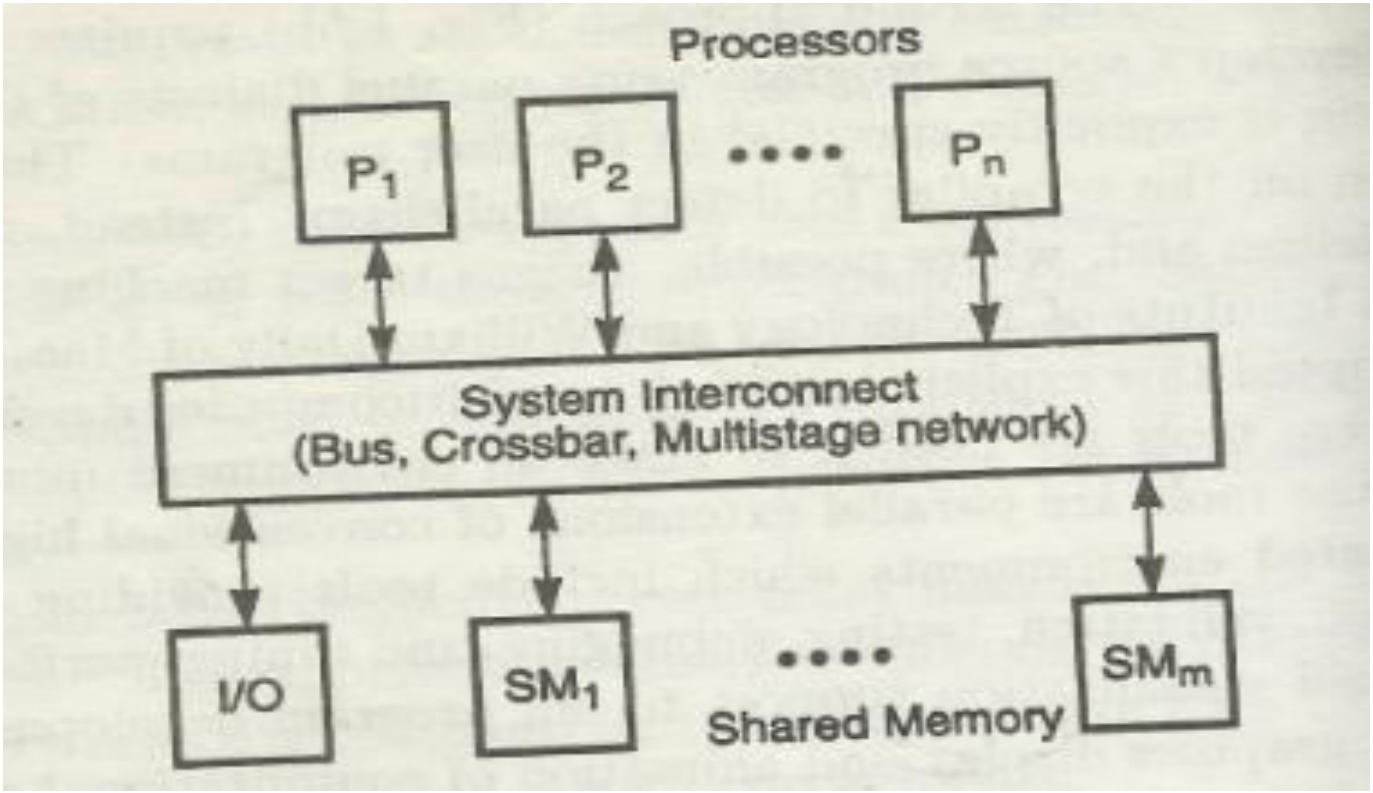


Two or more CPUs within a single computer system. The term also refers to the ability of a system to support more than one processor and/or the ability to allocate tasks between them

MIMD- Shared Memory Multiprocessors- UMA(Uniform Memory Access)

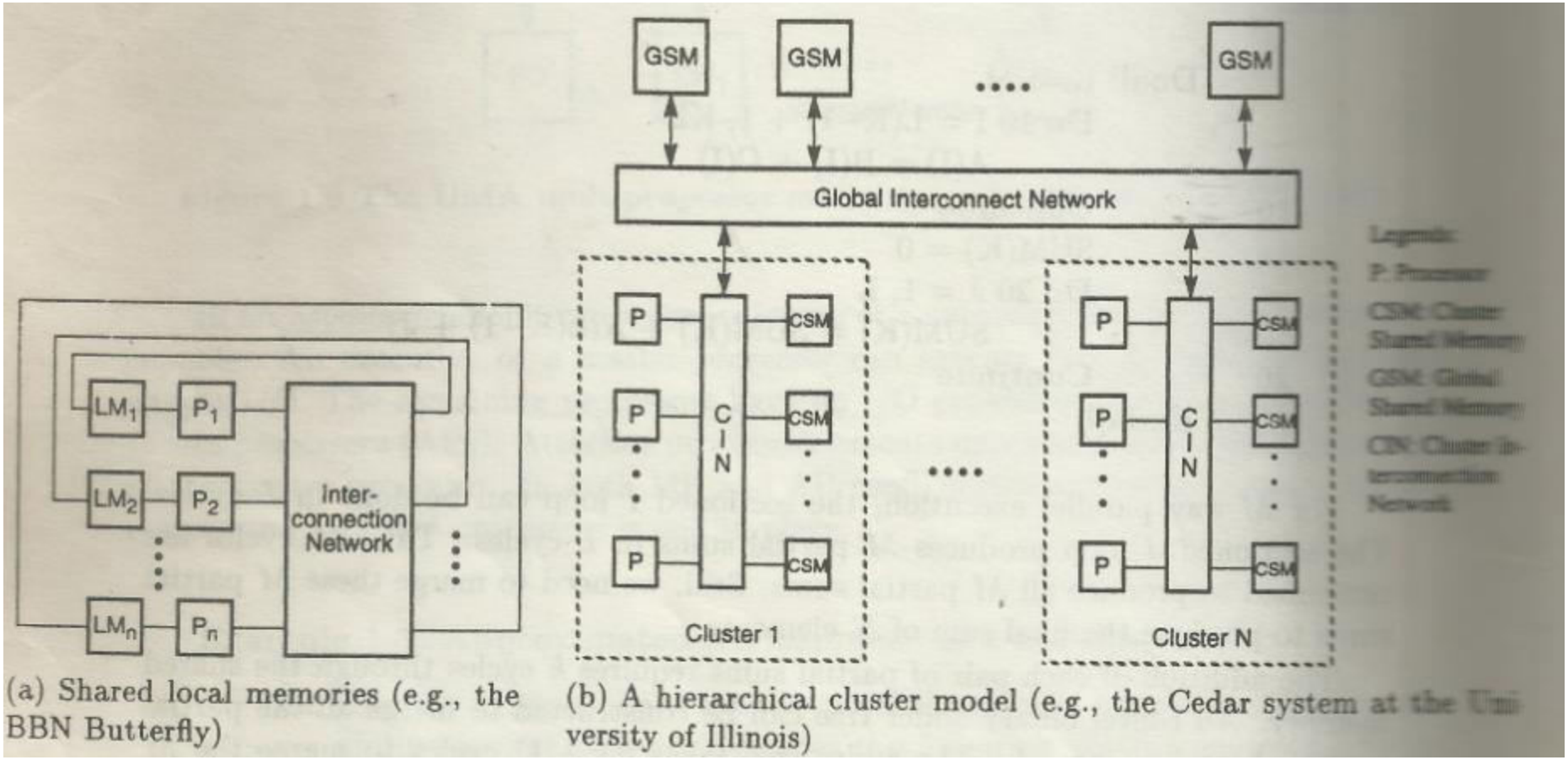
- **UMA means all processors have equal access time to all memory words(–used in general purpose computer)**
  - Each processor may have local cache
  - 1-- Symmetric UMA Multiprocessors
  - Most commonly represented by Symmetric Multiprocessor (SMP) machines
  - Identical processors
    - All processors have equal access to peripheral devices
    - The OS kernel (executive program) & I/O service routines can be run on any of the processors
    - 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.
  - 2—Asymmetric UMA Multiprocessors
    - Only one or a subset of processors are executive enabled
    - Remaining processors have no I/O capabilities and are called attached processors(AP)
    - Aps execute code under supervision of Master processor
    - Memory sharing of master and APs exist.

UMA Schematic Diagram



MIMD- Shared Memory Multiprocessors- NUMA (non-uniform memory access model)

- Often made by physically linking two or more SMPs
- One SMP directly access memory of another SMP
- Not all processors have equal access time to all memories
  - The shared memory is physically distributed to all processors
    - Called Local Memories
  - Collection of all local memories form global memory address space accessible all processors
  - Faster to access local memory of a processor by the processor than to access remote memory attached to another processor
- Memory access across link /interconnection networks is slower
- If cache coherency is maintained, then may also be called CC-NUMA - Cache Coherent NUMA



(a) Shared local memories (e.g., the BBN Butterfly)

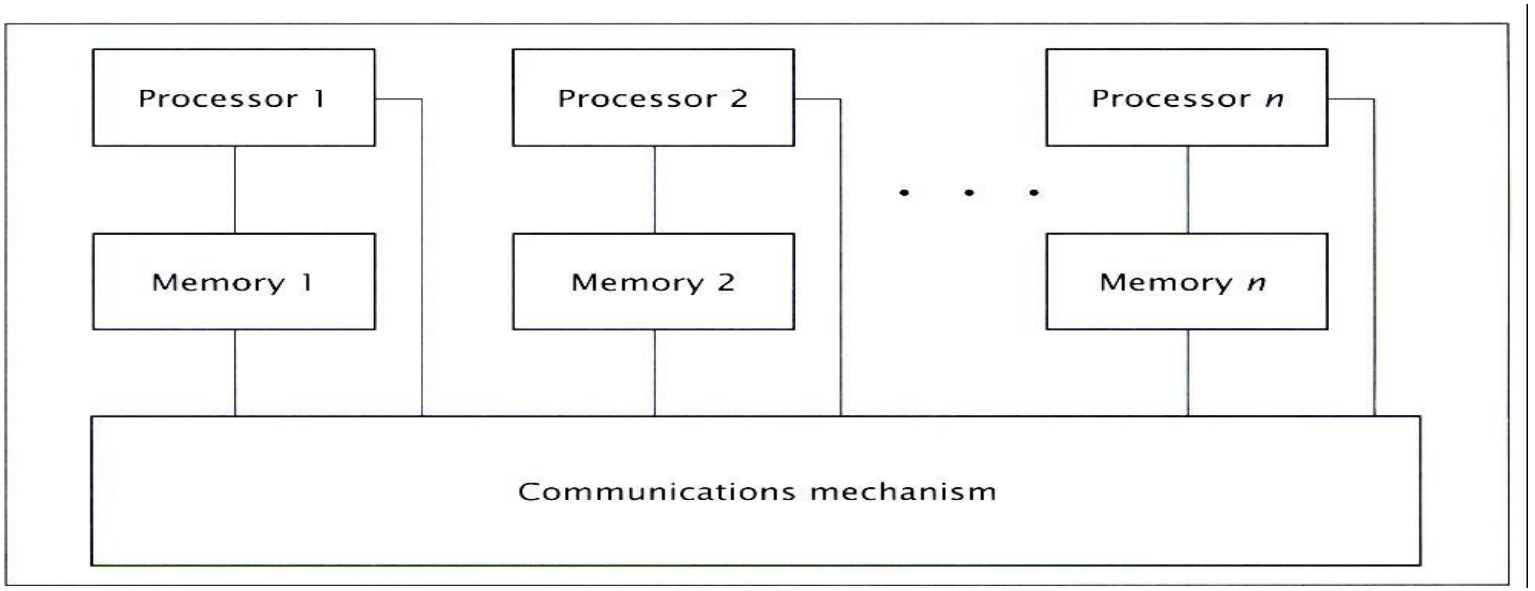
(b) A hierarchical cluster model (e.g., the Cedar system at the University of Illinois)

P:Processor, CSM: Cluster Shared Memory , CIN: Cluster Interconnection Network, GSM : Global Shared Memory, LM : (Shared)Local Memory

MIMD- Shared Memory Multiprocessors- NUMA (non-uniform memory access model)

- globally shared memory can be added to distributed memory
  - Memory access pattern :
    - Local memory access is fastest
    - Global shared memory access speed comes next
    - Access to remote processor memory slowest.
- Microprocessors can be hierarchically clustered
  - Each cluster is itself an UMA or NUMA multiprocessor
  - All processors belonging to the same cluster, uniformly access the cluster shared memory modules

NUMA Schematic Diagram



Multiprocessors- COMA (Cache Only Memory)

- A special case of NUMA
  - Distributed main memories are converted to cache
  - All cache form a global address space
  - Remote cache access assisted by distributed cache directories (D)
  - Depending on the interconnection network used, sometimes hierarchical directories may be used to locate copies of cache block
  - Initial data placement is not critical because data eventually migrate to where it will be used

MIMD- Shared Memory Multiprocessors- COMA (Cache Only Memory Access)

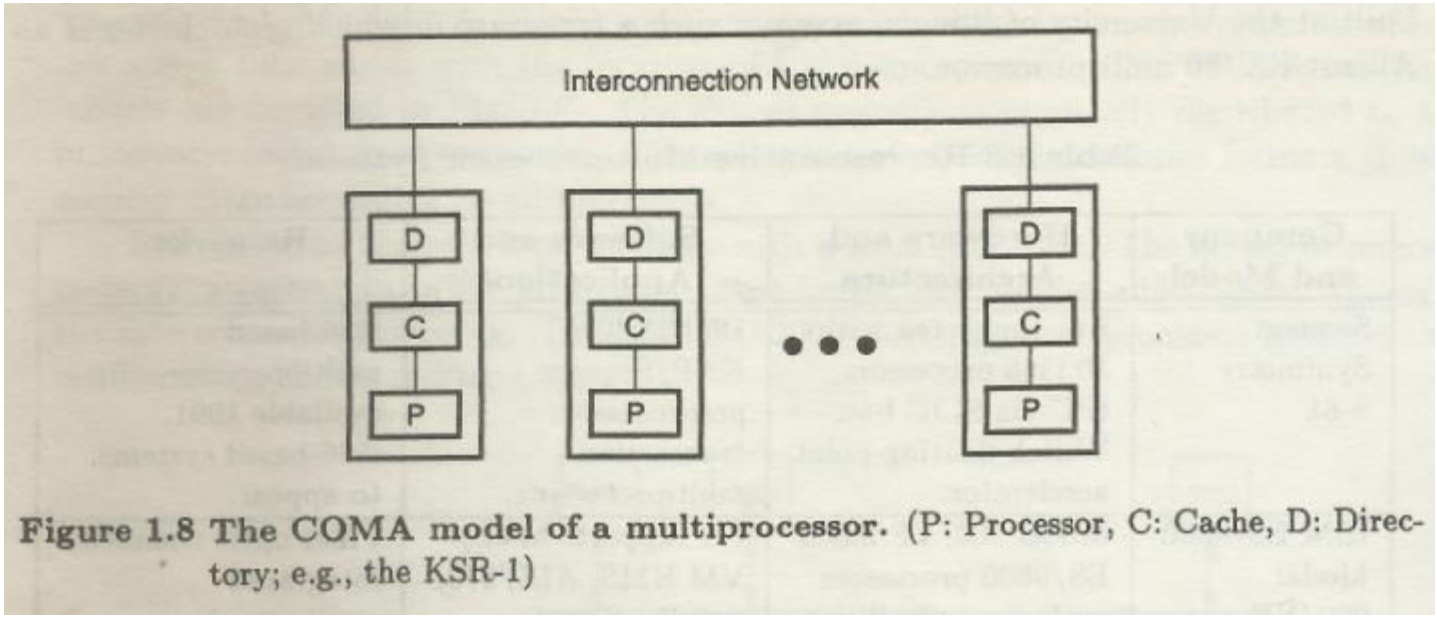
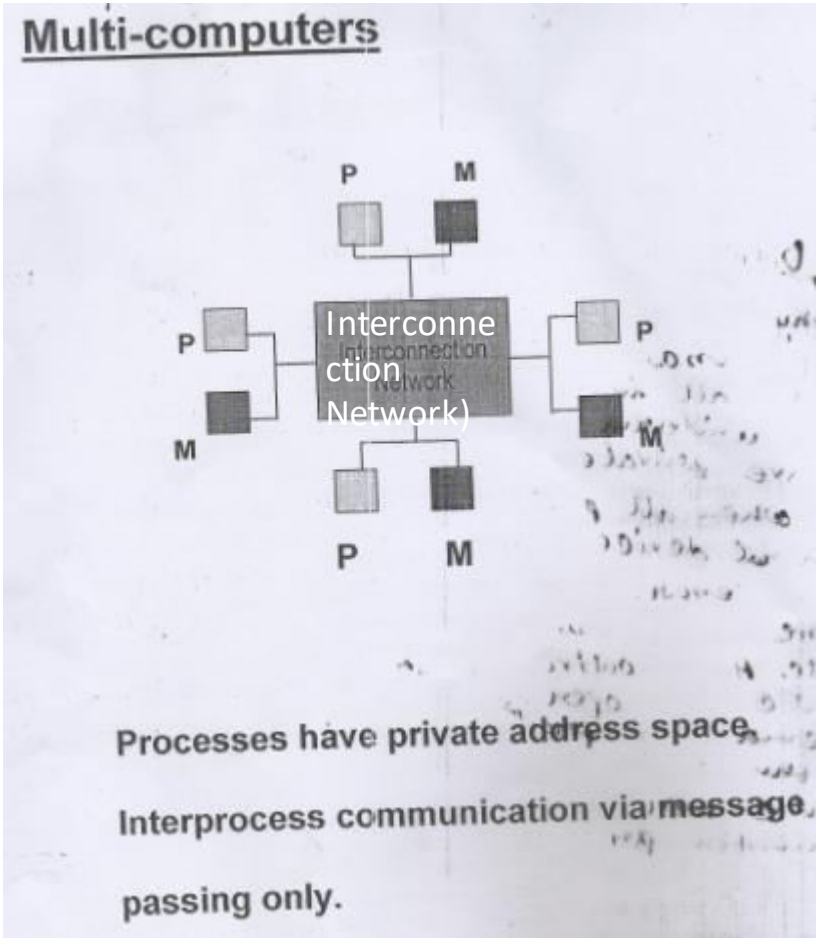


Figure 1.8 The COMA model of a multiprocessor. (P: Processor, C: Cache, D: Directory; e.g., the KSR-1)

Shared Memory: Pro and Con

- 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.

Salient Features of Each Type of MIMD-
Memory architectures of Distributed Memory Multiprocessors/Multicomputers



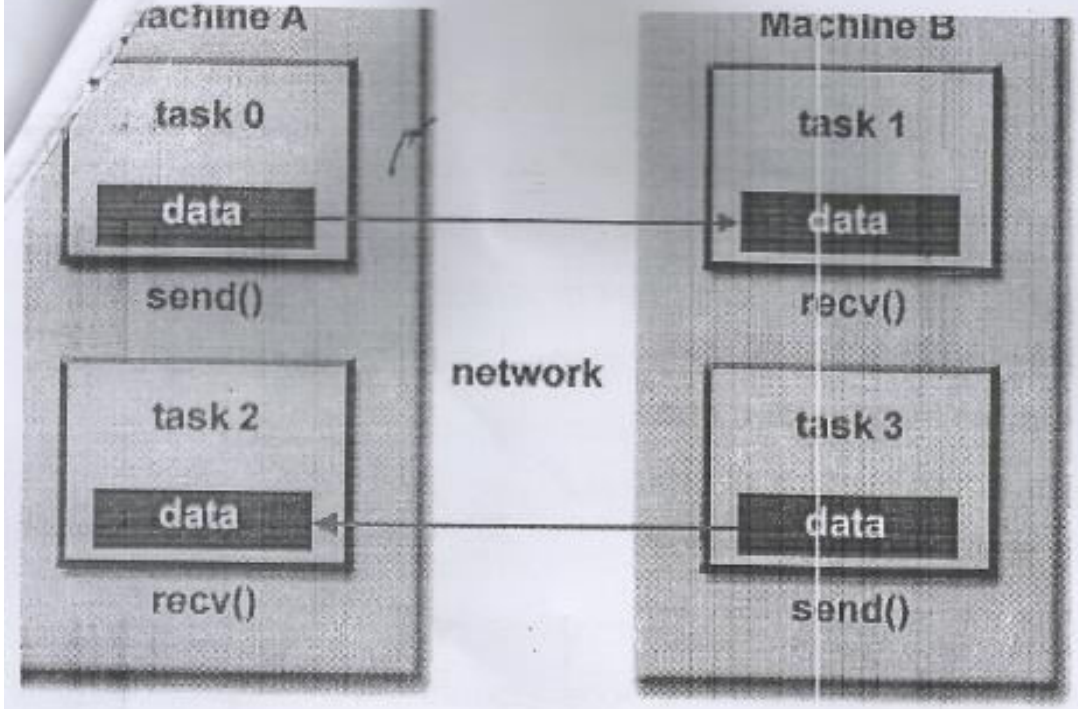
Multicomputers

- Inexpensive computers (Each processor with its own memory unit) connected by LAN.
  - Clusters of computer

Distributed Memory

- Distributed memory system require a communication network to connect inter processor memory
- Processors have their own memory. Memory address in one processor do not map to another processor(No global address across all processors)
- Each processor works independently with its own memory . Changes made in its own memory do not have effect on the memory of other processors . Hence no cache coherence apply.

Message Passing Model



- The message passing model demonstrates the following characteristics:
  - A set of tasks that use their own local memory during computation. Multiple tasks can reside on the same physical machine and/or across an arbitrary number of machines.
  - Tasks exchange data through communications by sending and receiving messages.
  - Data transfer usually requires cooperative operations to be performed by each process. For example, a send operation must have a matching receive operation.

Message Passing Implementation

Implementations:

- From a programming perspective, message passing implementations commonly comprise a library of subroutines that are imbedded in source code. The programmer is responsible for determining all parallelism.
- Historically, a variety of message passing libraries have been available since the 1980s. These implementations differed substantially from each other making it difficult for programmers to develop portable applications.
- In 1992, the MPI Forum was formed with the primary goal of establishing a standard interface for message passing implementations.
- Part 1 of the **Message Passing Interface (MPI)** was released in 1994. Part 2 (MPI-2) was released in 1996. Both MPI specifications are available on the web at http://www-unix.mcs.anl.gov/mpi/.
- MPI is now the "de facto" industry standard for message passing, replacing virtually all other message passing implementations used for production work. Most, if not all of the popular parallel computing platforms offer at least one implementation of MPI. A few offer a full implementation of MPI-2.
- For shared memory architectures, MPI implementations usually don't use a network for task communications. Instead, they use shared memory (memory copies) for performance reasons

Massively Parallel Computer

Massive parallel processing

Main article: Massive parallel processing

A massively parallel processor (MPP) is a single computer with many networked processors. MPPs have many of the same characteristics as clusters, but MPPs have specialized interconnect networks (whereas clusters use commodity hardware for networking). MPPs also tend to be larger than clusters, typically having "far more" than 100 processors.[29] In an MPP, "each CPU contains its own memory and copy of the operating system and application. Each subsystem communicates with the others via a high-speed interconnect."[30]

A cabinet from Blue Gene/L, ranked as the fourth fastest supercomputer in the world according to the 11/2008 TOP500 rankings. Blue Gene/L is a massively parallel processor.

Blue Gene/L, the fifth fastest supercomputer in the world according to the June 2009 TOP500 ranking, is an MPP.

Grid computing

Main article: Grid computing

Grid computing is the most distributed form of parallel computing. It makes use of computers communicating over the Internet to work on a given problem. Because of the low bandwidth and extremely high latency available on the Internet, grid computing typically deals only with embarrassingly parallel problems. Many grid computing applications have been created, of which SETI@home and Folding@Home are the best-known examples.[31]

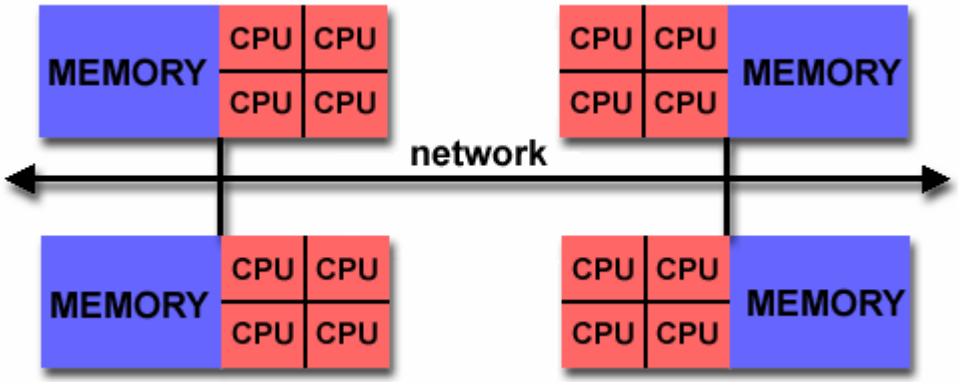
Most grid computing applications use middleware, software that sits between the operating system and the application to manage network resources and standardize the software interface. The most common grid computing middleware is the Berkeley Open Infrastructure for Network Computing (BOINC). Often, grid computing software makes use of "spare cycles", performing computations at times when a computer is idling.

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

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.

Different Types of Uniprocessor Computers

- Von Neumann Architecture
  - Slow due to sequential execution of instructions in programs
- Fast Pipelined Instruction execution, pipelined arithmetic computations and memory access
  - Useful in performing identical operations repeatedly over vector data stings (processor with multiple functional units)