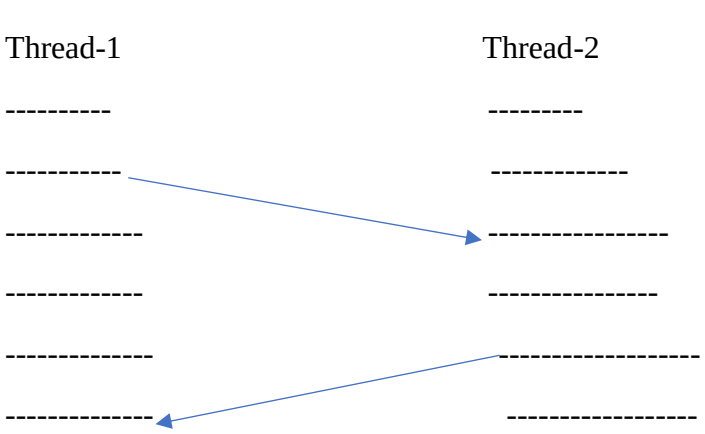
Notes-05

MIMD Architectures:

Informal: As we strive for higher performance, we consider the parallel execution of cooperating programs, or threads, on multiple processors. For example, we can sort two halves of a sequence of numbers on two independent processors **in parallel.** We can then merge the two sorted sequences to produce a single sorted sequence.

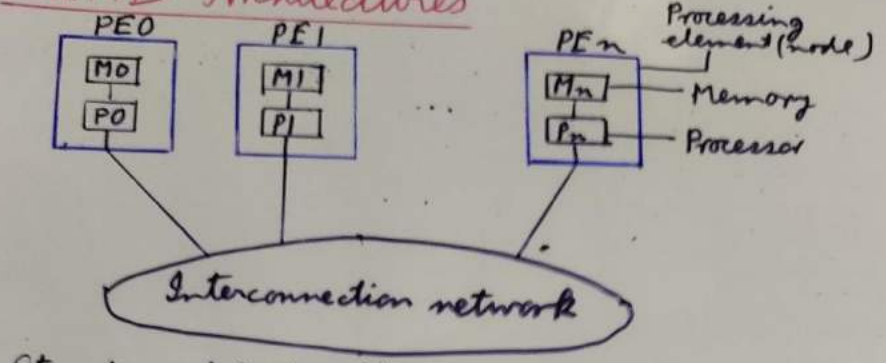
The gross picture is that each thread runs on a different processor. They need to exchange information; this is shown by the arrows between the threads.



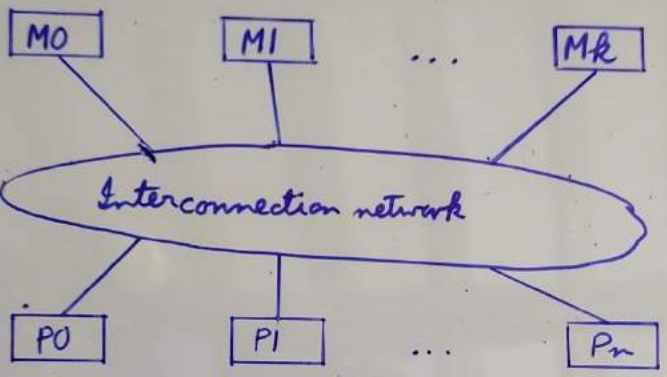
Thus, the two parallel threads need to communicate in order to transfer data between themselves; they cooperate to achieve a goal.

The communication can be through shared memory or through messages.

MIMD Architectures



Structure of Distributed Memory MIMD Architectures



Structure of Shared Memory MIMD Architectures

(1) Distributed memory (or message-passing)
→ Whenever interaction among PEs is necessary they send messages to each other. None of the PEs can ever access the memory module of another PE directly.

(2) Shared Memory → Any processor can directly access any memory module via an interconnection network. The set of memory modules defines a global address space which is shared among the processors.

MIMD: Interconnection Networks

Distributed memory MIMD architectures are often simply called multicomputers while shared memory MIMD architectures are referred to as multiprocessors.

According to their topology interconnection networks can be classified as static or dynamic networks.

Static networks → connection of switching units is fixed and typically realized as direct or point-to-point connections. These networks are also called direct networks.

Dynamic networks → communication links can be reconfigured by setting the active switching units of the system.

Multicomputers are typically based on static networks while dynamic networks are mainly employed in multiprocessors.

Distributed memory systems: ADVANTAGES

- Since processors work with their attached local memory module most of the time, the contention problem is not so severe as in shared memory systems. As a result, distributed memory multicomputers are highly scalable and good architectural candidates for building massively parallel computers.

- Processes cannot communicate through shared data structures and hence sophisticated synchronization techniques like monitors are not needed. Message passing solves not only communication but synchronization as well.

Distributed memory systems: DISADVANTAGES

- In order to achieve high performance in multicomputers special attention should be paid to load balancing.

- Message-passing-based communication and synchronization can lead to deadlock situations.

- Although there is no architectural bottleneck in multicomputers, message passing requires physical copying of data structures between processes. Intensive data copying can result in significant performance degradation.

SHARED MEMORY SYSTEMS: Advantages

- There is no need to partition either the code or the data; therefore uniprocessor programming techniques can easily be adapted in the multiprocessor environment. Neither new programming languages nor sophisticated compilers are needed to exploit shared memory systems.
- There is no need to physically move data when two or more processes communicate. The consumer process can access the data from the place where the producer stored it. As a result communication between processes is very efficient.

SHARED MEMORY SYSTEMS: Disadvantages

- Synchronized access to shared data structures requires special synchronizing constructs such as semaphores, conditional critical regions, monitors, and so on. Usually message-passing synchronization is simpler to understand and apply.
- The main disadvantage of shared memory systems is lack of scalability due to the contention problem. When several processors want to access the same memory module they must compete for the right to do so. The winner can access the memory, while the losers must wait. The larger the number of processors, the higher the probability of memory contention. Beyond a certain number of processors this probability is so high that adding a new processor to the system will not increase performance.

LOW SCALABILITY: Solutions

- Use of a high throughput low-latency interconnection network.
- Use of cache memories.
- The logically shared memory can be physically implemented as a collection of local memories. This new architecture type is called a virtual shared memory or distributed shared memory architecture.

In distributed shared memory systems, the local memories are components of a global address space and any processor can access the local memory of any other processor.

In distributed memory systems the local memories have separate address spaces and direct access to the local memory of a remote processor is prohibited.

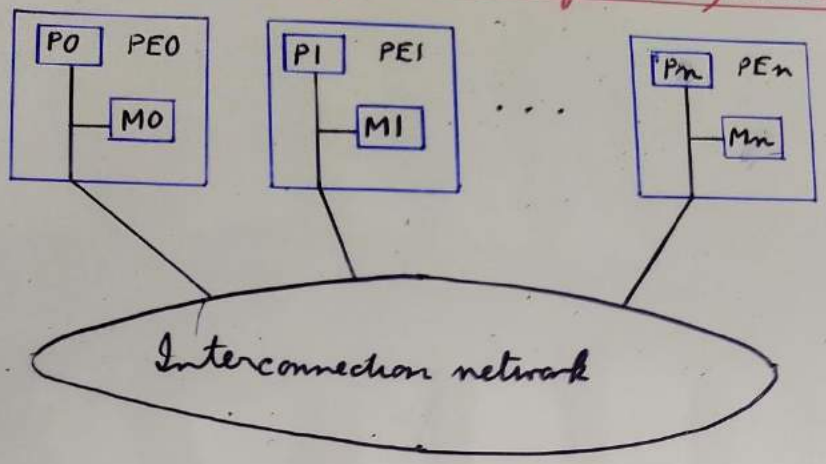
DISTRIBUTED SHARED MEMORY SYSTEMS: Classification

(i) Non-uniform memory access (NUMA) machines;

(ii) Cache-coherent non-uniform memory access (CC-NUMA) machines; and

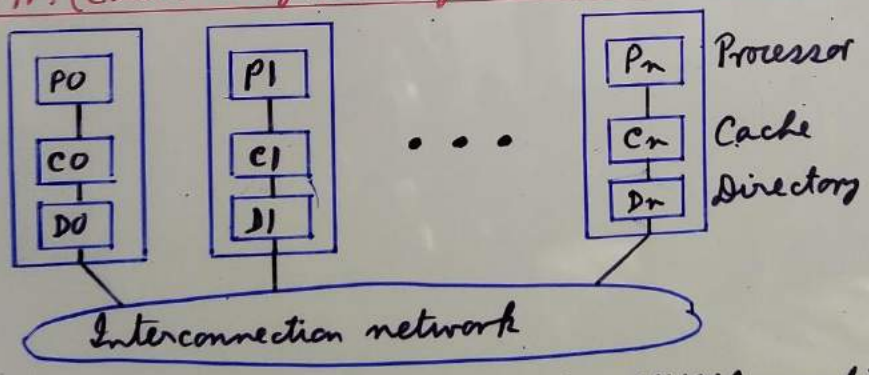
(iii) Cache-only memory access (COMA) machines.

NUMA (Non-uniform memory access) machines



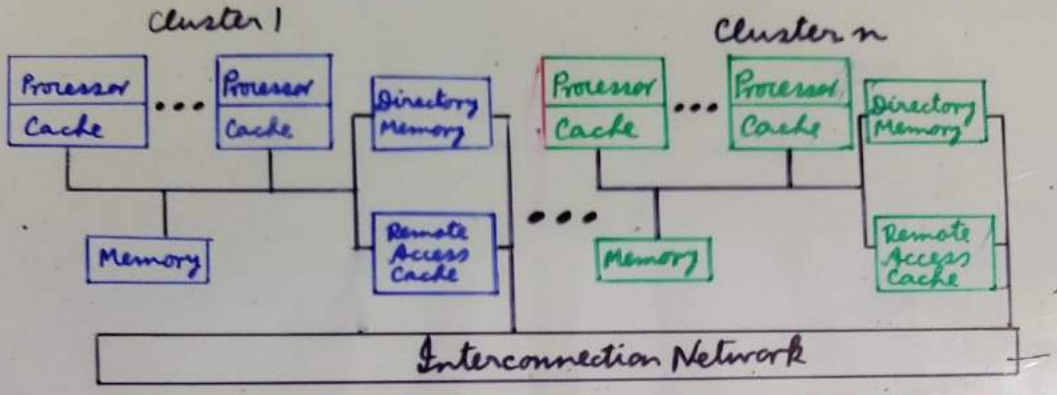
In NUMA machines, the shared memory is divided into as many blocks as there are processors in the system and each memory block is attached to a processor as a local memory with direct bus connection. As a result, whenever a processor addresses the part of the shared memory that is connected as local memory, access to that block is much faster than access to remote ones.

COMA (Cache-only memory access) machines



The COMA model is a special case of a NUMA machine, in which the distributed main memories are converted to caches. There is no memory hierarchy at each processor node. All the caches form a global address space. Remote cache access is assisted by the distributed cache directories.

CC-NUMA (Cache-coherent non-uniform memory access) machines



Stanford Dash [A CC-NUMA architecture]

CC-NUMA machines represent a compromise between the NUMA and COMA machines. Like the NUMA machines, the shared memory is constructed as a set of local memory blocks. However, in order to reduce the traffic on the interconnection network each processor node is supplied with a large cache memory block.

The Dash architecture is a large-scale CC-NUMA multiprocessor system. It consists of multiple microprocessor clusters connected through a scalable, low-latency interconnection network. Physical memory is distributed among the processing nodes in various clusters. The distributed memory forms a global address space.

For each memory block, the directory keeps track of remote nodes caching it. Directory memory and remote access cache facilitate prefetching and the directory-based coherence protocol.

Shared Memory MIMD Architectures

The distinguishing feature of shared memory systems is that no matter how many memory blocks are used in them and how these memory blocks are connected to the processors, the address spaces of these memory blocks are unified into a global address space which is completely visible to all processors.

Design issues

The three main design issues in increasing the scalability of shared memory systems are:

- Organization of memory
- Design of interconnection networks
- Design of cache coherent protocols

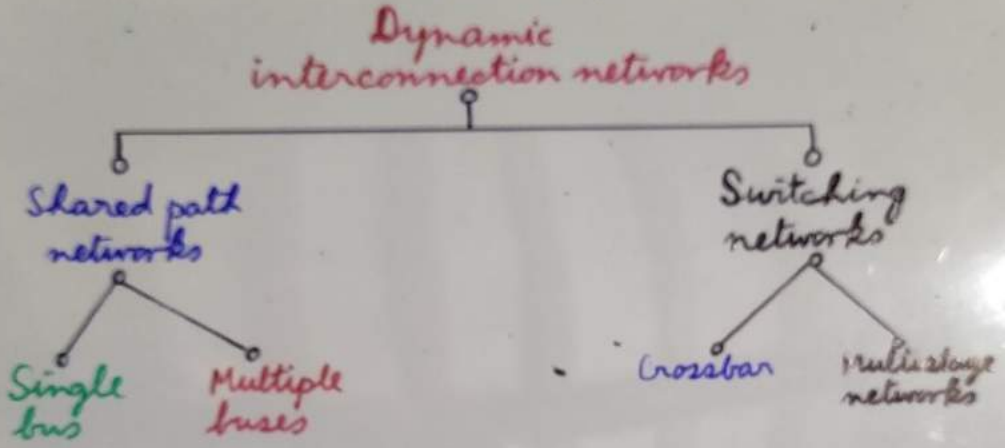
Organization of memory

Shared memory systems are basically classified according to their memory organization since this is the most fundamental design issue. Accordingly, shared memory systems can be divided into four main classes:

- Uniform memory access (UMA) machines
- Non-uniform memory access (NUMA) machines
- Cache-coherent non-uniform memory access (CC-NUMA) machines
- Cache-only memory access (COMA) machines

Shared Memory MIMD: Interconnection networks

The quality of the interconnection network has a decisive impact on the speed, size, and cost of the whole machine. Dynamic interconnection schemes are usually employed in multiprocessors.



Shared path networks: provide continuous connection among the processors and memory blocks. The continuous connection is shared among the processors which have to compete for its use.

Switching networks: do not provide a continuous connection among the processors and memory blocks. A switching mechanism enables processors to be temporarily connected to memory blocks.