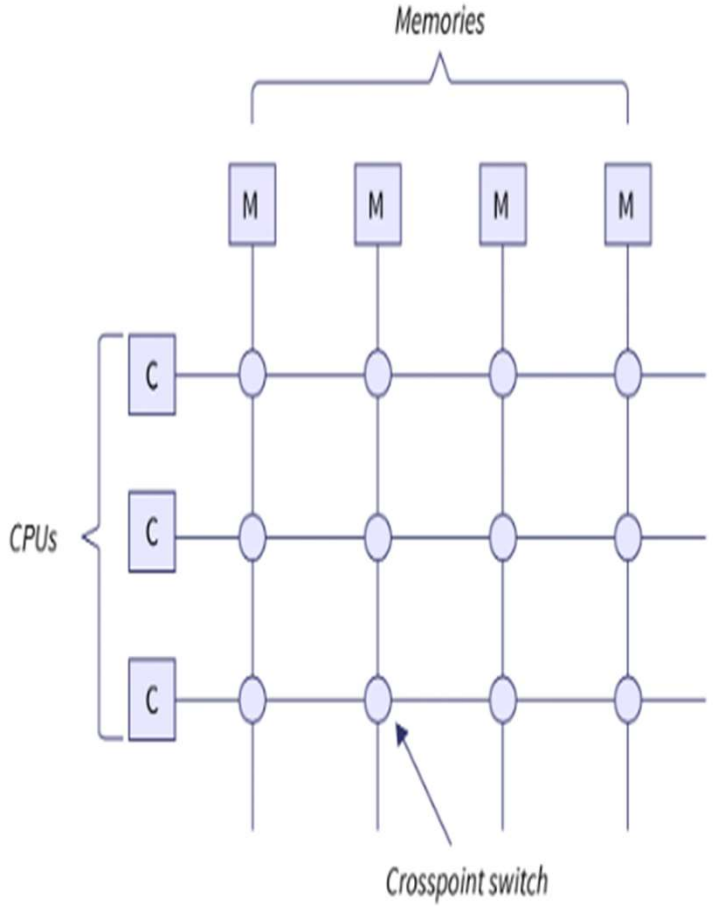
Switched Multiprocessors, Directories, Caching

Switched Multiprocessor:

- When we build multiprocessors with more than 64 processors.
- Divide memory into modules and connect them to the CPU.
- This is used to build a system as multiple clusters and connect clusters using an intercluster bus.
- From the figure, every intersection is a tiny electronic crosspoint switch that can be opened and closed to transfer data.



Advantages:

Scalability: Adding more processors is easier without congestion since each processor can have its own path to memory.
Reduced Contention: Switches allow multiple data transactions to happen simultaneously without much interference.
Parallel Processing: Efficiently supports parallel operations, as processors can work concurrently with fewer bottlenecks.

Directories in Distributed Shared Memory

Purpose:

- Directories are used to manage data location and access rights in a DSM system.
- They help keep track of which memory blocks are stored where and who needs access to them.

How Directories Work:

- Each memory block has an entry in the directory, recording its status (modified, shared, etc.) and which processors hold copies of it.

- When a processor requests data, the directory checks the block's status and either provides access directly or communicates with other processors to get the latest copy.

Types of Directory Organizations:

Centralized Directory: A single directory holds all the information for every memory block in the system. Simple but becomes a bottleneck as the number of processors grows.

Distributed Directory: The directory is spread across multiple nodes, with each node managing information about certain memory blocks. It scales better than a centralized directory.

Hierarchical Directory: Combines centralized and distributed approaches, organizing directories in a hierarchy to balance scalability and manageability.

Caching in Distributed Shared Memory

Objective of Caching in DSM: Caching helps improve performance by allowing processors to keep local copies of frequently accessed data, reducing access time and load on the shared memory.

Challenges in DSM Caching:

Cache Coherence: Ensuring that all copies of a cached data block reflect the most recent updates. In a distributed system, copies may exist in different processors' caches, so updates need to be synchronized.

Consistency Models: Defines the order in which updates to shared data are visible to processors. Examples include strict consistency, sequential consistency, and eventual consistency.