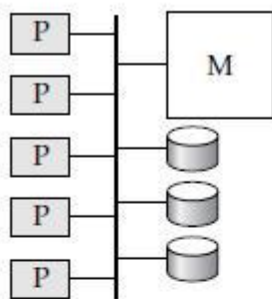


Parallel Database

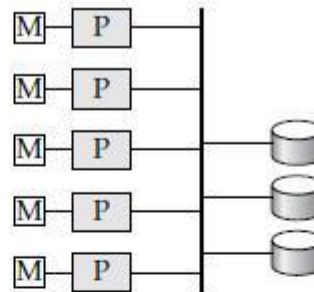
A **parallel database** system seeks to improve performance through parallelization of various operations, such as loading data, building indexes and evaluating queries. Although data may be stored in a distributed fashion, the distribution is governed solely by performance considerations. Parallel databases improve processing and input/output speeds by using multiple CPUs and disks in parallel. Centralized and client-server database systems are not powerful enough to handle such applications. In parallel processing, many operations are performed simultaneously, as opposed to serial processing, in which the computational steps are performed sequentially.

Parallel databases can be roughly divided into four categories:

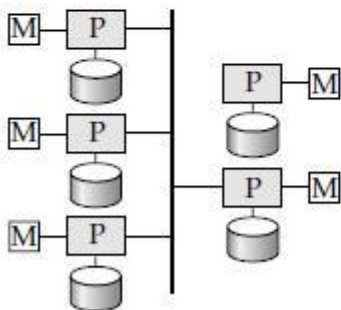
- Ø **Shared memory:** All the processors share a common memory (Figure a).
- Ø **Shared disk:** All the processors share a common set of disks (Figure b). Shared-disk systems are sometimes called **clusters**.
- Ø **Shared nothing:** The processors share neither a common memory nor common disk (Figure c).
- Ø **Hierarchical:** This model is a hybrid of the preceding three architectures (Figure d).



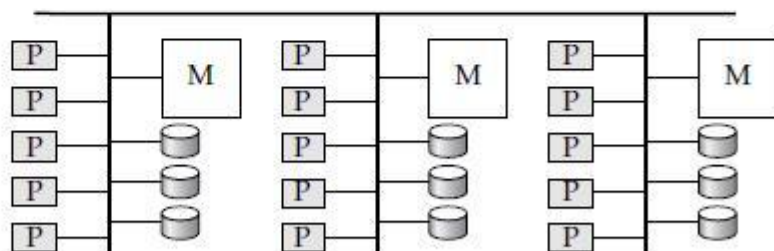
(a) shared memory



(b) shared disk



(c) shared nothing



(d) hierarchical

Shared Memory

- In **shared-memory** architecture, the processors and disks have access to a common memory, typically via a bus or through an interconnection network.
- Benefit of using shared memory is extreme efficient communication between processors - data in shared memory can be accessed by any processor without being moved with software.
- A processor can send messages to other processors much faster by using memory writes than by sending messages through communication medium.
- There is downside of shared memory as well. The architecture is not scalable beyond 32 or 64 processors. Reason behind this downside is bus or the interconnection network becomes the bottleneck. Adding more processors will make processors spend most of their time in waiting for their turn on the bus to access memory.
- Shared memory architectures usually have large memory caches at each processor, so that referencing of the shared memory is avoided whenever possible.
- Moreover, caches need to be coherent, that is, if a processor performs a write to a memory location, the data in that memory location should be either updated at or removed from any processor where the data is cached.

Shared Disk

- In the **shared-disk** model, all processors can access all disks directly via an interconnection network, but the processors have private memories.
- Shared disk has two advantages over shared memory. First is, since each processor has its own memory, the memory bus is not a bottleneck. Second is, it offers a cheap way to provide a degree of **fault tolerance**.
- **Fault tolerance:** If a processor (or its memory) fails, the other processor can take over its tasks, since the database is resident on disks that are accessible from all processors.
- The main problem with shared disk system is again scalability. Although the memory bus is no longer a bottleneck, the interconnection to the disk subsystem is now a bottleneck.
- Compared to share memory systems, shared disk systems can scale to a somewhat larger number of processors, but communication across processors is slower, since it has to go through a communication network.

Shared nothing

- In **shared nothing** system, each node of the machine consists of a processor, memory and one or more disks. The processors at one node may communicate with another processor at another node by a high speed interconnection network.
- A node functions as a server for the data on the disk.
- Moreover, the interconnection networks for shared nothing systems are usually designed to be scalable, so that their transmission capacity increases as more nodes are added.
- Consequently, shared nothing architectures are more scalable and can easily support a large number of processors.
- Main drawback of shared nothing systems are the costs of communication and of non-local disk access, which are higher than in a shared memory or shared disk architecture since sending data involves software interaction at both ends.

Hierarchical

- The **hierarchical architecture** comes with characteristics of shared memory, shared disk and shared nothing architectures. At the top level, the system consists of nodes connected by an interconnection network, and do not share disks or memory with one another.
- Thus, the top level is shared-nothing architecture.
- Attempts to reduce the complexity of programming such systems have yielded **distributed virtual-memory** architectures, where logically there is a single shared memory, the memory mapping hardware coupled with system software, allows each processor to view the disjoint memories as a single virtual memory,
- Such architectures are also referred to as a **nonuniform memory architecture (NUMA)**.