Types of Architecture Database
Sonia Ordoñez

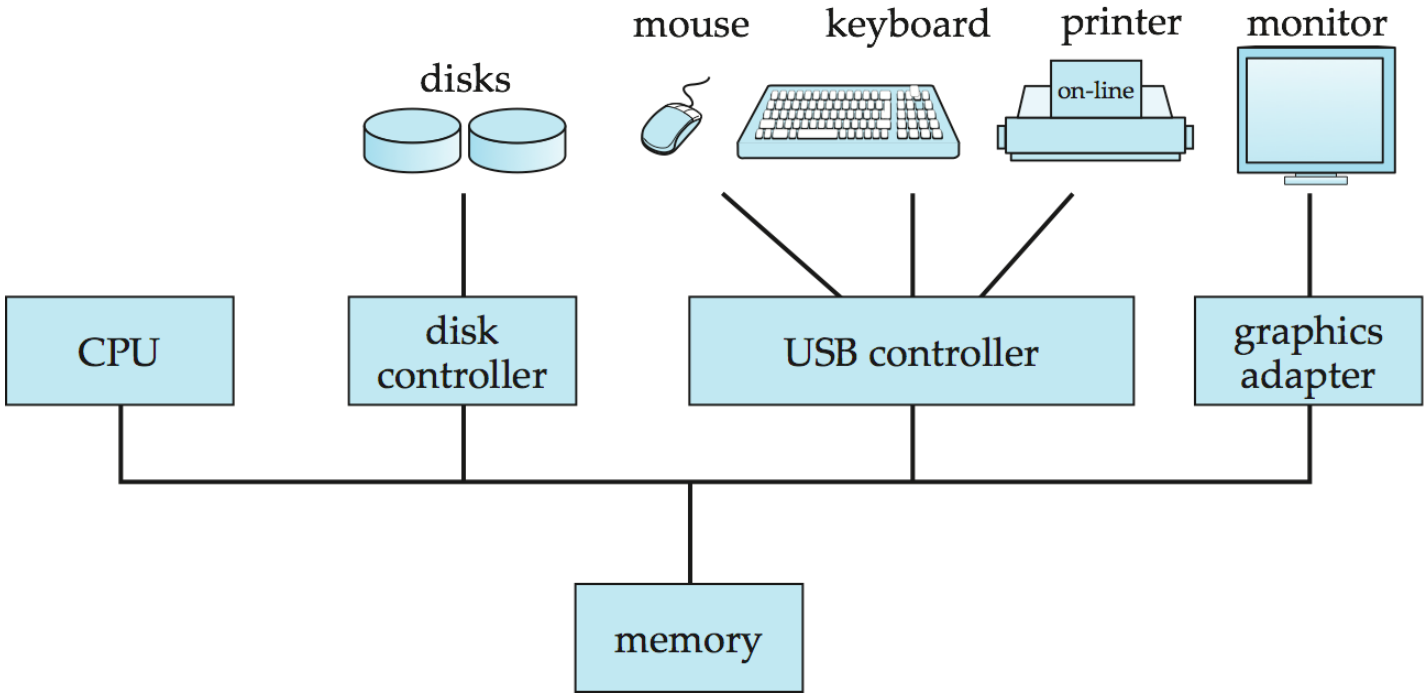
Database System Concepts, 6th Ed.

Capitulo 4: Database System Architectures

- Centralized and Client-Server Database
- Parallel Database
- Distributed Database
- Cluster Database
- Grid Database
- Cloud Dtabase
- P2P Database
- Memory Database

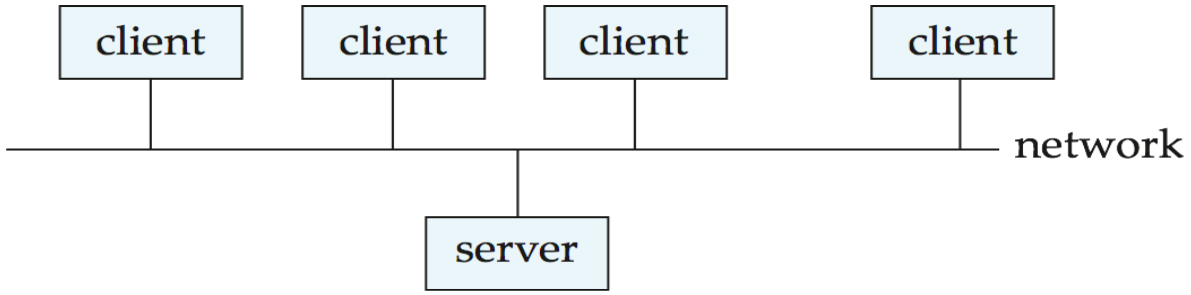
Centralized Systems

- Run on a single computer system and do not interact with other computer systems.
- General-purpose computer system: one to a few CPUs and a number of device controllers that are connected through a common bus that provides access to shared memory.
- Single-user system (e.g., personal computer or workstation): desk-top unit, single user, usually has only one CPU and one or two hard disks; the OS may support only one user.
- Multi-user system: more disks, more memory, multiple CPUs, and a multi-user OS. Serve a large number of users who are connected to the system vie terminals. Often called *server* systems.

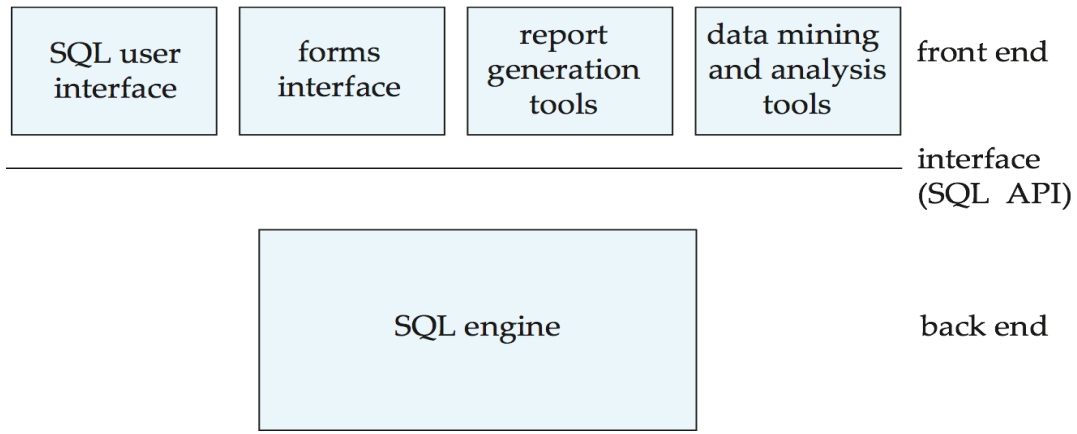


Client-Server Systems

- Server systems satisfy requests generated at *m* client systems, whose general structure is shown below:

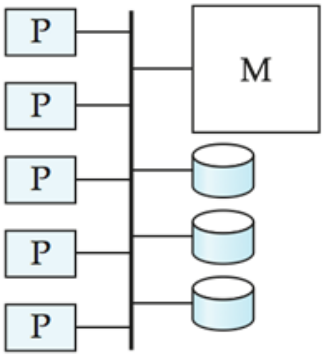


- Database functionality can be divided into:
  - **Back-end**: manages access structures, query evaluation & optimization, CC and recovery.
  - **Front-end**: consists of tools such as *forms*, *report-writers*, and graphical UI facilities.
- The interface between the front-end and the back-end is through SQL or through an application program interface

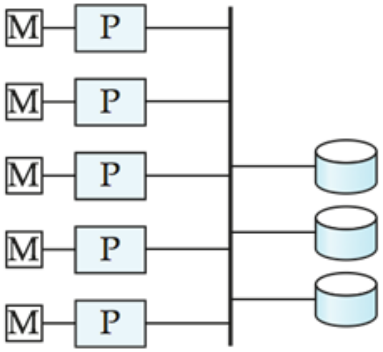


Parallel Database Architectures

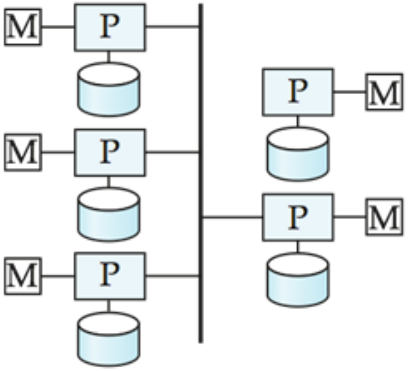
- **Shared memory** -- processors share a common memory
- **Shared disk** -- processors share a common disk, sometimes called clusters
- **Shared nothing** -- processors share neither a common memory nor common disk
- **Hierarchical** -- hybrid of the above architectures



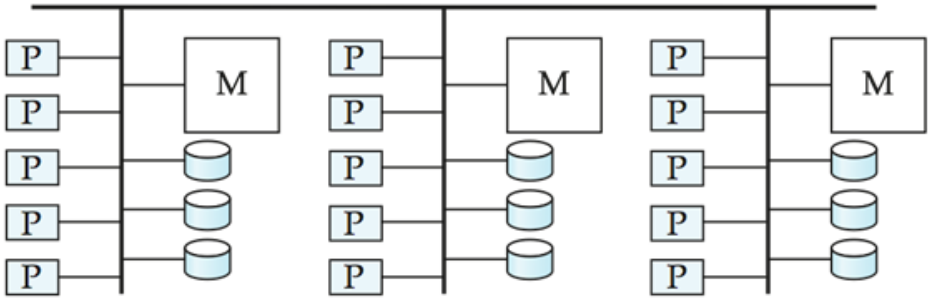
(a) shared memory



(b) shared disk



(c) shared nothing



(d) hierarchical

Parallel Databases

- I/O Parallelism

Reduce the time required to retrieve relations from disk by partitioning the relations on multiple disks.

- Interquery Parallelism

Different queries / transactions execute in parallel with one another
Increases transaction throughput

- Intraquery Parallelism

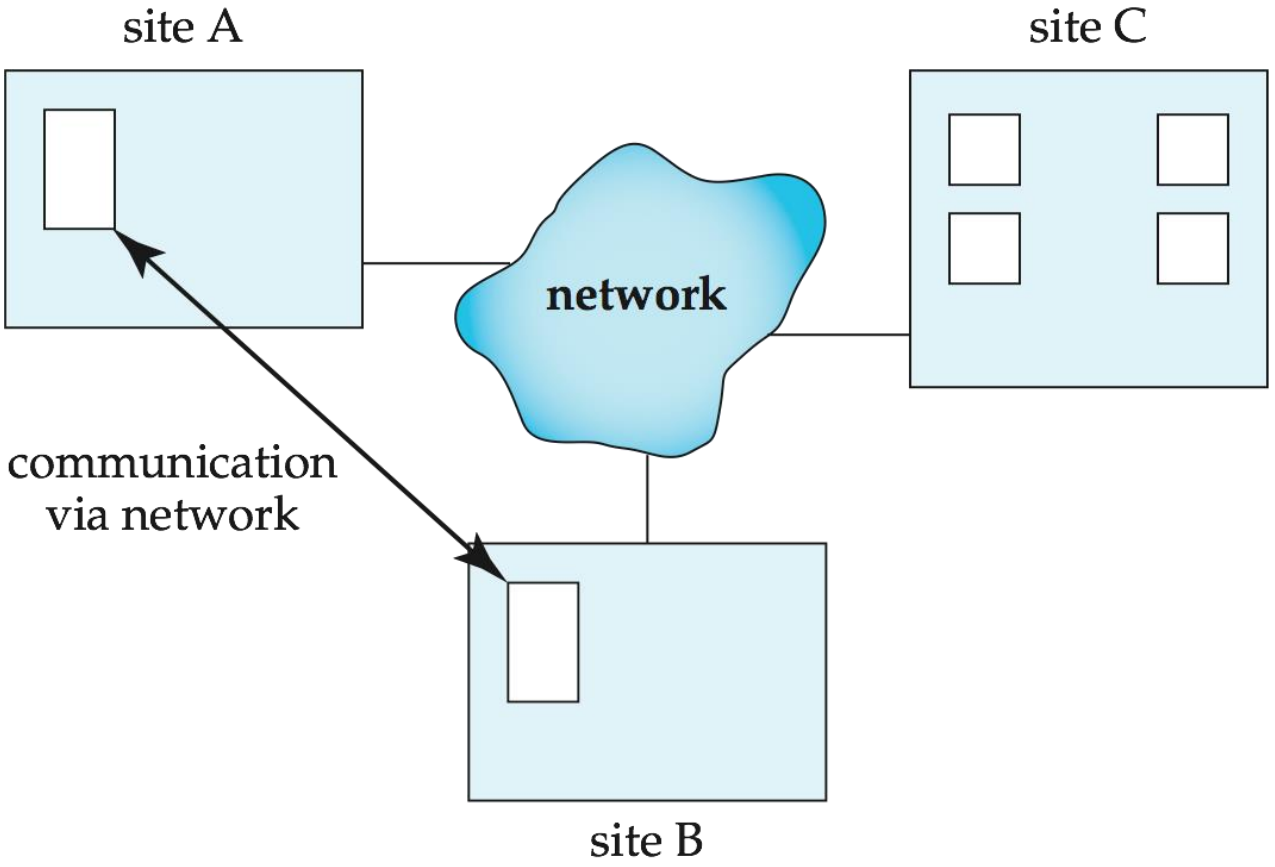
Execution of a single query in parallel on multiple processors / disks

- Intraoperation e Interoperation Parallelism

Sort, Join, Fragment-and-Replicate Join, Selection

Distributed Systems

- Data spread over multiple machines (also referred to as **sites**, **nodes**)
- Network interconnects the machines
- Data shared by users on multiple machines



Trade-offs in Distributed Systems

- Advantages
  - Sharing data
    - Users at one site able to access the data residing at some other sites.
  - Autonomy
    - Each site is able to retain a degree of control over data stored locally.
  - Higher system availability through redundancy
    - Data can be replicated at remote sites, and system can function even if a site fails.

- Disadvantages: added complexity for proper coordination among sites
  - Software development cost
  - Greater potential for bugs
  - Increased processing overhead

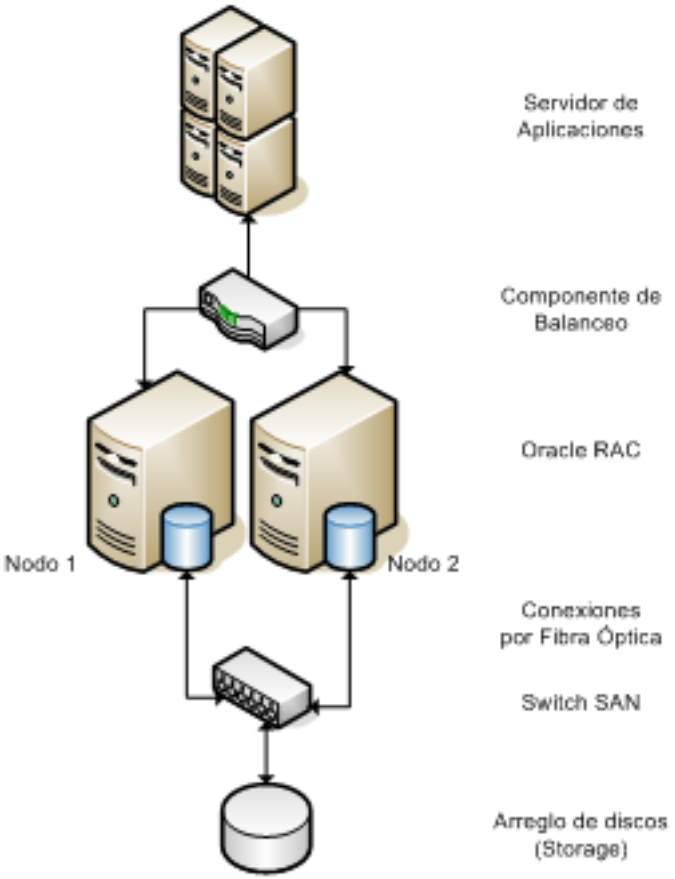
Distributed Databases

- Homogeneous distributed databases
  - Same software/schema on all sites, data may be partitioned among sites
  - Goal: provide a view of a single database, hiding details of distribution
- Heterogeneous distributed databases
  - Different software/schema on different sites
  - Goal: integrate existing databases to provide useful functionality
- Differentiate between *local* and *global* transactions
  - A local transaction accesses data in the *single* site at which the transaction was initiated.
  - A global transaction either accesses data in a site different from the one at which the transaction was initiated or accesses data in several different sites.

Cluster Database

Cluster

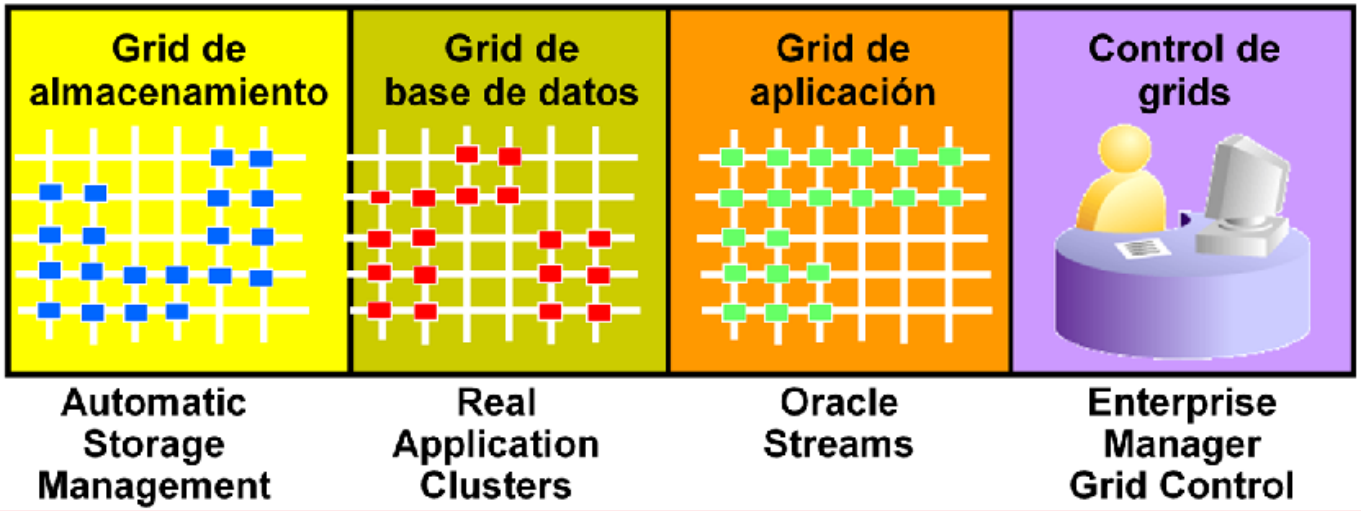
A cluster is a group of independent servers that cooperate behaving as if they were a single system.



- **An Clusters Database** , it is software that allows you to use a cluster of servers running multiple instances on the same database.
- The database files are stored on physical or logical disks attached to each node, so that all active instances can read or write them.
- RAC software manages data access so that changes in data are coordinated between instances and each instance sees consistent images of the database.
- The cluster interconnect allows instances to pass coordination information and data images to each other.

Grid Database

– Fácil de gestionar



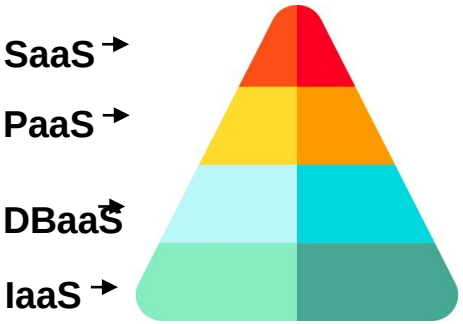
ORACLE

Cloud Database

The Cloud stack

All computing platforms are built on the concept of a technology stack, where one layer of technology is built on another. In the Cloud computing area, different categories of products bring Cloud computing to different levels of the stack.

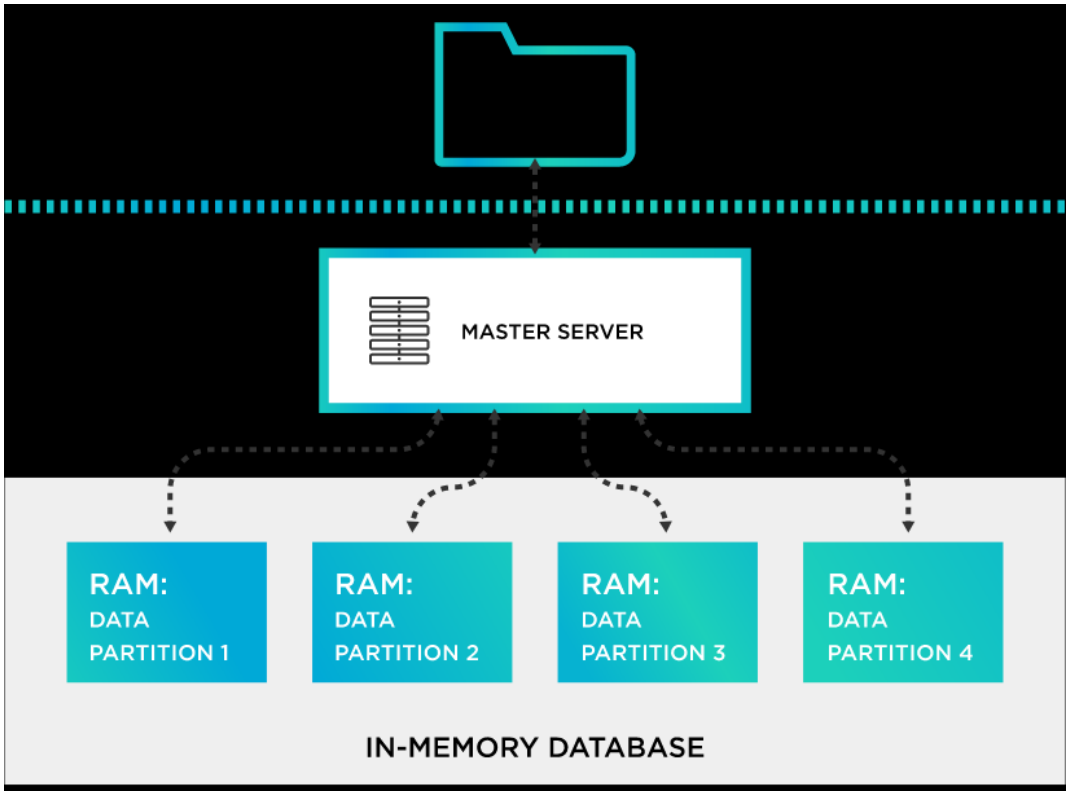
- There are four main categories of Cloud computing solutions



- Infrastructure-as-a-Service (IaaS), which gives users access to infrastructure components on a Cloud platform, such as operating systems and other system software
- Database-as-a-Service (DBaaS), which gives users access to databases running on a Cloud computing platform
- Platform-as-a-Service (PaaS), which gives users access to development and deployment environments running on a Cloud platform
- Software-as-a-Service (SaaS), which gives users access to solutions running on a Cloud computing platform

Fuente: https://www.oracle.com/technetwork/database/database-cloud/public/oracle-db-and-db-cloud-service-wp-1844127.pdf

Memory Database



Fuente: https://www.tibco.com/es/reference-center/what-is-an-in-memory-database

Memory Database

Nonvolatile Random Access Memory (NVRAM)

Another way to ensure data durability is through the use of non-volatile random access memory (NVRAM).

NVRAM retains data even after power is turned off. NVRAM is a popular solution used by in-memory databases to achieve data durability. In-memory databases use battery-backed static RAM or electrically erasable programmable ROM (EEPROM).

Fuente: https://www.tibco.com/es/reference-center/what-is-an-in-memory-database

End of Chapter