Dpto. Ciencias e Ingeniería de la Computación
Universidad Nacional del Sur

Elementos de Bases de Datos

Segundo Cuatrimestre 2015

Clase 20:

Arquitecturas para SMBD

Mg. María Mercedes Vitturini
[mvitturi@uns.edu.ar]

Arquitectura

"Any complex system is composed of subsystems that interact under the control of system design such that the system provides the expected behavior. When designing such a system, therefore, the logical approach is to *identify the subsystems that should compose the system, the interfaces of these subsystems, and the rules for interaction between the subsystems*. This is what software architecture aims to do."

A Concise Introduction to Software – Pankaj Jalote

"An important component of the design phase is the **architecture design**, which *describes the system's hardware, software, and network environment*. The architecture design flows primarily from the nonfunctional requirements, such as operational, performance, security, cultural, and political requirements..."

System Analysis and Design A. Dennis, B. Haley Wixom, R.Roth

DBMS - Arquitecturas

La "arquitectura" del DBMS está influenciada por el marco o *framework* que le da soporte, incluyendo:

- **Conexión en red:** permite distribuir las tareas entre las que se resuelven en el servidor y las que se ejecutan en el cliente (cliente–servidor).
- **Capacidad de procesamiento en paralelo:** capacidad del servidor de contar con elementos para trabajar en paralelo. Se aceleran las actividades: tiempos de respuesta y número de transacciones por unidad de tiempo.
- **Capacidades de distribución de datos:** que permite que
  - los datos residan donde se generan/ usan, permaneciendo accesibles a todos los sitios.
  - mantener múltiples copias de datos, que refuerza la disponibilidad

Arquitecturas de DBMS

Estilos arquitectónicos - Evolución

- *Las arquitecturas centralizadas*
  - Sistemas centralizados: monousuarios y multiusuario
  - Sistemas cliente-servidor
    - Servidor de transacciones
    - Servidor de datos
- *Arquitecturas paralelas*
- *Arquitecturas distribuidas*

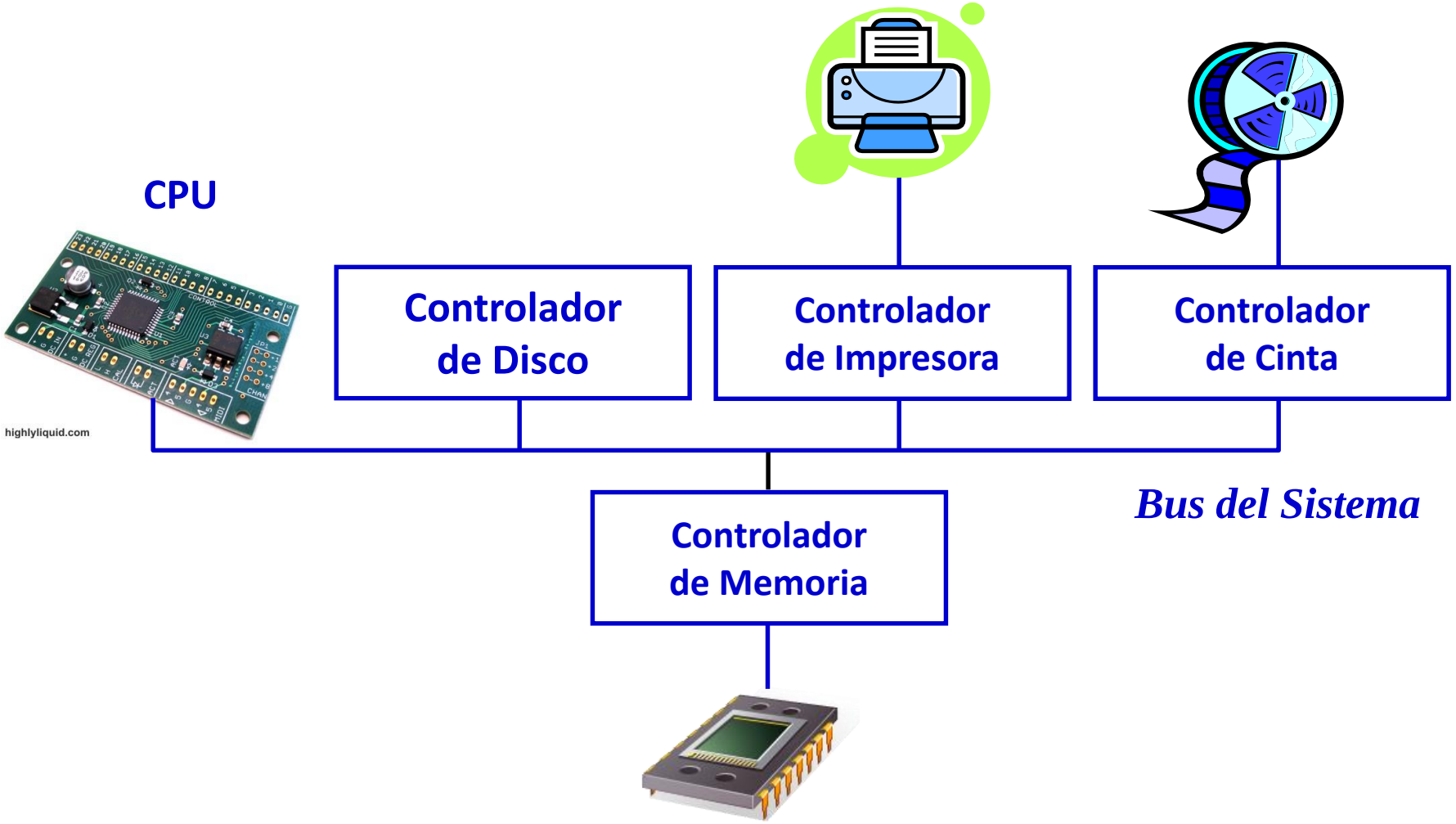
- Arquitecturas para Sistemas de Información

Sistemas Centralizado

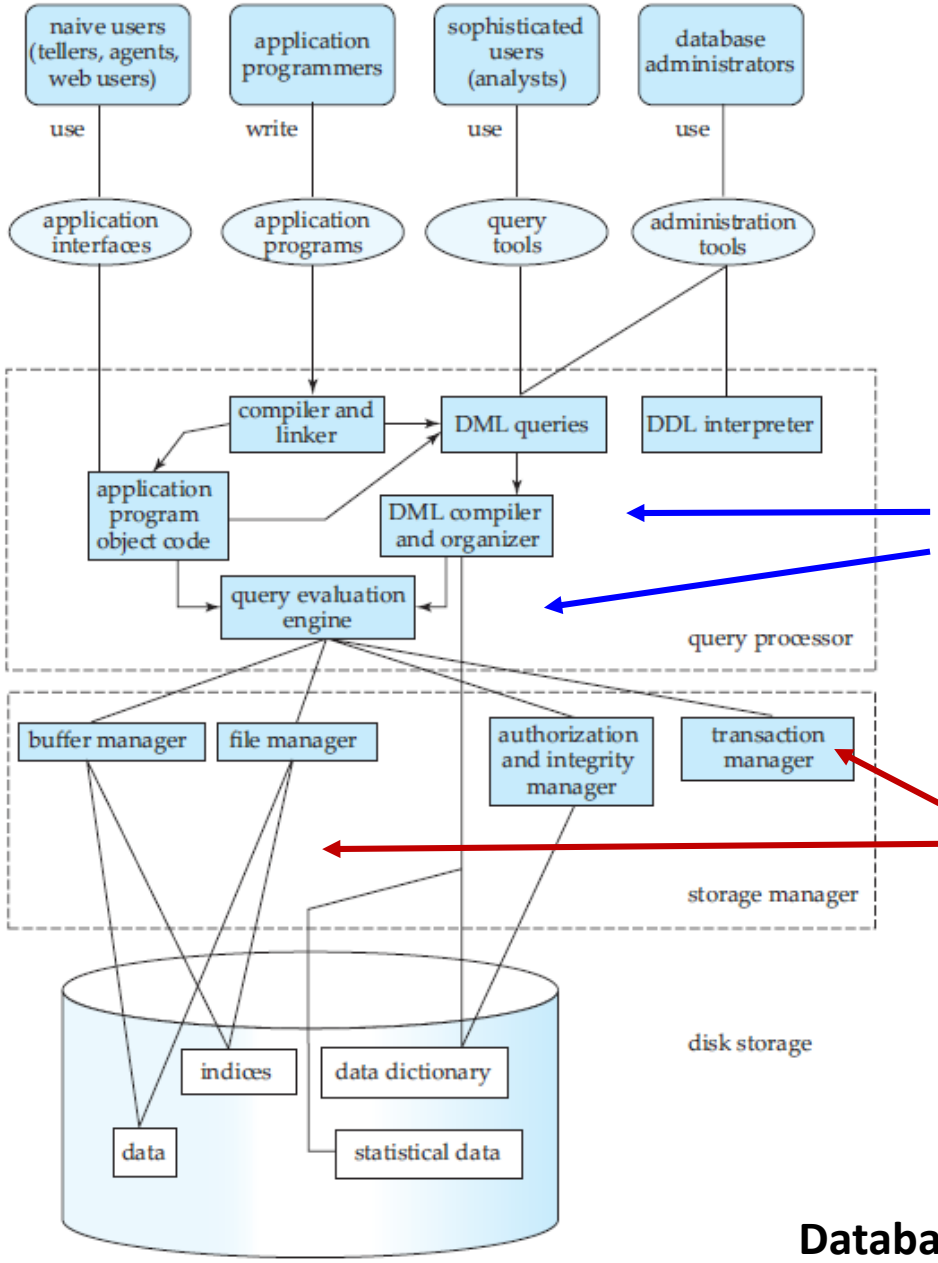
Considera una o unas pocas **CPU**, **memoria principal** y una serie de **controladores** conectados a través de un bus común que provee acceso a memoria compartida.

- Corre sobre una única computadora y no necesita interacción con otras.
- La CPU y los controladores ejecutan acciones de manera concurrente, compitiendo por el acceso a memoria.
- El uso de memorias caché reduce el número de accesos de la CPU a la memoria principal.
- Se distinguen los sistemas:
  - *mono-usuarios:* computadoras personales y workstation,
  - *multi-usuario.*

Sistema Centralizado



DBMS Centralizado



Procesador de Consulta

Manager Almacenamient.

Database Systems Concepts– A. Silberschatz

Figure 1.5 System structure.

Primeras Organizaciones Centralizados

Mono Usuario

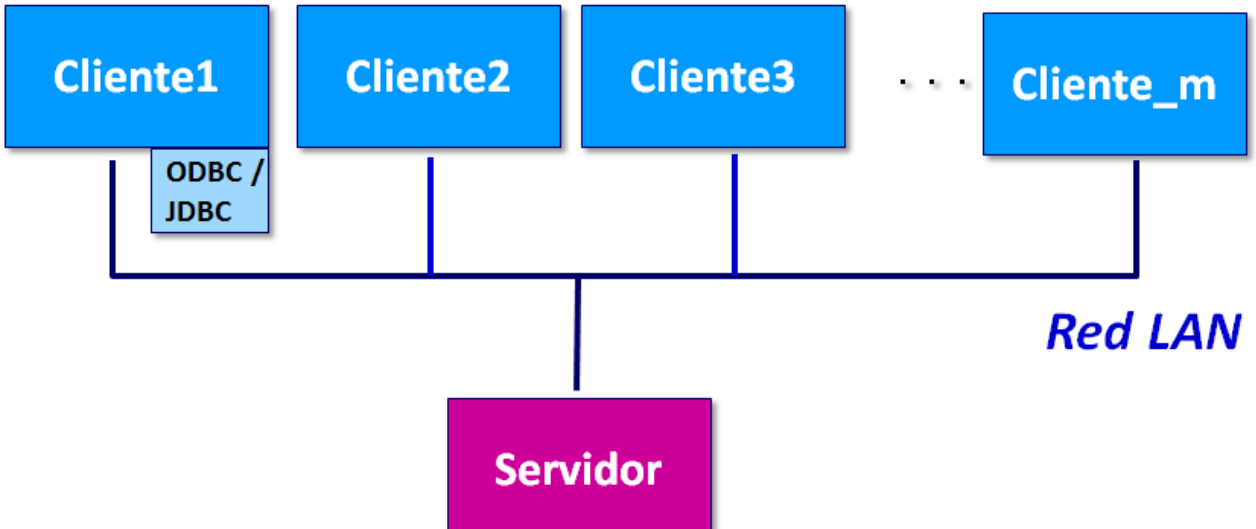
- Usados y administrados por una persona al mismo tiempo.
- Sistema operativo mono-usuario.
- No requieren control de concurrencia.
  - Ejemplos: Computadoras Personales, Estaciones de Trabajo (Workstations).

Multi Usuarios (mainfraimes)

- Atienden a varios usuarios que operan el sistema al mismo tiempo.
- Disponen de mayor capacidad de disco, memoria y CPU.
- El sistema operativo es multiusuario.
- Disponen de facilidades para multitareas.
- Los usuarios se conectan desde terminales.

Entorno Cliente-Servidor

- Es un tipo de organización ampliamente usada.
- Las computadoras personales (PC) remplazan a la terminal y asumen la responsabilidad de la funcionalidad asociada a la interface.
- El sistema actúa como un **sistema servidor** que satisface los requerimientos de los clientes.



DBMS y Cliente-Servidor

- Los **servicios del DBMS** se dividen en:
  - **Back-End (en el servidor):** manejar las estructuras de acceso, evaluación y optimización de consultas, control de concurrencia y recuperación ante fallos.
  - **Front-End (en el cliente)** abarcan herramientas como generadores de reportes y facilidades para el diseño de interfaces gráficas, herramientas de análisis y data mining.
- La interface entre back-end y front-end es a través de SQL o de los programas de aplicación.

Funcionalidades Front-End y Back-End

Front-End

- Interface de Usuario SQL
- Interface de Formularios
- Generador de Reportes
- Interface Gráfica

Interface (SQL + API)

- SQL Engine

Back-End

Arquitectura del Sistema Servidor

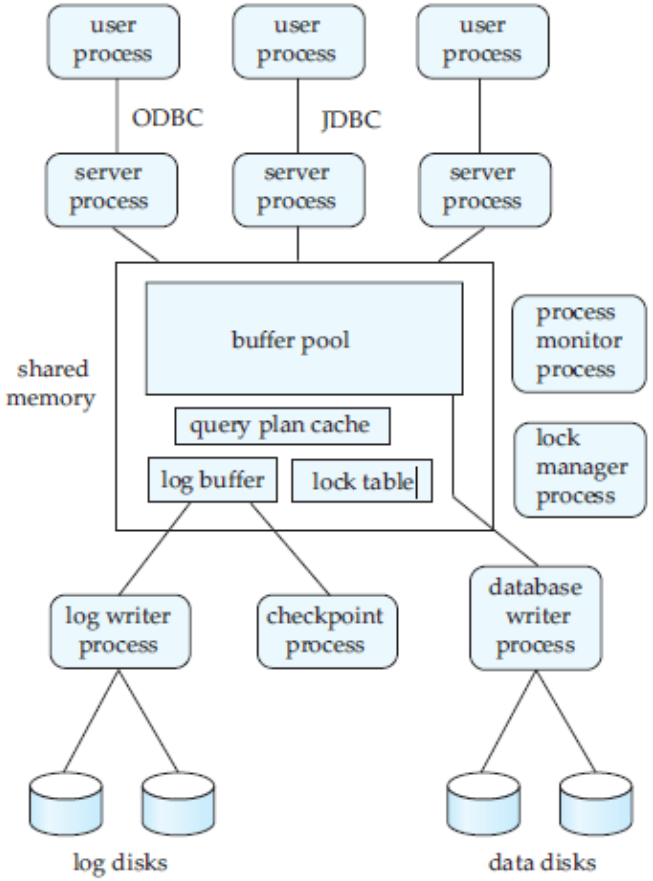
- **SISTEMA SERVIDOR DE TRANSACCIONES (o servidores de consulta):** proveen una interface para que el cliente les envíe requerimientos y retornar los resultados. La conexión con los clientes es vía ODBC

- **SISTEMA SERVIDOR DE DATOS:** los clientes interactúan con el servidor a través de requerimiento de lectura o escritura de datos, en unidad de archivo o páginas.

Una Arquitectura de Servidor de Transacciones

Procesos del servidor:

- Procesos de los clientes.
- Proceso lock manager.
- Proceso actualizador de datos
- Proceso actualizador registros log.
- Proceso de checkpointing.
- Proceso de monitoreo y recuperación.



Mainframe vs. Cliente-Servidor

Ventajas de las arquitecturas Cliente/Servidor

☺ Mejora la relación funcionalidad / costo.

☺ Es más simple de extender (*extensibles*).

☺ Mejor distribución de las estaciones de trabajo.

☺ Mejor integración con otras herramientas: formularios, integración con otras aplicaciones,

Sobre el almacenamiento de datos

Replicación – el sistema mantiene varias copias (réplicas) idénticas de un dato en varios nodos.

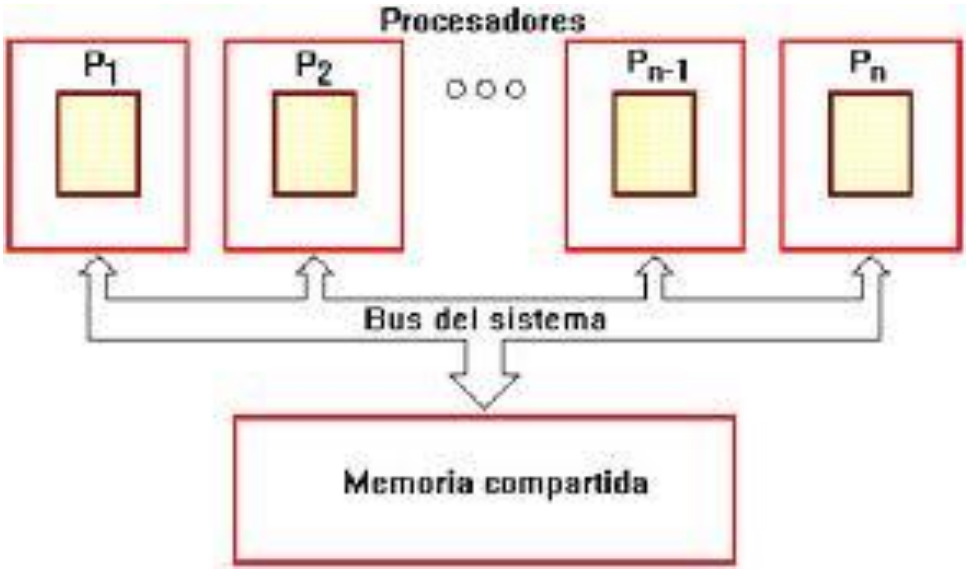
Mejora el tiempo de respuesta y tolerancia a fallos.

Fragmentación – una relación *r* se particiona en fragmentos almacenados en sitios diferentes.

Ejemplo cada sucursal mantiene datos de sus clientes

Replicación y Fragmentación – una relación *r* es particionada en varios fragmentos y el sistema mantiene varias copias idénticas de los fragmentos en distintos nodos.

Arquitecturas Paralelas



Arquitecturas Paralelas

Actualmente han resurgido las arquitecturas paralelas. Algunas de las razones:

- Han crecido los sistemas transaccionales y desde la Web surgen sitios con millones de usuarios que hacen que las compañías tengan **bases de datos cada vez más grandes**.
- Creció el interés y la demanda de sistemas de soporte de decisión que **consultan** y generen información de **bases de datos muy grandes (terabytes)**, que no tendrían tiempos de respuesta aceptables si se considera un único procesador.
- Los micro procesadores con nuevas **tecnologías multicore**.
- La naturaleza orientada a conjuntos de las consultas hace posible pensar en procesamiento paralelo.

Arquitecturas Paralelas

- Un servidor con arquitectura paralela considera:
  - **múltiples procesadores** y/o
  - **múltiples discos** y/o
  - **múltiples memorias**.
- El objetivo es mejorar tiempos de procesamiento y/o la velocidad de E/S.
- Para ellos, las operaciones se desarrollan en paralelo (simultáneamente).
- Surgen junto con la demanda de aplicaciones que usan bases de datos muy grandes (del orden de terabytes) y de miles de transacciones por minuto.

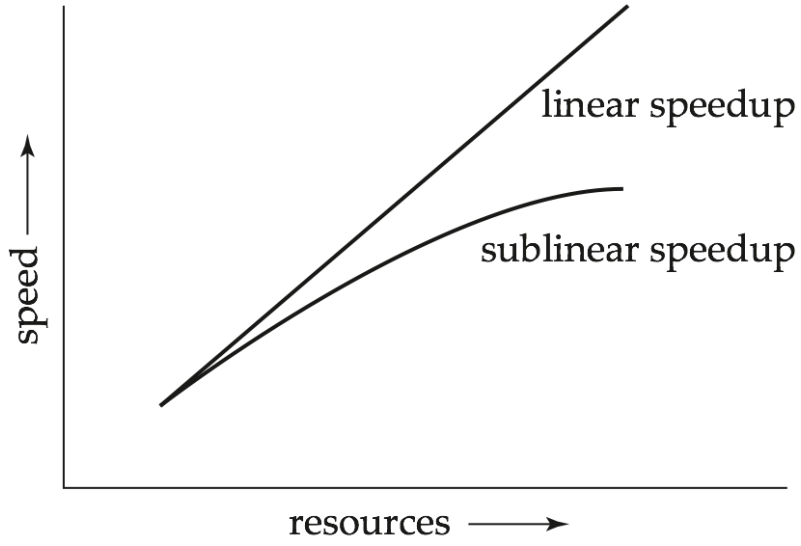
Sistemas Paralelos

- Dos organizaciones:
  - **Coarse-Grain** (granularidad gruesa): máquinas con un pequeño número de procesadores muy potentes.
  - **Massively parallel o Fine-Grain** (granularidad fina): máquinas con cientos de procesadores pequeños.
- Consideramos dos parámetros de performance para una arquitectura paralela
  - **Productividad** (Throughput): el número (cantidad) de tareas que se terminan en un período de tiempo.
  - **Tiempo de Retorno**: tiempo para completar una única tarea compleja.

Performance de Sistemas Paralelos

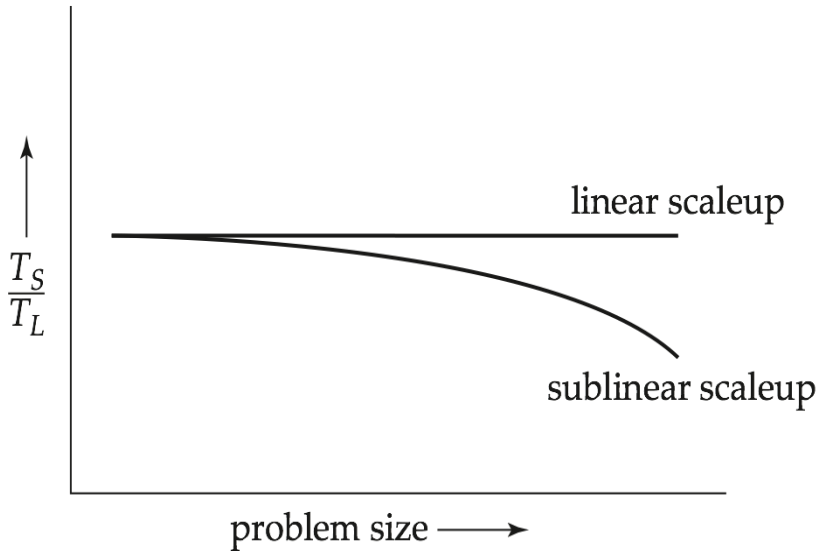
- Los sistemas paralelos pueden aumentar la productividad *procesando varias transacciones en paralelo*.
- El tiempo de retorno también se puede mejorar ya que pueden desarrollar *subtareas de una transacción en paralelo.*
- Cuestiones de estudio en sistemas paralelos:
  - **Speedup**: analiza el tiempo requerido por la misa tarea aumentando incrementa el paralelismo.
  - **Scaleup**: si se aumentan en proporción el problema y el sistema se mantiene el tiempo de respuesta.

Speedup



$$speedup = \frac{small\ system\ elapsed\ time}{large\ system\ elapsed\ time}$$

Scaleup



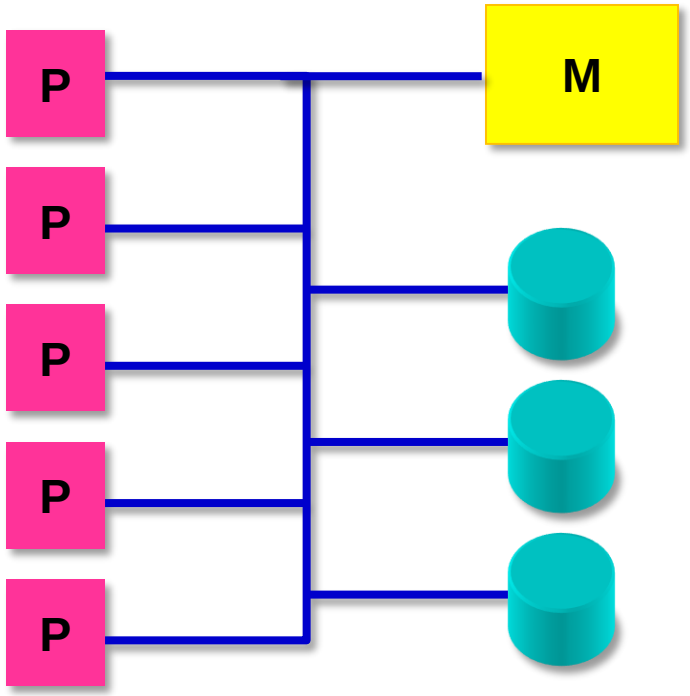
$$scaleup = \frac{small\ system\ small\ problem\ elapsed\ time}{big\ system\ big\ problem\ elapsed\ time}$$

Arquitecturas Paralelas

- Existen distintos modelos de arquitectura para máquinas paralelas:
  - **Memoria compartida**: todos los procesadores comparten una memoria común.
  - **Disco compartido**: todos los procesadores comparten un conjunto de discos en común (clusters).
  - **Nada compartido.**
  - **Jerárquico:** una combinación de los anteriores.

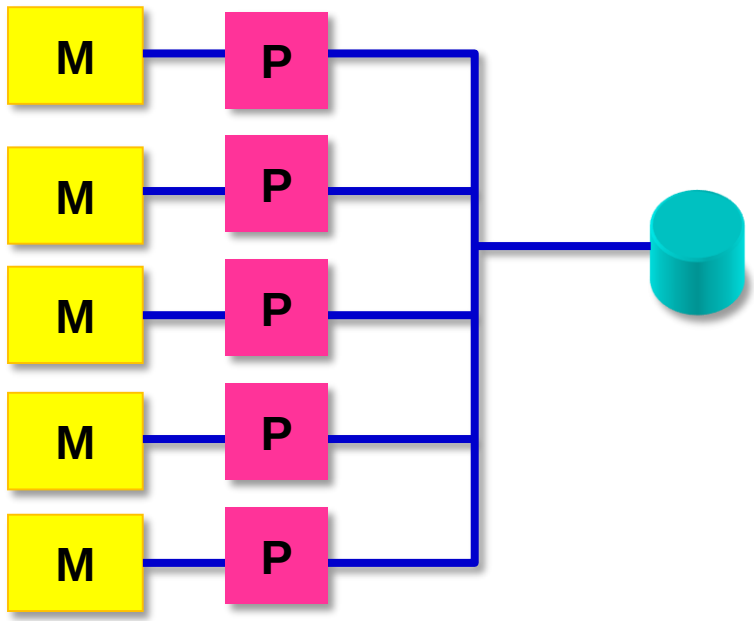
Arq. Paralelas – Memoria Compartida

- Procesadores y discos comparten la memoria en común vía bus de comunicación o interconexión de red.
  - **'+'** Es eficiente la comunicación entre procesadores, los datos compartidos pueden ser accedidos por cualquier procesador.
  - **'-'** No admite más de 32 a 64 procesadores ya que el bus de interconexión resulta un cuello de botella.

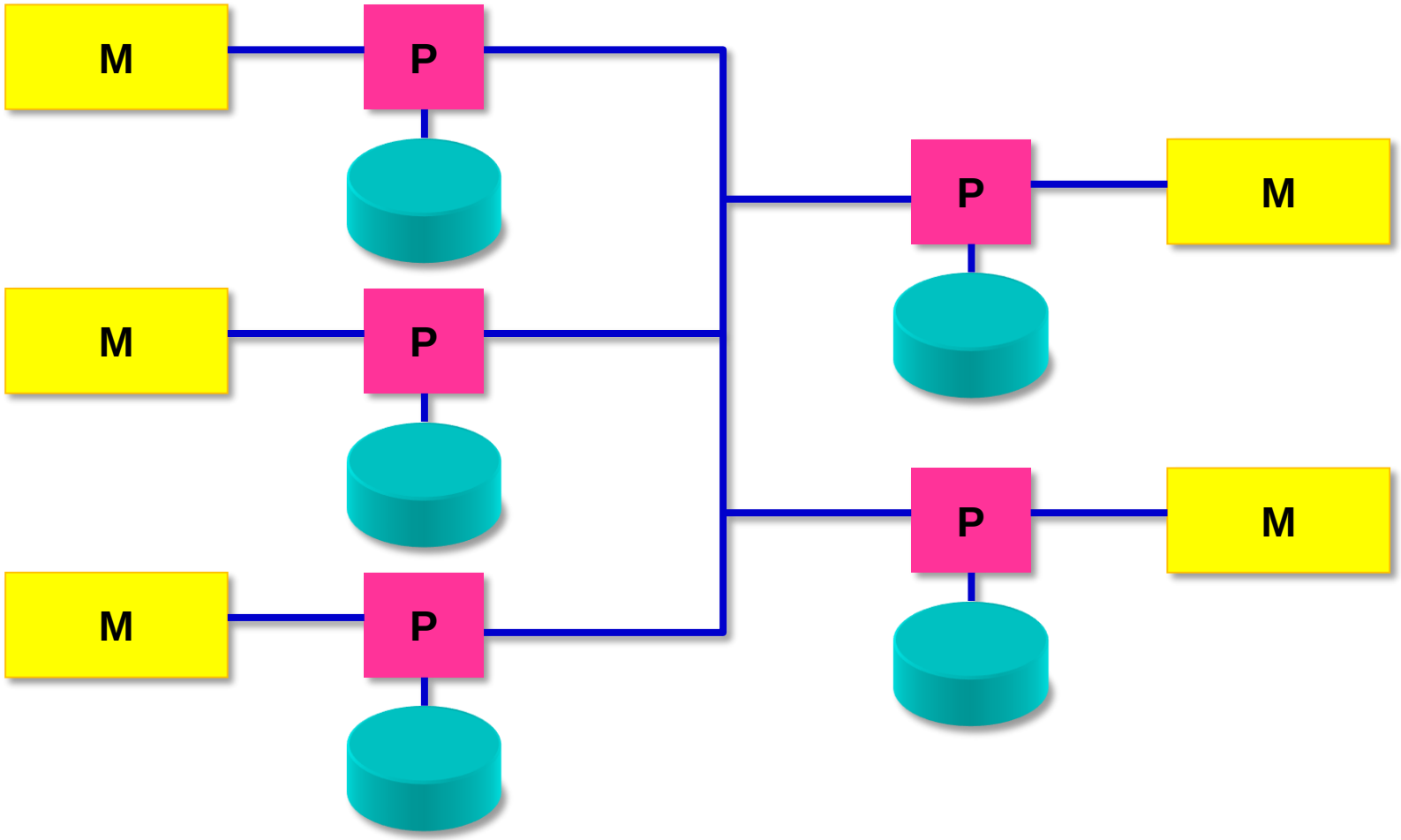


Arq. Paralelas – Disco Compartido

- Todos los procesadores comparten el medio de almacenamiento.
- Cada procesador cuenta con su memoria principal.
  - **'+'** el bus no es cuello de botella
  - **'+'** proporcionan tolerancia a fallas de procesador.
  - **'-'** el cuello de botella ocurre a nivel de conexión con el disco. Esto limita el crecimiento en cuanto a escalabilidad (*scaleup*).



Arq. Paralelas – Nada compartido

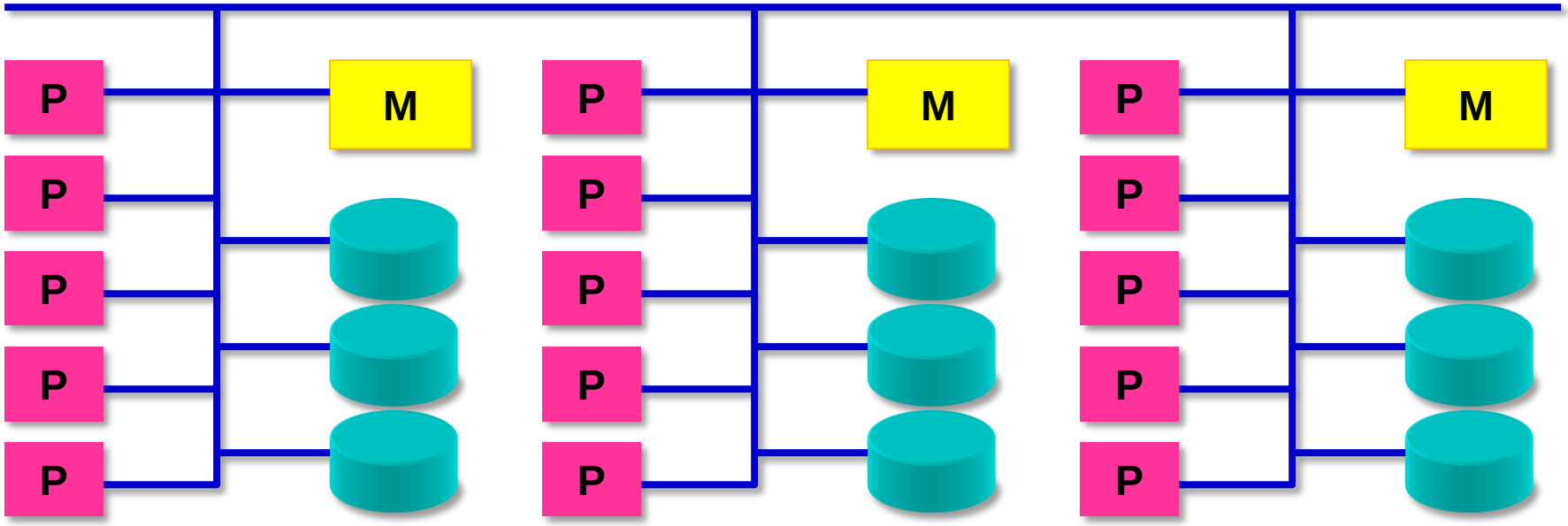


Arq. Paralelas – Nada compartido

- Cada nodo consiste de procesador, memoria y uno o más discos.
- La comunicación entre procesadores es a través de la red.
  - **'+'** Son ampliables y pueden soportar gran número de procesadores.
  - **'+'** minimiza las interfaces por recursos compartidos.
  - **'-'** La desventaja esta en el costo de comunicación y acceso a disco no locales.
  - Asume que la conexión entre nodos eficiente

Arq. Paralelas – Jerárquica

- Es un sistema híbrido entre memoria compartida, disco compartido y nada compartido.
- Las arquitecturas de memoria virtual distribuida han surgido de la idea de arquitecturas jerárquicas.



Arquitecturas Distribuidas

Organización para datos almacenados en varias computadoras (sitios o nodos) comunicadas entre si

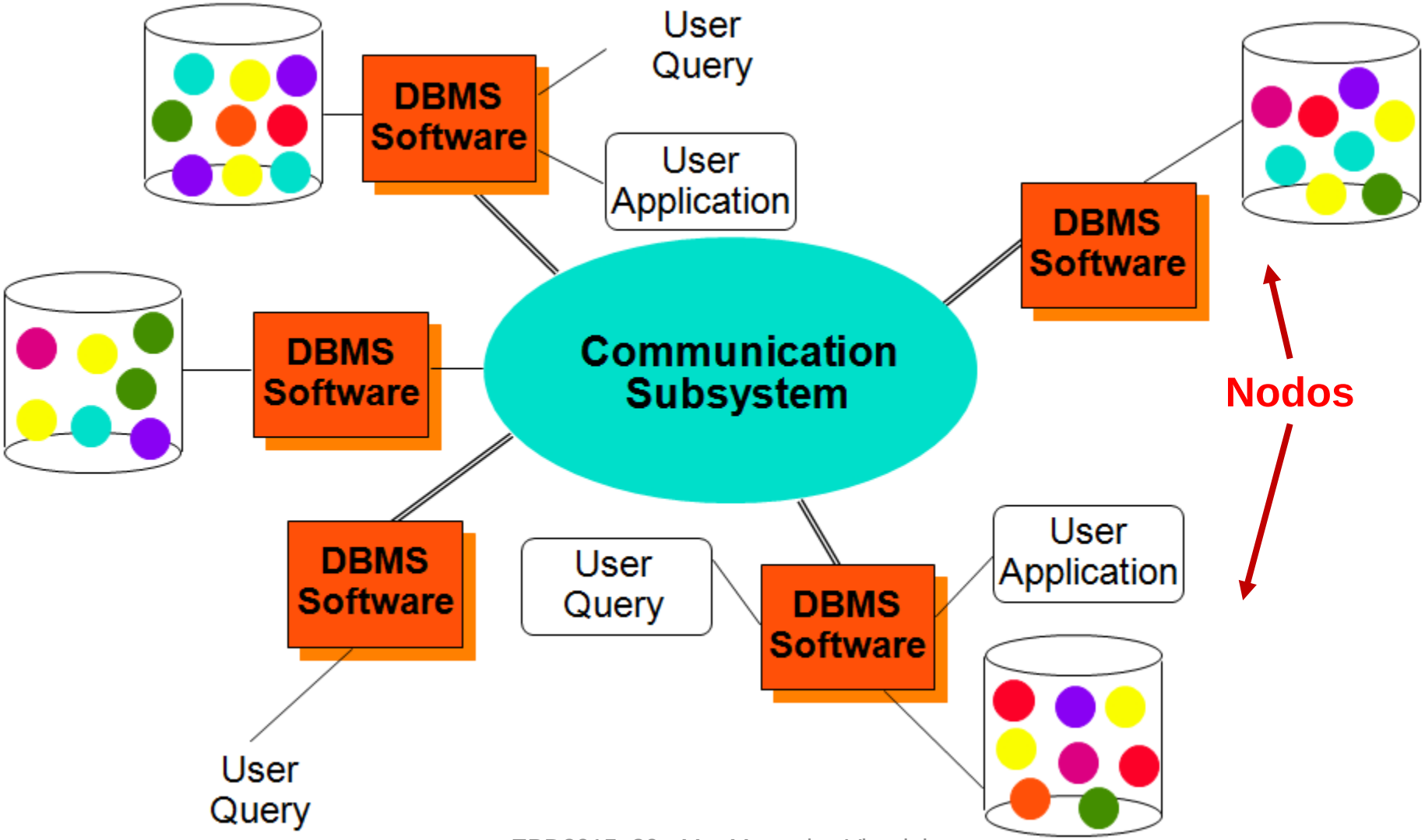
DBMS Distribuidos

En un *Sistema de Manejo de Base de Datos Distribuido* (*SMBDD o DDBMS*) **la base de datos se almacena en diversas computadoras, denominados sitios o nodos, comunicados entre si.**

Características

- Los nodos están en equipos que no comparten memoria ni disco y se conectan por la red.
- Los nodos generalmente se ubican geográficamente separados.
- Las computadoras pueden variar en tamaño y función.
- Las transacciones pueden acceder a datos que está físicamente en sitios distintos.

DBMS Distribuidos



Bases de Datos Distribuidas

- Bases de datos distribuidas **homogéneas**:
  - El mismo software/esquema en todos los sitios, los datos son particionados/replicados entre los sitios.
  - Objetivo: proveer la vista de una sola base de datos ocultando detalles de distribución.

- Bases de datos distribuidas **heterogéneas**:
  - Diferentes software/esquema en los sitios.
  - Objetivo: integrar distintas bases de datos para proveer una función útil.

Bases de Datos Distribuidas Homogéneas

BD Distribuidas Homogéneas – *características:*

- Todos los sitios tienen software de base de datos idéntico.
- Cada sitio sabe de la existencias de los otros y están de acuerdo en cooperar con la atención de los requerimientos de usuarios.
- Cada sitio sacrifica autonomía en beneficio de conservar un esquema idéntico.
- Para los usuarios es como si existiera un único sistema.

Bases de Datos Distribuidas Heterogéneas

BD Distribuidas Heterogéneas – *características:*

- Sitios distintos pueden usar esquemas y software de bases de datos distintos:
  - Diferencias en los esquemas afectan al procesamiento de consultas.
  - Diferencias en el software afectan principalmente al problema de procesamiento.
- Los sitios pueden no saber de la existencia de los otros y solo proveer facilidades acotadas de cooperación en procesamiento de transacciones.

Sistemas Distribuidos Homogéneos

Características

- Cada sitio conoce la existencia de los otros.
- Cada sitio provee un entorno para ejecutar tanto transacciones locales como globales.

TRANSACCIONES LOCALES Y TRANSACCIONES GLOBALES

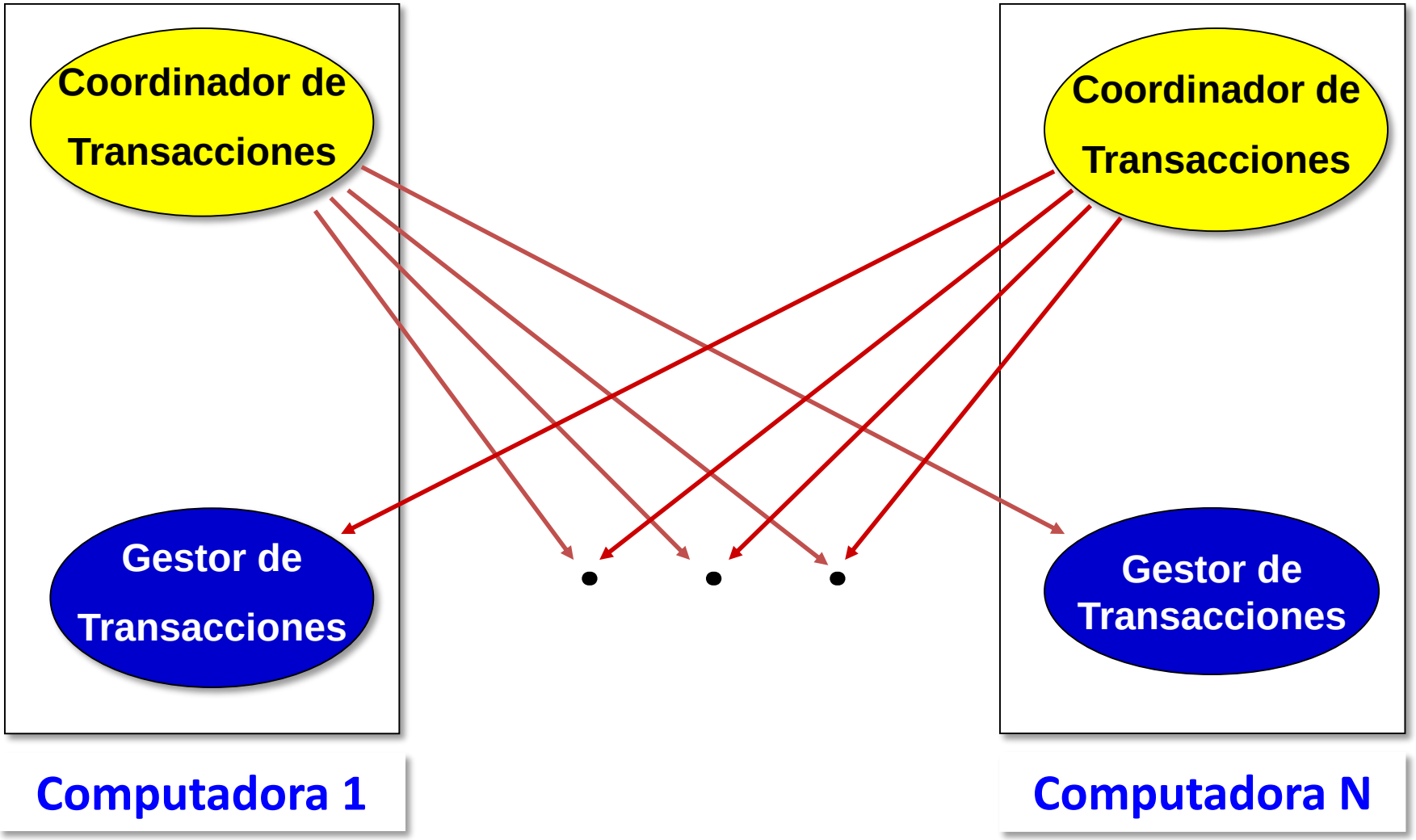
- **Una transacción local** accede a datos de un único sitio donde se inició la transacción.
- **Una transacción global** accede tanto a datos del sitio donde se originó como a datos de varios sitios diferentes.

Modelo de Transacciones Distribuido

Cada sitio cuenta con dos subsistemas:

- **El gestor de transacciones**: gestiona la ejecución de aquellas transacciones (o subtransacciones) que acceden a los datos almacenados en el sitio local.
  - Mantener la bitácora.
  - Participar en el esquema de control de concurrencia del sitio local.
- **El coordinador de transacciones**: encargado de la ejecución de las transacciones (locales y globales) iniciadas en el sitio local.
  - Iniciar las transacciones.
  - Dividir las transacciones en subtransacciones.
  - Coordinar la ejecución de subtransacciones en los distintos sitios.

Modelo Transaccional Distribuido Homogéneo



Sistemas Distribuidos - Ventajas

- **Datos compartidos –** los usuarios de un sitio pueden acceder a datos de otros sitios.
- **Autonomía –** cada sitio es capaz de mantener el control de los datos que están almacenados localmente. Existe un administrador global del sistema y cada sitio cuenta con un administrador local.
- **Mayor Disponibilidad –** si un sitio falla, el resto de los sitios puede continuar operando. Si la información se replica, el fallo de un sitio puede no afectar a todo el sistema.

Sistemas Distribuidos - Desventajas

- **Costo del Desarrollo de Software –** es más difícil y costoso implementar un sistema de base de datos distribuido.
- **Mayor Probabilidad de Errores –** puesto que los sitios que constituyen el sistema distribuido operan en paralelo, es más difícil asegurar la correctitud de los algoritmos.
- **Incremento en la Sobrecarga de Procesamiento –** el intercambio de mensajes y los cómputos adicionales para coordinar los sitios constituyen una sobrecarga importante.

Sistemas Distribuidos vs. Paralelos

- Los sistemas distribuidos se asemejan a los sistemas paralelos con estructura nada compartido.
- Sin embargo, generalmente están en lugares diferentes (geográficamente separados), son administradas independientemente y tienen una interconexión más lenta.
- Otra diferencia es el tipo de transacciones que admiten. Las *transacciones locales* acceden a datos de un único sitio mientras que las *transacciones globales* pueden acceder a datos que están en sitios diferentes a donde se inicia la transacción.

Tipos de Redes

- Las bases de datos distribuidas y los sistemas cliente/servidor se construyen sobre la base de **redes de comunicación.**
- La configuración y conectividad de la red influye en el comportamiento del sistema
- Existen dos tipos de redes:
  - **Redes de área local (LAN):** compuesta de procesadores distribuidos en áreas geográficas pequeñas.
  - **Redes de área amplia (WAN):** compuestas por procesadores distribuidos en áreas geográficas mayores.
- Entre ellas varían en velocidad y confiabilidad.

Diferentes Topologías de Redes

- **Red Totalmente Conectada**: existe un vínculo entre cada par de nodos.
- **Red Parcialmente Conectada**: algunos pares de nodos no están conectados.
- **Red Tipo árbol**: existe una estructura arbórea con un nodo raíz, sus hijos. Cada nodo puede tener 0 o más nodos hijos pero sólo un nodo padre.
- **Red Estrella:** existe un nodo al cual todos están conectados.
- **Red Anillo:** los nodos están conectados circularmente.

Arquitecturas para Sistemas de Información

Estilos Arquitectónicos para SI asistidos por SMBD

Sistemas de Información

Sistemas Información (SI): dado un negocio, la base de datos es apenas una parte de la estructura total, que considera:

- **Datos**: información persistente.
- **Procesos**: que manipulan los datos.
- **Personas**: que llevan adelante los procesos
- **Redes**: que transportan los datos dentro de una organización y fuera de ella.
- **Hardware**: CPU, discos, impresoras, etc.
- **Software de base**: sistema operativos, DBMS, programas de aplicación, etc.

Sistemas de Información

La **organización lógica** típica en "capas" o niveles de un SI identifica:

- **Capa de presentación (PL) :** se encarga de aspectos de presentación e interacción con el usuario.
- **Capa de Lógica de Negocio (BL):** se ocupa de representar a los conceptos y funcionalidades del negocio.
- **Capa de Acceso a Datos (DAL):** administra las conexiones con la capa de datos.
- **Capa de Datos (DL):** concentra la persistencia de datos en la base de datos

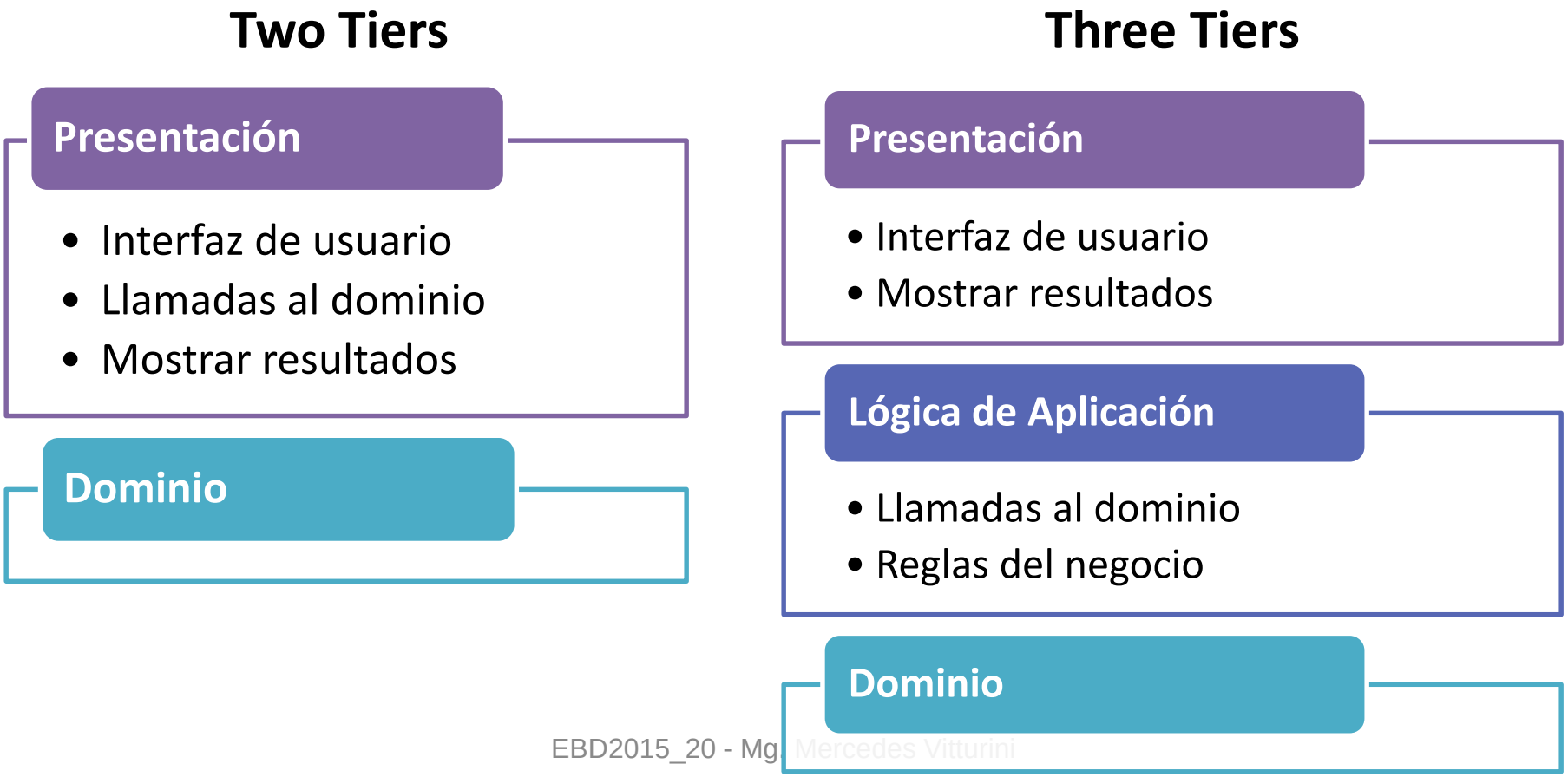
Organización Lógica en Capas

Componente Lógico	Responsabilidades
Capa de Presentación (PL) – Procesos de interfaz de usuario	Interactuar con distintos tipos de usuario. Interactuar con otras aplicaciones. Validación básica de las entradas al sistema. Presentar los resultados.
Capa de Negocio (BL) – Procesos de Negocio	Realizar cálculos y procedimientos de negocio. Hacer ejecutar las reglas de negocio. Representar entidades del negocio.
Capa de Acceso a Datos (DAL) – Procesos de acceso a datos	Proveer acceso a datos. Administrar las conexiones. Proveer seguridad en el acceso.
Capa de Datos (DL) – Procesos de datos	Almacenamiento y manipulación de datos (persistencia). Hacer cumplir las restricciones sobre los datos.

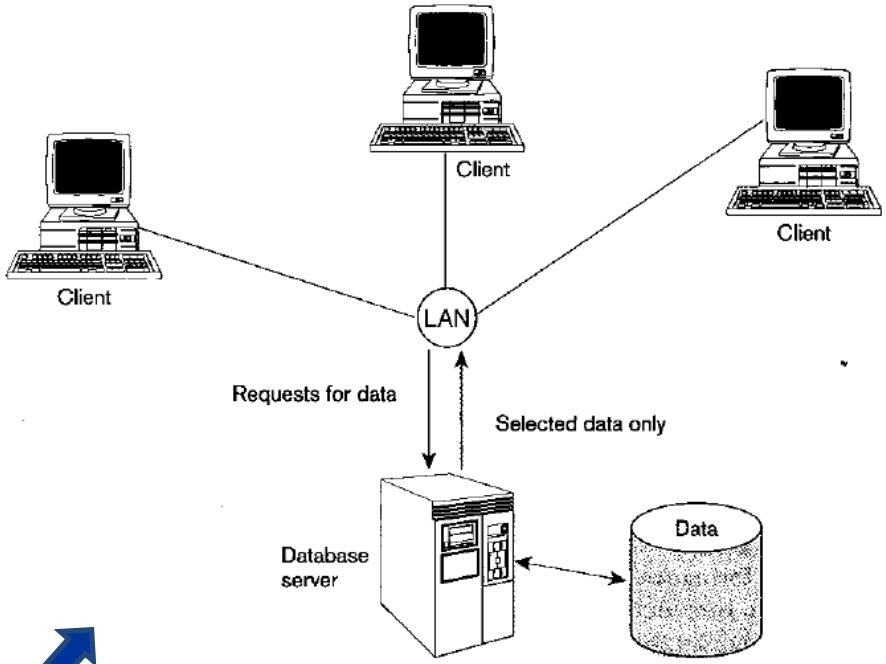
Tiers y Layers

Layer (capa): separación lógica del sistema.

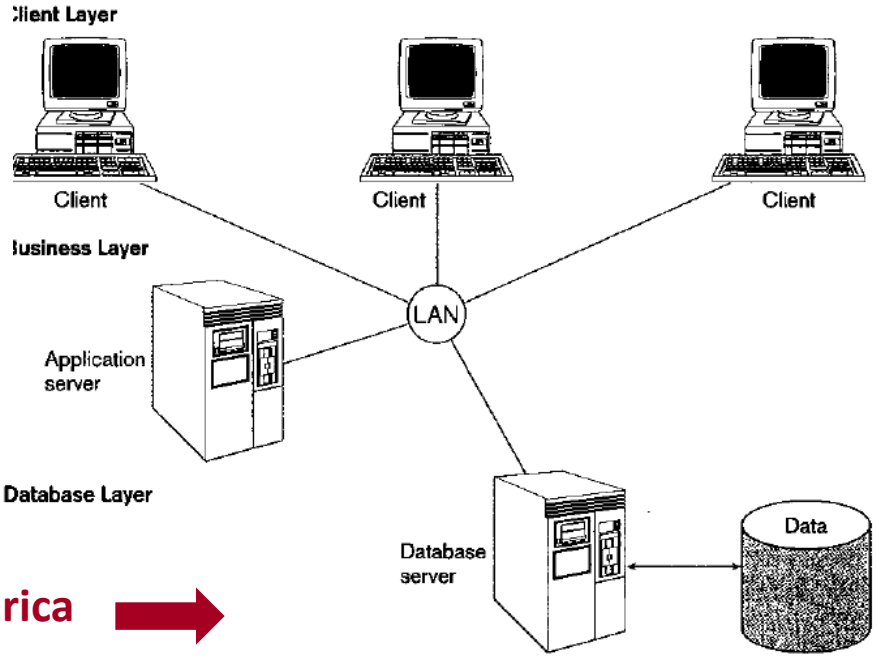
Tier: separación física del sistema (servidores).



Arquitecturas Cliente-Servidor



Arquitectura Servidor de BD Two Tier



Arquitectura Three Tier Genérica

Temas de la clase de hoy

- DBMS: Arquitecturas
  - Sistemas centralizados. Cliente Servidor.
  - Sistemas Paralelos
  - Sistemas distribuidos.

- **Bibliografía**
  - “Database Systems Concepts” – A. Silberschatz. Capítulos 20 y 22 (ed. 2005) / Capítulos 17 y 19 (ed. 2010).
  - “DataBase System – The Complete Book” – H. Molina, J. Ullman. Capítulo 20.
  - “Principles of Database and Knowledge-Base Systems” – J. Ullman. Capítulo 10.