

16 ARQUITECTURAS DE SISTEMAS DE BASES DE DATOS

En la arquitectura de un sistema de bases de datos se reflejan aspectos como

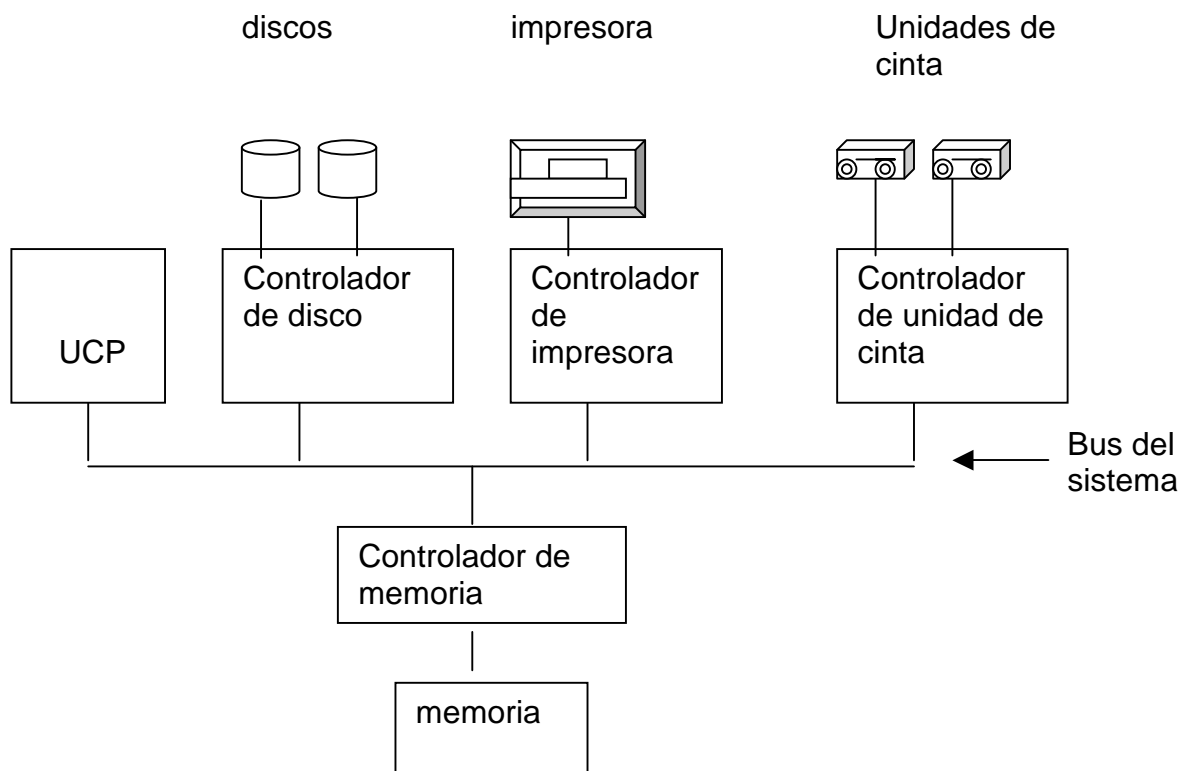
conexión en red (permite que unas tareas se ejecuten en un sistema servidor y otras se ejecuten en los sistemas clientes)

paralelismo (permite acelerar las actividades del sistema de base de datos, proporcionando a las transacciones unas respuestas más rápidas)

distribución (permite que los datos sean guardados donde han sido generados, pero continuar siendo accesible desde otros lugares o departamentos)

16.1 SISTEMAS CENTRALIZADOS

- ⇒ Se ejecutan en un único sistema informático sin interaccionar con ninguna otra computadora.
- ⇒ Una computadora moderna de propósito general consiste en una o unas pocas unidades centrales de procesamiento y un número determinado de controladores para los dispositivos que se encuentren conectados a través de un bus común, que proporciona acceso a la memoria compartida.



- ⇒ Las UCP poseen memorias caché locales donde se almacenan copias de ciertas partes de la memoria para acelerar el acceso a datos.

SISTEMAS DE GESTION DE BASES DE DATOS

- ⇒ Cada controlador de dispositivo se encarga de un tipo específico de dispositivos (unidades disco, monitor, tarjeta sonido etc.)
- ⇒ La UCP y los controladores de dispositivo pueden ejecutarse concurrentemente, lo que les lleva a competir por el acceso a memoria. La memoria caché de la UCP reduce esta disputa al necesitar acceder a la memoria compartida un número inferior de veces.
- ⇒ Se distinguen dos formas de utilizar las computadoras:

1. Sistemas monousuario (computadoras personales, estaciones de trabajo)

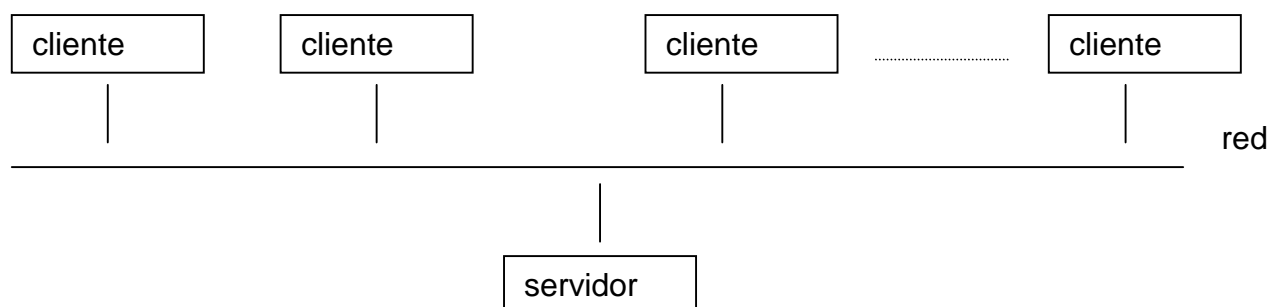
- ❖ Típicamente está compuesto por una unidad de sobremesa utilizada por una única persona que dispone de una sola UCP, uno o dos discos fijos y que trabaja con un sistema operativo que sólo permite un único usuario.
- ❖ No tienen control de concurrencia (no es necesario)
- ❖ No tienen facilidad de recuperación.
- ❖ La mayoría no admiten SQL y proporcionan un lenguaje de consulta muy simple (en algunos casos, variante de QBE)

2. Sistemas multiusuario (sistemas *servidores*)

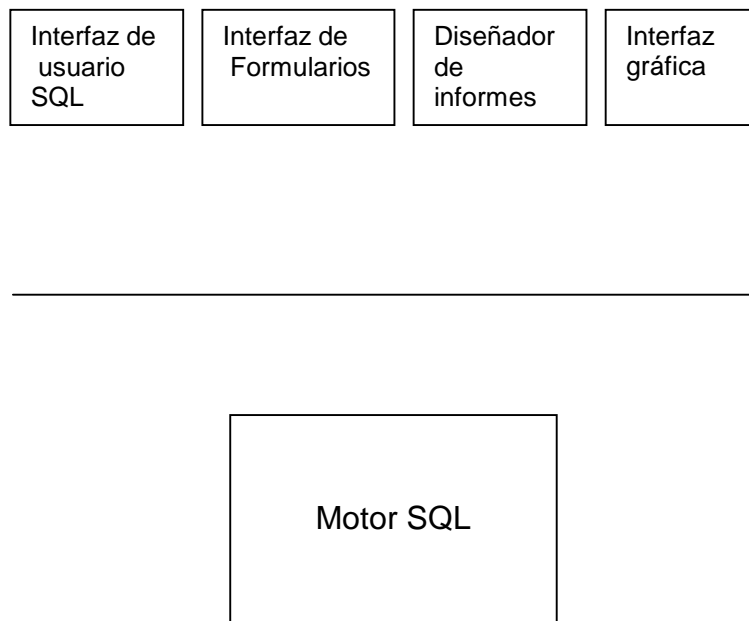
- ❖ Tiene más discos y más memoria. Puede disponer de varias UCP y trabaja con un sistema operativo multiusuario.
- ❖ Se encarga de dar servicio a un gran número de usuarios que están conectados al sistema a través de terminales.
- ❖ Sí tienen control de concurrencia.

16.2 SISTEMAS CLIENTE-SERVIDOR

- ⇒ Dado que las computadoras personales son más potentes, rápidas y baratas, los terminales conectados a un sistema central han sido sustituidos por computadores personales, siendo gestionada por éstos la interfaz con el usuario en lugar de hacerlo el sistema central.
- ⇒ Por tanto, los sistemas centralizados actúan como sistemas servidores que satisfacen las necesidades generadas por los sistemas clientes



⇒ La funcionalidad de una base de datos se divide en 2 partes:



1. Parte visible al usuario:

Está formada por herramientas como *formularios*, *diseñadores de informes* y facilidades gráficas de interfaz de usuario.

La **interfaz** entre la parte visible al usuario y el sistema subyacente puede ser SQL o una aplicación

2. Sistema subyacente:

Es el que gestiona el acceso a las estructuras, la evaluación y optimización de consultas, el control de concurrencia y la recuperación

Sist. servidores pueden dividirse en servidores de transacciones y servidores de datos

16.2.1 **Servidores de Transacciones (o servidores de consulta)**

- ❖ La arquitectura del servidor de transacciones responde a la división funcional entre la parte visible al usuario y el sistema subyacente (al contrario que los sistemas centralizados, donde no hay esta división)
- ❖ **Los clientes envían transacciones a los sistemas servidores, que se encargan de ejecutarlas y enviar de los resultados de vuelta a los clientes, que se encargan a su vez de mostrar los datos en pantalla**
- ❖ La norma **ODBC** (Conectividad abierta de bases de datos) es una interfaz de aplicación que permite que los clientes generen instrucciones SQL para enviarlas al servidor donde se ejecutan, es decir, permite que cualquier cliente puede conectarse a cualquier servidor sea del tipo que sea, siempre que ambos posean la interfaz ODBC.
- ❖ Existen también otras interfaces de programación de aplicaciones que utiliza el cliente para realizar llamadas a **procedimientos remotos de transacciones** sobre el servidor. Estas llamadas, aunque para el programador sean como llamadas normales a procedimientos, todas las llamadas a procedimientos remotos desde un cliente se engloban en una única transacción en el servidor. Así, si la transacción aborta, el servidor puede deshacer los efectos de las llamadas a procedimientos remotos individuales.
- ❖ Debido al abaratamiento de los ordenadores personales, cada vez se prefiere más esta arquitectura de tener conectados varios ordenadores personales a un sistema subyacente, lo que conlleva entre otras, las siguientes **ventajas**
 - una mejor funcionalidad respecto al coste (fraccionamiento de costes),
 - una mayor flexibilidad para localizar recursos y facilidades de expansión
 - mejor interfaces de usuario
 - mantenimiento más sencillo

16.2.2 Servidores de datos

- ❖ Se utilizan en redes de área local en las que se alcanza una alta velocidad de conexión entre los clientes y el servidor.
- ❖ ***Se envían los datos a las máquinas cliente, se realiza allí todo el procesamiento (que puede durar un tiempo) y después se envían los datos de vuelta al servidor.***
- ❖ Esta arquitectura necesita que los clientes posean todas las funcionalidades del sistema subyacente.
- ❖ Se utiliza mucho en los sistemas de bases de datos orientadas a objetos.
- ❖ Dado que el coste en tiempo de comunicación entre el cliente y el servidor es alto comparado al de acceso a una memoria local (milisegundos frente a menos de 100 nanosegundos), se utilizan diversas **ESTRATEGIAS PARA MINIMIZAR** este tiempo de comunicación:

- **Envío de páginas o envío de elementos**

Cuando se necesita un elemento (bien una página, que es de grano grueso, o una tupla, de grano fino) se envían además otros elementos que probablemente vayan a ser utilizados en un futuro próximo (preextracción)

- **Bloqueo:**

El problema de la preextracción reside en que el cliente adquiere también el bloqueo de todos los elementos preextraídos, incluso aunque no se esté accediendo a ellos, impidiendo innecesariamente el procesamiento de otros clientes. Esto se evita con las técnicas de liberación de bloqueos

- **Caché de datos**

Los datos que se envían al cliente a favor de una transacción se pueden alojar en una caché del cliente, si dispone de suficiente espacio libre. Pero hay que tener una buena comunicación con el servidor para asegurarse que los datos de la caché del cliente están al día.

- **Caché de bloqueos.**

Igualmente se pueden almacenar los bloqueos en la caché del cliente, pues es bastante difícil que un cliente necesite los datos que está utilizando otro cliente, pero si ello ocurre, el servidor debe comunicar a todos los clientes los bloqueos sobre el elemento de datos que se encuentren en las memorias caché de otros clientes. Esta tarea es bastante complicada si además se tiene que tener en cuenta los posibles fallos de la máquina.

16.3 SISTEMAS PARALELOS

- ⇒ Los sistemas paralelos mejoran la velocidad de procesamiento y de E/S mediante la utilización de UCP y discos en paralelo.
- ⇒ En el procesamiento paralelo se realizan muchas operaciones simultáneamente, mientras que en el procesamiento secuencial los distintos pasos computacionales han de ejecutarse en serie.
- ⇒ Una máquina **paralela de grano grueso** consiste en un pequeño número de potentes procesadores.
- ⇒ Una máquina **masivamente paralela o de grano fino** utiliza miles de procesadores más pequeños. Son capaces de soportar un grado de paralelismo mayor que las de grano grueso
- ⇒ Para medir el **rendimiento** de un sistema de bases de datos existen dos medidas principales:

SISTEMAS DE GESTION DE BASES DE DATOS

- ❖ La **productividad**: Número de tareas que pueden completarse en un intervalo de tiempo
- ❖ El **tiempo de respuesta**: Cantidad de tiempo que necesita para completar una única tarea a partir del momento en que se envíe.

Un sistema que procese un gran número de pequeñas transacciones, puede aumentar su productividad realizando muchas transacciones en paralelo).

Un sistema que procese transacciones largas, puede mejorar el tiempo de respuesta, así como la productividad, realizando en paralelo las distintas subtareas de cada transacción

16.3.1 **Ganancia de velocidad y ampliabilidad**

mediante el incremento del grado de paralelismo.

Ganancia de Velocidad: Ejecución en menos tiempo de una tarea : T_P/T_G

T_P : Tiempo que tarda **una tarea** en ejecutarse en un sistema paralelo con P recursos

T_G : Tiempo que tarda **misma tarea** en ejecutarse en sistema paralelo con G recursos (N veces mayor que P)

Ampliabilidad: Manejo de transacciones más largas : T_P/T_G

T_P : Tiempo que tarda **una tarea Q** en ejecutarse

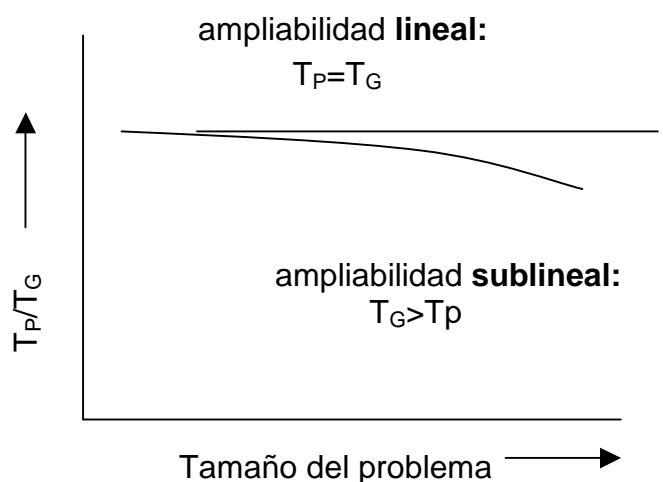
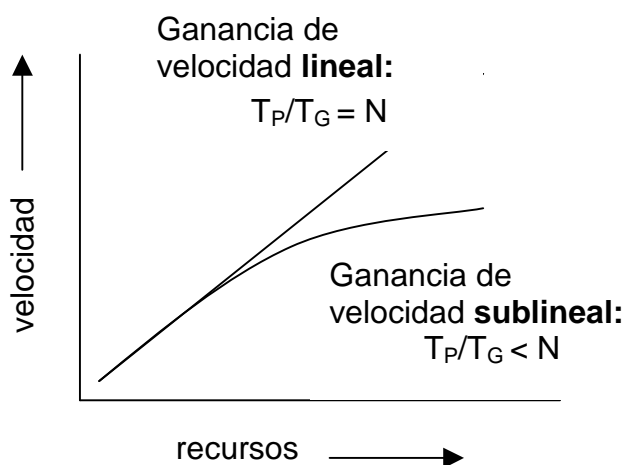
T_G : Tiempo que tarda **una tarea Q_N** en ejecutarse (N veces mayor que Q)

Según sea la forma de medir el tamaño del problema, tenemos:

- **Ampliabilidad Por lotes:** Aumenta el tamaño de la base de datos, y las tareas son trabajos más largos cuyos tiempos de ejecución dependen del tamaño de la base de datos
- **Ampliabilidad De transacciones:** Aumenta la velocidad con la que se envía las transacciones a la base de datos

Ganancia de velocidad respecto al incremento de recursos

Ampliabilidad respecto al crecimiento del tamaño del problema



- ❖ La **ampliabilidad** es el factor más importante para medir la **eficiencia** de un sistema paralelo de bases de datos
- ❖ Factores en **contra eficiencia** del paralelismo:

- **Costes de inicio**, si tiempo de inicio >>> tiempo real procesamiento (incide en la ganancia de velocidad)
- **Interferencia**, al haber competencia entre los procesos por acceder a los recursos compartidos (incide tanto en la ganancia de velocidad como en la ampliabilidad)
- **Sesgo**, al no poder dividir una tarea en partes exactamente iguales (distribución sesgada) y tener que medir el tiempo de servicio de esa tarea completa por la de su paso más lento (incide en la ganancia de velocidad)

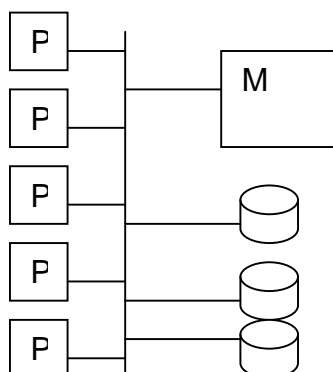
16.3.2 *Redes de interconexión*

Es el medio para comunicarse entre sí un conjunto de componentes (procesadores, memoria, disco).

- **Bus:** (Ethernet o una interconexión paralela)
Todos los componentes pueden enviar o recibir datos de un único bus
Trabajan bien sólo para pocos componentes
- **Malla:**
Cada componente está conectado con todos los nodos adyacentes.
Bidimensional: Con 4 nodos adyacentes
Tridimensional: Con 6 nodos adyacentes
Si no existe conexión directa entre los nodos, pueden comunicarse a través de mensajes a través de nodos intermedios
Un componente puede estar a $\sqrt{n}-1$ ó $\sqrt{n}/2$ enlaces de otros componentes
Su capacidad de comunicación es mayor cuanto más aumenta el paralelismo
- **Hipercubo**
Se asigna a cada componente un n^o binario, tal que dos componentes tienen una conexión directa si sus correspondientes representaciones binarias difieren en un solo bit.
Cada uno de los N componentes está conectado con otros $\log(N)$ componentes
El n^o máximo de enlaces que atraviesa un mensaje para comunicarse entre dos componentes cualquiera es $\log(N)$
El retardo de comunicación es significativamente menor que en una malla

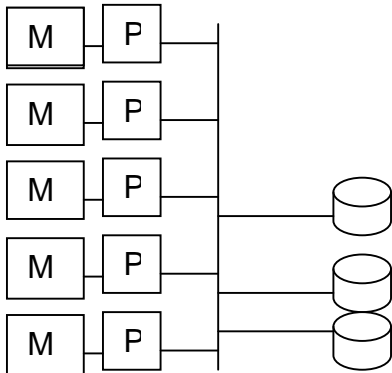
16.3.3 *Arquitecturas paralelas de bases de datos:*

16.3.3.1 *Memoria compartida*



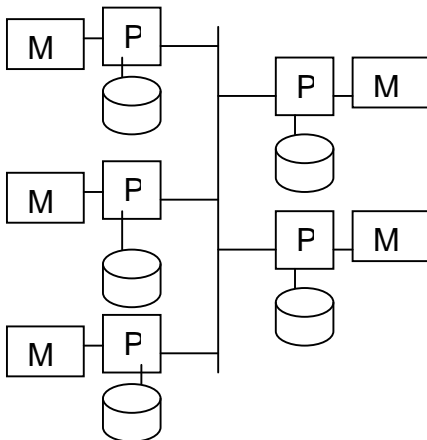
- ⇒ Todos los procesadores comparten una memoria común
- ⇒ Beneficio → Mayor velocidad de comunicación al no tener que recurrir al software
- ⇒ Inconveniente → El bus se convierte en un Cuello de botella (aunque estén dotados de memoria caché)

16.3.3.2 Disco compartido , agrupaciones



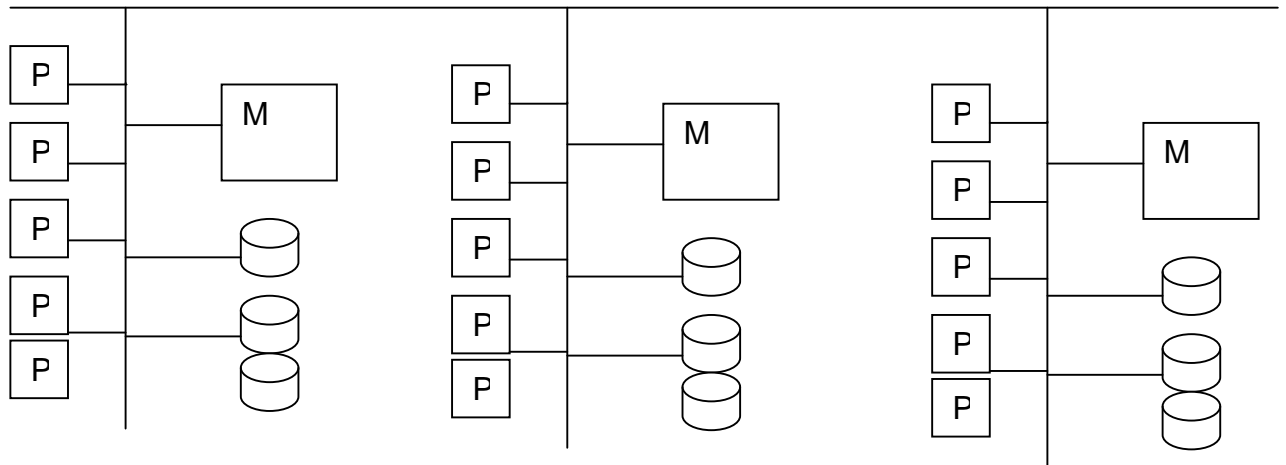
- ⇒ Todos los procesadores comparten un disco común
- ⇒ Ventajas respecto a la memoria compartida →
 - El bus no se convierte en cuello de botella, al tener cada procesador memoria propia
 - Forma barata *tolerancia ante fallos* : Si un procesador (o su memoria) falla, los demás procesadores pueden hacerse cargo de sus tareas, ya que la base de datos reside en los discos, a los que tienen acceso todos los procesadores.
- ⇒ Problema principal → La ampliabilidad: la interconexión con el subsistema de discos se convierte ahora en el nuevo cuello de botella.
- ⇒ Oracle Rdb utiliza ésta arquitectura

16.3.3.3 Sin compartimiento



- ⇒ Cada nodo de la máquina consta de un procesador, memoria y uno o más discos.
- ⇒ Los procesadores de nodos distintos pueden comunicarse a través de una red de conexión de alta velocidad
- ⇒ Un nodo funciona como un servidor de los datos almacenados en los discos que posee
- ⇒ Ventajas →
 - Todas las operaciones de E/S no van por la red, sólo va por la red las peticiones, los accesos a discos remotos y las relaciones de resultados.
 - Ampliabilidad. Pueden soportar un gran nº de procesadores
- ⇒ Inconveniente →
 - Coste de comunicación y de acceso a discos remotos (hace necesario el software en ambos sentidos)
- ⇒ Teradata, Grace y Gamma

16.3.3.4 Jerárquico



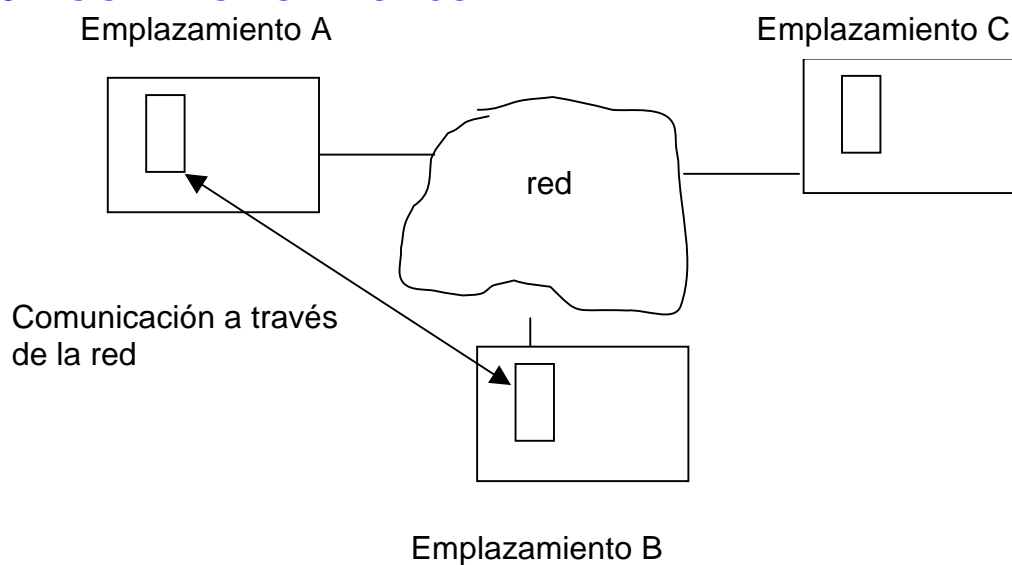
⇒ Combina las anteriores

⇒ Memoria virtual distribuida:

Una única memoria compartida desde el punto de vista lógico

Varios sistemas de memoria disjuntos desde el punto de vista físico, obteniéndose una única vista del área de memoria virtual mediante un hardware de asignación de memoria virtual en conjunción con un software extra

16.4 SISTEMAS DISTRIBUIDOS



- ⇒ La base de datos se almacena en varias computadoras (se les suele llamar **emplazamientos**).
- ⇒ Se comunican por redes de alta velocidad o telefónicas
- ⇒ No comparten ni memoria ni discos
- ⇒ Pueden variar en tamaño y función . Desde estaciones de trabajo a grandes sistemas
- ⇒ Diferencias con las BD paralelas sin compartimientos:
 - Las BD distribuidas se encuentran en lugares geográficos distintos.
 - Se administran de forma separada
 - Poseen interconexión más lenta
 - Se dan dos tipos de transacciones:
 - ❖ **Locales**→ Sólo accede a los datos del emplazamiento donde se inició la transacción
 - ❖ **Globales**→ Puede acceder a los datos de varios emplazamientos distintos.

16.4.1 *Ejemplo*

16.4.2 *Compromisos*

➤ **Ventajas**

- **Compartimiento de datos**→Proporcionan un entorno en el que los usuarios de un emplazamiento pueden ser capaces de acceder a los datos en otros emplazamientos
- **Autonomía**→Cada emplazamiento puede conservar un cierto control sobre los datos que tiene almacenados localmente con un grado de *autonomía local* diferente
- **Disponibilidad**→ Si en un SBD distribuido falla un emplazamiento, los restantes pueden continuar funcionando. Para ello debe ser capaz de detectar los fallos y solucionarlos. Esta disponibilidad es un factor **crucial** en las aplicaciones de **tiempo real**

➤ **Inconvenientes**→ El principal es la **complejidad** debida a

- **Coste de desarrollo del software**→La implementación es más difícil y costosa
- **Mayor probabilidad de errores**→ Es difícil asegurar la corrección de los algoritmos, del funcionamiento en cuanto a los fallos, así como de su recuperación (Todo ello debido a que los emplazamientos operan en paralelo)
- **Mayor sobrecarga de procesamiento**→ Debido al intercambio de mensajes y la coordinación entre los distintos emplazamientos

16.5 TIPOS DE REDES

Las bases de datos distribuidas y los sistemas cliente-servidor se constituyen en torno a las redes de comunicación

16.5.1 Redes de área local (LANs)

- ⇒ Cubren una zona geográfica relativamente pequeña y se utilizan normalmente en un entorno de oficina.
- ⇒ Una LAN típica puede consistir en:
 - Estaciones de trabajo, en un nº determinado
 - Uno o más sistemas servidores
 - Varios dispositivos periféricos compartidos (impresoras...)
 - Una o más pasarelas (procesadores especializados) que proporcionan acceso a otras redes.
 - Suele utilizarse un esquema de Ethernet
- ⇒ Mayor velocidad y tasa errores más pequeña que las WANs:
 - Velocidad varía entre 1 megabytes/seg y 1 Gigabit/seg.
 - Más normal es de 10 megabits/seg (Ethernet normal)
 - La red FDDI de fibra óptica y la Ethernet más rápida consiguen 100 megabits/seg
 - ATM (modo transferencia asíncrono) alcanza 144 megabits/seg
- ⇒ Los enlaces más frecuentes son: Par trenzado, cable coaxial de banda base o ancha, fibra óptica.

16.5.2 Redes de área amplia (WANs)

- ⇒ Unen computadores geográficamente distantes (Internet), que pueden diferir en el tipo, velocidad, longitud de palabra y sistema operativo.
- ⇒ Enlaces típicos son los circuitos telefónicos sustentados en líneas de fibra óptica y canales vía satélite.
 - T1 → 1,544 megabits/seg
 - T3 → 44,736 “
- ⇒ *Encaminadores*: Son sistemas que deciden que ruta siguen los mensajes a través de la red.
 - Ruta dinámica → incrementa eficiencia
 - Ruta estática → reduce riesgos seguridad ; más sencillo el cálculo del coste mantenimiento
- ⇒ *Módem*: Dispositivo que convierte los datos digitales que recibe en analógicos del sistema telefónico y viceversa.
 - Velocidad oscila entre 2.400 bits/seg y 32 Kilobits/seg
 - RDSI puede transmitir los datos punto a punto a una velocidad de 128 kilobits/seg
 - La red UNIX, UUCP permite intercambiar mensajes vía modem un nº limitado de veces. Estos se dirigen a otros sistemas cercanos y así se propagan a toda la red (públicos) o a su destino (privados)
- **WAN de conexión discontinua**: UUCP
- **WAN de conexión continua**: Internet Los sistemas distribuidos sólo pueden comunicarse de forma continua.