

Estrategias de Persistencia

Primer Cuatrimestre 2020.

Clase Nro 8.

Capitulo Nro. 20 de la bibliografia.

Arquitectura de los sistemas de base de datos.

En esta clase, nos enfocaremos en contenido teórico que será el último contenido que se incluye en el examen de la clase Nro 9 (Semana Proxima), a continuacion de algunos puntos del capítulo 20 a tener en cuenta y un cuestionario que deben realizar a modo de ejercicio para el examen.

***** RECORDATORIO*****

Semana proxima 03/06/20 19.30 hs primer examen de la materia. Nos conectaremos de manera sincronica para realizar el examen, no lleguen tarde al examen ya que esta configurado el inicio y el final.

Temas: Solo teorico, todo lo visto en los documentos compartidos, inclusive clase 8.

Arquitectura de los sistemas de base de datos.

Arquitectura Centralizada y Cliente – Servidor.

Los sistemas de bases de datos centralizados son aquellos que se ejecutan en un único sistema informático sin interactuar con ninguna otra computadora. Tales sistemas comprenden el rango desde los sistemas de bases de datos monousuario ejecutándose en computadoras personales hasta los sistemas de bases de datos de alto rendimiento ejecutándose en grandes sistemas. Por otro lado, los sistemas cliente –servidor tienen su funcionalidad dividida entre el sistema servidor y múltiples sistemas clientes.

Sistemas Centralizados.

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 encuentran conectados a través de un bus común, el cual proporciona acceso a la memoria compartida (Figura 20.1). Las CPU (unidades centrales de procesamiento) poseen memorias caché locales donde se almacenan copias de ciertas partes de la memoria para acelerar el acceso a los datos. Cada controlador de dispositivo se encarga de un tipo específico de dispositivos (por ejemplo, una unidad de disco, una tarjeta de sonido o un monitor). Las CPU y los controladores de dispositivos pueden ejecutarse concurrentemente compitiendo así por el acceso a la memoria. La

memoria caché reduce la disputa por el acceso a la memoria ya que la CPU necesita acceder a la memoria compartida un número de veces menor.

Se distinguen dos formas de utilizar las computadoras: como sistemas monousuario o multiusuario. En la primera categoría se encuentran las computadoras personales y las estaciones de trabajo. Un **sistema monousuario** típico es una unidad de sobremesa utilizada por una única persona que dispone de una sola CPU, de uno o dos discos fijos y que trabaja con un sistema operativo que sólo permite un único usuario. Por el contrario, un **sistema multiusuario** típico tiene más discos y más memoria, puede disponer de varias CPU 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.

Normalmente, los sistemas de bases de datos diseñados para funcionar sobre sistemas monousuario no suelen proporcionar muchas de las facilidades que ofrecen los sistemas multiusuario. En particular no tienen control de concurrencia, el cual no es necesario cuando solamente un usuario puede generar modificaciones. Las facilidades de recuperación en estos sistemas o no existen o son primitivas; por ejemplo, realizar una copia de seguridad de la base de datos antes de cualquier modificación. La mayoría de estos sistemas no admiten SQL y proporcionan un lenguaje de consulta muy simple que, en algunos casos, es una variante de QBE. En cambio, los sistemas de bases de datos diseñados para sistemas multiusuario soportan todas las características de las transacciones que se han estudiado antes.

Aunque hoy en día las computadoras de propósito general disponen de varios procesadores, utilizan **paralelismo de grano grueso**, disponiendo de unos pocos procesadores (normalmente dos o cuatro) que comparten la misma memoria principal. Las bases de datos que se ejecutan en tales máquinas habitualmente no intentan dividir una consulta simple entre los distintos procesadores, sino que ejecutan cada consulta en un único procesador posibilitando la concurrencia de varias consultas. Así, estos sistemas soportan una mayor productividad, es decir, permiten ejecutar un mayor número de transacciones por segundo, a pesar de que cada transacción individualmente no se ejecute más rápido.

Las bases de datos diseñadas para las máquinas monoprocesador ya disponen de multitarea permitiendo que varios procesos se ejecuten a la vez en el mismo procesador, usando tiempo compartido, mientras que de cara al usuario parece que los procesos se están ejecutando en paralelo. De esta manera

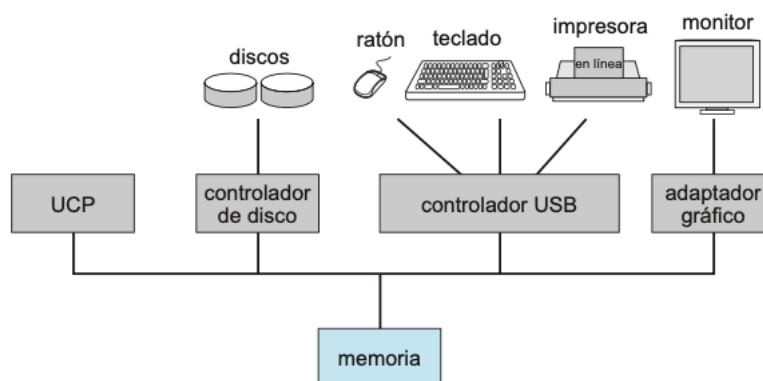


Figura 20.1 Un sistema centralizado.

desde un punto de vista lógico, las máquinas paralelas de grano grueso parecen ser idénticas a las máquinas monoprocesador, y pueden adaptarse fácilmente los sistemas de bases de datos diseñados para máquinas de tiempo compartido para que puedan ejecutarse sobre máquinas paralelas de grano grueso.

Por el contrario, las **máquinas paralelas de grano fino** disponen de un gran número de procesadores y los sistemas de bases de datos que se ejecutan sobre ellas intentan hacer paralelas las tareas simples (consultas, por ejemplo) que solicitan los usuarios. En el Apartado 20.3 se estudia la arquitectura de los sistemas de bases de datos paralelos.

Sistema Cliente – Servidor.

Como las computadoras personales son cada vez más rápidas, más potentes y más baratas, los sistemas se han ido distanciando de la arquitectura centralizada. Los terminales conectados a un sistema central han sido suplantados por computadoras personales. De igual forma, la interfaz de usuario, que solía estar gestionada directamente por el sistema central, está pasando a ser gestionada, cada vez más, por las computadoras personales. Como consecuencia, los sistemas centralizados actúan hoy como **sistemas servidores** que satisfacen las peticiones generadas por los *sistemas clientes*. En la Figura 20.2 se representa la estructura general de un sistema cliente-servidor.

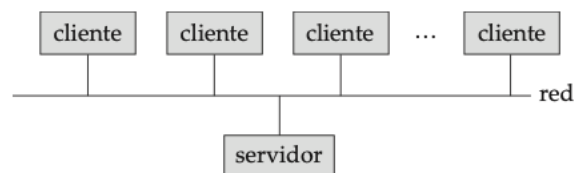
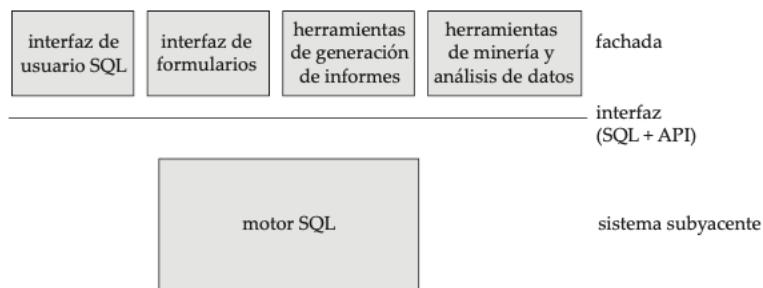


Figura 20.2 Estructura general de un sistema cliente-servidor.



La funcionalidad proporcionada por los sistemas de bases de datos se puede dividir a grandes rasgos en dos partes: la fachada y el sistema subyacente. El sistema subyacente gestiona el acceso a las estructuras, la evaluación y optimización de consultas, el control de concurrencia y la recuperación. La fachada de un sistema de base de datos está formado por herramientas como la interfaz de usuario con SQL, interfaces de formularios, diseñadores de informes y herramientas para la recopilación y análisis de datos. La interfaz entre la fachada y el sistema subyacente puede ser SQL o una aplicación.

Las normas como *ODBC* y *JDBC*, que se estudiaron en el Capítulo 3, se desarrollaron para hacer de interfaz entre clientes y servidores. Cualquier cliente que utilice interfaces ODBC o JDBC puede conectarse a cualquier servidor que proporcione esta interfaz.

Ciertas aplicaciones como las hojas de cálculo y los paquetes de análisis estadístico utilizan la interfaz cliente – servidor directamente para acceder a los datos del servidor subyacente. De hecho, proporcionan interfaces visibles especiales para diferentes tareas.

Los sistemas que trabajan con una gran cantidad de usuarios adoptan una arquitectura de tres capas, que se vio anteriormente en la Figura 1.7 (Capítulo 1), donde la fachada es el explorador Web que se comunica con un servidor de aplicaciones. El servidor de aplicaciones actúa como cliente del servidor de bases de datos.

Algunos sistemas de procesamiento de transacciones proporcionan una interfaz de **llamada a procedimientos remotos para transacciones** para conectar los clientes con el servidor. Estas llamadas aparecen para el programador como llamadas normales a procedimientos, pero todas las llamadas a procedimientos remotos hechas desde un cliente se engloban en una única transacción al servidor final. De este modo, si la transacción se cancela, el servidor puede deshacer los efectos de las llamadas a procedimientos remotos individuales.

Estructura de procesos del servidor de transacciones.

Hoy en día, un sistema servidor de transacciones típico consiste en múltiples procesos accediendo a los datos en una memoria compartida, como en la Figura 20.4. Los procesos que forman parte del sistema de bases de datos incluyen:

- **Proceso servidor.** Son procesos que reciben consultas del usuario (transacciones), las ejecutan, y devuelven los resultados. Las consultas deben enviarse a los procesos servidor desde la interfaz de usuario, o desde un proceso de usuario que ejecuta SQL incorporado, o a través de JDBC, ODBC o protocolos similares. Algunos sistemas de bases de datos utilizan un proceso distinto para cada sesión de usuario, y otros utilizan un único proceso de la base de datos para todas las sesiones del usuario, pero con múltiples hebras de forma que se pueden ejecutar concurrentemente múltiples consultas (una **hebra** es parecida a un proceso; sin embargo, varias hebras se pueden ejecutar concurrentemente como parte de un mismo proceso, y todas ellas en el mismo espacio de memoria virtual). Algunos sistemas de bases de datos utilizan una arquitectura híbrida, con procesos múltiples, cada uno de ellos con varias hebras.
- **Proceso gestor de bloqueos.** Este proceso implementa una función de gestión de bloqueos que incluye concesión de bloqueos, liberación de bloqueos y detección de interbloqueos.
- **Proceso escritor de bases de datos.** Existe uno o más procesos que vuelcan al disco los bloques de memoria intermedia modificados de forma continua.

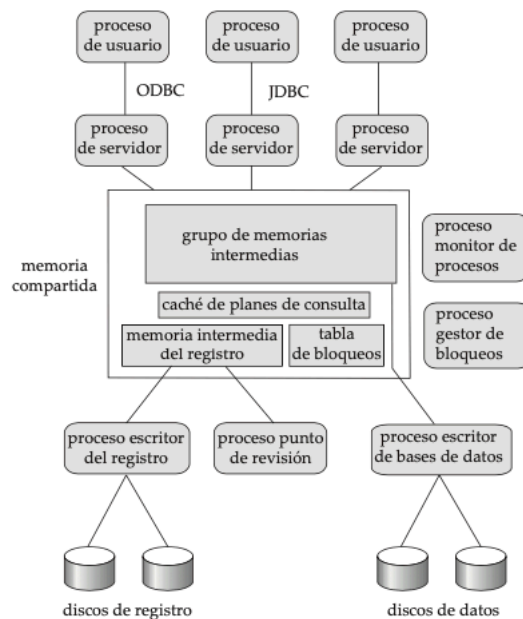


Figura 20.4 Estructura de la memoria compartida y de los procesos.

- **Proceso escritor del registro.** Este proceso genera entradas del registro en el almacenamiento estable a partir de la memoria intermedia del registro. Los procesos servidor simplifican la adición de entradas a la memoria intermedia del registro en memoria compartida y, si es necesario forzar la escritura del registro, le piden al proceso escritor del registro que vuelque las entradas del registro.
- **Proceso punto de revisión.** Este proceso realiza periódicamente puntos de revisión.
- **Proceso monitor de procesos.** Este proceso observa otros procesos y, si cualquiera de ellos falla, realiza acciones de recuperación para el proceso, tales como cancelar cualquier transacción que estuviera ejecutando el proceso fallido, y reinicia el proceso.

La memoria compartida contiene todos los datos compartidos, como:

- Grupo de memorias intermedias.
- Tabla de bloqueos.
- Memoria intermedia del registro, que contiene las entradas del registro que esperan a ser volcadas en el almacenamiento estable.
- Planes de consulta en caché, que se pueden reutilizar si se envía de nuevo la misma consulta.

Todos los procesos de la base de datos pueden acceder a los datos de la memoria compartida. Ya que múltiples procesos pueden leer o realizar actualizaciones en las estructuras de datos en memoria compartida, debe haber un mecanismo que asegure que sólo uno de ellos está modificando una estructura de datos en un momento dado, y que ningún proceso está leyendo una estructura de datos mientras otros la escriben. Tal **exclusión mutua** se puede implementar por medio de funciones del sistema operativo llamadas semáforos. Implementaciones alternativas, con menos sobrecargas, utilizan **instrucciones atómicas** especiales soportadas por el hardware de la computadora; un tipo de instrucción atómica comprueba una posición de la memoria y la establece a uno automáticamente. Se pueden encontrar más detalles sobre la exclusión mutua en cualquier libro de texto de un sistema operativo estándar. Los mecanismos de exclusión mutua también se utilizan para implementar pestillos.

Para evitar la sobrecarga del paso de mensajes, en muchos sistemas de bases de datos los procesos servidor implementan el bloqueo actualizando directamente la tabla de bloqueos (que está en memoria compartida), en lugar de enviar mensajes de solicitud de bloqueo a un proceso administrador de bloqueos. El procedimiento de solicitud de bloqueos ejecuta las acciones que realizaría el proceso administrador de bloqueos para procesar una solicitud de bloqueo. Las acciones de la solicitud y la liberación de bloqueos son como las del Apartado 16.1.4, pero con dos diferencias significativas:

- Dado que varios procesos servidor pueden acceder a la memoria compartida, se asegurará la exclusión mutua en la tabla de bloqueos.
- Si no se puede obtener un bloqueo inmediatamente a causa de un conflicto de bloqueos, el código de la solicitud de bloqueo sigue observando la tabla de bloqueos hasta percatarse de que se ha concedido el bloqueo. El código de liberación de bloqueo actualiza la tabla de bloqueos para indicar a qué proceso se le ha concedido el bloqueo.

Para evitar repetidas comprobaciones de la tabla de bloqueos, el código de solicitud de bloqueo puede utilizar los semáforos del sistema operativo para esperar una notificación de una concesión de bloqueo. El código de liberación de bloqueo debe utilizar entonces el mecanismo de semáforos para notificar a las transacciones que están esperando que sus bloqueos hayan sido concedidos.

Incluso si el sistema gestiona las solicitudes de bloqueo por medio de memoria compartida, aún utiliza el proceso administrador de bloqueos para la detección de interbloqueos.

Sistemas Paralelos.

Los sistemas paralelos mejoran la velocidad de procesamiento y de E/S porque la CPU y los discos funcionan en paralelo. Cada vez son más comunes las máquinas paralelas, lo que hace que cada vez sea más importante el estudio de los sistemas paralelos de bases de datos. La fuerza que ha impulsado a los sistemas paralelos de bases de datos ha sido la demanda de aplicaciones que han de manejar bases de datos extremadamente grandes (del orden de terabytes—esto es, 10^{12} bytes) o que tienen que procesar un número enorme de transacciones por segundo (del orden de miles de transacciones por segundo). Los sistemas de bases de datos centralizados o cliente-servidor no son suficientemente potentes para soportar tales aplicaciones.

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. Hoy en día, la mayoría de las máquinas de gama alta ofrecen un cierto grado de paralelismo de grano grueso: son comunes las máquinas con dos o cuatro procesadores. Las computadoras masivamente paralelas se distinguen de las máquinas paralelas de grano grueso porque son capaces de soportar un grado de paralelismo mucho mayor. Ya se encuentran en el mercado computadoras paralelas con cientos de CPU y discos.

Para evaluar el rendimiento de los sistemas de bases de datos existen dos medidas principales: (1) la **productividad**, número de tareas que pueden completarse en un intervalo de tiempo determinado, y (2) 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 mejorar la productividad realizando muchas transacciones en paralelo. Un sistema que procese transacciones largas puede mejorar el tiempo de respuesta y la productividad realizando en paralelo las distintas sub tareas de cada transacción.

Arquitectura paralelas de bases de datos.

Existen varios modelos de arquitecturas para las máquinas paralelas. En la Figura 20.8 se muestran algunos de los más importantes (en la figura, M quiere decir memoria, P procesador y los discos se dibujan como cilindros):

- **Memoria compartida.** Todos los procesadores comparten una memoria común (Figura 20.8a).
- **Disco compartido.** Todos los procesadores comparten un conjunto de discos común (Figura 20.8b). Algunas veces los sistemas de disco compartido se denominan **agrupaciones**.
- **Sin compartimiento.** Los procesadores no comparten ni memoria ni disco (Figura 20.8c).
- **Jerárquica.** Este modelo es un híbrido de las arquitecturas anteriores (Figura 20.8d).

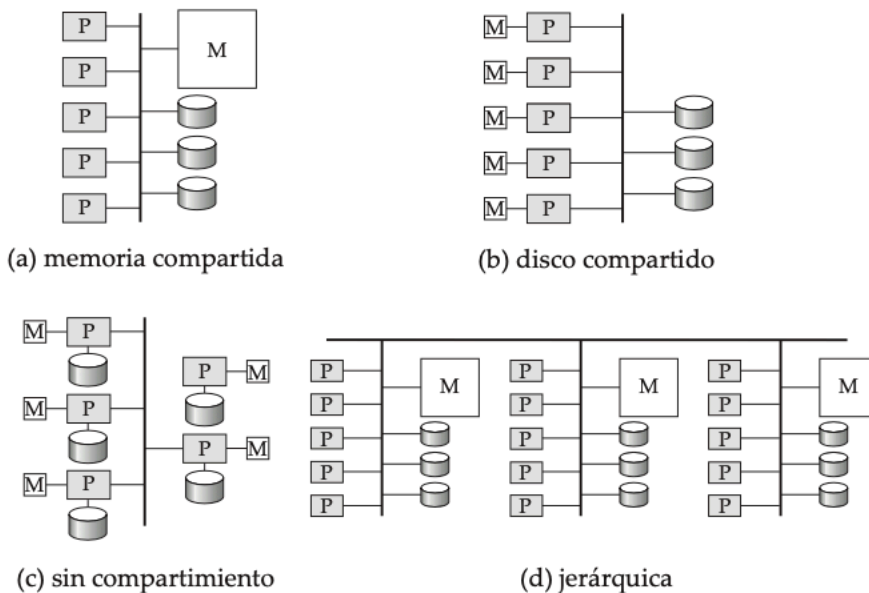


Figura 20.8 Arquitecturas paralelas de bases de datos.