

Consultas Distribuidas

M.C. Carlos Rojas Sánchez¹

¹Licenciatura en Informática
Universidad del Mar - Puerto Escodido

[Bases de Datos Distribuidas]

1 Metodología del procesamiento de consultas distribuidas

Consultas distribuidas

Las consultas distribuidas obtienen acceso a datos de varios orígenes de datos heterogéneos. Estos orígenes de datos pueden estar almacenados en el mismo equipo o en equipos diferentes.

Práctica 1

- Dos o más instancias de mysql en una misma computadora

Consultas distribuidas

- El costo de transmisión de datos en la red
- Repetición y fragmentación
- Procesamiento de intersección simple

Consultas distribuidas

Cuando una base de datos se encuentra en múltiples servidores y distribuye a un número determinado de nodos:

- El servidor recibe una petición de un nodo
- El servidor debe soportar el acceso concurrente
- El servidor muestra un resultado y le da un hilo a cada una de las máquinas nodo de la red local

Consultas distribuidas

- En una BD centralizada todos los datos residen en un solo sitio, por tanto el DBMS debe evaluar cada solicitud de datos y encontrar la manera más eficiente de acceder a los datos locales
- Por el contrario el DDBMS hace posible dividir una BD en varios fragmentos, lo que complica mas trabajo en las consultas por que el DDBMS debe decidir a qué fragmentos de la BD debe acceder

Consultas distribuidas

- La replicación de los datos complica aún más el problema de acceso, por que ahora la BD debe decidir a qué copia de la BD se debe acceder. El DDBMS utiliza técnicas de optimización de consultas para ocuparse de tales problemas y para garantizar un desempeño aceptable de la BD. Los costos asociados a una solicitud están en función:
 - Del costo del tiempo de acceso (E/S) implicado al acceder a los datos físicos guardados en disco
 - Del costo de comunicación asociado con la transmisión de datos entre nodos en sistemas de BDD
 - Del costo de tiempo de CPU asociado con la sobrecarga de procesamiento de manejar transacciones distribuidas

Almacenamiento Distribuido de Datos

- Réplica. El sistema conserva réplicas (copias) idénticas de la relación y guarda cada réplica en un sitio diferente. La alternativa a las réplicas es almacenar sólo una copia de la relación r
- Fragmentación. El sistema divide la relación en varios fragmentos y guarda cada fragmento en un sitio diferente

Transacciones Distribuidas

- Las transacciones locales son las que tienen acceso a los datos y los actualizan sólo en una base de datos local
- Las transacciones globales son las que tienen acceso a datos y los actualizan en varias bases de datos locales

Estructura del Sistema

- El gestor de transacciones administra la ejecución de las transacciones (o subtransacciones) que tienen acceso a los datos almacenados en un sitio local. Téngase en cuenta que cada una de esas transacciones puede ser una transacción local o parte de una transacción global
- El coordinador de transacciones coordina la ejecución de las diferentes transacciones (tanto locales como globales) iniciadas en ese sitio

Coordinador de transacciones

En cada una de esas transacciones el coordinador es responsable de:

- Inicio de la ejecución de la transacción
- División de la transacción en varias subtransacciones y distribución de esas subtransacciones a los sitios correspondientes para su ejecución
- Coordinación de la terminación de la transacción, que puede hacer que la transacción se comprometa en todos los sitios o que se aborte en todos los sitios

Proyecto 3er Parcial

- Nodo 1 Banco
 - Sucursales (id_s,nombre)
 - Transacciones(id_t,tipo)
- Nodo (2 y 3) Banco
 - Clientes (id_c,id_s,nombre)
 - Cuentas(id,id_c,moneto)
 - Transacciones(id_t,tipo)

Transacciones Distribuidas

- Se refieren a transacciones planas o anidadas que acceden a objetos administrados por múltiples servidores.
- Cuando una transacción distribuida llega a su fin, la propiedad de atomicidad de las transacciones requiere que todos los servidores involucrados produzcan el commit de la transacción o todos ellos la abortan.
- La manera en que el coordinador logra ésto, depende del protocolo elegido.

Transacciones Distribuidas

Transacciones distribuidas planas y anidadas

- En una transacción plana, el cliente hace requerimientos a más de un servidor. Cada transacción accede a los objetos en los servidores secuencialmente. El cliente de la transacción plana espera completar todos sus requerimientos antes de pasar a la próxima.
- En una transacción anidada, la transacción de mayor nivel puede abrir subtransacciones y, a su vez cada subtransacción puede abrir otras en niveles más bajos de anidamiento.

Ejemplo de una transacción bancaria anidada

- Sea una transacción distribuida donde el cliente transfiere \$10 de la cuenta A a C y \$20 de B a D.
- Las cuentas A y B están en el servidor X y las cuentas C y D están en el servidor Y.
- Si la transacción se estructura como un conjunto de cuatro transacciones anidadas, los cuatro requerimientos (dos depósitos y dos retiros) pueden correr en paralelo y el efecto total es lograr mayor rendimiento que una transacción simple ejecutando las cuatro operaciones secuencialmente.

Ejemplo de una transacción bancaria anidada

- T = abreTransacción
 - abreSubTransacción a.retiro(10);
 - abreSubTransacción b.retiro(20);
 - abreSubTransacción c.depósito(10);
 - abreSubTransacción d.depósito(20);
- cierraTransacción

Ejemplo de una transacción bancaria anidada

El coordinador de una transacción distribuida

- Los servidores que ejecutan requerimientos que son parte de una transacción distribuida necesitan poder comunicarse con otros para coordinar sus acciones cuando la transacción commits.
- Un cliente comienza la transacción enviando un requerimiento abreTransacción a un coordinador en algún servidor.
- El coordinador que es contactado lleva adelante la abreTransacción y retorna un identificador al cliente (éste debe ser único).
- El coordinador que abre la transacción se convierte en el coordinador para la transacción distribuida.



Ejemplo de una transacción bancaria anidada

- Cada uno de los servidores que administra un objeto accedido por la transacción es un participante en la transacción, se llamará participante.
- Los participantes son responsables en la cooperación con el coordinador para llevar adelante el protocolo de commit de la transacción.
- Durante el progreso de la transacción, el coordinador registra los participantes y éstos registran al coordinador.

Ejemplo de una transacción bancaria anidada

- Ejemplo: Un cliente cuya transacción (plana) involucra las cuentas A, B, C y D en los servidores Nodo 1, Nodo 2 . La transacción T del cliente transfiere \$3 de la cuenta B a D y \$4 de A a C.

Bibliografía I

-  M. Tamer Özsu and Patrick Valduriez.
Principles of Distributed Database Systems.
Springer, 2011.
-  Abraham Silberschatz, Henry F. Korth and S. Sudarshan.
Fundamentos de Bases de Datos.
McGRAW-HILL, 2002.