

Instituto Politécnico Nacional

Escuela Superior de Cómputo

Bases de Datos

Profesor: Zagal Flores Roberto Eswart

Autor: Velasco Huerta Angel Eduardo

Práctica no. -5

15/04/2021

Arquitecturas de MYSQL:

Modelo de arquitectura lógica:

A continuación, se muestran los principales elementos que componen al motor de bases de datos que utilizaremos en la implementación, y una breve, pero concisa descripción de el elemento en cuestión:

Motores de almacenamiento

Un motor de almacenamiento es el encargado de almacenar, gestionar y recuperar toda la información de una tabla. Es por ello que es de suma importancia que nosotros conozcamos la existencia de estos motores

El elemento más notable de la arquitectura de MySQL es la denominada arquitectura de motores de almacenamiento reemplazables. La idea de esa arquitectura es hacer una interfaz abstracta con funciones comunes de gestión de datos en el nivel físico. De ese modo, el gestor de almacenamiento puede intercambiarse, e incluso un mismo servidor MySQL puede utilizar diferentes motores de almacenamiento para diferentes bases de datos o para diferentes tablas en la misma base de datos.

Los conectores

Los conectores son bibliotecas en diferentes lenguajes de programación que permiten la conexión (remota o local) con servidores MySQL y la ejecución de consultas.

MySQL Connector/ODBC : Es el nombre de la familia de controladores MySQL ODBC (anteriormente llamados Controladores MyODBC) que proporcionan acceso a una base de datos MySQL usando el estándar industrial de Conectividad de Base de Datos Abierta (Open Database Connectivity).

El gestor de conexiones

La gestión de conexiones es responsable de mantener las múltiples conexiones de los clientes. Un gestor de conexiones inexistente o laxo simplemente crearía una conexión por cada cliente conectado. No obstante, las conexiones consumen recursos de máquina, y crearlas y destruirlas son también procesos costosos. Por eso, el gestor de conexiones de MySQL puede configurarse para limitar el número de conexiones concurrentes, y también implementa un pool de conexiones.

Para poder cumplir con el requisito no funcional especificado previamente, será importante conocer este elemento, pues nos permitirá determinar el número de conexiones que se podrán realizar en nuestra base de datos.

El gestor de conexiones también se ocupa de la autenticación de los usuarios. La autenticación por defecto se basa en el nombre de usuario, la máquina desde la que se conecta y la password.

El procesamiento y optimización de consultas

Cada vez que una consulta llega al gestor de MySQL, se analiza sintácticamente y se produce una representación intermedia de la misma. A partir de esa representación, MySQL toma una serie de decisiones, que pueden incluir el determinar el orden de lectura de las tablas, el uso de ciertos índices, o la reescritura de la consulta en una forma más eficiente.

La caché de consultas

MySQL implementa un caché de consultas, donde guarda consultas y sus resultados enteros. De este modo, el procesador de consultas, antes ni siquiera de plantear la optimización, busca la consulta en la caché, para evitarse realizar el trabajo en el caso de que tenga suerte y encuentre la consulta en la caché.

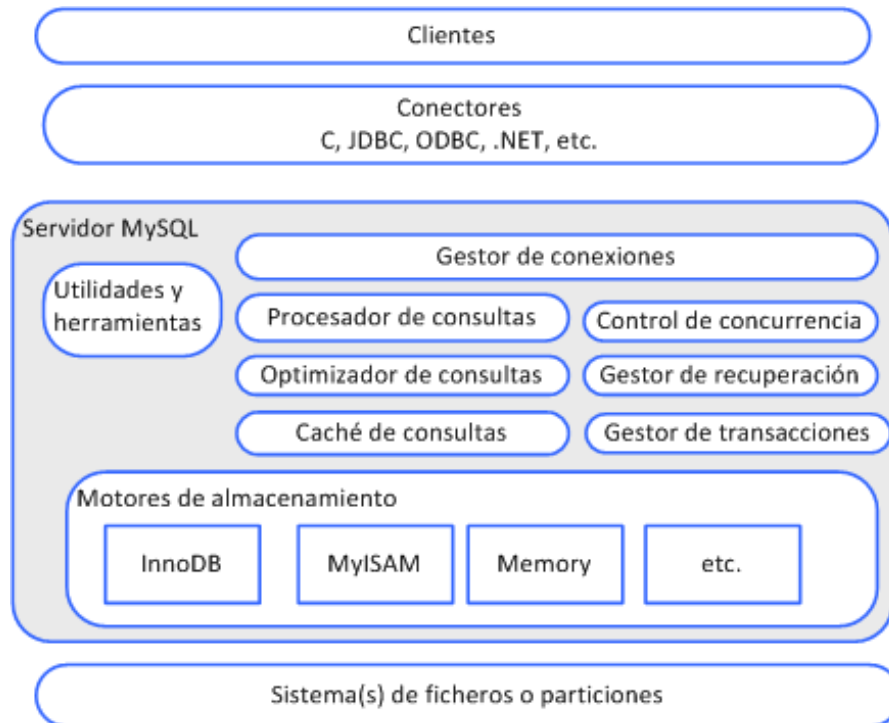
El Control de Concurrencia

El control de concurrencia en un gestor de bases de datos es simplemente el mecanismo que se utiliza para evitar que lecturas o escrituras simultáneas a la misma porción de datos terminen en inconsistencias o efectos no deseados. El mecanismo que se utiliza para controlar este acceso es el de los bloqueos (**locks**).

La gestión de transacciones y recuperación

La gestión de transacciones permite dotar de semántica “todo o nada” a una consulta o a un conjunto de consultas que se declaran como una sola transacción. Es decir, si hay algún problema y parte de la consulta o algunas de las consultas no consiguen llevarse a cabo, el servidor anulará el efecto parcial de la parte que ya haya sido ejecutada. La recuperación permite “volver hacia atrás” (**rollback**) partes de una transacción.

Con la siguiente imagen, se denotan estos elementos descritos:



ACID:

Las transacciones son un conjunto de instrucciones SQL que tienen la cualidad de ejecutarse como una unidad, es decir, o se ejecutan todas o no se ejecuta ninguna. Si una transacción tiene éxito, todas las modificaciones de los datos realizados durante la transacción se guardan en la base de datos. Si una transacción contiene errores los cambios no se guardarán en la base de datos.

Atomicity: Es la propiedad que asegura que la operación se ha realizado o no, y por lo tanto ante un fallo del sistema no puede quedar a medias. Es decir, si una operación necesita varios pasos para ser completada se ejecutan todos los pasos o ninguno.

Consistency: Esta propiedad está ligada a la integridad referencial, es decir solo se pueden escribir datos válidos respetando los tipos de datos declarados y la integridad referencial. Consiste en mantener la integridad de los datos de la base de datos.

Isolation: Asegura que una operación no puede afectar a otras. Con esto se asegura que varias transacciones sobre la misma información sean independientes y no generen ningún tipo de error. Es decir que cada operación está aislada.

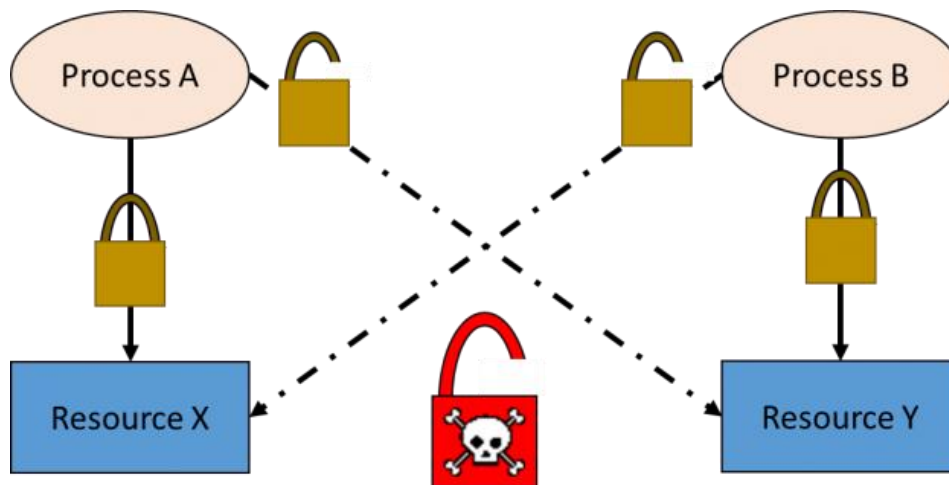
Durability: Es la propiedad que asegura que, una vez realizada la operación, ésta persistirá. Para conseguir esta característica es necesario tener sistemas de backup, unidades SAI por si

hay apagones de electricidad, prácticamente, es salvar los cambios una vez se verificó que la instrucción es válida.

Deadlocks:

MySQL permite bloquear tablas para que diversas queries puedan ejecutarse al mismo tiempo sin distorsionar sus resultados. Este bloqueo puede ser de escritura o de lectura.

Un Deadlock (también llamado "bloqueo mutuo", "abrazo mortal", "punto muerto", etc.) sucede cuando dos o más transacciones intentan hacer bloqueos de claves en orden opuesto, son un problema clásico de las bases de datos transaccionales, pero no son peligrosos a menos que sean tan frecuentes que no se puedan ejecutar en absoluto ciertas transacciones. En pocas palabras sucede porque transacciones diferentes, no pueden continuar porque están bloqueando algo que la otra necesita.



COMMIT, ROLLBACK and AUTOCOMMIT

Por defecto, MySQL se ejecuta con el modo autocommit activado. Esto significa que en cuanto ejecute un comando que actualice (modifique) una tabla, MySQL almacena la actualización en disco.

Tras deshabilitar el modo autocommit poniendo la variable AUTOCOMMIT a cero, debe usar COMMIT para almacenar los cambios en disco o ROLLBACK si quiere ignorar los cambios hechos desde el comienzo de la transacción. Con START TRANSACTION, autocommit permanece deshabilitado hasta el final de la transacción con COMMIT o ROLLBACK. El modo autocommit vuelve a su estado previo.

Cambios en MySQL Versión:

Resumen de algunos de los bugs importantes corregidos en la versión 8.0.23:

1. La operación crear tabla que especificaba la opción de compression ya no se permite, y lo marca como un error.
2. Una transacción de InnoDB se hizo inconsistente, después de crear una tabla usando otro motor de almacenamiento con una expresión de tabla que usa InnoDB ahora marcar error.
3. Se repararon instancias de la operación SELECT COUNT * cuando esta se usaba en una tabla con información no comprometida.
4. Se mejoraron los procesos de limpiado de página en el proceso sync_flush
5. Un puntero inválido causaba errores en MySQL con la configuración DISABLE_PSA_MEMORY
6. La mayoría de los bugs corregidos, son acerca del cambio de variables internas en MySQL
7. Se arreglaron problemas con el comando alter table pues este solía fallar a una tabla que no estaba particionada.
8. Bug con Windows que corría el servicio de MySQL con fallos, causando errores en la conexión de memoria compartida.
9. Fallos con el editor de configuraciones de MySQL causaba que este tuviera problemas con el campo password.

Conclusiones de MySQL:

MySQL parece tener muchas funciones que están en el fondo de la arquitectura, que son de extrema importancia para el manejo de dato, con propiedades que pueden servir prácticamente en cualquier proyecto, a mi parecer es un gestor robusto, y que esta en constante actualización, aunque algo que me hizo dudar, es que en las últimas versiones de MySQL, el reporte de cambios era gigante, y parecía tener problemas con sentencias que hasta en un uso básico, podían dar problema, esto me hizo pensar que posiblemente haya que encontrar una versión estable del mismo si se planea realizar un proyecto importante, pero en cuestión de las características del manejador, creo que cumple con lo necesario para considerarse como un sistema robusto.

Referencias:

- Documentacion oficial de MySQL:

<https://dev.mysql.com/doc/relnotes/mysql/8.0/en/news-8-0-23.html#mysqld-8-0-23-bug>

- Openstax.com: <https://cnx.org/contents/-jPvAviC@1/Visi%C3%B3n-general-de-la-arquitectura-de-MySQL-5-1>
- UNAM, Ventajas de uso de MYSQL
https://www.redisibd.unam.mx/rss/ponencias/SunMySql_engine.pdf