

Entendiendo y Optimizando MySQL

Jesús Espino García

Grupo de Usuarios de Linux
Universidad Carlos III de Madrid.



10 de Noviembre de 2010

Índice

1 Introducción

- Introducción
- Arquitectura

2 Esquemas e índices

3 Componentes

4 Las queries

5 Escalando MySQL

Introducción

- Sistema de Gestión de Bases de Datos.
- GPL.
- Escrito en C y C++.
- Multiplataforma.
- Mas de 6 millones de instalaciones.

Índice

1 Introducción

■ Introducción

■ **Arquitectura**

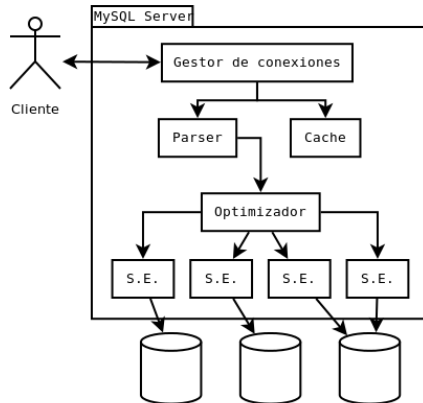
2 Esquemas e índices

3 Componentes

4 Las queries

5 Escalando MySQL

Arquitectura



Índice

- 1 Introducción
- 2 Esquemas e índices
 - Introducción
 - Normalización
 - Índices
- 3 Componentes
- 4 Las queries
- 5 Escalando MySQL

Introducción

- Nuestros esquemas e índices dependen de la funcionalidad.
- Es tan importante el que, como el como.
- Todo el diseño debe tener en cuenta los casos de uso.
- Es muy diferente un diseño para lectura, escritura o baja latencia.
- Todo deriva en un compromiso entre el rendimiento en diferentes situaciones.

Índice

- 1 Introducción
- 2 Esquemas e índices
 - Introducción
 - Normalización
 - Índices
- 3 Componentes
- 4 Las queries
- 5 Escalando MySQL

Normalización

- Reestructuración de nuestras tablas.
- Busca eliminar redundancia.
- Se aplican una serie de "formas normales".
- Normalmente es una buena política.

Des-normalización

- A veces la normalización es ineficiente.
- La redundancia puede producir incrementos de rendimiento significativos.
- Bases de datos con mucha lectura y poca escritura.
- Bases de datos con necesidades de latencias muy bajas.

Índice

- 1 Introducción
- 2 Esquemas e índices
 - Introducción
 - Normalización
 - Índices
- 3 Componentes
- 4 Las queries
- 5 Escalando MySQL

Índices

- Estructuras auxiliares para búsquedas.
- Aceleran las consultas (cuando tienen datos suficientes).
- Pueden resolver la consulta entera (si los datos necesarios están contenidos).
- Hacen referencia a uno o mas campos.
- Los índices de varios campos tienen un orden concreto.
- $(A,B) \neq (B,A)$
- Cada índice incrementa el espacio consumido y decrementa la velocidad de escritura.

Arboles B

- Este es tipo de índice mas habitual.
- El formato interno del árbol depende del S.E.
- Este tipo de índice permite las siguientes consultas:
 - El valor completo del índice.
 - Valores en la parte izquierda del índice.
 - Rangos de valores.
 - Una parte exacta (a la izquierda), y el resto como un rango.
 - Consultas de solo el índice.

Tablas hash

- Para cada columna, se calcula un hash y se asocia al índice.
- Solo permite búsquedas exactas.
- Las búsquedas son muy rápidas.
- Es el tipo por defecto del S.E. Memory.
- Este modo no esta disponible en MyISAM o InnoDB, pero se puede "emular".

Spatial indexes

- Índices especiales para GIS.

Full text

- Índices para búsquedas sobre el contenido.
- Solo disponibles en MyISAM (por ahora).

Clustered indexes

- No es otro tipo de índice, es un concepto.
- Consiste en incluir los datos de la fila, dentro del índice de la clave primaria.
- Esto permite que la búsqueda de la clave primaria de como resultado la fila, sin necesidad de ningún salto extra.
- InnoDB implementa este tipo de árbol B.

Coverage indexes

- Extraer los datos directamente del índice.
- Solo si todos los datos están contenidos en el índice.
- Supone un incremento significativo del rendimiento.

Índice

1 Introducción

2 Esquemas e índices

3 Componentes

- El servidor
 - La cache
 - El parser
 - El optimizador
 - Storage Engines

4 Las queries

5 Escalando MySQL

El servidor

- Espera las conexiones de los usuarios.
- Hace la autenticación de usuarios.
- Comprueba si la consulta esta en la cache.
- Si la consulta esta en la cache, devuelve el resultado.
- Si la consulta no esta en la cache, la pasa al parser.

Índice

1 Introducción

2 Esquemas e índices

3 Componentes

- El servidor
- **La cache**
- El parser
- El optimizador
- Storage Engines

4 Las queries

5 Escalando MySQL

La cache

- La cache almacena resultados asociados a "hash" de consultas.
- Cualquier modificación en una tabla relacionada con una consulta, caduca esa consulta.

Índice

1 Introducción

2 Esquemas e índices

3 Componentes

- El servidor
- La cache
- **El parser**
- El optimizador
- Storage Engines

4 Las queries

5 Escalando MySQL

El parser

- Recibe una consulta en SQL, y la convierte en una estructura de árbol.
- Pasa la estructura de árbol al optimizador.

Índice

1 Introducción

2 Esquemas e índices

3 Componentes

- El servidor
- La cache
- El parser
- **El optimizador**
- Storage Engines

4 Las queries

5 Escalando MySQL

El optimizador

- Utiliza la estructura de árbol para hacer optimizaciones.
- Mediante datos estadísticos del S.E. y algoritmos de optimización hace diferentes pruebas.
- Escoge la prueba que le haya dado un valor mas óptimo.
- Algunos ejemplos de algoritmos de optimización:
 - Reordenación de Joins
 - Aplicación de reglas algebraicas.
 - Evaluación y reducción de expresiones constantes.
 - ...
- Como resultado del optimizador se obtiene el "execution plan" que se aplica a los S.E.

Índice

1 Introducción

2 Esquemas e índices

3 Componentes

- El servidor
- La cache
- El parser
- El optimizador
- **Storage Engines**

4 Las queries

5 Escalando MySQL

Introducción

- Recibe peticiones simples de acceso a datos.
- Mediante estas operaciones simples se satisface el "execution plan".
- Cada S.E. tiene funcionalidades y capacidades diferentes.

MyISAM

- El tipo por defecto de MySQL.
- Muy rápido
- Lock por tablas
- Índices B-Tree y Full-Text
- Tablas comprimidas (solo lectura)

InnoDB

- Transaccional.
- Lock por filas.
- Integridad referencial.
- Índices B+Tree con clustered index.

Memory (Heap)

- Tablas en memoria.
- Muy rápidas.
- Se pierden todos los datos al reiniciar el mysql.
- Índices Hash y B-Tree.

CSV

- Lock por tabla.
- Gestiona los datos como ficheros CSV.
- Útil para gestionar datos compartidos con otras aplicaciones que solo entienden CSV.

BlackHole

- Transaccional.
- Lock por filas.
- Descarta cualquier insert.
- El sistema de logs sigue funcionando.

Archive

- Transaccional.
- Lock por filas.
- Base de datos orientada a escritura.
- Ideal para almacenar logs.

Federated

- Tabla virtual, traduce las consultas a consultas DBMS.
- Por ahora solo permite acceso a otros MySQLs.

Otros

- Maria
- BerkeleyDB
- Merge
- NDB
- Falcon
- ...

Índice

- 1 Introducción
- 2 Esquemas e índices
- 3 Componentes
- 4 **Las queries**
 - Las queries
 - Estrategia de ejecución
 - Analizando el rendimiento
- 5 Escalando MySQL

Las queries

- El cliente envía una query al servidor.
- Este comprueba si esta en la cache.
- Si esta, devuelve el resultado.
- Si no lo esta, pasa la query al parser.
- Este convierte la query en un árbol y lo pasa al optimizador.
- El optimizador procesa el árbol y obtiene un "execution plan".
- Se ejecuta el execution plan sobre los S.E. que corresponda.
- Si es necesario se procesa el resultado en una tabla temporal.
- Se almacena en la cache el resultado, y se devuelve al cliente.

Índice

- 1 Introducción
- 2 Esquemas e índices
- 3 Componentes
- 4 Las queries
 - Las queries
 - **Estrategia de ejecución**
 - Analizando el rendimiento
- 5 Escalando MySQL

Estrategia de ejecución

- Toda query en mysql es un "join".
- Mysql crea un bucle de ejecución recursivo.
- Si tiene que buscar en n tablas busca en la primera.
- Cuando encuentra un resultado empieza a buscar en la segunda.
- Así hasta la ultima, en la cual, al encontrar resultado, devuelve la fila.
- Al terminar cualquiera de las tablas, hace backtracking.

Sobre la estrategia de ejecución

- Algunas optimizaciones se entienden por la estrategia de mysql.
- Es muy importante el orden de las tablas.
- Es muy importante la elección correcta de los índices.

Índice

- 1 Introducción
- 2 Esquemas e índices
- 3 Componentes
- 4 Las queries**
 - Las queries
 - Estrategia de ejecución
 - Analizando el rendimiento**
- 5 Escalando MySQL

Errores comunes

- Solicitar mas columnas de las que necesitas.
- Solicitar mas filas de las que necesitas.
- Solicitar datos no indexados.

Analizando el rendimiento

- Log de consultas lentas
- Explain para ver el execution plan
- SHOW STATUS LIKE 'last_query_cost';
- El query cost es por lo que se guía el optimizador para elegir el execution plan.
- El profiler de mysql.

Forzando las cosas

- Normalmente no es necesario forzar las cosas.
- `STRAIGHT_JOIN` fuerza el orden de los joins al especificado en la consulta.
- `USE INDEX`, `IGNORE INDEX` o `FORCE INDEX` fuerza, usa o ignora un índice concreto.
- `SQL_CACHE` o `SQL_NO_CACHE` le dice al mysql si es o no candidato para cache.

Índice

1 Introducción

2 Esquemas e índices

3 Componentes

4 Las queries

5 Escalando MySQL

- Escalado vertical
- Escalado horizontal
- Escalado hacia atrás
- Escalado con cluster

Escalado vertical

- Ampliar o mejorar el hardware.
- MySQL no se adapta muy bien al escalado vertical.
- Depende mucho del tipo de uso que se haga del MySQL.

Índice

- 1 Introducción
- 2 Esquemas e índices
- 3 Componentes
- 4 Las queries
- 5 Escalando MySQL**
 - Escalado vertical
 - **Escalado horizontal**
 - Escalado hacia atrás
 - Escalado con cluster

Escalado horizontal

- Distribución de los datos.
- Varias opciones:
 - Replicación (y uso de los esclavos para lectura).
 - Distribución (por clave)
 - Distribución (por funcionalidad)
 - Combinación de las anteriores.

Índice

1 Introducción

2 Esquemas e índices

3 Componentes

4 Las queries

5 Escalando MySQL

- Escalado vertical
- Escalado horizontal
- **Escalado hacia atrás**
- Escalado con cluster

Escalado hacia atrás

- Eliminación de datos ya no necesarios.
- Datos de carácter histórico que pueden ser eliminados o migrados.

Índice

1 Introducción

2 Esquemas e índices

3 Componentes

4 Las queries

5 Escalando MySQL

- Escalado vertical
- Escalado horizontal
- Escalado hacia atrás
- **Escalado con cluster**

Escalado con cluster

- MySQL Cluster es una implementación de distribución de datos transparente.
- Distribuye los datos entre un conjunto de nodos.
- Da buen rendimiento para consultas simples y pocos datos.
- Se comporta mal con consultas complejas y que requieran comunicación entre nodos.

Referencias

- Manual oficial de mysql (mysql.com).
- O'Reilly - High Performance MySQL (Second Edition).
- Apress - Pro MySQL.

¿Qué queda en el tintero?

- Alta disponibilidad
- Replicación
- Optimizaciones de configuración.
- Optimizaciones de SSOO.
- Optimizaciones de Hardware.
- ...

Dudas

...

Fin

Gracias por venir.