

Diseño y Administración de Datos

Monografía de Adscripción

**Sistemas de Bases de Datos Como Soporte de Entornos de
Simulación de Algoritmos de Sistemas Operativos:**

El Caso de MySQL

Alumno: Nelson Fabián Rodríguez

Profesor: Mgter. David L. la Red Martínez

Índice

Índice.....	2
Lista de Figuras.....	4
Lista de Tablas.....	5
Introducción.....	6
¿Qué es MySQL?	6
Historia de MySQL.....	6
Características Principales.....	8
Ventajas	9
Desventajas	10
Estabilidad de MySQL.....	10
Dimensiones Máximas de las Tablas MySQL	11
El Servidor MySQL Incrustado (Embedded).....	12
Fuentes de Información Acerca de MySQL.....	12
Las Listas de Correo de MySQL.....	12
Soporte por IRC (Internet Relay Chat) de la Comunidad MySQL	13
Soporte Por Parte de la Comunidad en los Foros de MySQL	13
Cumplimiento de los Estándares Por Parte de MySQL	14
Estándares Utilizados por MySQL.....	14
Selección de Modos SQL	14
Extensiones MySQL al Estándar SQL.....	15
Diferencias en MySQL del Estándar SQL.....	15
Subconsultas	15
SELECT INTO TABLE	15
Transacciones y Operaciones Atómicas	16
Claves Foráneas (Foreing Keys)	17
Vistas.....	17
Cómo Trata MySQL las Restricciones (Constraints).....	17
Sistemas Operativos Soportados Por MySQL	18
Panorámica de Programas MySQL	20
Especificar Opciones de Programa	20
El Servidor Extendido de MySQL mysqld-max.....	21

El Gestor de Instancias de MySQL	22
El Sistema de Privilegios de Acceso de MySQL	22
Qué Hace el Sistema de Privilegios.....	22
Cómo Funciona el Sistema de Privilegios	22
Los Ficheros de Registro (Log) de MySQL	23
La Caché de Consultas de MySQL	23
Replicación en MySQL	24
Motores de Almacenamiento de MySQL y Tipos de Tablas	24
MySQL Cluster	26
Procedimientos Almacenados y Funciones	27
Disparadores (Triggers)	28
Vistas (Views)	28
Conectores	28
El Paquete de Pruebas MySQL Test.....	29
Conclusiones	29
Bibliografía Web.....	30

Lista de Figuras

Figura 1: phpMyAdmin.....	10
Figura 2: Clustering con MySQL.....	27

Lista de Tablas

Tabla 1: Tamaños máximos de archivos según Sistema Operativo.	11
Tabla 2: Soporte BDB y/o NDB.	21
Tabla 3: Archivos de Registro.	23

Introducción

Conscientes de la necesidad que la educación tiene de incorporar las tecnologías de la información y la comunicación a los procesos de enseñanza-aprendizaje en las distintas disciplinas de estudio, se propuso la utilización de herramientas web para el desarrollo de entornos de simulación de algoritmos de sistemas operativos, para ser utilizados por los alumnos de la cátedra “Sistemas Operativos” dictada en la Carrera de Lic. En Sistemas de Información perteneciente a la Facultad de Ciencias Exactas y Naturales y Agrimensura.

Para lograr el registro del uso de dicho entorno web por parte del alumnado de la cátedra es necesario la utilización de una base de datos con características de uso e implementación que soporten la tecnología de plataforma web.

En este caso se optó por el uso de la base de datos MySQL; a continuación se describen las características más importantes de este motor de base de datos.

¿Qué es MySQL?

Es un sistema de gestión de bases de datos relacional, fue creada por la empresa sueca MySQL AB, la cual tiene el copyright del código fuente del servidor SQL, como así también de la marca.

MySQL es un software de código abierto, licenciado bajo la GPL de la GNU, aunque MySQL AB distribuye una versión comercial, en lo único que se diferencia de la versión libre, es en el soporte técnico que se ofrece, y la posibilidad de integrar este gestor en un software propietario, ya que de otra manera, se vulneraría la licencia GPL.

El lenguaje de programación que utiliza MySQL es Structured Query Language (SQL) que fue desarrollado por IBM en 1981 y desde entonces es utilizado de forma generalizada en las bases de datos relacionales.

Historia de MySQL

MySQL surgió alrededor de la década del 90, Michael Widenis comenzó a usar mSQL para conectar tablas usando sus propias rutinas de bajo nivel (ISAM). Tras unas primeras pruebas, llegó a la conclusión de que mSQL no era lo bastante flexible ni rápido para lo que necesitaba, por lo que tuvo que desarrollar nuevas funciones. Esto resultó en una interfaz SQL a su base de datos, totalmente compatible a mSQL.

No se sabe con certeza el origen del nombre MySQL; por un lado se dice que en sus librerías han llevado el prefijo “my” durante los diez últimos años, por otra parte, la hija de uno de los desarrolladores se llama My. Así que no está claramente definido cuál de estas dos causas han dado lugar al nombre de este conocido gestor de bases de datos.

MySQL en los últimos años ha tenido un crecimiento vertiginoso. Es la base de datos de código abierto más popular del mundo. Código abierto significa que todo el mundo puede acceder al código fuente, es decir, al código de programación de MySQL, esto significa que también todos pueden contribuir con ideas, elementos, mejoras o

sugerir optimizaciones. Y así es que MySQL ha pasado de ser una pequeña base de datos a una completa herramienta. Su rápido desarrollo se debe en gran medida a la contribución de mucha gente al proyecto, así como la dedicación del equipo de MySQL.

A diferencia de los proyectos propietarios, en los que el código fuente es desarrollado por un número reducido de personas y se protege atentamente, los proyectos de código abierto no excluyen a nadie interesado en aportar ideas, si disponen de los conocimientos necesarios.

Lo que en un tiempo se consideró como un sencillo juguete para uso en sitios Web, se ha convertido en la actualidad en una solución viable y de misión crítica para la administración de datos.

MySQL es un sistema de administración de bases de datos relacional (RDBMS). Se trata de un programa capaz de almacenar una enorme cantidad de datos de gran variedad y de distribuirlos para cubrir las necesidades de cualquier tipo de organización, desde pequeños establecimientos comerciales a grandes empresas y organismos administrativos. MySQL compite con sistemas RDBMS propietarios como Oracle, SQL Server y DB2. Esto no quita la posibilidad de que podamos utilizarla para fines académicos.

MySQL incluye todos los elementos necesarios para instalar el programa, preparar diferentes niveles de acceso de usuario, administrar el sistema y proteger los datos. Se puede desarrollar aplicaciones de bases de datos en la mayor parte de lenguajes de programación utilizados en la actualidad y ejecutarlos en casi todos los sistemas operativos. MySQL utiliza el lenguaje de consulta estructurado (SQL).

Antes MySQL se consideraba como la opción ideal de sitios web; sin embargo, ahora incorpora muchas de las funciones necesarias para otros entornos y conserva su gran velocidad. MySQL es una base de datos robusta que se puede comparar con una base de datos comercial, es incluso más veloz en el procesamiento de las transacciones y dispone de un sistema de permisos elegante y potente, y ahora, además, incluye un motor de almacenamiento InnoDB compatible con ACID, además dispone de store procedures, triggers, vistas.

MySQL es rápido, y una solución accesible para administrar correctamente los datos de una empresa. MySQL AB es la compañía responsable del desarrollo de MySQL, dispone de un sistema de asistencia eficiente y a un precio razonable, y, como ocurre con la mayor parte de las comunidades de código abierto, se puede encontrar una gran cantidad de ayuda en la Web.

Actualmente son muchas las razones para escoger a MySQL como una solución de misión crítica para la administración de datos:

- *Costo:* MySQL es gratuito para la mayor parte de los usos y su servicio de asistencia resulta económico.
- *Asistencia:* MySQL AB ofrece contratos de asistencia a precios razonables y existe una nutrida y activa comunidad MySQL.
- *Velocidad:* MySQL es mucho más rápido que la mayoría de sus rivales.
- *Potencia:* SQL es un lenguaje muy potente para consulta de bases de datos; usar un motor ahorra una enorme cantidad de trabajo.
- *Funcionalidad:* MySQL dispone de muchas de las funciones que exigen los desarrolladores profesionales, como compatibilidad completa con ACID, compatibilidad para la mayor parte de SQL ANSI, volcados online, duplicación, funciones SSL e integración con la mayor parte de los entornos de programación.
- *Portabilidad:* MySQL se ejecuta en la inmensa mayoría de sistemas operativos y, la mayor parte de los casos, los datos se pueden transferir de un sistema a otro sin dificultad. SQL es también un lenguaje estandarizado, de modo que las consultas hechas usando SQL son fácilmente portables a otros sistemas y plataformas. Esto, unido al uso de C/C++ proporciona una portabilidad enorme.
- *Facilidad de uso:* MySQL resulta fácil de utilizar y de administrar. Las herramientas de MySQL son potentes y flexibles, sin sacrificar su capacidad de uso.
- *Seguridad:* en forma de permisos y privilegios, determinados usuarios tendrán permiso para consulta o modificación de determinadas tablas. Esto permite compartir datos sin que peligre la integridad de la base de datos o protegiendo determinados contenidos.
- *Accesibilidad:* acceso a las bases de datos de forma simultánea por varios usuarios y/o aplicaciones.

Características Principales

Inicialmente, MySQL carecía de algunos elementos esenciales en las bases de datos relacionales, tales como integridad referencial y transacciones. A pesar de esto, atrajo a los desarrolladores de páginas web con contenido dinámico, debido a su simplicidad, de tal manera que los elementos faltantes fueron complementados por la vía de las aplicaciones que la utilizan. Poco a poco estos elementos faltantes, están siendo incorporados tanto por desarrolladores internos, como por desarrolladores de software libre.

En las últimas versiones se pueden destacar las siguientes características principales:

- El principal objetivo de MySQL es velocidad y robustez.
- Soporta gran cantidad de tipos de datos para las columnas.
- Gran portabilidad entre sistemas, puede trabajar en distintas plataformas y sistemas operativos.
- Cada base de datos cuenta con 3 archivos: Uno de estructura, uno de datos y uno de índice y soporta hasta 32 índices por tabla.

- Aprovecha la potencia de sistemas multiproceso, gracias a su implementación multihilo.
- Flexible sistema de contraseñas (passwords) y gestión de usuarios, con un muy buen nivel de seguridad en los datos.
- El servidor soporta mensajes de error en distintas lenguas.

Ventajas

Ya se ha visto que para acceder a bases de datos es mucho más útil usar un motor o servidor que hace las funciones de intérprete entre las aplicaciones y usuarios con las bases de datos.

Esta utilidad se traduce en ventajas, entre las que podemos mencionar las siguientes:

- *Accesibilidad*: acceso a las bases de datos de forma simultánea por varios usuarios y/o aplicaciones.
- *Seguridad*: en forma de permisos y privilegios, determinados usuarios tendrán permiso para consulta o modificación de determinadas tablas. Esto permite compartir datos sin que peligre la integridad de la base de datos o protegiendo determinados contenidos.
- *Potencia*: SQL es un lenguaje muy potente para consulta de bases de datos, usar un motor nos ahorra una enorme cantidad de trabajo.
- *Portabilidad*: SQL es también un lenguaje estandarizado, de modo que las consultas hechas usando SQL son fácilmente portables a otros sistemas y plataformas. Esto, unido al uso de C/C++ proporciona una portabilidad enorme. MySQL está escrito en C y C++ y probado con multitud de compiladores y dispone de APIs para muchas plataformas diferentes.
- *Escalabilidad*: es posible manipular bases de datos enormes, del orden de seis mil tablas y alrededor de cincuenta millones de registros, y hasta 32 índices por tabla.
- *Conectividad*: es decir, permite conexiones entre diferentes máquinas con distintos sistemas operativos. Es corriente que servidores Linux o Unix, usando MySQL, sirvan datos para ordenadores con Windows, Linux, Solaris, etc. Para ello se usa TCP/IP, tuberías, o sockets Unix.
- *Facilidad multihilo*: puede beneficiarse de sistemas multiprocesador.
- *Tipos*: permite manejar multitud de tipos para columnas.
- *Registros*: permite manejar registros de longitud fija o variable.
- *Rendimiento*: velocidad al realizar las operaciones, lo que le hace uno de los gestores con mejor rendimiento.
- *Costo*: bajo costo en requerimientos para la elaboración de bases de datos, ya que debido a su bajo consumo puede ser ejecutado en una máquina con escasos recursos sin ningún problema.
- *Facilidad*: facilidad de configuración e instalación.
- *Multiplataforma*: soporta gran variedad de Sistemas Operativos.
- *Confiabilidad*: baja probabilidad de corromper datos, incluso si los errores no se producen en el propio gestor, sino en el sistema en el que está.
- *Conectividad y seguridad*.

Desventajas

Un gran porcentaje de las utilidades de MySQL no están documentadas.

No es intuitivo, como otros programas (ACCESS).

Con el fin de hacer el manejo de MySQL un poco más "amigable", hay programas como el PHPMyAdmin, que es un front-end vía web para el acceso a las bases de datos y sus tablas.

Depende del servidor web Apache por lo que pedirá instalarlo también. Para acceder, una vez arrancado el Apache, basta con poner en el navegador `http://localhost/phpmyadmin` y pedirá un nombre de usuario y una contraseña para el acceso a las bases de datos, como se ha tenido que poner en el `.my.cnf` por ejemplo (ver Figura 1).

Una vez que se permite el acceso, se tiene la posibilidad de consultar las tablas, insertar filas, modificarlas... y muchas otras opciones vía web y muy cómodamente.

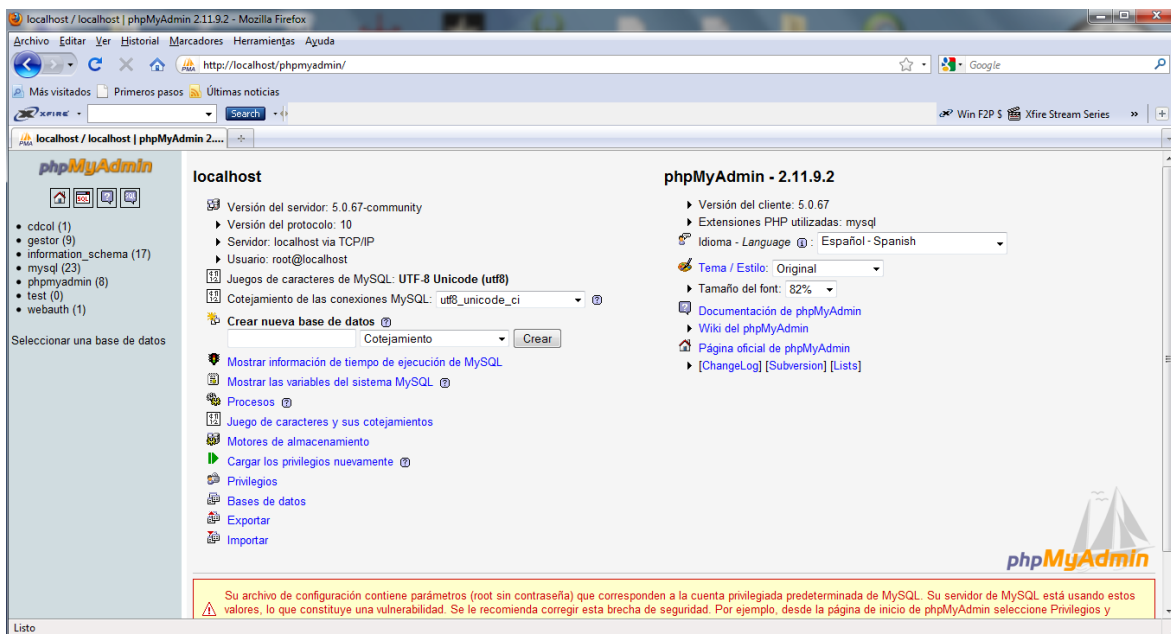


Figura 1: phpMyAdmin.

Estabilidad de MySQL

El código original se remonta a los principios de los años 80. En TcX, la predecesora de MySQL AB, el código MySQL ha funcionado en proyectos desde mediados de 1996 sin ningún problema. Cuando el software de base de datos MySQL fue distribuido entre un público más amplio, los nuevos usuarios rápidamente encontraron trozos de código no probados. Cada nueva versión desde entonces ha tenido pocos problemas de portabilidad incluso considerando que cada nueva versión ha tenido muchas nuevas funcionalidades.

El diseño de MySQL Server es multi capa, con módulos independientes. Algunos de los últimos módulos se listan a continuación con una indicación de su situación de testeo:

- Replicación (estable): hay grandes grupos de servidores usando replicación en producción, con buenos resultados. Se trabaja para mejorar características de replicación en MySQL 5.x.
- InnoDB tablas (estable): el motor de almacenamiento transaccional InnoDB es estable y usado en grandes sistemas de producción con alta carga de trabajo.
- BDB tablas (estable): el código Berkeley DB es muy estable, pero todavía se lo está mejorando con el interfaz del motor de almacenamiento transaccional BDB en MySQL Server.
- Búsquedas Full-text (estable): la búsquedas Full-text es ampliamente usada.
- MyODBC 3.51 (estable): MyODBC 3.51 usa ODBC SDK 3.51 y es usado en sistemas de producción ampliamente. Algunas cuestiones surgidas parecen ser cuestión de las aplicaciones que lo usan e independientes del controlador ODBC o la base de datos subyacente.

Dimensiones Máximas de las Tablas MySQL

En MySQL 5.0, usando el motor de almacenamiento MyISAM, el máximo tamaño de las tablas es de 65536 terabytes. Por lo tanto, el tamaño efectivo máximo para las bases de datos en MySQL usualmente los determinan los límites de tamaño de ficheros del sistema operativo, y no por límites internos de MySQL.

El motor de almacenamiento InnoDB mantiene las tablas en un espacio que puede ser creado a partir de varios ficheros. Esto permite que una tabla supere el tamaño máximo individual de un fichero. Este espacio puede incluir particiones de disco, lo que permite tablas extremadamente grandes. El tamaño máximo del espacio de tablas es 64 TB.

La siguiente tabla lista algunos ejemplos de límites de tamaño de ficheros de sistemas operativos. Esto es sólo una burda guía y no pretende ser definitiva. Para la información más actual, asegúrese de consultar la documentación específica de su sistema operativo (ver Tabla 1).

Sistema operativo	Tamaño máximo de fichero
Linux 2.2-Intel 32-bit	2GB (LFS: 4GB)
Linux 2.4	(usando sistema de ficheros ext3) 4TB
Solaris 9/10	16TB
Sistema de ficheros NetWare w/NSS	8TB
win32 w/ FAT/FAT32	2GB/4GB
win32 w/ NTFS	2TB (posiblemente mayor)
MacOS X w/ HFS+	2TB

Tabla 1: Tamaños máximos de archivos según Sistema Operativo.

En Linux 2.2, se puede utilizar tablas MyISAM mayores de 2 GB usando el parche para LFS (Large File Support) en el sistema de ficheros ext2. En Linux 2.4 y posteriores, existen parches para ReiserFS soportando grandes archivos (hasta 2 TB). La mayoría de distribuciones Linux se basan en el kernel 2.4 o 2.6 e incluyen todos los parches LFS

necesarios. Con JFS y XFS, se permiten ficheros mayores de un petabyte para Linux. Sin embargo, el tamaño máximo de ficheros todavía depende de diversos factores, uno de ellos el sistema de ficheros usado para almacenar tablas MySQL.

Otros métodos para cambiar los límites de tamaño de ficheros para tablas MyISAM son:

- Si una tabla es de sólo lectura, puede usar myisampack para comprimirla, myisampack normalmente comprime una tabla al menos un 50%, lo que permite, a efectos prácticos, tablas mucho mayores, myisampack puede mezclar múltiples tablas en una misma tabla.
- MySQL incluye la biblioteca MERGE que permite tratar una colección de tablas MyISAM con una estructura idéntica en una tabla MERGE.

El Servidor MySQL Incrustado (Embedded)

La biblioteca del servidor incrustado libmysqld permite que MySQL Server pueda trabajar con una gran cantidad de dominios de aplicaciones. Usando esta biblioteca, los desarrolladores pueden añadir MySQL Server en varias aplicaciones y dispositivos electrónicos, donde el usuario final no tiene conocimiento que hay una base de datos subyacente.

MySQL Server incrustado es ideal para uso tras aplicaciones en Internet, kioscos públicos, combinación de hardware/software en llaveros, servidores de alto rendimiento de Internet, bases de datos autocontenidas distribuidas en CD-ROM, y así, muchos usuarios de libmysqld se benefician de la licencia dual de MySQL. Para los que no quieran estar ligados a la licencia GPL, el software está disponible con licencia comercial.

Fuentes de Información Acerca de MySQL

Las Listas de Correo de MySQL

Las listas de correo de MySQL son las siguientes:

- **anuncios:** esta lista es para anuncios de nuevas versiones de MySQL y programas relacionados. Esta es una lista de tráfico bajo y a la que todos los usuarios de MySQL deberían suscribirse.
- **Mysql:** esta es la lista principal para discusión sobre MySQL en general. Hay que tener en cuenta que algunas cuestiones es mejor discutir las en listas más especializadas. Si se postea en una lista equivocada, podría no obtenerse respuesta.
- **Bugs:** esta lista es para gente que desee estar informada sobre cuestiones reportadas desde la última versión de MySQL o que deseen estar activamente implicadas en el proceso de buscar bugs y arreglarlos.
- **temas internos:** esta lista es para gente que trabaja en el código de MySQL. Este también es el forum para discutir acerca del desarrollo de MySQL y para publicar parches.
- **Mysqldoc:** esta lista es para gente que trabaja en la documentación de MySQL: gente de MySQL AB, traductores, y otros miembros de la comunidad.
- **pruebas de rendimiento:** esta lista es para cualquiera interesado en temas de rendimiento. La discusión se concentra en rendimiento de bases de datos (no sólo

de MySQL), pero también incluye categorías más amplias como rendimiento del kernel, sistema de ficheros, tipos de discos, etc.

- empaquetadores: esta lista es para discusiones acerca de empaquetar y distribuir MySQL. Este es el forum usado por mantenedores de distribuciones para intercambiar ideas sobre empaquetar MySQL y asegurar que MySQL parece tan similar como sea posible en todas las plataformas y sistemas operativos soportados.
- Java: esta lista es para discusiones acerca de MySQL Server y Java. Normalmente se usa para discutir acerca de JDBC, incluyendo el connector/J de MySQL.
- win32: esta lista es para todos los temas acerca del software MySQL en sistemas operativos Microsoft, tales como Windows 9x, Me, NT, 2000, SP y 2003.
- Myodbc: esta lista es para todos los tópicos acerca de conectar al MySQL Server con ODBC.
- herramientas gui: esta lista es para todos los temas acerca de herramientas GUI MySQL, incluyendo MySQL Administrator y el cliente gráfico MySQL Control Center.
- Cluster: esta lista es para discusión acerca de MySQL Cluster.
- Dotnet: esta lista es para discusión acerca de MySQL Server y la plataforma .NET. La mayoría de discusiones es acerca del Connector/NET MySQL.
- Plusplus: esta lista es para tópicos acerca de programación con la API C++ para MySQL.
- Perl: esta lista es para tópicos acerca de soporte Perl para MySQL con DBD:mysql.

Soporte por IRC (Internet Relay Chat) de la Comunidad MySQL

Adicionalmente a las listas de correo MySQL, puede encontrar una comunidad experimentada en IRC (Internet Relay Chat). Estos son canales que algunos de los canales conocidos:

- freenode (consulte <http://www.freenode.net/> para servidores).
- #mysql Básicamente preguntas sobre MySQL , pero también de otras bases de datos y preguntas generales sobre SQL. Preguntas sobre PHP, Perl o C en combinación con MySQL son frecuentes.

Soporte Por Parte de la Comunidad en los Foros de MySQL

El último recurso de soporte para la comunidad son los foros en <http://forums.mysql.com>.

Hay una variedad de foros disponibles, agrupados en las siguientes categorías generales:

- Migración.
- Uso de MySQL.
- Conectores MySQL.
- Tecnología MySQL.
- Negocios.

Cumplimiento de los Estándares Por Parte de MySQL

El estándar SQL ha ido evolucionando desde 1986 y existen varias versiones. Frecuentemente, "SQL:92" se refiere al estándar publicado en 1992, "SQL:1999" se refiere al estándar publicado en 1999, y "SQL:2003" se refiere a la versión actual del estándar.

Se usará la frase "el estándar SQL" para referirnos a la versión actual del estándar SQL en cualquier momento.

Uno de los fines principales del producto es continuar el trabajo hacia el cumplimiento del estándar SQL, pero sin sacrificar velocidad o fiabilidad. Se pueden añadir extensiones a SQL o soporte para funcionalidades no SQL si esto aumenta la usabilidad de MySQL Server para un gran número de usuarios. La interfaz HANDLER en MySQL Server 4.0 es un ejemplo de esta estrategia.

Continúa soportando bases de datos transaccionales y no transaccionales para satisfacer uso crítico 24/7 y uso pesado en entornos Web o log.

MySQL Server fue diseñado originalmente para trabajar con bases de datos de tamaño medio (de 10 a 100 millones de registros, o unos 100 MB por tabla) en máquinas pequeñas.

Hoy MySQL Server soporta bases de datos de tamaño de terabytes, pero el código todavía puede compilarse en una versión reducida adecuada para dispositivos hand-held o incrustados. El diseño compacto de MySQL Server hace el desarrollo en ambas direcciones posible sin ningún conflicto en el árbol fuente.

Actualmente, no se trata soporte en tiempo real, aunque la capacidad de replicación en MySQL ofrece funcionalidades significativas.

Existe soporte para clusters de bases de datos a través de soluciones de terceras partes, así como la integración de tecnología NDB Cluster, disponible desde la versión 4.1.2.

Estándares Utilizados por MySQL

- Se está intentando soportar el estándar ANSI/ISO completamente, pero sin hacer concesiones a la velocidad y calidad del código.
- Niveles ODBC 0-3.51.

Selección de Modos SQL

MySQL Server puede operar en distintos modos SQL y puede aplicar dichos modos de forma diferente para distintos clientes. Esto permite a una aplicación adaptar el funcionamiento del servidor a sus propios requerimientos.

Los modos definen la sintaxis que MySQL debe soportar y qué clase de validaciones debe efectuar a los datos. Esto hace más fácil usar MySQL en un conjunto de entornos diferentes y usar MySQL junto con otros servidores de bases de datos.

Puede inicializar el modo SQL por defecto inicializando mysqld con la opción `--sql-mode="modes"`. Empezando en MySQL 4.1., se puede cambiar el modo tras inicializar mediante la variable `sql_mode` con un comando `SET [SESSION|GLOBAL] sql_mode='modes'`.

Extensiones MySQL al Estándar SQL

MySQL Server incluye algunas extensiones que probablemente no se encontrará en otras bases de datos SQL. Se debe tener en cuenta que si se lo usa, su código no será portable a otros servidores SQL. En algunos casos, se puede escribir código que incluya extensiones MySQL, pero siendo portable, mediante comentarios de la forma `/*! ... */`. En ese caso, MySQL parsea y ejecuta el código dentro de los comentarios como si fuera cualquier otro comando de MySQL, pero otros servidores SQL ignorarán la extensión. Por ejemplo:

```
SELECT /*! STRAIGHT_JOIN */ col_name FROM table1,table2 WHERE ...
```

Si se añade un número de versión tras el carácter '!', la sintaxis dentro del comentario se ejecuta sólo si el número de versión de MySQL es igual o mayor que el especificado:

```
CREATE /*!32302 TEMPORARY */ TABLE t (a INT);
```

Eso significa que si se tiene la Versión 3.23.02 o posterior, MySQL Server usa la palabra clave TEMPORARY.

Diferencias en MySQL del Estándar SQL

Se trata que MySQL Server siga los estándares ANSI SQL y el estándar ODBC SQL, pero MySQL Server ejecuta operaciones de forma distinta en algunos casos:

- Para columnas VARCHAR, los espacios finales se eliminan cuando el valor se guarda. (Arreglado en MySQL 5.0.3).
- En algunos casos, las columnas de tipo CHAR se convierten en columnas VARCHAR cuando define una tabla o altera su estructura. (Arreglado en MySQL 5.0.3).
- Los privilegios para una tabla no se eliminan automáticamente cuando se borra una tabla. Se debe usar explícitamente un comando REVOKE para quitar los privilegios de una tabla.
- La función CAST() no soporta conversión a REAL o BIGINT.
- SQL estándar necesita que las cláusulas HAVING en un comando SELECT puedan referirse a columnas en la cláusula GROUP BY. Esto no se permite antes de la versión MySQL 5.0.2.

Subconsultas

MySQL 4.1 soporta sub-consultas y tablas derivadas. Una "sub-consulta" es un comando SELECT anidado en otro comando. Una tabla "derivada" (una vista sin nombre) es una subconsulta en la cláusula FROM de otra consulta de subconsultas".

Para versiones MySQL anteriores a la 4.1, la mayoría de subconsultas pueden reescribirse usando joins u otros métodos.

SELECT INTO TABLE

MySQL Server no soporta la sintaxis de extensiones Sybase SQL:

```
SELECT ... INTO TABLE ....
```

En su lugar, MySQL Server soporta la sintaxis estándar SQL:

`INSERT INTO ... SELECT ...`, que básicamente es lo mismo.

Transacciones y Operaciones Atómicas

MySQL Server (versiones 3.23-max y todas las versiones 4.0 y posteriores) soportan transacciones con los motores transaccionales InnoDB y BDB . InnoDB proporciona completa compatibilidad ACID.

Los otros motores no transaccionales en MySQL Server (como MyISAM) siguen un paradigma diferente para integridad de datos llamado "operaciones atómicas". En términos transaccionales, tablas MyISAM operan en modo `AUTOCOMMIT=1`. Operaciones atómicas a menudo ofrecen integridad comparable con mejor rendimiento.

MySQL Server soporta ambos paradigmas, puede decidir si su aplicación necesita la velocidad de operaciones atómicas o el uso de características transaccionales. Esta elección puede hacerse para cada tabla.

Como se ha dicho, el compromiso entre tipos de tablas transaccionales y no transaccionales reside principalmente en el rendimiento. Tablas transaccionales tienen requerimientos significativamente mayores para memoria y espacio de disco, y mayor carga de CPU. Por otra parte, tipos de tablas transaccionales como InnoDB también ofrecen muchas características significativas. El diseño modular de MySQL Server permite el uso concurrente de distintos motores de almacenamiento para cumplir distintos requerimientos y mostrarse óptimo en todas las situaciones.

El paradigma transaccional tiene sus ventajas y desventajas. Muchos usuarios y desarrolladores de aplicaciones dependen en la facilidad con la que pueden solucionar problemas donde un aborto parece ser o es necesario. Sin embargo, incluso si el paradigma de operaciones atómicas es desconocido o se está más familiarizado con las transacciones, se debe considerar el beneficio de la velocidad que pueden ofrecer las tablas no transaccionales, que puede ser de tres a cinco veces más rápido que las más optimizadas tablas transaccionales.

En las situaciones en las que la integridad es de máxima importancia, MySQL Server ofrece integridad a nivel de transacción incluso para tablas no transaccionales. Si se bloquea tablas con `LOCK TABLES`, todas las actualizaciones se bloquean hasta que se hacen las comprobaciones necesarias. Si se obtiene un bloqueo `READ LOCAL` (el contrario a un bloqueo de escritura) para una tabla que permita inserciones concurrentes al final de la tabla, las lecturas están permitidas, así como las inserciones de otros clientes. Los registros insertados no puede verlos el cliente que tenga el bloqueo hasta que lo libere. Con `INSERT DELAYED`, se puede encolar inserciones en una cola local, hasta que los bloqueos se liberan, sin tener que esperar el cliente a que acabe la inserción.

"Atómico", en el sentido en que nos referimos, no es nada mágico. Se trata que puede asegurar que mientras cada actualización específica está ejecutándose, ningún otro usuario puede interferir con ellas, y que nunca puede haber un rollback automático (lo que puede ocurrir con tablas transaccionales si no se es muy cuidadoso). MySQL Server garantiza que no hay dirty reads (lecturas sucias).

Claves Foráneas (Foreign Keys)

En MySQL Server 3.23.44 y posteriores, el motor InnoDB soporta chequeo para restricciones de claves foráneas, incluyendo CASCADE, ON DELETE, y ON UPDATE.

Restricciones de claves foráneas ofrecen distintos beneficios a los diseñadores de bases de datos:

- Suponiendo un diseño adecuado de las relaciones, las restricciones de claves foráneas hacen más difícil que un programador introduzca inconsistencias en la base de datos.
- Chequeo centralizado de restricciones por el servidor de base de datos hace que sea innecesario realizar esos chequeos en la parte de la aplicación, eliminando la posibilidad que distintas aplicaciones puedan no chequear todas las restricciones de la misma forma.
- Usando actualizaciones y borrados en cascada puede simplificarse el código de aplicación.
- Reglas diseñadas correctamente para claves foráneas pueden ayudar a documentar las relaciones entre tablas.

Hay que tener en cuenta que estos beneficios tienen el coste de un trabajo adicional para el servidor de base de datos para poder realizar todas las comprobaciones necesarias. Chequeos adicionales por parte del servidor afectan al rendimiento, lo que puede ser lo suficientemente malo para algunas aplicaciones como para evitarlo todo lo posible. (Algunas grandes aplicaciones comerciales han codificado la lógica de claves foráneas en el nivel de aplicación por esta razón).

MySQL proporciona a diseñadores de bases de datos la posibilidad de elegir qué paradigma elegir.

Se debe tener en cuenta que el uso de claves foráneas puede provocar algunos problemas:

- El soporte de claves foráneas arregla muchas cuestiones relacionadas con la integridad, pero todavía es necesario diseñar las claves cuidadosamente para evitar reglas circulares o combinaciones incorrectas de borrados en cascada.

Vistas

Las vistas son útiles para permitir acceder a los usuarios a un conjunto de relaciones (tablas) como si fueran una sola, y limitar su acceso a las mismas. También se pueden usar las vistas para restringir el acceso a registros (un subconjunto de una tabla particular).

Para control de acceso a columnas, puede usar el sofisticado sistema de privilegios de MySQL Server.

Cómo Trata MySQL las Restricciones (Constraints)

MySQL permite trabajar con tablas transaccionales, que permiten hacer un rollback, y con tablas no transaccionales que no lo permiten. Es por ello que las restricciones son algo distintas en MySQL respecto a otras bases de datos. Se debe tratar el caso en el que se

insertan o actualizan muchos registros en una tabla no transaccional en la que los cambios no pueden deshacerse cuando ocurre un error.

La filosofía básica es que MySQL Server trata de producir un error para cualquier cosa que detecte mientras parsea un comando que va a ejecutarse, y trata de recuperarse de cualquier error que ocurra mientras se ejecuta el comando. MySQL lo hace en la mayoría de casos, pero todavía no en todos.

Las opciones en MySQL cuando ocurre un error son parar el comando en medio de la ejecución o recuperarse lo mejor posible del problema y continuar. Por defecto, el servidor utiliza esta última opción. Esto significa, por ejemplo, que el servidor puede cambiar algunos valores ilegales por el valor legal más próximo.

A partir de la versión 5.0.2 de MySQL, hay disponibles varios modos SQL para proporcionar un mayor control sobre cómo aceptar valores incorrectos y si continuar ejecutando el comando o abortarlo cuando ocurre el error. Usando estas opciones, puede configurar MySQL Server para actuar en un modo más tradicional como otros servidores de bases de datos que rechazan datos incorrectos.

Los modos SQL pueden cambiarse en tiempo de ejecución, lo que permite a los clientes individuales seleccionar el comportamiento más apropiado para sus requerimientos.

Sistemas Operativos Soportados Por MySQL

Se ha utilizado GNU Autoconfig, de modo que es posible portar MySQL a todos los sistemas modernos que tengan un compilador de C++ y una implementación funcional de subprocesos (threads) POSIX. El soporte de subprocesos es necesario para el servidor para compilar únicamente el código del cliente, no se requiere más que el compilador de C++.

MySQL ha sido compilado correctamente en las siguientes combinaciones de sistemas operativos y paquetes de subprocesos. Se puede ver que, para varios sistemas operativos, el soporte nativo de subprocesos funciona solamente en las versiones más recientes.

- AIX 4.x, 5.x con subprocesos nativos.
- BSDI 2.x with con el paquete MIT-pthreads.
- BSDI 3.0, 3.1 y 4.x con subprocesos nativos.
- Digital Unix 4.x con subprocesos nativos.
- FreeBSD 2.x con el paquete MIT-pthreads.
- FreeBSD 3.x and 4.x con subprocesos nativos.
- FreeBSD 4.x con LinuxThreads.
- HP-UX 10.20 con el paquete DCE threads o MIT-pthreads.
- HP-UX 11.x con subprocesos nativos.
- Linux 2.0+ con LinuxThreads 0.7.1+ o glibc 2.0.7+ para varias arquitecturas de CPU.
- Mac OS X. Consulte Sección 2.12.2, “Notas sobre Mac OS X”.
- NetBSD 1.3/1.4 Intel y NetBSD 1.3 Alpha (requiere GNU make).
- Novell NetWare 6.0. Consulte Sección 2.6, “Instalar MySQL sobre NetWare”.
- OpenBSD > 2.5 con subprocesos nativos. OpenBSD < 2.5 con el paquete MIT-pthreads.

- OS/2 Warp 3, FixPack 29 y OS/2 Warp 4, FixPack 4.
- SCO OpenServer 5.0.X con una versión del paquete FSU Pthreads.
- SCO UnixWare 7.1.x.
- SCO Openserver 6.0.x.
- SGI Irix 6.x con subprocesos nativos.
- Solaris 2.5 y posteriores con subprocesos nativos en SPARC y x86.
- SunOS 4.x con el paquete MIT-pthreads package.
- Tru64 Unix.
- Windows 9x, Me, NT, 2000, XP, y 2003.

No todas las plataformas son igualmente aptas para ejecutar MySQL. Los siguientes factores determinan si una plataforma está más o menos bien preparada para un servidor MySQL con alto volumen de carga y para misiones crítica:

- Estabilidad general de la biblioteca de subprocesos. Una plataforma puede tener una excelente reputación en otras situaciones, pero MySQL es estable como lo sea la biblioteca de subprocesos que utiliza la plataforma, aun cuando cualquier otro aspecto sea perfecto.
- La capacidad del núcleo o kernel del sistema operativo y de la biblioteca de subprocesos para aprovechar sistemas de multiprocesamiento simétrico (SMP). En otras palabras, cuando un proceso crea un subproceso, éste debería poderse ejecutar en una CPU diferente a la del proceso original.
- La capacidad del núcleo o kernel del sistema operativo y de la biblioteca de subprocesos para ejecutar varios subprocesos que bloquean y liberan mutexes frecuentemente en una pequeña región crítica sin excesivos cambios de contexto. Si la implementación de `pthread_mutex_lock()` es muy proclive a consumir tiempo de CPU, esto afectará en gran manera a MySQL. Si no se previene este problema, añadir más CPUs hará todavía más lento a MySQL.
- El rendimiento y la estabilidad general del sistema de ficheros.
- Si se emplean grandes tablas, la capacidad del sistema de ficheros para gestionar eficientemente archivos de gran tamaño.
- El nivel de experiencia que los desarrolladores de MySQL AB posean sobre una determinada plataforma. Si la conocen bien, habilitan optimizaciones específicas y soluciones en tiempo de compilación. Además pueden proporcionar consejos sobre cómo configurar el sistema en forma óptima para MySQL.
- El volumen de pruebas realizadas por MySQL AB sobre configuraciones similares.
- La cantidad de usuarios que han ejecutado MySQL con éxito en la misma plataforma y en configuraciones similares. Si este número es alto, las probabilidades de encontrar sorpresas específicas de la plataforma son mucho menores.

En base a estos criterios, las mejores plataformas para ejecutar MySQL en este momento son x86 con SuSE Linux (kernel versión 2.4 o 2.6), y ReiserFS (o cualquier distribución de Linux similar) y SPARC con Solaris (2.7-9). FreeBSD aparece en tercer lugar, pero es de esperar que se integre al lote principal cuando se mejore la biblioteca de subprocesos. También las otras plataformas donde MySQL se compila y ejecuta en la actualidad podrían

ser incluidas en la categoría principal, pero no con el mismo nivel de estabilidad y rendimiento. Esto requiere un esfuerzo por parte de los desarrolladores de MySQL en cooperación con los desarrolladores de los sistemas operativos y de bibliotecas de componentes de las que depende MySQL.

Panorámica de Programas MySQL

MySQL AB proporciona varios tipos de programas:

El servidor MySQL y los scripts de inicio del servidor:

- `mysqld` es el servidor MySQL.
- `mysqld_safe`, `mysql.server`, y `mysqld_multi` son scripts de inicio del servidor.
- `mysql_install_db` inicializa el directorio "data" y las bases de datos que MySQL instala por defecto.

Programas cliente que acceden al servidor:

- `mysql` es un programa cliente que proporciona una interfaz de línea de comandos para ejecutar sentencias SQL en modo interactivo o por lotes.
- `mysqladmin` es un cliente para administración.
- `mysqlcheck` ejecuta operaciones de mantenimiento de tablas.
- `mysqldump` y `mysqlhotcopy` son utilidades para copia de respaldo.
- `mysqlimport` realiza importación de ficheros de datos.
- `mysqlshow` muestra información relativa a tablas y bases de datos.

Programas que operan independientemente del servidor:

- `myisamchk` ejecuta operaciones de mantenimiento de tablas.
- `myisampack` genera tablas comprimidas, de sólo lectura.
- `mysqlbinlog` es una herramienta para procesar archivos de registro binario (binary logs).
- `mysqlerror` informa el significado de un código de error.

Especificar Opciones de Programa

Al ejecutar programas MySQL se les pueden indicar opciones en varias formas:

- En la línea de comandos, a continuación del nombre del programa. Este es el modo más común para opciones que se aplican a una ejecución específica del programa.
- En un fichero de opciones, que el programa lee al iniciarse. Esto es lo acostumbrado para opciones que se desea que el programa use cada vez que se ejecuta.
- En variables de entorno. Esto es útil para opciones que se desean aplicar cada vez que el programa se ejecuta, si bien en la práctica, para este propósito, es más común emplear ficheros de opciones.

Los programas MySQL determinan qué opciones les fueron suministradas examinando en primer lugar las variables de entorno, luego los ficheros de opciones, y, finalmente, la línea de comandos. Si una opción se especifica más de una vez, la última tiene precedencia. Esto

significa que las variables de entorno tienen la prioridad más baja, y la línea de comandos, la más alta.

El Servidor Extendido de MySQL `mysqld-max`

El servidor MySQL-Max es una versión del servidor MySQL `mysqld` compilada para añadir características adicionales.

La distribución a usar depende de la plataforma:

- Para Windows, las distribuciones binarias de MySQL incluyen ambos servidores (`mysqld.exe`) y el servidor MySQL-Max (`mysqld-max.exe`), por lo que no es necesario adquirir ninguna distribución especial.
- Para Linux, si se instala MySQL utilizando una distribución RPM, se debe usar el RPM MySQL-server en primer lugar para instalar una versión estándar del servidor llamada `mysqld`. A continuación se debe usar el RPM MySQL-Max para instalar el servidor llamado `mysqld-max`. El RPM MySQL-Max presupone que el RPM con el servidor normal está instalado.
- Todas las otras distribuciones MySQL-Max contienen un único servidor llamado `mysqld` pero que tiene las características adicionales incluidas.

Las distribuciones binarias de MySQL-Max son útiles para aquéllos que quieran instalar programas precompilados. Si se compila MySQL a partir de una distribución fuente, se puede construir un servidor propio de estilo Max activando las mismas características en tiempo de configuración que usan las distribuciones binarias de MySQL-Max al ser creadas.

Los servidores MySQL-Max incluyen el motor de almacenamiento BerkeleyDB (BDB) cuando es posible, pero no todas las plataformas soportan BDB. Los servidores MySQL-Max para Solaris, Mac OS X, y Linux (en la mayoría de plataformas) incluyen soporte para el motor de almacenamiento NDB Cluster. Hay que tener en cuenta que el servidor debe reiniciarse con la opción `ndbcluster` para ejecutar el servidor como parte de un MySQL Cluster. La Tabla 2 muestra en qué plataformas los binarios de MySQL-Max incluyen soporte para BDB y/o NDB Cluster.

Sistema	Soporte BDB	Soporte NDB
AIX 4.3	N	N
HP-UX 11.0	N	N
Linux-Alpha	N	S
Linux-IA-64	N	N
Linux-Intel	S	S
Mac OS X	N	N
NetWare	N	N
SCO OSR5	S	N
Solaris-SPARC	S	S
Solaris-Intel	N	S
UnixWare	S	N
Windows NT/2000/XP	S	N

Tabla 2: Soporte BDB y/o NDB.

El Gestor de Instancias de MySQL

El MySQL Instance Manager (IM), es el demonio que corre en el puerto TCP/IP, el cual provee monitoreo y administración de las instancias del servidor de datos MySQL. MySQL Instance Manager está disponible para sistemas operativos basados en Unix.

MySQL Instance Manager se incluye en las distribuciones de MySQL desde la versión 5.0.3, y puede usarse en lugar del script `mysqld_safe` para arrancar y parar MySQL Server, incluso desde una máquina remota. MySQL Instance Manager implementa la funcionalidad (y la mayoría de la sintaxis) del script `mysqld_multi`.

El típico ciclo de arranque/cierre para un servidor MySQL con el MySQL Instance Manager habilitado es como sigue:

- El MySQL Instance Manager se arranca con el script `/etc/init.d/mysql`.
- El MySQL Instance Manager arranca todas las instancias y las monitoriza.
- Si una instancia de un servidor cae, el MySQL Instance Manager la reinicia.
- Si el MySQL Instance Manager se cierra (por ejemplo con el comando `/etc/init.d/mysql stop`), todas las instancias se apagan con el MySQL Instance Manager.

El Sistema de Privilegios de Acceso de MySQL

Qué Hace el Sistema de Privilegios

La función primaria del sistema de privilegios de MySQL es autenticar un usuario conectándose desde un equipo dado, y asociar dicho usuario con privilegios en una base de datos tales como `SELECT`, `INSERT`, `UPDATE`, y `DELETE`.

Funcionalidad adicional incluye la habilidad de tener usuarios anónimos y de dar privilegios para funciones específicas de MySQL tales como `LOAD DATA INFILE` y operaciones administrativas.

Cómo Funciona el Sistema de Privilegios

El sistema de privilegios de MySQL asegura que todos los usuarios pueden ejecutar sólo la operación permitida a los mismos. Como usuario, cuando se conecta a un servidor MySQL, la identidad se determina mediante el equipo desde el que se conecta y el nombre de usuario que se especifique. Cuando se efectúen peticiones tras conectar, el sistema otorga privilegios acorde a la identidad y lo que se quiera hacer.

MySQL considera tanto el nombre de usuario y el equipo a la hora de identificar, ya que no hay razón para asumir que un nombre de usuario pertenece a la misma persona en cualquier sitio de Internet. Por ejemplo, el usuario `joe` que conecta desde `office.com` no tiene porqué ser la misma persona que el usuario `joe` que conecta desde `elsewhere.com`. MySQL trata esto permitiendo distinguir usuarios en diferentes equipos que tienen el mismo nombre. Se puede otorgar un conjunto de privilegios para conexiones de `joe` desde `office.com`, y un conjunto distinto para conexiones de `joe` desde `elsewhere.com`.

El control de acceso de MySQL implica dos etapas:

- El servidor comprueba si debe permitir la conexión.
- Asumiendo que se conecta, el servidor comprueba cada comando que se ejecuta para ver si tiene suficientes permisos para hacerlo. Por ejemplo, si se intenta seleccionar registros de una tabla en una base de datos o eliminar una tabla de la base de datos, el servidor verifica que se tenga el permiso SELECT para la tabla o el permiso DROP para la base de datos.

Si los permisos cambian mientras se está conectado, estos cambios no tienen porqué tener efecto inmediatamente para el siguiente comando que se ejecute.

El servidor guarda información de privilegios en las tablas de permisos de la base de datos mysql (esto es, en la base de datos llamada mysql). El servidor MySQL lee el contenido de dichas tablas en memoria cuando arranca y las vuelve a leer nuevamente cada vez que se conecta el cliente. Las decisiones acerca de control de acceso se basan en las copias en memoria de las tablas de permisos.

Los Ficheros de Registro (Log) de MySQL

MySQL tiene varios archivos de registro diferentes que pueden ayudar a encontrar lo que está ocurriendo en mysqld (ver Tabla 3).

Archivo de registro	Tipo de información registrado en el archivo
El registro de error	Registra problemas encontrados iniciando, ejecutando, o parando <code>mysqld</code> .
El registro de consultas	Registra las conexiones de clientes establecidas, y las sentencias ejecutadas.
El registro de actualizaciones The update log	Registra las sentencias que cambian datos. Este registro está ya en desuso.
El registro binario	Registra todas las sentencias que cambian datos. También utilizado para replicación.
El registro de lentitud	Registra todas las sentencias que tardaron más de <code>long_query_time</code> segundos en ejecutarse, o no utilizaron índices.

Tabla 3: Archivos de Registro.

Por defecto, todos los registros son creados en el directorio de datos de mysqld. Se puede forzar a mysqld a que cierre y reabra los archivos de registro (o en algunos casos, cambiar a un nuevo registro) mediante el volcado de registros. El volcado de registros ocurre cuando se ejecuta la sentencia FLUSH LOGS o el comando mysqladmin flush-logs or mysqladmin refresh.

La Caché de Consultas de MySQL

MySQL 5.0 Server proporciona una query cache. Cuando se usa, la query cache almacena el texto de una consulta SELECT junto con el resultado que se le envió al cliente. Si se recibe una consulta idéntica posteriormente, el servidor devuelve el resultado de la caché de consultas en lugar de parsear y ejecutar la consulta de nuevo.

La caché de consultas es muy útil en un entorno donde se tiene tablas que no cambian frecuentemente y donde el servidor recibe muchas consultas idénticas. Esta es la típica situación de muchos servidores Web que generan muchas páginas dinámicas basadas en contenido de bases de datos.

La caché de consultas no devuelve datos antiguos. Cuando las tablas se modifican, cualquier entrada relevante en la caché de consultas se elimina.

La caché de consultas no funciona en un entorno donde se tiene muchos servidores mysqld actualizando las mismas tablas MyISAM.

Replicación en MySQL

Las características de MySQL 5 soportan replicación asíncrona unidireccional: un servidor actúa como maestro y uno o más actúan como esclavos (sto contrasta con la replicación síncrona que es una característica de MySQL Cluster). El servidor maestro escribe actualizaciones en el fichero de log binario, y mantiene un índice de los ficheros para rastrear las rotaciones de logs. Estos logs sirven como registros de actualizaciones para enviar a los servidores esclavos. Cuando un esclavo se conecta al maestro, informa al maestro de la posición hasta la que el esclavo ha leído los logs en la última actualización satisfactoria. El esclavo recibe cualquier actualización que ha tenido lugar desde entonces, y se bloquea y espera para que el master le envíe nuevas actualizaciones.

Un esclavo servidor puede servir como maestro si quiere preparar una cadena de replications de replicación.

La replicación unidireccional tiene beneficios para la robustez, velocidad, y administración del sistema:

- La robustez se incrementa con un escenario maestro/esclavo. En caso de problemas con el maestro, se puede cambiar al esclavo como copia de seguridad.
- Puede conseguirse un mejor tiempo de respuesta dividiendo la carga de consultas de clientes a procesar entre los servidores maestro y esclavo. Se puede enviar consultas SELECT al esclavo para reducir la carga de proceso de consultas del maestro. Sin embargo, las sentencias que modifican datos deben enviarse siempre al maestro, de forma que el maestro y el esclavo no se desincronicen. Esta estrategia de balanceo de carga es efectiva si dominan consultas que no actualizan datos, pero este es el caso más habitual.
- Otro beneficio de usar replicación es que puede realizar copias de seguridad usando un servidor esclavo sin molestar al maestro. El maestro continúa procesando actualizaciones mientras se realiza la copia de seguridad.

Motores de Almacenamiento de MySQL y Tipos de Tablas

MySQL soporta varios motores de almacenamiento que tratan con distintos tipos de tablas. Los motores de almacenamiento de MySQL incluyen algunos que tratan con tablas transaccionales y otros que no lo hacen:

- MyISAM trata tablas no transaccionales. Proporciona almacenamiento y recuperación de datos rápida, así como posibilidad de búsquedas fulltext. MyISAM se soporta en todas las configuraciones MySQL, y es el motor de almacenamiento por defecto a no ser que tenga una configuración distinta a la que viene por defecto con MySQL.

- El motor de almacenamiento MEMORY proporciona tablas en memoria. El motor de almacenamiento MERGE permite una colección de tablas MyISAM idénticas ser tratadas como una simple tabla. Como MyISAM, los motores de almacenamiento MEMORY y MERGE tratan tablas no transaccionales y ambos se incluyen en MySQL por defecto.
- Los motores de almacenamiento InnoDB y BDB proporcionan tablas transaccionales. BDB se incluye en la distribución binaria MySQL-Max en aquellos sistemas operativos que la soportan. InnoDB también se incluye por defecto en todas las distribuciones binarias de MySQL 5.0. En distribuciones fuente, puede activar o desactivar estos motores de almacenamiento configurando MySQL a su gusto.
- El motor de almacenamiento EXAMPLE es un motor de almacenamiento "tonto" que no hace nada. Puede crear tablas con este motor, pero no puede almacenar datos ni recuperarlos. El objetivo es que sirva como ejemplo en el código MySQL para ilustrar cómo escribir un motor de almacenamiento. Como tal, su interés primario es para desarrolladores.
- NDB Cluster es el motor de almacenamiento usado por MySQL Cluster para implementar tablas que se particionan en varias máquinas. Está disponible en distribuciones binarias MySQL-Max 5.0. Este motor de almacenamiento está disponible para Linux, Solaris, y Mac OS X. Añadiremos soporte para este motor de almacenamiento en otras plataformas, incluyendo Windows en próximas versiones.
- El motor de almacenamiento ARCHIVE se usa para guardar grandes cantidades de datos sin índices con una huella muy pequeña.
- El motor de almacenamiento CSV guarda datos en ficheros de texto usando formato de valores separados por comas.
- El motor de almacenamiento FEDERATED se añadió en MySQL 5.0.3. Este motor guarda datos en una base de datos remota.

Una base de datos puede contener tablas de distintos tipos.

Las tablas transaccionales (TSTs) tienen varias ventajas sobre las no transaccionales (NTSTs):

- Más seguras. Incluso si MySQL cae o tiene problemas de hardware, puede recuperar los datos, mediante recuperación automática o desde una copia de seguridad más el log de transacciones.
- Puede combinar varios comandos y aceptarlos todos al mismo tiempo con el comando COMMIT (si autocommit está desactivado).
- Puede ejecutar ROLLBACK para ignorar los cambios (si autocommit está desactivado).
- Si falla una actualización, todos los cambios se deshacen (con tablas no transaccionales, todos los cambios son permanentes.)
- Motores de almacenamiento transaccionales pueden proporcionar mejor concurrencia para tablas que tienen varias actualizaciones concurrentes con lecturas.

Tablas no transaccionales tienen varias ventajas al no tener una sobrecarga transaccional:

- Más rápidas.
- Menor requerimiento de espacio.
- Menos memoria para actualizaciones.

MySQL Cluster

MySQL Cluster es una versión de alta disponibilidad, alta redundancia de MySQL adaptada para el entorno de computación distribuida. Usa el motor de almacenamiento NDB Cluster para permitir la ejecución de varios servidores MySQL en un cluster. Este motor de almacenamiento está disponible en las distribuciones binarias de MySQL 5.0 y en los RPMs compatibles con las distribuciones Linux más modernas.

Los sistemas operativos en que MySQL Cluster está disponible son Linux, Mac OS X, y Solaris (algunos usuarios han reportado éxito al ejecutar MySQL Cluster en FreeBSD, aunque no está soportada oficialmente por MySQL AB). Se está trabajando para hacer que Cluster se ejecute en todos los sistemas operativos soportados por MySQL, incluyendo Windows.

MySQL Cluster es una tecnología que permite clustering de bases de datos en memoria en un entorno de no compartición. La arquitectura de no compartición permite que el sistema funcione con hardware barato, y sin ningún requerimiento especial de hardware o software. Tampoco tienen ningún punto único de fallo porque cada componente tiene su propia memoria y disco.

MySQL Cluster integra el servidor MySQL estándar con un motor de almacenamiento clusterizado en memoria llamado NDB. El término NDB se refiere a la parte de la inicialización específica al motor de almacenamiento, mientras que MySQL Cluster se refiere a la combinación de MySQL y el nuevo motor de almacenamiento.

Un MySQL Cluster consiste en un conjunto de máquinas, cada una ejecutando un número de procesos incluyendo servidores MySQL, nodos de datos para NDB Cluster, servidores de administración, y (posiblemente) programas especializados de acceso a datos. La relación de estos componentes en un cluster se muestra en la Figura 2.

Todos estos programas funcionan juntos para formar un MySQL Cluster. Cuando se almacenan los datos en el motor NDB Cluster, las tablas se almacenan en los nodos de datos. Tales tablas son directamente accesibles desde todos los otros servidores MySQL en el cluster. Por lo tanto, en una aplicación de pago que almacene datos en un cluster, si una aplicación actualiza el salario de un empleado, todos los otros servidores MySQL que acceden a estos datos pueden ver el cambio inmediatamente.

Los datos almacenados en los nodos de datos de MySQL Cluster pueden replicarse: el cluster puede tratar fallos de nodos de datos individuales sin otro impacto a parte de abortar unas pocas transacciones debido a la pérdida de estado de transacción. Como las aplicaciones transaccionales se supone que tratan fallos transaccionales, esto no debería ser un problema. Todos estos programas funcionan juntos para formar un MySQL Cluster.

Cuando se almacenan los datos en el motor NDB Cluster, las tablas se almacenan en los nodos de datos. Tales tablas son directamente accesibles desde todos los otros servidores MySQL en el cluster. Por lo tanto, en una aplicación de pago que almacene datos en un

cluster, si una aplicación actualiza el salario de un empleado, todos los otros servidores MySQL que acceden a estos datos pueden ver el cambio inmediatamente.

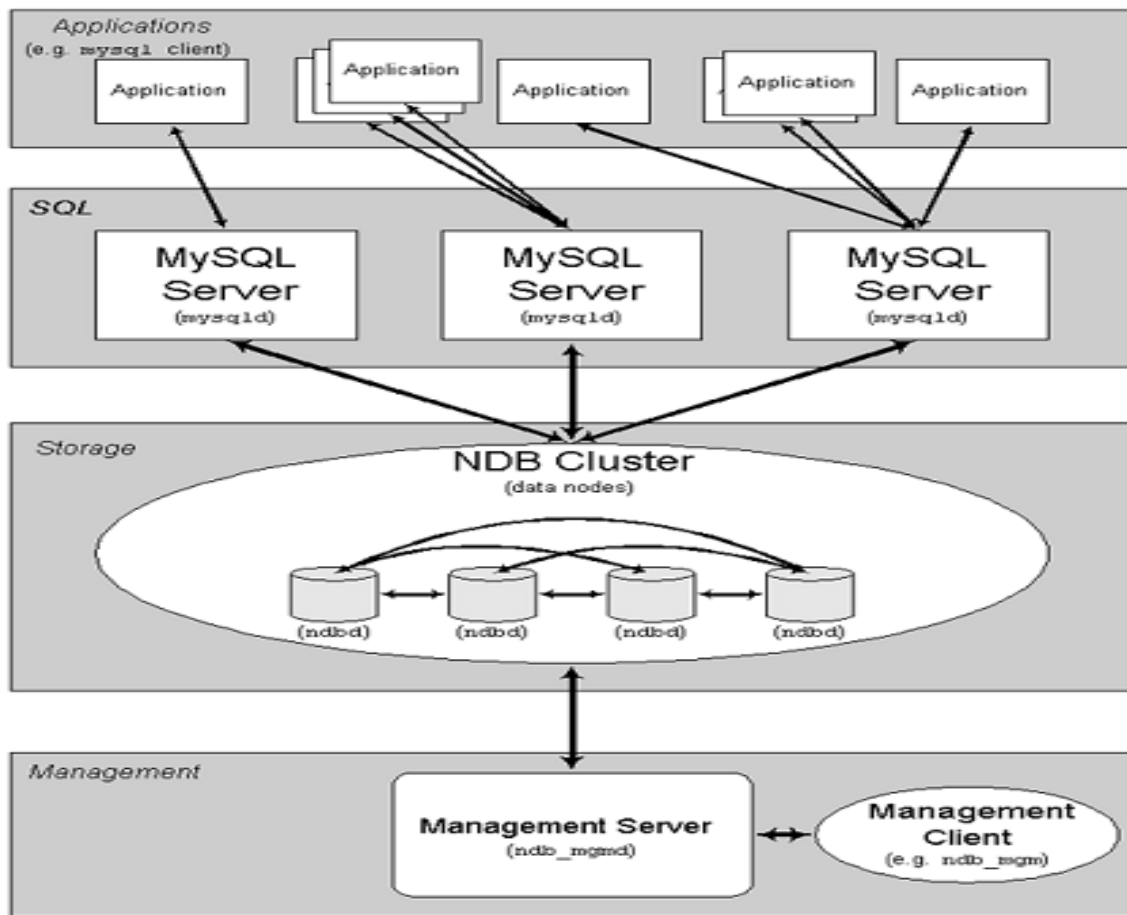


Figura 2: Clustering con MySQL.

Al llevar MySQL Cluster al mundo Open Source, MySQL proporciona tratamiento de datos clusterizado con alta disponibilidad, alto rendimiento, y escalabilidad disponible para todo el que lo necesite.

Procedimientos Almacenados y Funciones

Los procedimientos almacenados y funciones son nuevas funcionalidades de la versión de MySQL 5.0. Un procedimiento almacenado es un conjunto de comandos SQL que pueden almacenarse en el servidor. Una vez que se hace, los clientes no necesitan reemplazar los comandos individuales pero pueden en su lugar referirse al procedimiento almacenado.

Algunas situaciones en que los procedimientos almacenados pueden ser particularmente útiles:

- Cuando múltiples aplicaciones cliente se escriben en distintos lenguajes o funcionan en distintas plataformas, pero necesitan realizar la misma operación en la base de datos.

- Cuando la seguridad es muy importante. Los bancos, por ejemplo, usan procedimientos almacenados para todas las operaciones comunes. Esto proporciona un entorno seguro y consistente, y los procedimientos pueden asegurar que cada operación se loguea apropiadamente. En tal entorno, las aplicaciones y los usuarios no obtendrían ningún acceso directo a las tablas de la base de datos, sólo pueden ejecutar algunos procedimientos almacenados.

Los procedimientos almacenados pueden mejorar el rendimiento ya que se necesita enviar menos información entre el servidor y el cliente. El intercambio que hay es que aumenta la carga del servidor de la base de datos ya que la mayoría del trabajo se realiza en la parte del servidor y no en el cliente. Se debe considerar esto si muchas máquinas cliente (como servidores Web) se sirven de sólo uno o pocos servidores de bases de datos.

Los procedimientos almacenados permiten tener bibliotecas o funciones en el servidor de base de datos. Esta característica es compartida por los lenguajes de programación modernos que permiten este diseño interno, por ejemplo, usando clases.

MySQL sigue la sintaxis SQL2003 para procedimientos almacenados, que también usa IBM DB2.

Disparadores (Triggers)

A partir de MySQL 5.0.2 se incorporó el soporte básico para disparadores (triggers). Un disparador es un objeto con nombre dentro de una base de datos el cual se asocia con una tabla y se activa cuando ocurre en ésta un evento en particular.

Vistas (Views)

Las vistas (incluyendo vistas actualizables) fueron introducidas en la versión 5.0 del servidor de base de datos MySQL y en él se puede realizar lo siguiente:

- Creación o modificación de vistas con `CREATE VIEW` o `ALTER VIEW`.
- Eliminación de vistas con `DROP VIEW`.
- Obtención de información de definición de una vista (metadatos) con `SHOW CREATE VIEW`.

Conectores

Los Conectores MySQL son controladores (drivers) que proporcionan a los programas cliente conectividad con el servidor MySQL. Existen actualmente varios conectores MySQL:

- Connector/ODBC: proporciona soporte a nivel de controlador para la conexión con un servidor MySQL usando la API de Conectividad de Bases de Datos Abierta (ODBC por sus siglas en inglés). Con este controlador la conexión ODBC es posible desde las plataformas Windows, Unix y Mac OS X.
- Connector/NET: permite a los desarrolladores crear aplicaciones .NET usando los datos almacenados en una base de datos MySQL. Connector/NET implementa una

interfaz ADO.NET totalmente funcional y proporciona soporte para su uso con herramientas compatibles con ADO.NET. Las aplicaciones que se desee usen Connector/NET pueden escribirse en cualquier lenguaje .NET soportado.

- Plugin Visual Studio MySQL: trabaja con Connector/NET y Visual Studio 2005. Este plugin es un proveedor DDEX, lo que significa que se pueden usar herramientas de manipulación de esquemas y datos dentro de Visual Studio para crear y editar objetos dentro de una base de datos MySQL.
- Connector/J: proporciona soporte de controlador para conectar con MySQL desde una aplicación Java usando la API de Conectividad con Bases de Datos Java estándar (JDBC).
- Connector/MXJ: es una herramienta que permite poner en marcha y administrar fácilmente el servidor y la base de datos MySQL a través de una aplicación Java.
- Connector/PHP: es un controlador para conectar Windows con PHP. Proporciona las extensiones mysql y mysqli para su uso con MySQL 5.0.18 y posteriores.

El Paquete de Pruebas MySQL Test

El sistema de testeo incluido en distribuciones fuente Unix y distribuciones binarias hace que sea posible para los usuarios y desarrolladores realizar tests de regresión en código MySQL. Estos tests pueden ejecutarse en Unix o en Windows (usando el entorno Cygwin) si el servidor se ha compilado bajo Cygwin. No pueden ejecutarse en un entorno Windows nativo.

El conjunto actual de casos de uso no testea todo en MySQL, pero debería atrapar la mayoría de bugs obvios en el código de proceso SQL, temas de bibliotecas del SO, y es bastante útil para testear replicación.

Conclusiones

MySQL otorga todas las funcionalidades que hoy en día se espera de un motor de base de datos, MySQL incluye todos los elementos necesarios para instalar el programa, preparar diferentes niveles de acceso de usuario, administrar el sistema y proteger los datos.

Es posible desarrollar las aplicaciones de bases de datos en la mayor parte de los lenguajes de programación utilizados en la actualidad y ejecutarlas en casi todos los sistemas operativos utilizando MySQL.

MySQL también dispone de procedimientos almacenados (store procedures), triggers y vistas, además, utiliza el lenguaje de consulta estructurado (SQL).

Es una base de datos relacional la cual permite obtener ciertas ventajas como: costo, asistencia, velocidad, potencia, portabilidad, escalabilidad, potencia, seguridad, conectividad, etc. Todo esto sumado a que es un software de código abierto, licenciado bajo la GPL de la GNU.

Estas características hacen de MySQL una herramienta muy útil a la hora del uso de algún tipo de base de datos relacional, tanto para el ámbito comercial, como para el desarrollo de aplicaciones de enseñanza-aprendizaje.

Bibliografía Web

http://www.uco.es/~el2pamuj/docs/research/national_cong/jp2005_2.pdf

<http://www.jorgesanchez.net/bd/mysql.pdf>

<http://www.mysql-hispano.org/articulos/num43/analisis-comparativo.pdf>