

Replicación de datos en MySQL

Mantén activa tu aplicación web ante
un fallo en los servidores



Introducción

Hoy en día, muchos negocios dependen del buen funcionamiento de sus **aplicaciones web** en Internet, que deben estar accesibles las 24 horas del día durante los 365 días del año. Un fallo en estas aplicaciones puede significar la pérdida de una importante cantidad de dinero, de ahí la importancia de implementar mecanismos que garanticen su buen funcionamiento en caso de sufrir problemas.

Entre estos mecanismos que se pueden adoptar está la replicación de los datos en varios servidores, una buena alternativa para tener disponible la información cuando un servidor se cae. Este mecanismo no suplanta a las **copias de seguridad**, sino que nos garantiza que la información esté siempre disponible.

A lo largo de este documento, nos centraremos en la replicación de los datos almacenados en bases de datos MySQL y veremos cómo configurar nuestros servidores para que lleven a cabo esta replicación.

Beneficios de la replicación de los datos

Entre los beneficios que nos ofrece la replicación de la información, podemos destacar los siguientes aspectos.

1.- Disponibilidad

Supongamos el caso de una aplicación que cuenta con una base de datos replicada en dos servidores. Si uno de esos servidores callera, la aplicación seguiría funcionando ya que podría obtener la información del otro servidor. Esto no ocurre cuando la base de datos está sólo disponible en un servidor.

2.- Fiabilidad

Como hemos comentado anteriormente, la replicación no sustituye al backup de la información, pero al estar la información en varios servidores se convierte en un excelente sistema para recuperar la información cuando existan fallos.

3.- Rendimiento

La replicación de los datos también trae consigo una mejora en el rendimiento a la hora de realizar las consultas, ya que las distintas peticiones se van repartiendo entre los servidores donde se encuentran las bases de datos.

4.- Soporte para un mayor número de usuarios

Al estar la carga repartida entre varios servidores, el sistema de base de datos admite un mayor número de usuarios accediendo a la información.

Instalación MySQL en un entorno Linux

Antes de avanzar con este White Paper, veremos el proceso de instalación del software MySQL en un servidor con sistema operativo Linux. Para ello, una vez que estamos conectados a la consola, debemos ejecutar la siguiente instrucción:

```
sudo apt-get install mysql-server mysql-client
```

Este proceso nos pedirá una contraseña para el usuario root. Le asignamos la que nosotros queramos y la dejamos a buen recaudo, ya que nos hará falta para conectarnos al servidor MySQL y poder ejecutar instrucciones en él.

Conexión al servidor MySQL desde consola

A lo largo de todo este libro blanco veremos ciertas instrucciones MySQL que deben ser ejecutadas bien mediante un gestor de base de datos como puede ser phpMyAdmin, o bien mediante la interfaz de MySQL de línea de comandos.

Para los que no tengan acceso a un gestor, a continuación os explicamos cómo conectaros a esta interfaz para ejecutar las instrucciones.

Para llevar a cabo esta conexión, lo único que debemos hacer es ejecutar la siguiente instrucción desde la consola del servidor.

```
mysql -h localhost -u root -p
```

Lo que hacemos con esa instrucción es indicarle que nos queremos conectar al servidor MySQL alojado en esa máquina mediante el usuario root.

A continuación nos pedirá que introduzcamos el password de ese usuario, que es el que hemos indicado en el proceso de instalación de MySQL. Si todo ha ido bien, deberemos ver algo similar a la siguiente imagen.

```
login as: root
root@~# mysql -h localhost -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 12061
Server version: 5.1.61 Source distribution

Copyright (c) 2000, 2011, Oracle and/or its affiliates. All
rights reserved.

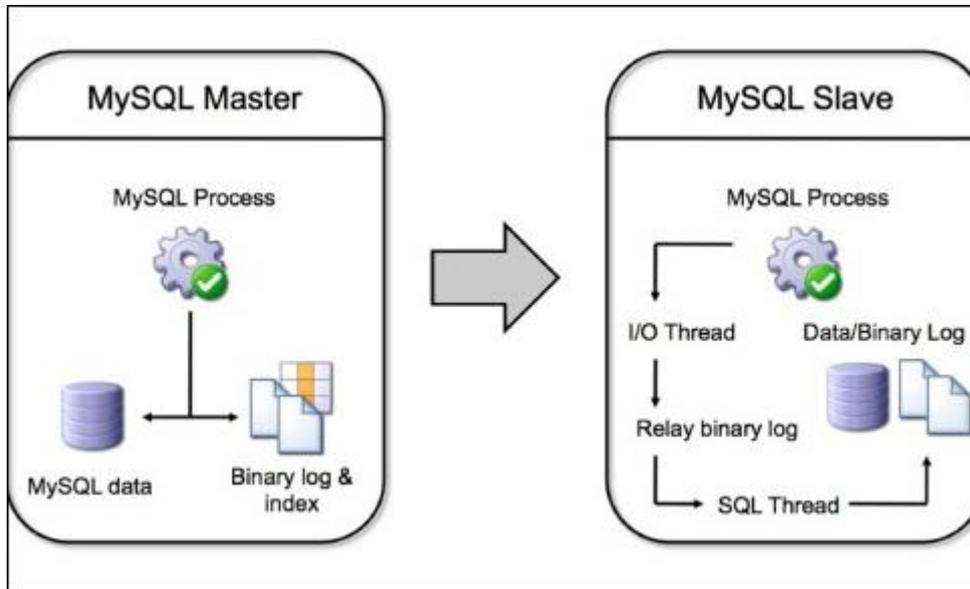
Oracle is a registered trademark of Oracle Corporation and/or
its affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current
statement.

mysql>
```

Tipos de replicación que nos podemos encontrar

a) Modelo Maestro – Esclavo



En la replicación Maestro – Esclavo todas las actualizaciones que se llevan a cabo sobre el servidor Maestro, son replicadas automáticamente en el Esclavo. Este tipo de replicación es útil cuando necesitamos realizar operaciones de selección, inserción, actualización o eliminación de datos en el Maestro, mientras que el servidor Esclavo está pensado únicamente para realizar operaciones de selección. Si en el Esclavo realizáramos cualquier otra operación que no fuera de selección de datos, estos cambios no se verían reflejados en el Maestro, cosa que sí ocurre a la inversa.

El funcionamiento básico para la replicación de la información está en hacer uso del fichero binario de transacciones para almacenar en el Maestro todos los cambios que se han realizado sobre la base de datos (updates, deletes, creates...), para que después el servidor Esclavo los lea y replique esos cambios en la base de datos que almacena.

En nuestro ejemplo utilizaremos un servidor Maestro y otro Esclavo, aunque perfectamente podríamos tener un número mayor de Esclavos.

Para crear esta estructura, lo primero que debemos hacer es configurar el servidor maestro para que almacene el log binario con todos los cambios realizados sobre la base de datos y asignarle un identificador único.

Para ello editamos el fichero “**my.cnf**”, que lo podemos encontrar en la ruta “**/etc/mysql**” una vez que hayamos hecho la instalación del servidor MySQL. En ese archivo añadimos o editamos las siguientes líneas:

```
server-id=1

log-bin=mysql-bin

max-binlog-size=500M

expire_logs_days=4

innodb_flush_log_at_trx_commit=1
```

A continuación explicamos lo más importante de las líneas anteriores.

- **server-id**: Identificador que tendrá el servidor.
- **log-bin**: Nombre que le asignamos al log binario de las transacciones.
- **max-binlog-size**: Mediante este parámetro le indicamos el tamaño máximo del log.
- **expire_logs_days**: Número de días que tienen que pasar para que se borre el log.
- **innodb_flush_log_at_trx_commit**: Cuando toma el valor de 1, cada vez que se confirma una transacción se graba en el fichero de registro y se vuelca a disco.

Una vez guardados los cambios es necesario reiniciar el servicio de MySQL, cosa que haremos desde la línea de comandos mediante la instrucción:

```
service mysqld restart
```

A continuación debemos crear un usuario con privilegios de replicación. Para ello nos conectamos a la interfaz de línea de comandos de MySQL, tal y como hemos explicado anteriormente, y ejecutamos la siguiente instrucción.

```
CREATE USER 'usuario'@'host' IDENTIFIED BY 'password';
GRANT REPLICATION SLAVE ON nombre_bdd.* TO 'usuario'@'host' IDENTIFIED BY 'password';
```

En esa misma interfaz, ejecutamos el siguiente comando para obtener los datos necesarios para la configuración del Esclavo:

```
SHOW MASTER STATUS;
```

Esto nos devolverá un resultado similar al que os dejamos en la imagen.

```
+-----+-----+-----+-----+
| File           | Position | Binlog_Do_DB | Binlog_Ignore_DB |
+-----+-----+-----+-----+
| mysql-bin.000001 |      98 | dbmysnsconfig |                   |
+-----+-----+-----+-----+
```

Hecho todo lo anterior, es hora de ponerse a configurar el servidor Esclavo. Para ello editamos su fichero de configuración MySQL “**my.cnf**”. Ahí añadiremos o editaremos la siguiente línea.

```
server-id=2
```

Con esa línea lo que le estamos indicando es el identificador que tendrá dentro de la estructura que estamos creando.

Tras el cambio y una vez que hayamos reiniciado el servidor MySQL Esclavo, nos conectamos a su interfaz de línea de comando, de la misma forma que hicimos en el servidor Maestro para ejecutar las instrucciones que os mostramos a continuación:

```
CHANGE MASTER TO  
MASTER_HOST='host',  
MASTER_USER='user',  
MASTER_PASSWORD='password',  
MASTER_PORT=3306,  
MASTER_LOG_FILE='mysql-BIN.000001',  
MASTER_LOG_POS=98;
```

El significado de cada una de las anteriores instrucciones es el siguiente:

- **MASTER_HOST**: El nombre del host o dirección IP del servidor Maestro.
- **Master_user**: Es el usuario que hemos creado anteriormente con privilegios de replicación.
- **Master_password**: La clave que le hemos asignado al usuario.
- **Master_port**: Indicamos el puerto donde escucha el servicio MySQL.
- **Master_log_file**: El nombre del log donde el Master va escribiendo las transacciones realizadas. Lo obtenemos de ejecutar “Show Master Status”, como hemos visto en un paso anterior.
- **Master_log_pos**: La posición del Maestro. Es un valor que también obtenemos al ejecutar “Show Master Status”.

Para finalizar el proceso de replicación, sólo falta ejecutar la siguiente instrucción desde la interfaz MySQL del servidor Esclavo.

```
Start Slave
```

b) Modelo Maestro – Múltiples esclavos

Perfectamente podríamos tener un número mayor de Esclavos. Para ello simplemente tenemos que repetir la configuración explicada anteriormente por cada uno de los Esclavos que queramos tener.

c) Modelo Maestro – Maestro

La replicación en modo “Maestro - Maestro” no está soportada oficialmente por MySQL y según la experiencia de los técnicos de acens no es válida para entornos de producción, por los muchos riesgos que conlleva. Por tanto, nunca debería ser aplicada en producción, y debido a ello la hemos eliminado este apartado en el White Paper.