



KHAI GIẢNG Khóa Học Lập Trình **RUBY**

- ✓ Lập trình hướng đối tượng với **Ruby**
- ✓ Tổng quan và thực hành về **GIT**
- ✓ Xây dựng website với framework **Ruby on Rails**
- ✓ Lập trình nâng cao với **Ruby on Rails** trong hai dự án thực tế.



Follow



Giới thiệu về MySQL Replication

Khái niệm Replication

MySQL Replication là một quá trình cho phép bạn dễ dàng duy trì nhiều bản sao của dữ liệu MySQL bằng cách cho họ sao chép tự động từ một master tạo ra một cơ sở dữ liệu slave. Điều này rất hữu ích vì nhiều lý do bao gồm việc tạo điều kiện cho sao lưu cho dữ liệu, một cách để phân tích nó mà không sử dụng các cơ sở dữ liệu chính, hoặc chỉ đơn giản là một phương tiện để mở rộng ra.

Replication mặc định là không đồng bộ, slave không cần phải kết nối vĩnh viễn để nhận được cập nhật từ master. Tùy thuộc vào cấu hình, bạn có thể sao chép tất cả các cơ sở dữ liệu, cơ sở dữ liệu đã chọn, hoặc thậm chí bảng được lựa chọn trong một cơ sở dữ liệu. Thật vậy, Replication có ý nghĩa là “nhân bản”, là có một phiên bản giống hệt phiên bản đang tồn tại, đang sử dụng. Với một cơ sở dữ liệu có nhu cầu lưu trữ lớn, thì đòi hỏi cơ sở dữ liệu phải toàn vẹn, không bị mất mát trước những sự cố ngoài dự đoán là rất cao. Vì vậy, người ta nghĩ ra khái niệm (slave) “nhân bản”, tạo một phiên bản cơ sở dữ liệu giống hệt cơ sở dữ liệu đang tồn tại, và lưu trữ ở một nơi khác, để phòng có sự cố.

Server master lưu trữ phiên bản cơ sở dữ liệu phục vụ ứng dụng. Server slave lưu trữ phiên bản cơ sở dữ liệu “nhân bản”. Quá trình nhân bản từ master sang slave gọi là replication.

Tất cả các thay đổi trên cơ sở dữ liệu master sẽ được ghi lại dưới dạng file log binary, slave đọc file log đó, thực hiện những thao tác trong file log, việc ghi, đọc và thực thi trong file log này dưới dạng binary được thực hiện rất nhanh.

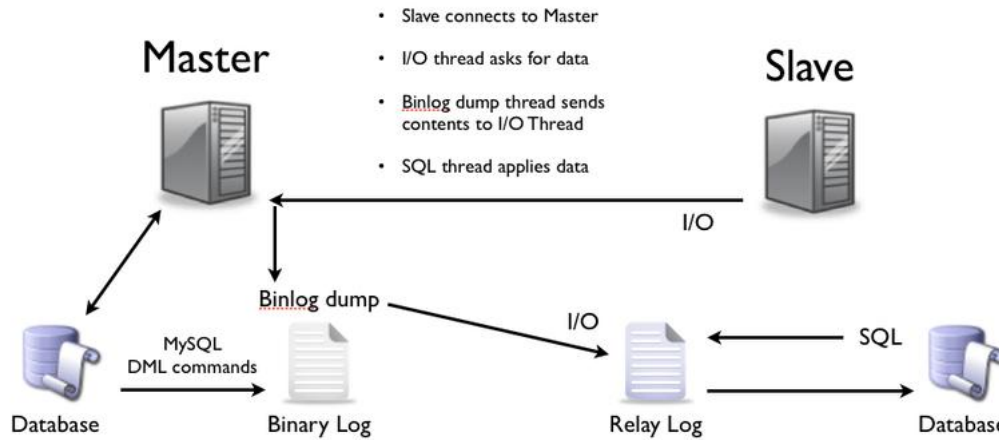
Ưu điểm của replication trong mysql

- Giảm tải cho cơ sở dữ liệu server master, tải trọng của server được phân tải cho các con slave, cải thiện hiệu năng cho toàn hệ thống. Trong môi trường này, tất cả các quá trình ghi và cập nhật đều phải diễn ra trên server master, bên cạnh đó quá trình đọc được diễn ra trên một hoặc nhiều con slave. Chính vì vậy mô hình này giúp tăng đáng kể hiệu năng của toàn hệ thống.
- Tính bảo mật dữ liệu cao - vì dữ liệu được sao chép đến các slave, và các slave có thể tạm dừng quá trình sao chép, nó có thể chạy các dịch vụ sao lưu trên các slave mà không làm hư hỏng dữ liệu tổng thể tương ứng.



- Tính phân tích - dữ liệu trực tiếp có thể được tạo ra trên master, trong khi phân tích các thông tin có thể xảy ra trên các slave mà không ảnh hưởng đến hiệu suất của master.
- Tính phân phối dữ liệu từ xa - bạn có thể sử dụng replication để tạo ra một bản sao của dữ liệu cho một trang web từ xa để sử dụng, mà không cần truy cập thường xuyên vào con master.

Mô hình MySQL Replication và cách thức hoạt động



Replication dựa trên các con master lưu giữ theo dõi tất cả những thay đổi cơ sở dữ liệu của nó (cập nhật, xóa, vv) trong bản ghi nhị phân của nó. Các bản ghi nhị phân phục vụ như là các record của tất cả các sự kiện làm thay đổi cấu trúc cơ sở dữ liệu hoặc nội dung (dữ liệu) từ thời điểm các máy chủ đã bắt đầu thực thi. Thông thường, câu SELECT không được ghi lại bởi vì chúng không phải thay đổi cấu trúc cũng như nội dung của cơ sở dữ liệu.

Mỗi slave kết nối đến các master yêu cầu một bản sao của bản ghi nhị phân. Đó là, nó kéo các dữ liệu từ các master, chứ không phải là master đẩy dữ liệu đến các slave. Các slave cũng thực hiện các sự kiện từ các bản ghi nhị phân mà nó nhận được. Quá trình này lặp đi lặp lại những thay đổi ban đầu cũng giống như nó đã được thực hiện trên master. Bảng được tạo ra hoặc cấu trúc thay đổi và dữ liệu đã chèn hay đã xóa và kể cả cập nhật thì đều giống hệt theo những thay đổi mà ban đầu đã được thực hiện trên master.

Chi tiết quá trình thực thi trong Replication như sau:

- Luồng Binlog dump: Các master tạo 1 luồng và gửi nội dung binary log đến một slave khi các slave kết nối với master. Luồng này có thể được xác định trong đầu ra của query `SHOW PROCESSLIST` trên master như là 1 luồng Binlog Dump. Các binary log có được trên bản ghi nhị phân của master đọc các sự kiện đó và gửi đi cho slave. Ngay sau khi sự kiện được đọc, khóa được phát hành ngay cả khi sự kiện được gửi tới slave.
- Luồng Slave I/O: Khi thông báo Slave được ban hành trên slave server, các slave tạo một luồng I/O, cái mà kết nối với server master và hỏi nó để nó gửi thông tin bản ghi cập nhật nó vào trong log nhị phân. Luồng slave I/O đọc sự cập nhật trên luồng Binlog Dump của master gửi và sao chép chúng vào 1 file local - file mà bao hàm cả những log trễ (Relay Log). Các trạng thái của luồng này được thể hiện như là `Slave_IO_running` trong output `SHOW SLAVE STATUS` hoặc là `Slave_running` trong output của `SHOW STATUS`.
- Luồng Slave SQL: Các slave tạo ra một luồng SQL để đọc cái log trễ, cái này sau đó sẽ được ghi vào luồng slave I/O và thực thi các sự kiện chứa trong đó

Cách cài đặt MySQL Slave Replication

```
sudo nano /etc/mysql/my.cnf
```

Bước đầu tiên phải tìm đến phần trông như sau để binding server master localhost chẳng hạn:

```
bind-address          = 127.0.0.1
```

Thay thế địa chỉ IP local thành địa chỉ của server. Ví dụ:

```
bind-address          = 12.34.56.789
```

Thay đổi tiếp theo đề cập đến các server-id, nằm trong phần [mysqld]. Bạn có thể chọn bất kì số nào, ví dụ đơn giản nhất có thể đặt là 1.

```
server-id              = 1
```

Tiếp theo đặt đường dẫn file log cho mysql, tất cả các sự kiện của slave được lưu trữ trong đường dẫn này. Tìm đến dòng log_bin :

```
log_bin                = /var/log/mysql/mysql-bin.log
```

Cuối cùng chúng ta cần phải chỉ định cơ sở dữ liệu sẽ được nhân bản trên các máy slave. Bạn có thể chỉ định thay vì một mà là nhiều các slave bằng cách lặp lại dòng này cho tất cả các cơ sở dữ liệu mà bạn cần:

```
binlog_do_db           = newdatabase
```

Sau khi chỉnh sửa xong chúng ta cần lưu lại file cấu hình và khởi động lại mysql

```
sudo service mysql restart
```

Bước tiếp theo chúng ta cần phải mở MySQL mà cấp quyền cho các slave, bạn có thể đặt tên, mật khẩu cho các slave tùy ý:

```
GRANT REPLICATION SLAVE ON *.* TO 'slave_user'@'%' IDENTIFIED BY 'password';  
Sau đó kiểm tra bằng cách:  
FLUSH PRIVILEGES;
```

Một điều quan trọng nữa, bạn cần phải mở 1 tab mới và lựa chọn cơ sở dữ liệu của bạn.

```
USE newdatabase;
```

Sau đó, quan trọng nhất là bạn cần phải khóa cơ sở dữ liệu mà sau này các slave để ở chế độ chỉ đọc

```
FLUSH TABLES WITH READ LOCK;
```

Để kiểm tra cơ sở dữ liệu chúng ta gõ câu lệnh sau:

```
mysql> SHOW MASTER STATUS;
+-----+-----+-----+-----+
| File           | Position | Binlog_Do_DB | Binlog_Ignore_DB |
+-----+-----+-----+-----+
| mysql-bin.000001 |      107 | newdatabase  |                   |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Position ở đây có nghĩa là vị trí mà bắt đầu các slave sao lưu dữ liệu. Căn cứ vào cơ sở dữ liệu đã bị khóa, chúng ta export cơ sở dữ liệu ra 1 file sql để tiện dùng cho bước 2.

```
mysqldump -u root -p --opt newdatabase > newdatabase.sql
```

Bây giờ quay trở lại cửa sổ ban đầu và mở khóa cơ sở dữ liệu của bạn:

```
UNLOCK TABLES;
QUIT;
```

Về cơ bản cấu hình máy chủ đã tạm ổn

Bước 2: Cấu hình cơ sở dữ liệu slave

Đăng nhập vào server slave, mở mysql là tạo cơ sở dữ liệu với tên giống hệt cơ sở dữ liệu master:

```
CREATE DATABASE newdatabase;
EXIT;
```

Import cơ sở dữ liệu mà đã export ở bước 1.

```
mysql -u root -p newdatabase < /path/to/newdatabase.sql
```

Chúng ta cần cấu hình những con slave giống hệt như cách mà chúng ta cấu hình con master. Tuy nhiên cũng cần chỉnh sửa một số thông số cho phù hợp như server-id:

```
server-id          = 2
relay-log          = /var/log/mysql/mysql-relay-bin.log
log_bin            = /var/log/mysql/mysql-bin.log
binlog_do_db       = newdatabase
```

Khởi động lại mysql của con slave:

```
sudo service mysql restart
```

Bước tiếp theo chúng ta cần phải cấp quyền và cho phép nhân bản ở bên trong MySQL shell. Bật lại MySQL shell và thay thế các thông tin như sau:

```
CHANGE MASTER TO MASTER_HOST='12.34.56.789',MASTER_USER='slave_user', MASTER_PASSWORD='password', MASTER_LOG_FILE=''
```

1. Chỉ định các máy chủ hiện tại như là slave của server master
2. Cung cấp thông tin đăng nhập chuẩn cho các máy chủ
3. Cuối cùng chỉ định cho các máy slave biết rằng cần phải sao lưu từ file log nào và đăng nhập từ vị trí mà đã định nghĩa trong position nào.

Sau đó chúng ta active server slave:

```
START SLAVE;  
  
// Kiểm tra bằng cách:  
SHOW SLAVE STATUS\G  
  
// Nếu có vấn đề trong kết nối bạn có thể thử start slave bằng cách:  
SET GLOBAL SQL_SLAVE_SKIP_COUNTER = 1; SLAVE START;
```

Trên đây là tất cả những gì tôi tìm hiểu được về MySQL Replication. Nó còn nhiều các góc cạnh khác nhau, đây chỉ là một trong những khía cạnh tôi tìm hiểu một cách khái quát.


Tài liệu tham khảo


[Mysql replication](#)


How To Set Up Master Slave Replication in MySQL


VIBLO



 _____

 _____

 _____

 _____

