# Linux MySQL高可用

## 一，MySQL复制简介

使用MySQL复制功能可以将主服务器上的数据复制到多台的服务器。默认情况下，复制是异步传输方式，从服务器不需要总是连接主服务去更新数据。也就是说，数据更新可以在远距离连接的情况下进行，甚至在使用拨号网络的临时连接环境下也可以进行。根据自定义设置，我们可以对所有的数据库或部分数据库甚至是部分数据表进行复制。通过主从复制在企业级应用环境中就不必再当心数据库的单点故障，当一台服务器宕机时，其他服务器一样可以提供非常稳定、可靠的数据服务。

MySQL 复制的优势如下：

1，高性能：通过将请求分配给多台不同的服务器提高性能与速度。在这种环境中，所有对数据库的写操作必须提供给主服务器，但读操作可以被平均分配给多台从服务器。

2，数据安全：数据是从主服务器复制到从服务器，而且从服务器可以随时暂停复制，这样我们就可以实现数据备份与还原。

3，远程数据分享：如果企业拥有多处位于其他地理位置的分公司，而这些分公司希望共享总部数据库资源，就可以使用复制实现数据的共享，而且分公司不必时时连接主服务器，可以仅在需要时进行复制。

MySQL 复制有两种核心的格式，一种是基于SQL语句的复制SBR（Statement Based Replication）,另一种是基于行的复制RBR（Row Based Replication）,另外，有时也可以使用混合模式复制。MySQL 5.6 默认采用的是基于语句的复制。

MySQL 服务器之间的复制是基于二进日志机制的。在主服务器上，当MySQL实例进行写操作时会同时生成一条操作事件日志并写入二进制日志中，而从服务器则负责读取主服务器上的二进制日志，并在从服务器上重新执行该事件，从而实现复制数据至本地服务器。

当主服务器开启二进制日志功能后，所有的SQL语句都将被记录至日志。从服务器复制这些二进制条目，并根据自己的需要决定哪些语句需要重新在服务器上执行，而哪些语句不需要再执行，我们无法控制主服务器仅记录特定语句到日志文件中。如果进行其他设置，所有主服务器的日志事件都将在从服务器上重新执行，当然，你可以配置从服务器仅执行主服务器中的部分日志事件。因为从服务器会记录二进制日志的进度与位置（比如执行到了第几条语句），所以从服务器可以断开与服务器之间的链接，并在重新建立连接后继续进行复制工作。

在具体的操作过程中，主服务器与从服务器都需要配置一个唯一的ID编号。另外，所有的从服务器必须设置主服务器的主机名、日志文件名、文件位置等参数。

## 二，MySQL实例

演示步骤中的命令提示符为master，则代表指令在MySQL主服务器上运行，提示符为slave1则代表指令在MySQL 从服务器上运行）。整个过程分为以下几个步骤进行。

1，主服务器上开启二进制日志功能，设置唯一的服务器ID编号，这些设置需要重启MsSQL服务。

2，在所有的从服务器上设置唯一的服务器ID编号，这些设置需要重启MySQL服务。

3，在主服务器上为不同的从服务器创建可以读取主服务器日志文件的账户，或使用相同的统一账户。

4，进行数据复制前，需要记录主服务器上二进制日志的位置标记。

### 数据复制环境中主服务器的设置

在实际的生产环境中，可能在还没有部署数据复制前，数据库中就已经存在大量的数据。所以，这里事先创建一个测试用的数据库及数据表，用来掩饰如何对已经存在的数据进行数据同步备份。

[root@master ~]# mysql -u root -p

mysql> create database hr;

mysql> use hr;

mysql> create table employees(

-> employee\_id INT NOT NULL AUTO\_INCREMENT,

-> name char(20) NOT NULL,

-> e\_mail varchar(50),

-> PRIMARY KEY(employee\_id));

mysql> INSERT INTO employees values

-> [(1,'TOM','tom@example.com'),](mailto:(1,'TOM','tom@example.com'),)

-> [(2,'Jerry','jerry@example.com');](mailto:(2,'Jerry','jerry@example.com');)

mysql> exit

我们需要在主服务器上开启二进制日志并设置服务器编号，服务器编号必须是1至232-1之间的证书，根据自己的实际情况设置。进行这些设计需要关闭MySQL数据库并编辑my.cnf或my.ini文件，然后在[mysqld]设置段添加相应的配置选项。

[root@master ~]# vim /etc/my.cnf

**[mysqld]**

**log-bin=Jacob-bin**

**server-id=254**

[root@master ~]# service mysql restart

[root@master ~]# firewall-cmd --set-default-zone=trusted

**注意：在配置文件中不可以使用skip-networking参数选项，否则从服务器将无法与主服务器进行连接并复制数据。**

### 数据复制环境中从服务器的设置

如果从服务器的ID编号没有设置，或从服务器ID编号与主服务器有冲突，就必须关闭MySQL服务，并重新编辑配置文件，设置唯一的服务器编号，最后重启MySQL服务。如果有多台从服务器，则所有的服务器ID编号都必须是唯一的。可以考虑将服务器的ID编号与服务器的IP地址关联。如采用IP地址的后3位作为MySQL服务器的编号。

[root@slave1 ~]# vim /etc/my.cnf

**[mysqld]**

**server-id=2**

[root@slave1 ~]# service mysqld restart

[root@slave1 ~]# firewall-cmd --set-default-zone=trusted

对复制而言，MySQL从服务器上二进制日志功能是不需要开启的。但是，你可以通过启用服务器的二进制日志功能，实现数据备份与恢复。此外，在一些更复杂的拓扑环境中，MySQL从服务器也可以扮演其他从服务器的主服务器。

### 创建复制账号

执行的护具复制时，所有的从服务器都需要使用账户与密码连接MySQL主服务器，所以在主服务器上鼻息存在至少一个用户账户及相应的密码供服务器连接。这个账户必须拥有REPLICATION SLAVE权限，你可以为不同的从服务器创建不通的账户与密码，也可以使用同一的账户与密码。MySQL可以使用CREATE USER语句创建用户，使用GRANT语句为账户赋权。如果该用户仅为数据库复制所使用，则该账户仅需要REOLICATION SLAVE权限即可。

[root@master ~]# mysql -u root -p

mysql> CREATE USER ['slave\_cp'@'%.example.com'](mailto:'slave_cp'@'%.example.com') IDENTIFIED BY

-> 'SlaveAdmin';

mysql> GRANT REPLICATION SLAVE ON \*.\* TO

-> ['slave\_cp'@'%.example.com';](mailto:'slave_cp'@'%.example.com';)

mysql> exit

### 获取主服务器的二进制日志信息

在进行主从数据复制之前，需要了解一些主服务器的二进制日志文件的基本信息，这些信息在对从服务器的设置中需要用到，它包括主服务器二进制文件名称及当前日志记录的位置，这样从服务器就可以知道从哪里开机进行复制操作。

[root@master ~]# mysql -u root -p

mysql> FLUSH TABLES WITH READ LOCK;

mysql> SHOW MASTER STATUS;

**-----------------------------------------------------------------------------------------------------------------------**

**| File | Position | Binlog\_Do\_DB | Binlog\_Ignore\_DB | Executed\_Gtid\_Set |**

**-----------------------------------------------------------------------------------------------------------------------**

**| jacob-log.000001 | 1276 | | | |**

**-----------------------------------------------------------------------------------------------------------------------**

mysql> UNLOCK TABLES;

其中，File列显示的是二进制日志文件名，Position为当前日志记录位置。

FLUSH TABLES WITH READ LOCK 命令的作用是对所有数据库的表执行只读锁定，只读锁定后所有的数据库的写操作将被拒绝，但读操作可以继续。执行锁定可以防止在查看二进制日志信息的同时有人对数据进行修改操作，最后使用INLOCK TABLES 语句对全局锁执行结束操作。

### 对现有数据库进行快照备份

### 配置从服务器连接主服务器进行数据复制

### 数据同步验证