主从数据库

一、什么是主从同步?

影响MySQL-A(主master数据库的操作，在数据库执行后，都会写入本地的日志系统A中。

假设，实时的将变化了的日志系统中的数据库事件操作，在MYSQL-A的3306端口，通过网络发给MYSQL-B(从slave)。

MYSQL-B收到后，写入本地日志系统B，然后一条条的将数据库事件在数据库中完成。

那么，MYSQL-A的变化，MYSQL-B也会变化，这样就是所谓的MYSQL的复制，即MYSQL replication。

日志系统A，其实它是MYSQL的日志类型中的二进制日志，也就是专门用来保存修改数据库表的所有动作，即bin log。【注意MYSQL会在执行语句之后，释放锁之前，写入二进制日志，确保事务安全】

日志系统B，并不是二进制日志，由于它是从MYSQL-A的二进制日志复制过来的，并不是自己的数据库变化产生的，有点接力的感觉，称为中继日志，即relay log。

可以发现，通过上面的机制，可以保证MYSQL-A和MYSQL-B的数据库数据一致，但是时间上肯定有延迟，即MYSQL-B的数据是滞后的(即便不考虑什么网络的因素，MYSQL-A的数据库操作是可以并发的执行的，但是MYSQL-B只能从relay log中读一条，执行下。因此MYSQL-A的写操作很频繁，MYSQL-B很可能跟不上)。

读写分离的基本原理就是让主数据库处理事务性增、改、删操作（INSERT、UPDATE、DELETE）操作，而从数据库处理SELECT查询操作;因此写库只有一个，读库有多个，采用日志同步的方式实现主库和多个读库的数据同步。主库是包含事物性的，而从库是复制的所以不包含事务性

master的写操作，slaves被动的进行一样的操作，保持数据一致性，那么slave是否可以主动的进行写操作?

- 假设slave可以主动的进行写操作，slave又无法通知master，这样就导致了master和slave数据不一致了。因此slave不应该进行写操作，至少

是slave上涉及到复制的数据库不可以写。实际上，这里已经揭示了读写分离的概念。

数据库的读写分离的好处?

- 1. 将读操作和写操作分离到不同的数据库上，避免主服务器出现性能瓶颈;
- 2. 主服务器进行写操作时，不影响查询应用服务器的查询性能，降低阻塞，提高并发;
- 3. 数据拥有多个容灾副本，提高数据安全性，同时当主服务器故障时，可立即切换到其他服务器，提高系统可用性;

二、服务器(ubuntu)配置主从数据库操作

①准备两台服务器，分别查看服务ip(ifconfig →inet addr(本机ip))，

在/etc/hosts文件配置mastter和slave的ip和域名，执行sudo vim /etc/hosts

```
#
127.0.0.1 localhost.localdomain VM-0-8-ubuntu
127.0.0.1 localhost

# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
ff02::3 ip6-allhosts


主数据库ip  mysql-master
从数据库ip1 mysql-slave1
从数据库ip2 mysql-slave2
从数据库ip3 mysql-slave3
```

分别在主服务器和从服务器都配置上面操作，重启服务器让hosts文件生效，分别在服务器ping不同的域名：ping mysql-master

②安装mysql 数据库(sudo apt-get install mysql-server)

③修改my.cnf文件，ubuntu下mysql默认的配置文件是/etc/mysql/my.cnf

master服务器上添加mysqld部分：

```
sudo vim /etc/mysql/my.cnf
[mysqld]
bind-address=0.0.0.0
server-id=1
log-bin=mysql-bin-log
```

```
!includedir /etc/mysql/conf.d/
!includedir /etc/mysql/mysql.conf.d/

[mysqld]
bind-address=0.0.0.0
server-id=1
log-bin=mysql-bin-log
```

slave服务器添加mysqld如下

```
!includedir /etc/mysql/conf.d/
!includedir /etc/mysql/mysql.conf.d/

[mysqld]
server-id = 2
```

注意：master和slave里添加的server-id不能相同

重启两台服务器的数据库让my.cnf文件生效：sudo service mysql restart

④在主服务器上创建一个mysql用户，这个用户要配置到从服务器中用来执行复制任务。连接到主服务器上的mysql server，执行下面两条SQL

```
mysql> create user 'slave'@'mysql-slave' identified by 'slavepassword';
mysql> grant replication slave on *.* to 'slave'@'mysql-slave';
```

第一条语句创建了一个名叫slave的用户，并只能从mysql-slave(就是第一步在hosts里设置的域名和ip，mysql-slave对应的是就slave的ip)主机登录进来，slavepassword是密码;(可以设置多个slave，但是账号和@后的域名改成对应的slave，如：

```
create user 'slave1'@'mysql-slave1' identified by 'slavepassword';
create user 'slave2'@'mysql-slave3' identified by 'slavepassword';)
```

二条语句给该用户赋予了replication slave权限

⑤确保两mysql server中的数据一致性

```
mysql> flush tables;
Query OK, 0 rows affected (0.00 sec)
mysql> show master status;
```

```
mysql> create user 'slave'@'mysql-slave' identified by 'slave';
Query OK, 0 rows affected (0.00 sec)

mysql> grant replication slave on *.* to 'slave'@'mysql-slave';
Query OK, 0 rows affected (0.00 sec)

mysql> flush tables;
Query OK, 0 rows affected (0.01 sec)

mysql> show master status;
+----------------------+----------+--------------+------------------+-------------------+
| File                 | Position | Binlog_Do_DB | Binlog_Ignore_DB | Executed_Gtid_Set |
+----------------------+----------+--------------+------------------+-------------------+
| mysql-bin-log.000001 |      757 |              |                  |                   |
+----------------------+----------+--------------+------------------+-------------------+
1 row in set (0.00 sec)
```

slave里配置master_log_file用的就是master里的File;master_log_pos 用的就是Position

⑥在slave配置信息并开始复制

登录到mysql-slave上的mysql server，并输入下面的命令

```
mysql> change master to
    ->  master_host='mysql-master',
    ->  master_user='slave',
    ->  master_password='slave',
    ->  master_log_file='mysql-bin-log.000001',
    ->  master_log_pos=757;
Query OK, 0 rows affected, 2 warnings (0.00 sec)
```

change master to

master_host 是第一步hosts里配置的主域名(ip),

master_port=master的port,

master_user 是第三步SQL语句创建的用户,

master_password是用户的密码,

master_log_file是master里的File,

master_log_pos是master里的Position;

使用start slave;命令来让从服务器开始复制。

可以使用show slave status\G;来查看复制的信息

```
mysql> start slave;
Query OK, 0 rows affected (0.01 sec)

mysql> show slave status\G
*************************** 1. row ***************************
               Slave_IO_State: Waiting for master to send event
                  Master_Host: mysql-master
                  Master_User: slave
                  Master_Port: 3306
                Connect_Retry: 60
              Master_Log_File: mysql-bin-log.000001
          Read_Master_Log_Pos: 757
               Relay_Log_File: VM-0-17-ubuntu-relay-bin.000048
                Relay_Log_Pos: 324
        Relay_Master_Log_File: mysql-bin-log.000001
             Slave_IO_Running: Yes
            Slave_SQL_Running: Yes
              Replicate_Do_DB:
          Replicate_Ignore_DB:
           Replicate_Do_Table:
       Replicate_Ignore_Table:
      Replicate_Wild_Do_Table:
  Replicate_Wild_Ignore_Table:
                   Last_Errno: 0
                   Last_Error:
                 Skip_Counter: 0
          Exec_Master_Log_Pos: 757
              Relay_Log_Space: 239738
              Until_Condition: None
               Until_Log_File:
```

当 Slave_IO_Running: Yes;Slave_SQL_Running: Yes都是Yes状态时，表示同步成功

如果当Slave_SQL_Running: No时，解决如下

1.程序可能在slave上进行了写操作

2.也可能是slave机器重起后，事务回滚造成的.

一般是事务回滚造成的：

解决办法：

```
mysql> stop slave ;//关闭复制
mysql> set GLOBAL SQL_SLAVE_SKIP_COUNTER=1;
mysql> start slave ;
```