**MySQL主主复制与高可用**

# MySQL Replication 概述

MySQL Replication 俗称MySQL AB复制、主从复制、主从同步，是MySQL官方推荐的数据同步技术。数据同步基本过程为从数据库会实时去读取主数据库的二进制日志文件，按照日志中记录对从库进行同样的操作，以达到数据同步效果。

MySQL Replication优点：

* 通过增加从服务器来提高数据库平台的可靠性。在主服务器上执行写入和更新，在从服务器上向外提供读功能，可以动态地调整从服务器的数量，从而调整数据库平台的高性能。
* 提高数据安全性，因为数据已复制到从服务器，主数据库数据异常时，可以将从服务器复制进程终止来达到保护数据完整性的特点。
* 在主服务器上生成实时数据，而在从服务器上分析这些数据，从而缓解主服务器的性能压力。

## MySQL复制类型

**异步复制（Asynchronous replication）**

MySQL默认的复制即是异步的，主库在执行完客户端提交的事务后会立即将结果返给给客户端，并不关心从库是否已经接收并处理了事务，这样就会有一个问题，主库如果down掉了，此时主上已经提交的事务可能并没有传到从库服务器上，如果此时，强行将从提升为主，可能会导致新主上的数据不完整。默认情况下MySQL5.5/5.6/5.7和mariaDB10.0/10.1的复制功能是异步的。

**全同步复制（Fully synchronous replication）**

指当主库执行完一个事务，所有的从库都执行了该事务才返回给客户端。因为需要等待所有从库执行完该事务才能返回，所以全同步复制的性能必然会收到严重的影响，返回客户端的响应速度也会被拖慢。

**半同步复制（Semisynchronous replication）**

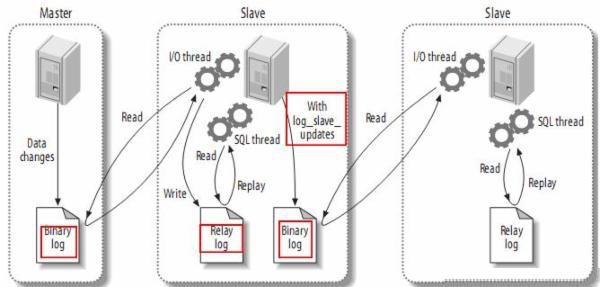
MySQL5.5由Google贡献的补丁才开始支持半同步复制（semi Replication）模式，介于异步复制和全同步复制之间，主库在执行完客户端提交的事务后不是立刻返回给客户端，而是等待至少一个从库接收到并写到relay log中才返回给客户端。相对于异步复制，半同步复制提高了数据的安全性，同时它也造成了一定程度的延迟.所以，半同步复制最好在低延时的网络中使用。当出现超时情况时，源主服务器会暂时切换到异步复制模式，直到至少有一台设置为半同步复制模式的从服务器及时收到信息为止。

半同步复制模式在主服务器和从服务器同时启用，否则主服务器默认使用异步复制模式。

## MySQL 支持的复制方式

* 基于 SQL 语句的复制(statement-based replication, SBR)：在主服务器上执行的SQL语句，在从服务器上执行同样的SQL语句，效率比较高。
* 基于行的复制(row-based replication, RBR)：主服务器把表的行变化作为事件写入到二进制日志中，主服务器把代表了行变化的事件复制到从服务器中。
* 混合模式复制(mixed-based replication, MBR)：先采用基于语句的复制，一旦发现基于语句无法精确复制时，再采用行。

## 复制的工作过程



1. 在每个事务更新完成数据之前，Master会在二进制日志（Binary log，binlog）中记录这些改变数据的语句，当MySQL将事务串行的写入二进制日志完成后，Master通知存储引擎提交事务并将数据写入磁盘。
2. Slave开启一个I/O工作线程，在Master之间建立一个连接，然后开始Binlog dump process,Binlog dump process会从Master的二进制日志中读取操作事件，如果已经跟Master达到一致状态，它会催眠并等待Master产生新的操作事件，I/O线程将这些改变数据的事件写入自己的中继日志。
3. SQL Slave Thread（SQL工作线程）是处理MySQL Replication过程的最后一步。SQL线程从中继日志读取操作事件，重放其中的事件从而更新Slave服务器的数据，使slave与Master中的数据保持一致，只要该线程与I/O线程保持一致，中继日志通常会在OS的缓存中，所以中继日志的开销很小。

# MySQL主主高可用方案

* MySQL主主+Keepalived
* MySQL+DRBD+Heartbeat

在企业中，数据库高可用一直是企业的重中之重，中小企业很多都是使用mysql主主方案或者一主多从+写分离（三期课程已学习）等，但是单主存在单点故障，从库切换成主库需要作改动。因此，如果是双主或者多主，就会增加mysql入口，增加高可用。

主主方案实现思路

1. 两台mysql都可读写，互为主备。默认只使用一台masterA负责数据的写入，另一台masterB备用处于备用状态；
2. masterA是masterB的主库，masterB又是masterA的主库，它们互为主从；
3. 两台主库之间做高可用,可以采用keepalived等方案，使用VIP对外提供服务；

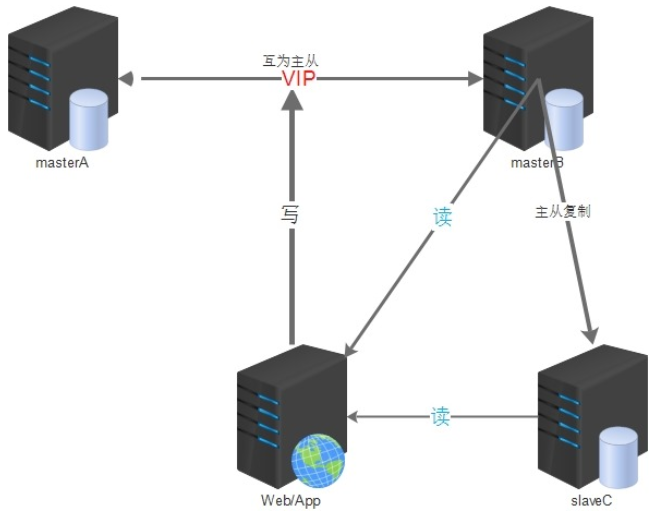
4、所有提供服务的从服务器与masterB进行主从同步（双主多从）;

5、建议采用高可用策略的时候，masterA或masterB均不因宕机恢复后而抢占VIP（非抢占模式）；

这样做可以在一定程度上保证主库的高可用,在一台主库down掉之后,可以在极短的时间内切换到另一台主库上，尽可能减少主库宕机对业务造成的影响，减少了主从同步给生产主库带来的压力；

但是也有几个不足的地方:

* masterB可能会一直处于空闲状态（可以用它当从库，负责部分查询）；
* 主库后面提供服务的从库要等masterB先同步完了数据后才能去masterB上去同步数据，这样可能会造成一定程度的同步延时；



配置部署：准备两台mysql服务器192.168.200.111和192.168.200.112

第1台机器：安装mysql并配置

[root@localhost ~]# yum -y install mariadb mariadb-devel mariadb-server

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

[mysqld]

server-id=1

log-bin=mysql-binlog #打开二进制功能

log-slave-updates=true

max\_binlog\_size=1024M #binlog单文件最大值

**auto\_increment\_offset = 1** **#**在主主同步配置时，需要将两台服务器的auto\_increment\_increment增长量都配置为2，而要把auto\_increment\_offset分别配置为1和2.这样才可以避免两台服务器同时做更新时自增长字段的值之间发生冲突

auto\_increment\_increment = 2 #奇数ID，1,3,5,7…

replicate-ignore-db = information\_schema #不需要同步的数据库

replicate-ignore-db = performance\_schema

replicate-ignore-db = test

replicate-ignore-db = mysql

relay-log=relay-log-bin #开启中继日志

relay-log-index=slave-relay-bin.index

**[root@localhost ~]# systemctl restart mariadb**

**[root@localhost ~]# mysql**

**MariaDB [(none)]> grant replication slave on \*.\* to 'repl'@'192.168.200.112' identified by '123456';**

**MariaDB [(none)]> grant replication slave on \*.\* to 'repl'@'centos02' identified by '123456';**

**Query OK, 0 rows affected (0.00 sec)**

**MariaDB [(none)]> flush privileges;**

**Query OK, 0 rows affected (0.00 sec)**

**MariaDB [(none)]> show master status;**

**+---------------------+----------+--------------+------------------+**

**| File | Position | Binlog\_Do\_DB | Binlog\_Ignore\_DB |**

**+---------------------+----------+--------------+------------------+**

**| mysql-binlog.000001 | 486 | | |**

**+---------------------+----------+--------------+------------------+**

**1 row in set (0.00 sec)**

第2台机器

**[root@localhost ~]# yum -y install mariadb mariadb-devel mariadb-server**

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

**[mysqld]**

**server-id = 2**

**log-bin=mysql-binlog**

**log-slave-updates=true**

**max\_binlog\_size=1024M**

**auto\_increment\_offset = 2**

**auto\_increment\_increment = 2 #偶数ID**

**replicate-ignore-db = information\_schema**

**replicate-ignore-db = performance\_schema**

**replicate-ignore-db = test**

**replicate-ignore-db = mysql**

**relay-log=relay-log-bin**

**relay-log-index=slave-relay-bin.index**

**[root@localhost ~]# systemctl restart mariadb**

**[root@localhost ~]# mysql**

**MariaDB [(none)]> grant replication slave on \*.\* to 'repl'@'192.168.200.111' identified by '123456';**

**MariaDB [(none)]> grant replication slave on \*.\* to 'repl'@'centos01' identified by '123456';**

**Query OK, 0 rows affected (0.00 sec)**

**MariaDB [(none)]> flush privileges;**

**Query OK, 0 rows affected (0.00 sec)**

**MariaDB [(none)]> show master status;**

**+---------------------+----------+--------------+------------------+**

**| File | Position | Binlog\_Do\_DB | Binlog\_Ignore\_DB |**

**+---------------------+----------+--------------+------------------+**

**| mysql-binlog.000001 | 486 | | |**

**+---------------------+----------+--------------+------------------+**

**1 row in set (0.00 sec)**

## 特别参数说明

**log-slave-updates = true #将复制事件写入binlog,一台服务器既做主库又做从库此选项必须要开启**

masterA自增长ID

**auto\_increment\_offset = 1**

**auto\_increment\_increment = 2**  #奇数ID

masterB自增加ID

**auto\_increment\_offset = 2**

**auto\_increment\_increment = 2 #偶数ID**

测试环境，保证没数据写入。

第1台机器（111）

**MariaDB [(none)]> stop slave;**

**Query OK, 0 rows affected, 1 warning (0.00 sec)**

**MariaDB [(none)]> change master to master\_host='192.168.200.112',master\_port=3306,master\_user='repl',master\_password='123456',master\_log\_file='mysql-binlog.000001',master\_log\_pos=889;**

**Query OK, 0 rows affected (0.04 sec)**

**MariaDB [(none)]> start slave;**

**Query OK, 0 rows affected (0.01 sec)**

**MariaDB [(none)]> show slave status\G;**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 1. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Slave\_IO\_State: Waiting for master to send event**

**Master\_Host: 192.168.200.112**

**Master\_User: repl**

**Master\_Port: 3306**

**Connect\_Retry: 60**

**Master\_Log\_File: mysql-binlog.000001**

**Read\_Master\_Log\_Pos: 486**

**Relay\_Log\_File: relay-log-bin.000002**

**Relay\_Log\_Pos: 532**

**Relay\_Master\_Log\_File: mysql-binlog.000001**

**Slave\_IO\_Running: Yes**

**Slave\_SQL\_Running: Yes**

第2台机器（112）

**MariaDB [(none)]> stop slave;**

**Query OK, 0 rows affected (0.01 sec)**

**MariaDB [(none)]> change master to master\_host='192.168.200.111',master\_port=3306,master\_user='repl',master\_password='123456',master\_log\_file='mysql-binlog.000001',master\_log\_pos=486;**

**Query OK, 0 rows affected (0.13 sec)**

**如果有出现ERROR 1201 (HY000): Could not initialize master info structure; more error messages can be found in the MariaDB error log报错信息，执行reset slave再做或者将中继日志信息删除了重新配置**

**MariaDB [(none)]> start slave;**

**Query OK, 0 rows affected (0.01 sec)**

**MariaDB [(none)]> show slave status\G;**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 1. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Slave\_IO\_State: Waiting for master to send event**

**Master\_Host: 192.168.200.111**

**Master\_User: repl**

**Master\_Port: 3306**

**Connect\_Retry: 60**

**Master\_Log\_File: mysql-binlog.000004**

**Read\_Master\_Log\_Pos: 486**

**Relay\_Log\_File: relay-log-bin.000002**

**Relay\_Log\_Pos: 532**

**Relay\_Master\_Log\_File: mysql-binlog.000004**

**Slave\_IO\_Running: Yes**

**Slave\_SQL\_Running: Yes**

第1台机器

**MariaDB [(none)]> create database test01;**

**Query OK, 1 row affected (0.01 sec)**

第2台机器

**MariaDB [(none)]> show databases;**

**+--------------------+**

**| Database |**

**+--------------------+**

**| information\_schema |**

**| mysql |**

**| performance\_schema |**

**| test |**

**| test01 |**

**+--------------------+**

**5 rows in set (0.00 sec)**

**MariaDB [(none)]> create database test02;**

**Query OK, 1 row affected (0.00 sec)**

第1台机器

**MariaDB [(none)]> show databases;**

**+--------------------+**

**| Database |**

**+--------------------+**

**| information\_schema |**

**| mysql |**

**| performance\_schema |**

**| test |**

**| test01 |**

**| test02 |**

**+--------------------+**

**6 rows in set (0.00 sec)**

# MySQL高可用方案

masterA（111）配置

**[root@localhost ~]# yum -y install keepalived**

**[root@localhost ~]# cat /etc/keepalived/keepalived.conf**

**! Configuration File for keepalived**

**global\_defs {**

**router\_id LVS\_MASTER-A**

**}**

**vrrp\_script mysql {**

**script "/opt/mysql.sh"**

**interval 2 #执行脚本的时间间隔**

**weight -5 #mysql down时，将优先级降低5（-5）**

**fall 2 #检测失败的最大次数，2表示请求失败两次就认为节点资源故障**

**rise 1 #定义请求成功的次数，1表示一次成功就认为节点资源恢复**

**}**

**vrrp\_instance VI\_1 {**

**state MASTER**

**interface ens33**

**virtual\_router\_id 51**

**priority 100**

**nopreempt**

**advert\_int 1**

**authentication {**

**auth\_type PASS**

**auth\_pass 1111**

**}**

**track\_script {**

**mysql**

**}**

**virtual\_ipaddress {**

**192.168.200.254**

**}**

**}**

**[root@localhost ~]# cat /opt/mysql.sh**

**#!/bin/bash**

**counter=$(netstat -na|grep "LISTEN"|grep "mysql"|wc -l)**

**if [ "${counter}" -eq 0 ]; then**

**systemctl stop keepalived**

**fi**

**[root@localhost ~]# chmod +x /opt/mysql.sh**

**[root@localhost ~]# systemctl start keepalived**

**[root@localhost ~]# ip a | grep ens33**

**2: ens33: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc pfifo\_fast state UP group default qlen 1000**

**inet 192.168.200.111/24 brd 192.168.200.255 scope global ens33**

**inet 192.168.200.254/32 scope global ens33**

**[root@localhost ~]# tail -f /var/log/messages**

**Feb 12 17:23:23 localhost Keepalived\_vrrp[70087]: VRRP\_Instance(VI\_1) Transition to MASTER STATE**

**Feb 12 17:23:24 localhost Keepalived\_vrrp[70087]: VRRP\_Instance(VI\_1) Entering MASTER STATE**

**Feb 12 17:23:24 localhost Keepalived\_vrrp[70087]: VRRP\_Instance(VI\_1) setting protocol iptable drop rule**

**Feb 12 17:23:24 localhost Keepalived\_vrrp[70087]: VRRP\_Instance(VI\_1) setting protocol VIPs.**

**Feb 12 17:23:24 localhost Keepalived\_vrrp[70087]: Sending gratuitous ARP on ens33 for 192.168.200.254**

**Feb 12 17:23:24 localhost Keepalived\_vrrp[70087]: VRRP\_Instance(VI\_1) Sending/queueing gratuitous ARPs on ens33 for 192.168.200.254**

**Feb 12 17:23:24 localhost Keepalived\_vrrp[70087]: Sending gratuitous ARP on ens33 for 192.168.200.254**

masterB（112）配置

**[root@localhost ~]# cat /etc/keepalived/keepalived.conf**

**! Configuration File for keepalived**

**global\_defs {**

**router\_id LVS\_MASTER-B**

**}**

**vrrp\_script mysql {**

**script "/opt/mysql.sh"**

**interval 2**

**weight -5**

**fall 2**

**rise 1**

**}**

**vrrp\_instance VI\_1 {**

**state BACKUP**

**interface ens33**

**virtual\_router\_id 51**

**priority 99**

**advert\_int 1**

**authentication {**

**auth\_type PASS**

**auth\_pass 1111**

**}**

**track\_script {**

**mysql**

**}**

**virtual\_ipaddress {**

**192.168.200.254**

**}**

**}**

**[root@localhost ~]# cat /opt/mysql.sh**

**#!/bin/bash**

**counter=$(netstat -na|grep "LISTEN"|grep "mysql"|wc -l)**

**if [ "${counter}" -eq 0 ]; then**

**systemctl stop keepalived**

**fi**

**[root@localhost ~]# chmod +x /opt/mysql.sh**

**[root@localhost ~]# systemctl start keepalived**

**[root@localhost ~]# tail -f /var/log/messages**

**Feb 12 17:28:02 localhost Keepalived\_healthcheckers[70075]: WARNING - script '/root/shutdown.sh' is not executable for uid:gid 0:0 - disabling.**

**Feb 12 17:28:02 localhost Keepalived\_healthcheckers[70075]: SECURITY VIOLATION - check scripts are being executed but script\_security not enabled.**

**Feb 12 17:28:02 localhost kernel: ip\_set: protocol 6**

**Feb 12 17:28:02 localhost kernel: IPVS: [wrr] scheduler registered.**

**Feb 12 17:28:02 localhost Keepalived\_healthcheckers[70075]: Activating healthchecker for service [192.168.200.254]:3306**

**Feb 12 17:28:02 localhost Keepalived\_vrrp[70076]: VRRP\_Instance(VI\_1) removing protocol VIPs.**

**Feb 12 17:28:02 localhost Keepalived\_vrrp[70076]: VRRP\_Instance(VI\_1) removing protocol iptable drop rule**

**Feb 12 17:28:02 localhost Keepalived\_vrrp[70076]: Using LinkWatch kernel netlink reflector...**

**Feb 12 17:28:02 localhost Keepalived\_vrrp[70076]: VRRP\_Instance(VI\_1) Entering BACKUP STATE**

**Feb 12 17:28:02 localhost Keepalived\_vrrp[70076]: VRRP sockpool: [ifindex(2), proto(112), unicast(0), fd(10,11)]**

1. 测试VIP转移

masterA配置

**[root@localhost ~]# systemctl stop mariadb**

**[root@localhost ~]# ip a | grep ens33**

**2: ens33: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc pfifo\_fast state UP group default qlen 1000**

**inet 192.168.200.111/24 brd 192.168.200.255 scope global ens33**

**[root@localhost ~]# ps aux | grep mysql**

**root 83651 0.0 0.0 112720 980 pts/2 S+ 14:30 0:00 grep --color=auto mysql**

masterB配置

**[root@localhost ~]# ip a | grep ens33**

**2: ens33: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc pfifo\_fast state UP group default qlen 1000**

**inet 192.168.200.112/24 brd 192.168.200.255 scope global ens33**

**inet 192.168.200.254/32 scope global ens33**

**[root@localhost ~]# tail -f /var/log/messages**

**Feb 13 14:30:02 localhost Keepalived\_vrrp[79095]: VRRP\_Instance(VI\_1) Entering MASTER STATE**

**Feb 13 14:30:02 localhost Keepalived\_vrrp[79095]: VRRP\_Instance(VI\_1) setting protocol VIPs.**

**Feb 13 14:30:02 localhost Keepalived\_vrrp[79095]: Sending gratuitous ARP on ens33 for 192.168.200.254**

**Feb 13 14:30:02 localhost Keepalived\_vrrp[79095]: VRRP\_Instance(VI\_1) Sending/queueing gratuitous ARPs on ens33 for 192.168.200.254**

1. 在远程客户端测试

mysql服务器授权

**[root@localhost ~]# mysql -uroot**

**MariaDB [(none)]> grant all on \*.\* to 'root'@'192.168.200.%' identified by '123456';**

**MariaDB [(none)]> flush privileges;**

通过vip登陆测试

**[root@localhost ~]# mysql -uroot -p123456 -h 192.168.200.254**

**Welcome to the MariaDB monitor. Commands end with ; or \g.**

**Your MariaDB connection id is 5796**

**Server version: 5.5.56-MariaDB MariaDB Server**

**Copyright (c) 2000, 2017, Oracle, MariaDB Corporation Ab and others.**

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

**MariaDB [(none)]> show databases;**

**+--------------------+**

**| Database |**

**+--------------------+**

**| information\_schema |**

**| mysql |**

**| performance\_schema |**

**| test |**

**| test01 |**

**| test02 |**

**+--------------------+**

**6 rows in set (0.00 sec)**