

MySQL 的主从复制

Organization: 千锋教育 Python 教学部

Date : 2019-10-23

Author: [张旭](#)

一、什么是主从复制

MySQL 主从复制是其最重要的功能之一. 主从复制是指一台服务器充当主数据库服务器, 另一台或多台服务器充当从数据库服务器, 主服务器中的数据自动复制到从服务器之中. 对于多级复制, 数据库服务器即可充当主机, 也可充当从机.

MySQL主从复制的基础是主服务器对数据库修改记录二进制日志, 从服务器通过主服务器的二进制日志自动执行更新.

二、主从复制的类型

1. 基于语句的复制:

主服务器上面执行的语句在从服务器上面再执行一遍.

存在的问题: 时间上可能不完全同步造成偏差, 执行语句的用户也可能是不同一个用户.

2. 基于行的复制:

把主服务器上面改编后的内容直接复制过去, 而不关心到底改变该内容是由哪条语句引发的.

存在的问题: 比如一个工资表中有一万个用户, 我们把每个用户的工资+1000, 那么基于行的复制则要复制一万行的内容, 由此造成的开销比较大, 而基于语句的复制仅仅一条语句就可以了.

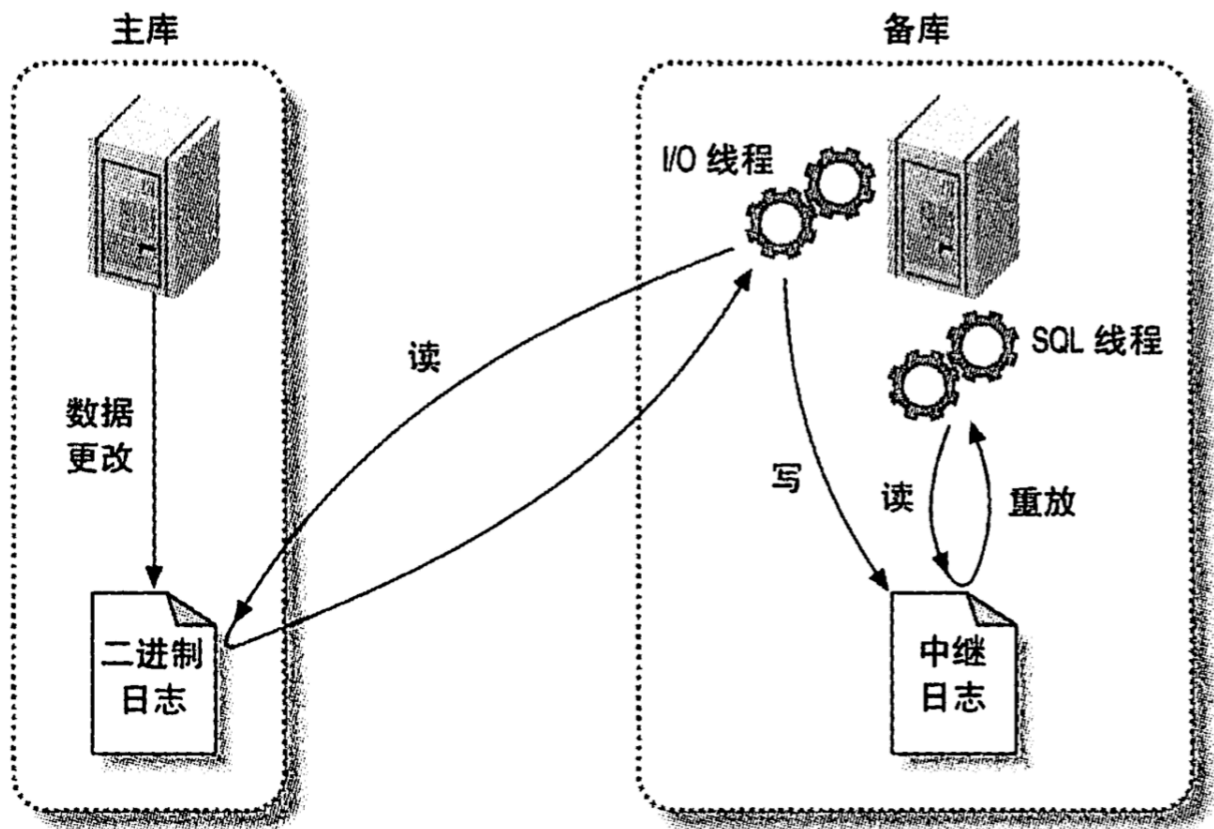
3. 混合类型的复制:

MySQL 默认使用基于语句的复制, 当基于语句的复制会引发问题的时候就会使用基于行的复制, MySQL会自动进行选择.

在MySQL主从复制架构中, 读操作可以在所有的服务器上面进行, 而写操作只能在主服务器上面进行. 主从复制架构虽然给读操作提供了扩展, 可如果写操作也比较多的话 (多台从服务器还要从主服务器上面同步数据), 单主模型的复制中主服务器势必会成为性能瓶颈.

三、主从复制原理

1. 主服务器上面的任何修改都会保存在二进制日志 Binary log 里面.
2. 从服务器上面启动一个 I/O thread (实际上就是一个主服务器的客户端进程), 连接到主服务器上面请求读取二进制日志, 然后把读取到的二进制日志写到本地的一个 Relay log 里面.
3. 从服务器上面开启一个 SQL thread 定时检查 Relay log, 如果发现有更改变立即把更改的内容在本机上面执行一遍.
4. 如果一主多从的话, 这时主库既要负责写, 又要负责同时为多个从库提供二进制日志. 所以可以稍做调整, 将二进制日志只给某一从, 这一从再开启二进制日志并将自己的二进制日志再发给其它从. 或者是干脆这个从不记录只负责将二进制日志转发给其它从, 这样架构起来性能可能要好得多, 而且数据之间的延时应该也稍微要好一些. 工作原理图如下:



四、主从复制的配置

最简单的复制模式就是一主一从的复制模式了, 这样一个简单的架构只需要如下几个步骤:

主机上的操作:

1. 在主机上建立一个用户, 专门用来进行主从复制

```
1 | GRANT REPLICATION SLAVE ON *.* TO 'backup'@'%' IDENTIFIED BY '123';
```

2. 建立一个主节点, 开启 binlog, 设置服务器id

在主机上找到 `/etc/mysql/my.cnf` 文件, 并打开. 将 `[mysqld]` 的段落改为如下内容

```

1  [mysqld]
2  # 主机绑定 0.0.0.0 确保从机可以访问。生产环境下出于安全考虑
   # 会绑定内网地址
3  bind-address = 0.0.0.0
4
5  # 设置主机 ID
6  server-id    = 1
7
8  # 主机读写模式：0 表示读写都可以
9  read-only    = 0
10
11 # 主机的 bin log 的路径
12 log-bin      = /var/log/mysql-bin.log

```

3. 然后重启 MySQL 服务: `sudo systemctl restart mysql`

4. 检查二进制日志是否启动

进入 MySQL, 执行如下语句.

```

1  show variables like 'log_%';

```

看到 log_bin 的值为 `ON` 表示成功

5. 查看 master 状态

```

1  show master status;
2
3  +-----+-----+-----+
4  | File           | Position | ... |
5  +-----+-----+-----+
6  | mysql-bin.000001 | 434      | ... |
7  +-----+-----+-----+

```

记下 File 和 Position 的值

6. 建立一个从节点, 设置服务器id;

同样, 再从机上找到 `/etc/mysql/my.cnf` 文件, 并打开, 添加如下内容:

```
1 [mysqld]
2 server-id = 2 # 设置从机的 id
3 read-only = 1 # 从机设置为只读模式
```

然后重启: `sudo systemctl restart mysql`

7. 再从机上进入 mysql, 并执行如下语句:

```
1 -- 指定主机
2 change master to
3 master_host='192.168.1.10',
4 master_user='backup',
5 master_password='123',
6 master_log_file='mysql-bin.000001',
7 master_log_pos=434;
8
9 -- 启动从库
10 start slave;
11
12 -- 查看从库
13 show slave status\G;
```

看到从机状态中包含如下信息说明大功告成:

```
1 Slave_IO_Running: Yes
2 Slave_SQL_Running: Yes
```