第 5 章 主从复制

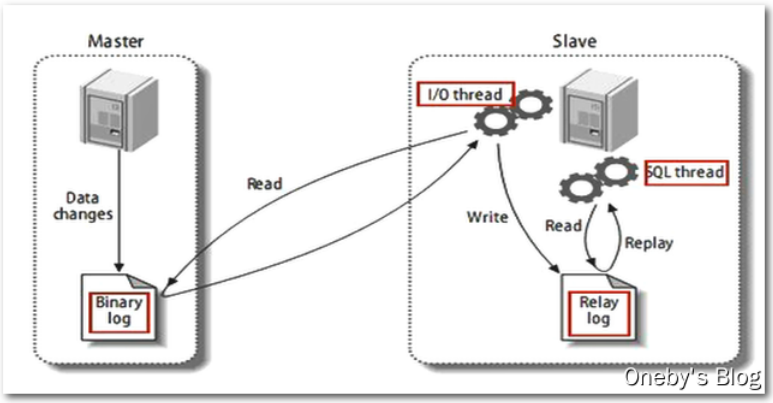
第 5 章 主从复制

1、复制的基本原理

> 复制的基本原理

slave会从master读取binlog来进行数据同步，主从复制的三步骤

1. master将改变记录到二进制日志（binary log）。这些记录过程叫做**二进制日志事件**（binary log events）
2. slave将master的binary log events拷贝到它的**中继日志（relay log）**
3. **slave重做中继日志中的事件**，将改变应用到自己的数据库中。MySQL复制是异步的且串行化的



2、复制的基本原则

1. 每个slave只有一个master
2. 每个slave只能有一个唯一的服务器ID
3. 每个master可以有多个salve

3、复制最大问题

因为发生多次 IO， 存在延时问题

4、一主一从常见配置

> **前提：mysql 版本一致，主从机在同一网段下**

ping 测试

- Linux 中 ping Windows

```
[root@Heygo 桌面]# ping 10.206.207.131
PING 10.206.207.131 (10.206.207.131) 56(84) bytes of data.
64 bytes from 10.206.207.131: icmp_seq=1 ttl=128 time=1.27 ms
64 bytes from 10.206.207.131: icmp_seq=2 ttl=128 time=0.421 ms
64 bytes from 10.206.207.131: icmp_seq=3 ttl=128 time=1.12 ms
64 bytes from 10.206.207.131: icmp_seq=4 ttl=128 time=0.515 ms
^C
--- 10.206.207.131 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3719ms
rtt min/avg/max/mdev = 0.421/0.835/1.279/0.373 ms
[root@Heygo 桌面]#
```

- Windows 中 ping Linux

```
C:\Users\Heygo>ping 192.168.152.129

```

```
正在 Ping 192.168.152.129 具有 32 字节的数据:
来自 192.168.152.129 的回复: 字节=32 时间<1ms TTL=64
来自 192.168.152.129 的回复: 字节=32 时间<1ms TTL=64
来自 192.168.152.129 的回复: 字节=32 时间=1ms TTL=64
来自 192.168.152.129 的回复: 字节=32 时间<1ms TTL=64

192.168.152.129 的 Ping 统计信息:
    数据包: 已发送 = 4, 已接收 = 4, 丢失 = 0 (0% 丢失),
往返行程的估计时间(以毫秒为单位):
    最短 = 0ms, 最长 = 1ms, 平均 = 0ms
```

> **主机修改 my.ini 配置文件（Windows）**

主从都配置都在 [mysqld] 节点下，都是小写，以下是老师的配置文件

```
[mysqld]

# The TCP/IP Port the MySQL Server will listen on
port=3306
server-id=1
log-bin=D:/devSoft/MySQLServer5.5/data/mysqlbin
log-err=D:/devSoft/MySQLServer5.5/data/mysqlerr

#Path to installation directory. All paths are usually resolved relative to this.
basedir="D:/devSoft/MySQLServer5.5/"
tmpdir="D:/devSoft/MySQLServer5.5/"

#Path to the database root
datadir="D:/devSoft/MySQLServer5.5/Data/"
read-only=0
binlog-ignore-db=mysql
binlog-do-db=ww
```

以下两条为必须配置

- 配置主机 id

```
server-id=1
```

- 启用二进制日志

```
log-bin=C:/Program Files (x86)/MySQL/MySQL Server 5.5/log-bin/mysqlbin
```

以下为非必须配置

- 启动错误日志

```
log-err=C:/Program Files (x86)/MySQL/MySQL Server 5.5/log-bin/mysqlerr
```

- 根目录

```
basedir="C:/Program Files (x86)/MySQL/MySQL Server 5.5/"
```

- 临时目录

```
tmpdir="C:/Program Files (x86)/MySQL/MySQL Server 5.5/"
```

- 数据目录

```
datadir="C:/Program Files (x86)/MySQL/MySQL Server 5.5/Data/"
```

- 主机，读写都可以

```
read-only=0
```

- 设置不要复制的数据库

```
binlog-ignore-db=mysql
```

- 设置需要复制的数据

```
binlog-do-db=需要复制的主数据库名字
```

> 从机修改 my.cnf 配置文件（Linux）

- 【必须】从服务器唯一ID

```
server-id=2
```

- 【可选】启用二进制文件

> 修改配置文件后的准备工作

因修改过配置文件，主机+从机都重启 mysql 服务

- Windows

```
net stop mysql
net start mysql
```

- Linux

```
service mysqld restart
```

主机从机都关闭防火墙

- Windows 手动关闭防火墙
- 关闭虚拟机 linux 防火墙

```
service iptables stop
```

> 在 Windows 主机上简历账户并授权 slave

- 创建用户， 并赋予从机 REPLICATION 权限（从主机的数据库表中复制表）

```
GRANT REPLICATION SLAVE ON *.* TO '备份账号'@'从机器数据库 IP' IDENTIFIED BY '账号密码';
```

```
GRANT REPLICATION SLAVE ON *.* TO 'Heygo'@'192.168.152.129' IDENTIFIED BY '123456';
```

- 刷新权限信息

```
flush privileges;
```

```
mysql> flush privileges;
Query OK, 0 rows affected (0.00 sec)
```

- 通过 `select * from mysql.user where user='Heygo'\G;` 命令可查看：从机只有 `Repl_slave_priv` 权限为 `Y`，其余权限均为 `N`

```
mysql> select * from mysql.user where user='Heygo'\G;
*************************** 1. row ***************************
                  Host: %
                  User: Heygo
              Password: *6BB4837EB74329105EE4568DDA7DC67ED2CA2AD9
```

```
           Select_priv: N
           Insert_priv: N
           Update_priv: N
           Delete_priv: N
           Create_priv: N
             Drop_priv: N
           Reload_priv: N
         Shutdown_priv: N
          Process_priv: N
             File_priv: N
            Grant_priv: N
       References_priv: N
            Index_priv: N
            Alter_priv: N
          Show_db_priv: N
            Super_priv: N
 Create_tmp_table_priv: N
      Lock_tables_priv: N
          Execute_priv: N
       Repl_slave_priv: Y
      Repl_client_priv: N
      Create_view_priv: N
        Show_view_priv: N
   Create_routine_priv: N
    Alter_routine_priv: N
      Create_user_priv: N
            Event_priv: N
          Trigger_priv: N
Create_tablespace_priv: N
              ssl_type:
            ssl_cipher:
           x509_issuer:
          x509_subject:
         max_questions: 0
           max_updates: 0
       max_connections: 0
  max_user_connections: 0
                plugin:
 authentication_string: NULL
1 row in set (0.00 sec)
```

- 查询 master 的状态，将 File 和 Position 记录下来，在启动 Slave 时需要用到这两个参数

```
show master status;
```

```
mysql> show master status;
+------------------+----------+--------------+------------------+
| File             | Position | Binlog_Do_DB | Binlog_Ignore_DB |
+------------------+----------+--------------+------------------+
| mysql-bin.000001 |      107 | mysql        |                  |
+------------------+----------+--------------+------------------+
1 row in set (0.00 sec)
```

> **在 Linux 从上验证是否能登陆主机的 MySQL**

- 在从机上执行 `mysql -h 10.206.207.131 -uHeygo -p` 命令，发现无法连接主机的 MySQL 数据库

```
[root@Heygo 桌面]# mysql -h 10.206.207.131 -uHeygo -p
Enter password:
ERROR 1130 (HY000): Host 'windows10.microdone.cn' is not allowed to connect to this MySQL server
```

- 查阅资料发现：当你远程登录 MySQL 时，使用的账号要有特殊要求，如果要使用某个账号来远程登录，必须将账号的 host 属性值更改成 `%`。我敲，阳哥怎么就成功了呢？可以看到：我们在执行了 `GRANT REPLICATION SLAVE ON *.* TO 'Heygo'@'192.168.152.129' IDENTIFIED BY '123456';` 命令之后，Heygo 账户的 host 属性为 `192.168.152.129`

```
mysql> select user,host,plugin from user;
+-------+-----------------+--------+
| user  | host            | plugin |
+-------+-----------------+--------+
| root  | localhost       |        |
| root  | 192.168.152.129 |        |
| Heygo | 192.168.152.129 |        |
+-------+-----------------+--------+
3 rows in set (0.00 sec)
```

- 于是我先使用 `update user set host = '%' where user = 'Heygo';` 命令将 Heygo 账户的 `host` 字段设置为 `%`；然后使用 `flush privileges;` 命令刷新权限信息

```
mysql> update user set host = '%' where user = 'Heygo';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> flush privileges;
Query OK, 0 rows affected (0.00 sec)

mysql> select user,host,plugin from user;
+-------+-----------------+---------+
| user  | host            | plugin |
+-------+-----------------+---------+
| root  | localhost       |        |
| root  | 192.168.152.129 |        |
| Heygo | %               |        |
+-------+-----------------+---------+
3 rows in set (0.00 sec)
```

- 在 Linux 从机上使用 `mysql -h 10.206.207.131 -uHeygo -p` 命令能够成功连接上主机上的 MySQL 数据库。我敲，谜一样

```
[root@Heygo 桌面]# mysql -h 10.206.207.131 -uHeygo -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 33
Server version: 5.5.15-log MySQL Community Server (GPL)

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

> **在 Linux 从机上配置需要复制的主机**

- 从机进行认证

```
CHANGE MASTER TO
MASTER_HOST='主机 IP',
MASTER_USER='创建用户名',
MASTER_PASSWORD='创建的密码',
MASTER_LOG_FILE='File 名字',
MASTER_LOG_POS=Position数字;
```

```
CHANGE MASTER TO
MASTER_HOST='10.206.207.131',
MASTER_USER='Heygo',
MASTER_PASSWORD='123456',
MASTER_LOG_FILE='mysql-bin.000001',
MASTER_LOG_POS=107;
```

- 启动从服务器复制功能

```
start slave;
```

- 查看从机复制功能是否启动成功：使用 `show slave status\G;` 命令查看 `Slave_SQL_Running:Yes` 和 `Slave_IO_Running:Yes` 说明从机连接主机成功（第一次测试没有成功，这是隔了半年之后的测试，因此某些数据会有出入）

```
mysql> show slave status\G;
*************************** 1. row ***************************
               Slave_IO_State: Waiting for master to send event
                  Master_Host: 10.206.207.131
                  Master_User: Heygo
                  Master_Port: 3306
                Connect_Retry: 60
              Master_Log_File: mysql-bin.000052
          Read_Master_Log_Pos: 4274
               Relay_Log_File: mysqld-relay-bin.000063
                Relay_Log_Pos: 2998
```

```
        Relay_Master_Log_File: mysql-bin.000052
             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: 4274
              Relay_Log_Space: 4749
              Until_Condition: None
               Until_Log_File:
                Until_Log_Pos: 0
           Master_SSL_Allowed: No
           Master_SSL_CA_File:
           Master_SSL_CA_Path:
              Master_SSL_Cert:
            Master_SSL_Cipher:
               Master_SSL_Key:
        Seconds_Behind_Master: 0
Master_SSL_Verify_Server_Cert: No
                Last_IO_Errno: 0
                Last_IO_Error:
               Last_SQL_Errno: 0
               Last_SQL_Error:
  Replicate_Ignore_Server_Ids:
             Master_Server_Id: 1
                  Master_UUID:
             Master_Info_File: /var/lib/mysql/master.info
                    SQL_Delay: 0
          SQL_Remaining_Delay: NULL
      Slave_SQL_Running_State: Slave has read all relay log; waiting for the slave I/O thread to update it
           Master_Retry_Count: 86400
                  Master_Bind:
      Last_IO_Error_Timestamp:
     Last_SQL_Error_Timestamp:
               Master_SSL_Crl:
           Master_SSL_Crlpath:
           Retrieved_Gtid_Set:
            Executed_Gtid_Set:
                Auto_Position: 0
1 row in set (0.00 sec)
```

- 如何停止从服务复制功能

```
stop slave;
```