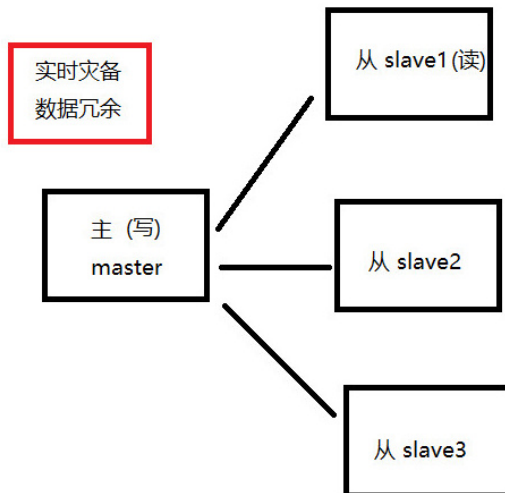


一.主从复制(replication)的概念

当master(主)库的数据发生变化的时候,变化会实时的同步到slave(从)库。(从3.5的版本开始)

二.主从同步有什么好处?

- (1)水平扩展数据库的负载能力。
- (2)容错, 高可用。Failover(失败切换)/High Availability
- (3)数据备份



三.实现过程

3.1 在master机器上,主从同步事件会被写到特殊的log文件中(binary-log);在slave机器上, slave读取主从同步事件,并根据读取的事件变化,在slave库上做相应的更改。

3.2 在master机器上,主从同步事件会被写到特殊的log文件中(binary-log);

主从同步事件有3种形式:statement、row、mixed。

statement: 会对数据库操作的sql语句写入到binlog中。

row: 会将每一条数据的变化写入到binlog中。

mixed: statement与row的混合。Mysql决定什么时候写statement格式的,什么时候写row格式的binlog

3.3在master机器上的操作

当master上的数据发生改变的时候,该事件(insert、update、delete)变化会按照顺序写入到binlog中。

binlog dump线程

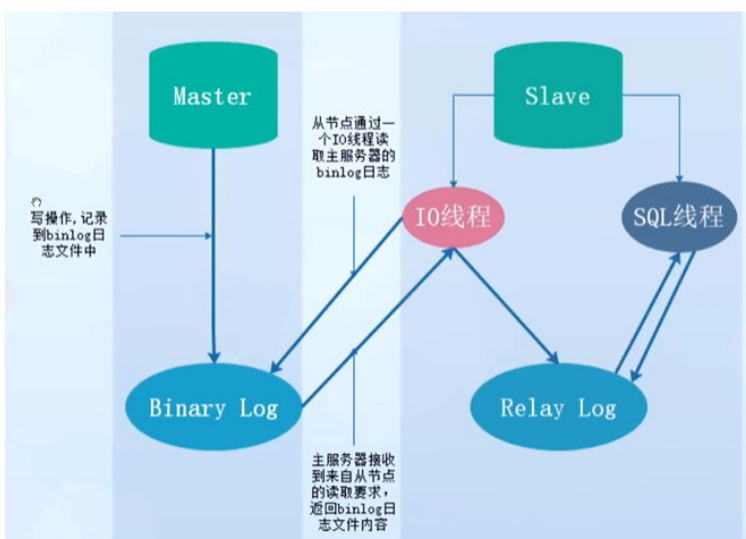
当slave连接到master的时候, master机器会为slave开启binlog dump线程。当master 的 binlog发生变化的时候, binlog dump线程会通知slave,并将相应的binlog内容发送给slave。

3.4在slave机器上的操作

当主从同步开启的时候, slave上会创建2个线程(I/O线程 / SQL线程)。

I/O线程 : 该线程连接到master机器, master机器上的binlog dump线程会将binlog的内容发送给该I/O线程。该I/O线程接收到binlog内容后,再将内容写入到本地的relay log(中继日志)。

SQL线程 : 该线程读取I/O线程写入的relay log,并且根据relay log的内容对slave数据库做相应的操作。



四.配置主从命令(这里应用的是mysql8.0版本,且ip为 192.168.100.135 | 192.168.100.136 两台机器)

配置主从的步骤

1. 确保从数据库与主数据库里的数据一致
2. 在主数据库里创建一个同步账户授权给从数据库使用
3. 配置主数据库(修改配置文件)与开启binlog日志
4. 配置从数据库(修改配置文件)
5. 配置从节点

MySQL8.0新加了很多功能,其中在用户管理中增加了角色的管理,默认的密码加密方式也做了调整,由之前的sha1改为了sha2,同时加上5.7的禁用用户和用户过期的设置,这样方便用户的管理和权限的管理,也增加了用户的安全性。

4.1 mysql> show variables like "%binlog_format%";

```
+-----+
| Variable_name | Value |
+-----+
| binlog_format | ROW   |
+-----+
1 row in set, 1 warning (0.03 sec)`
默认新版本中的复制模式默认都是行级别;
```

4.2查看binlog日志所在位置.(8.0版本默认是开启binlog的状态)

查看binlog是否开启,以及当前server_id(注意server_id代表当前实例的唯一标识,不能与其他实例相同);

```
mysql> show variables like "%log_bin%";
```

```
+-----+-----+
| Variable_name | Value |
+-----+-----+
| log_bin       | ON    |
| log_bin_basename | D:\mysql8.0.19\mysql-8.0.19-win64\data\binlog |
| log_bin_index  | D:\mysql8.0.19\mysql-8.0.19-win64\data\binlog.index |
| log_bin_trust_function_creators | OFF   |
| log_bin_use_v1_row_events | OFF   |
| sql_log_bin    | ON    |
+-----+-----+
6 rows in set, 1 warning (0.00 sec)
```

4.3查看当前数据库的server_id

```
mysql> show variables like "%server_id%";
```

```
+-----+-----+
| Variable_name | Value |
+-----+-----+
| server_id     | 1     |
| server_id_bits | 32    |
+-----+-----+
2 rows in set, 1 warning (0.00 sec)
```

4.4创建用于专门主从复制用的用户.(主库创建)

```
create user "repl"@"192.168.100.%" identified by "mysql";
```

从库所在服务器链接一下,链接成功,给用户授权.

```
grant replication slave on *.* to "repl"@"192.168.15.%";
```

查看权限

```
show grants for "repl"@"192.168.100.%";
```

4.5将主数据库中的数据进行mysqldump备份,

- 1.直接进行数据库备份 添加参数 --all-databases -master-data=2 --single-transaction 参数

参数说明:

--all-databases #表示将所有的数据库全部进行数据备份

-master-data=2 #将当前数据库中文件的binlog节点信息注释到备份文件中.

--single-transaction #会在备份的时候添加一个瞬时的锁定,为数据库创建一个快照,然后再备份过程中不会对数据的写入造成影响.

语句如下:

(1)备份所有数据库:

```
mysqldump --all-databases --master-data=2 --single-transaction -u root -p -P 3306 > ./dbdump.db
```

(2)备份单个数据库:

```
mysqldump -h127.0.0.1 -uroot -p -P 3306 php45 > d:/backupfile.sql
```

Enter password: ****

(3)同时备份多个MySQL数据库

```
mysqldump -h127.0.0.1 -uroot -ppass --databases php45 php46 > ./multibackupfile.sql
```

2.还原sql到数据库

还原MySQL数据库的命令。还原当前备份名为backupfile.sql的数据库

mysql -h127.0.0.1 -uroot -p -P 3306 数据库名 < backupfile.sql (将已经备份好的数据全部恢复到数据库中,有则覆盖无则添加)

4.6修改加密方式(caching_sha2_password)

mysql8.0重置了加密放置

```
mysql> select user,host,plugin from mysql.user;
+-----+-----+-----+
| user          | host          | plugin          |
+-----+-----+-----+
| mysql.infoschema | localhost    | caching_sha2_password |
| mysql.session  | localhost    | caching_sha2_password |
| mysql.sys      | localhost    | caching_sha2_password |
| root           | localhost    | caching_sha2_password |
+-----+-----+-----+
```

(1)修改加密方式(对于一些配置不是支持的很好),配置文件中修改

[mysqld]

default_authentication_plugin=mysql_native_password

修改后从起mysql服务

```
mysql> create user "slave1"@"192.168.100.%" identified by "mysql";
Query OK, 0 rows affected (0.02 sec)
```

```
mysql> select user,host,plugin from mysql.user;
+-----+-----+-----+
| user          | host          | plugin          |
+-----+-----+-----+
| slave1        | 192.168.100.% | mysql_native_password |
| mysql.infoschema | localhost    | caching_sha2_password |
| mysql.session  | localhost    | caching_sha2_password |
| mysql.sys      | localhost    | caching_sha2_password |
| root           | localhost    | caching_sha2_password |
+-----+-----+-----+
5 rows in set (0.00 sec)
```

(2)或者直接手动定义加密方式

```
mysql> create user "slave1"@"192.168.100.%" identified with mysql_native_password by "mysql";
FLUSH PRIVILEGES; #刷新权限
```

4.7 进行用户授权

查看权限:

```
mysql> show grants for 'slave1'@"192.168.100.%"
+-----+
| Grants for slave1@192.168.100.% |
+-----+
| GRANT USAGE ON *.* TO `slave1`@`192.168.100.%` |
+-----+
1 row in set (0.00 sec)
```

(1)为username@localhost用户赋予超级用户权限:

```
grant all privileges on *.* to 'slave1'@"192.168.100.%" with grant option;
```

(2)grant: 授权

(1)all privileges:所有的权限

(2)on *.*: 在哪个数据库的那个表

(3)to username@localhost: 对哪个用户的哪个主机

(4)with grant option: 是不是 将username用户自己本身的权限赋给其他账户

MySQL8.0的用户授权和之前有所区别,老版本的常用授权语句在8.0中会报错:

--MySQL8.0之前版本:

```
GRANT ALL ON *.* TO `root`@"127.0.0.1" IDENTIFIED BY 'password' WITH GRANT OPTION;
```

```
--MySQL8.0版本:
--创建账号密码
CREATE USER `root`@`127.0.0.1` IDENTIFIED BY 'passowrd';
--授予权限
GRANT ALL ON *.* TO `root`@`127.0.0.1` WITH GRANT OPTION;
--删除权限
REVOKE all privileges ON database.tablename FROM 'username'@'host';
--修改密码
ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'password';
```

(3)普通用户权限添加如下:

```
grant usage,select,insert,update,delete,create temporary tables,execute on php45.* to 'slave1'@'192.168.100.%';
//此时没有with grant option 表示不给其他用户赋权限
```

```
flush privileges;
```

```
usage:无权限,当你想创建一个没有权限的用户时候,指定usage
show:的权限
view:视图的权限(mysql8.0+赋权限出错)ERROR 3619 (HY000): Illegal privilege level specified for VIEW
create temporary tables:创建临时表的权限
excute: 执行的权限
REPLICATION SLAVE ON: 主从的权限
```

五.开始配置

/*创建账号sql:

```
create user 'username'@'localhost' identified by 'password';
```

```
授权grant [权限] on 数据库名.数据表名 to 'username'@'localhost' identified by 'password';
```

*/

5.1(主机配置)

```
mysql> CREATE USER 'slave1'@'192.168.100.%' IDENTIFIED WITH mysql_native_password BY 'slave1';
Query OK, 0 rows affected (0.04 sec)
```

```
mysql> select user,host from mysql.user;
```

```
mysql> GRANT REPLICATION SLAVE ON *.* TO 'slave1'@'192.168.100.%';
Query OK, 0 rows affected, 1 warning (0.00 sec)
```

对于MySQL的主从复制来说最重要的主要就是binlog日志, 所以我们就需要开启binlog日志, 并设置server-id的值。需要重启服务器之后才生效 二进制日志, 也就是我们常说的binlog。

```
mysql> show variables like "%log_bin%";
```

```
+-----+-----+
| Variable_name | Value |
+-----+-----+
| log_bin       | ON    |
| log_bin_basename | /usr/local/mysql/data/binlog |
| log_bin_index  | /usr/local/mysql/data/binlog.index |
| log_bin_trust_function_creators | OFF |
| log_bin_use_v1_row_events | OFF |
| sql_log_bin    | ON    |
+-----+-----+
6 rows in set (0.03 sec)
```

作用就是

1. 增量备份(不是所有数据备份, 而是最近的写操作)
2. 用于MySQL主从复制

主要就是下配置文件中添加如下配置

```
[mysqld]
log-bin=mysql-bin //开启二进制文件
server-id=1 //这里自己随意定义
```

重启mysql服务: service mysql restart

查看当前的binlog日志以及节点,方便配置从服务器

```
mysql> show master status;
```

```
+-----+-----+-----+-----+-----+
| File           | Position | Binlog_Do_DB | Binlog_Ignore_DB | Executed_Gtid_Set |
+-----+-----+-----+-----+-----+
| mysql-bin.000001 |      155 |              |                  |                  |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

5.2从机配置

修改配置文件,开启二进制

```
log-bin=mysql-bin
```

```
server-id = 2 ////一定不能和主机重复
```

```
relay_log = /usr/local/mysql/data/mysql-relay-bin
```

```
relay_log_index = /usr/local/mysql/data/mysql-relay-bin.index
```

```
log_slave_updates = 1
```

```
read_only = 1
```

指定主节点的ip, 端口, 用户

```
change master to master_host='192.168.81.132',--主服务器的ip
```

```
master_port=3306,--端口
```

```
master_user='slave1',--用户名
```

```
master_password='slave1',--密码
```

```
master_log_file='mysql-bin.000001',--指定主服务器的binlog日志
```

```
master_log_pos=155;--目前日志的节点
```

完整命令:

```
mysql> change master to master_host='192.168.100.143',master_port=3306,master_user='slave1',
master_password='slave1',master_log_file='mysql-bin.000001',master_log_pos=155;
Query OK, 0 rows affected, 2 warnings (0.03 sec)
```

```
change master to master_host='192.168.100.143',master_port=3306,master_user='slave1',master_password='slave1',
master_log_file='mysql-bin.000001',master_log_pos=322;
```

开启从服务器:

```
mysql> start slave;
```

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

查看状态:

```
mysql> show slave status\G;
```

```
***** 1. row *****

Slave_IO_State:
Master_Host: 192.168.100.143
Master_User: slave1
Master_Port: 3306
Connect_Retry: 60
Master_Log_File: mysql-bin.000001
Read_Master_Log_Pos: 155
Relay_Log_File: localhost-relay-bin.000001
Relay_Log_Pos: 4
Relay_Master_Log_File: mysql-bin.000001
Slave_IO_Running: Yes
Slave_SQL_Running: Yes
```

如果出现错误:

```
stop slave;
```

```
reset slave all;
```

从新查看主机的binlog日志(show master status),看是否填错了