TTS 11.0 COOKBOOK

(NSD RDBMS2 DAY02)

版本编号 11.0

2019-06

达内 IT 培训集团

NSD RDBMS2 DAY02

1. 案例 1：实现 MySQL 读写分离

- **问题**
  - 搭建一主一从结构
  - 配置 maxscale 代理服务器
  - 测试配置

- **方案**

使用 4 台虚拟机，如图-1 所示。其中 192.168.4.51 和 192.168.4.52，分别提供读、写服务，均衡流量，通过主从复制保持数据一致性，由 MySQL 代理 192.168.4.57 面向客户端提供服务，收到 SQL 写请求时，交给主服务器处理，收到 SQL 读请求时，交给从服务器处理。在客户机 192.168.4.50 测试配置。

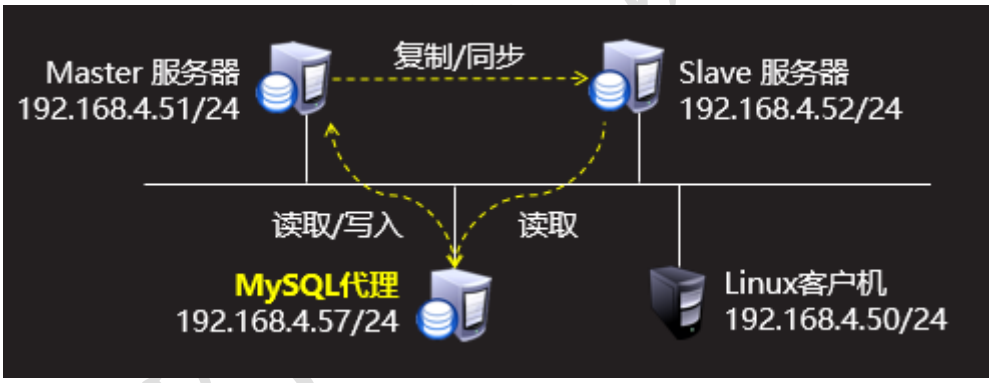

图 - 1

- **步骤**

实现此案例需要按照如下步骤进行。

步骤一：搭建 MySQL 一主一从同步结构

1）配置主服务器 192.168.4.51

```
]# vim /etc/my.cnf
[mysqld]
server_id=51  //指定服务器 ID 号
log-bin=master51      //启用 binlog 日志，并指定文件名前缀
...
[root@master10 ~]# systemctl restart mysqld     //重启 mysqld
```

2）主服务器授权用户，并查看 binlog 日志信息

```
]# mysql -uroot -p123456
mysql> grant all on *.* to 'repluser'@'%' identified by '123456';
Query OK, 0 rows affected, 1 warning (0.00 sec)

mysql> show master status;
+-----------------+----------+--------------+------------------+----------------
---+
| File      | Position | Binlog_Do_DB | Binlog_Ignore_DB | Executed_Gtid_Set |
+-----------------+----------+--------------+------------------+----------------
---+
| master51.000001 |      449 |              |                  |                   |
+-----------------+----------+--------------+------------------+----------------
---+
1 row in set (0.00 sec)
```

3）配置从服务器 192.168.4.52

```
]# vim /etc/my.cnf
[mysqld]
 server_id=52          //指定服务器 ID 号，不要与 Master 的相同
:wq

]# systemctl restart  mysqld
```

4)配置从服务器 192.168.4.52,指定主服务器信息,日志文件、偏移位置(参考 MASTER 上的状态输出)

```
]# mysql -uroot -p123456
mysql> change master to master_host='192.168.4.51',
    -> master_user='repluser',
    -> master_password='123456',
    -> master_log_file='master51.000001',
    -> master_log_pos=449;
Query OK, 0 rows affected, 2 warnings (0.01 sec)

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: 192.168.4.51
                  Master_User: repluser
                  Master_Port: 3306
                Connect_Retry: 60
              Master_Log_File: master51.000001
          Read_Master_Log_Pos: 738
               Relay_Log_File: slave20-relay-bin.000002
                Relay_Log_Pos: 319
        Relay_Master_Log_File: master51.000001
             Slave_IO_Running: Yes      //IO 线程 YES
            Slave_SQL_Running: Yes      //SQL 线程 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: 738
              Relay_Log_Space: 528
              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: 10
                  Master_UUID: 95ada2c2-bb24-11e8-abdb-525400131c0f
             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 more updates
           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
         Replicate_Rewrite_DB:
                 Channel_Name:
           Master_TLS_Version:
1 row in set (0.00 sec)
```

5）测试配置，在主服务器本机创建数据库 aa 库

```
]# mysql -uroot -p123456
 mysql> create database aa;
 Query OK, 1 row affected (0.00 sec)

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| aa                 |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
5 rows in set (0.00 sec)
```

6）从服务器上查看，有 aa 库

```
mysql> show databases;

+--------------------+
| Database           |
+--------------------+
```

```
| information_schema |
| aa                 |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
5 rows in set (0.00 sec)
```

步骤二：配置 maxscale 代理服务器

1）环境准备

关闭防火墙和 SElinux，保证 yum 源可以正常使用，安装提供服务的软件

```
]# rpm -ivh maxscale-2.1.2-1.rhel.7.x86_64.rpm      //安装 maxscale
warning: maxscale-2.1.2-1.rhel.7.x86_64.rpm: Header V4 RSA/SHA1 Signature, key ID
8167ee24: NOKEY
Preparing...                          ################################# [100%]
Updating / installing...
   1:maxscale-2.1.2-1                 ################################# [100%]
```

2）修改主配置文件

```
]# vim /etc/maxscale.cnf
[maxscale]
threads=auto          //运行的线程的数量

[server1]         //定义数据库服务器
type=server
address=192.168.4.51      //主服务器 ip
port=3306
protocol=MySQLBackend

[server2]
type=server
address=192.168.4.52      //从服务器 IP
port=3306
protocol=MySQLBackend


[MySQL Monitor]               //定义监控的数据库服务器
type=monitor
module=mysqlmon
servers=server1, server2      //监控的数据库列表，不能写 ip
user=maxscalemon                  //监控用户
passwd=123qqq...A             //密码
monitor_interval=10000



#[Read-Only Service]      //不定义只读服务
#type=service
#router=readconnroute
#servers=server1
#user=myuser
#passwd=mypwd
#router_options=slave
```

```
[Read-Write Service]            //定义读写分离服务
type=service
router=readwritesplit
servers=server1, server2
user=maxscalerouter             //路由用户
passwd=123qqq…A                 //密码
max_slave_connections=100%

[MaxAdmin Service]          //定义管理服务
type=service
router=cli

#[Read-Only Listener]       //不定义只读服务使用的端口号
#type=listener
#service=Read-Only Service
#protocol=MySQLClient
#port=4008

[Read-Write Listener]           //定义读写服务使用的端口号
type=listener
service=Read-Write Service
protocol=MySQLClient
port=4006

[MaxAdmin Listener]         //管理服务使用的端口号
type=listener
service=MaxAdmin Service
protocol=maxscaled
socket=default
port=4016 //手动添加，不指定时使用的是默认端口在启动服务以后可以知道默认端口是多少
```

3）添加授权用户

根据 maxscale.cnf 文件配置，在主/从服务器上添加对应的授权用户，因为 2 台数据库服务器是主从同步结构，只在主数据库服务器添加用户即可，从服务器会自动同步

```
mysql> grant replication slave,replication client on *.* to  maxscalemon@'%' identified by "123qqq…A"; //授权监控用户

mysql> grant select on mysql.* to maxscalerouter@"%" identified by "123qqq…A"; //授权路由用户
```

4）查看授权用户

分别在主/从服务器上面查看

```
mysql> select user,host from mysql.user where user like “maxscale%”;
+----------------+------+
| user           | host |
+----------------+------+
| maxscalemon    | %    |
| maxscalerouter | %    |
+----------------+------+
2 rows in set (0.00 sec)
```

在代理服务器 57 主机，测试授权用户

```
]# yum -y  install mariadb  //安装提供 mysql 命令的软件包
]# mysql -h 192.168.4.51 -umaxscalemon -p123qqq…A
```

```
]# mysql -h 192.168.4.52 -umaxscalemon -p123qqq…A
]# mysql -h 192.168.4.51 -umaxscalerouter -p123qqq…A
]# mysql -h 192.168.4.52 -umaxscalerouter -p123qqq…A
```

5）启动服务代理服务

```
]# maxscale -f  /etc/maxscale.cnf
]# ps -C  maxscale        //查看进程
PID TTY          TIME CMD
17930 ?       00:00:00 maxscale

]# netstat  -antup | grep :4006  //查看读写分离端口
tcp6       0      0 :::4006      :::*                    LISTEN      17930/maxscale

]# netstat  -antup | grep :4016  //查看管理服务端口
tcp6       0      0 :::4016       :::*                    LISTEN      17930/maxscale
```

步骤三：测试配置

1）查看监控信息（在主机 57 本机自己访问自己）

```
]# maxadmin  -uadmin -pmariadb -P4016

MaxScale> list  servers
Servers.
-------------------+-----------------+-------+-------------+--------------------
Server             | Address         | Port  | Connections | Status
-------------------+-----------------+-------+-------------+--------------------
server1            | 192.168.4.51    |  3306 |           0 | Master, Running
server2            | 192.168.4.52    |  3306 |           0 | Slave, Running
-------------------+-----------------+-------+-------------+--------------------
```

2）在主服务器上添加访问数据连接用户

在主服务器添加即可，从服务器会自动同步数据

```
mysql> create database gamedb;
mysql> create table gamedb.a(id int);
mysql> grant select,insert on  gamedb.* to  yaya66@"%" identified by "123qqq...A";
```

客户端连接代理服务 57 访问数据

```
]# mysql -h192.168.4.57 -P4006 -uyaya66 -p123qqq...A
mysql> select  * from gamedb.a;
mysql> insert into gamedb.a values(99);
mysql> select  * from gamedb.a;
mysql> select  * from gamedb.a;
Empty set (0.00 sec)
mysql>
mysql> insert into gamedb.a values(99);
Query OK, 1 row affected (0.06 sec)
mysql>
mysql> select  * from gamedb.a;
+------+
| id   |
+------+
|   99 |
+------+
1 row in set (0.00 sec)
```

3）验证 57 主机的数据读写分离功能

在从服务器添加新纪录

```
Mysql> insert into gamedb.values(52);
Mysql> select  * from mysql> select  * from gamedb.a;
+------+
| id   |
+------+
|   99 |
|   52 |
+------+
```

在主服务器查看记录

```
Mysql> select  * from mysql> select  * from gamedb.a;
+------+
| id   |
+------+
|   99 |
+------+
```

客户端连接代理服务器 57 访问数据

```
]# mysql -h192.168.4.57 -P4006 -uyaya66 -p123qqq...A
Mysql> select  * from mysql> select  * from gamedb.a;
+------+
| id   |
+------+
|   99 |
|   52 |
+------+
```

2. 案例 2：配置 MySQL 多实例

- **问题**
  - **在主机 192.168.4.57 上：**
  - **配置第 1 个 MySQL 实例**
  - 实例名称 mysqld1、端口 3307
  - 数据库目录/dir2、pid 文件 mysqld1.pid
  - 错误日志 mysqld1.err、socket 文件 mysqld1.socket
  - **配置第 2 个 MySQL 实例**
  - 实例名称 mysqld2、端口 3308
  - 数据库目录/dir1、pid 文件 mysqld2.pid
  - 错误日志 mysqld2.err、socket 文件 mysqld2.socket

步骤一：配置多实例（192.168.4.57 上操作）

什么是多实例：

在一台物理主机上运行多个数据库服务，可以节约运维成本，提高硬件利用率

1）解压软件、修改目录名、设置 PATH 路径

```
]# yum -y  install libaio
]# useradd  mysql
]# tar -zxvf mysql-5.7.20-linux-glibc2.12-x86_64.tar.gz
]# mv mysql-5.7.20-linux-glibc2.12-x86_64 /usr/local/mysql
]# PATH=/usr/local/mysql/bin:$PATH
]# vim /etc/bashrc
   export PATH=/usr/local/mysql/bin:$PATH
:wq
```

2）编辑主配置文件/etc/my.cnf

每个实例要有独立的数据库目录、监听端口号、实例名称和独立的 sock 文件

```
]# vim /etc/my.cnf
[mysqld_multi]        //启用多实例
mysqld = /usr/local/mysql/bin/mysqld_safe       //指定进程文件路径
mysqladmin = /usr/local/mysql/bin/mysqladmin    //指定管理命令路径
user = root       //指定进程用户

[mysqld1]     //实例进程名称
port=3307     //端口号
datadir=/dir1     //数据库目录 ， 要手动创建
socket=/dir1/mysqld1.sock      //指定 sock 文件的路径和名称
pid-file=/dir1/mysqld1.pid     //进程 pid 号文件位置
log-error=/dir1/mysqld1.err        //错误日志位置

[mysqld2]
port=3308
datadir=/dir2
socket=/dir2/mysqld2.sock
pid-file=/dir2/mysqld2.pid
log-error=/dir2/mysqld2.err
:wq
```

3）创建数据库目录

```
]# mkdir  /dir2
]# mkdir  /dir1
```

4）启动多实例

首次启动服务会做数据初始化 并初始和提示数据库管理员本机登录密码

```
[root@host57 ~]# mysqld_multi  start 1 //启动实例 1

Installing new database in /dir1

2019-06-13T10:46:29.307866Z 0 [Warning] TIMESTAMP with implicit DEFAULT value is deprecated.  Please  use  --explicit_defaults_for_timestamp  server  option  (see documentation for more details).
2019-06-13T10:46:30.997233Z 0 [Warning] InnoDB: New log files created, LSN=45790
2019-06-13T10:46:31.436904Z 0 [Warning] InnoDB: Creating foreign key constraint system tables.
2019-06-13T10:46:31.582129Z 0 [Warning] No existing UUID has been found, so we assume that this is the first time that this server has been started. Generating a new UUID: 816bf015-8dc8-11e9-b492-525400cffedc.
2019-06-13T10:46:31.605276Z 0 [Warning] Gtid table is not ready to be used. Table 'mysql.gtid_executed' cannot be opened.
```

```
    2019-06-13T10:46:31.606321Z 1 [Note] A temporary password is generated for
root@localhost: ly#LryiFE5fT  管理员本机登录密码

    ]# ls /dir1 //查看数据库目录文件列表
    auto.cnf   ib_buffer_pool   ibdata1   ib_logfile0   ib_logfile1   ibtmp1   mysql
mysql3307.log        mysql3307.pid        mysql3307.sock        mysql3307.sock.lock
performance_schema  sys

    ]# mysqld_multi  start 2  //启动实例 2


    Installing new database in /dir1

    2019-06-13T10:56:55.580796Z 0 [Warning] TIMESTAMP with implicit DEFAULT value is
deprecated.  Please  use  --explicit_defaults_for_timestamp  server  option  (see
documentation for more details).
    2019-06-13T10:56:57.199217Z 0 [Warning] InnoDB: New log files created, LSN=45790
    2019-06-13T10:56:57.571839Z 0 [Warning] InnoDB: Creating foreign key constraint
system tables.
    2019-06-13T10:56:57.708168Z 0 [Warning] No existing UUID has been found, so we assume
that this is the first time that this server has been started. Generating a new UUID:
f69f30fa-8dc9-11e9-8a17-525400cffedc.
    2019-06-13T10:56:57.724096Z 0 [Warning] Gtid table is not ready to be used. Table
'mysql.gtid_executed' cannot be opened.
    2019-06-13T10:56:57.724677Z 1 [Note] A temporary password is generated for
root@localhost: qedTjrZs*8ma  管理员本机登录密码

    ]# ls /dir1 //查看数据库目录文件列表
    auto.cnf   ib_buffer_pool   ibdata1   ib_logfile0   ib_logfile1   ibtmp1   mysql
mysql3308.log        mysql3308.pid        mysql3308.sock        mysql3308.sock.lock
performance_schema  sys
```

5）查看端口

```
    ]# netstat -utnlp  | grep :3307
    tcp6       0      0 :::3307                 :::*                    LISTEN
1151/mysqld

    ]# netstat -utnlp  | grep :3308
    tcp6       0      0 :::3308                 :::*                    LISTEN
1339/mysqld

    ]# netstat -utnlp | grep mysqld
    tcp6       0      0 :::3307                 :::*                    LISTEN
1151/mysqld
    tcp6       0      0 :::3308                 :::*                    LISTEN
1339/mysqld

    # ps -C mysqld
      PID TTY          TIME CMD
     1151 pts/1    00:00:00 mysqld
     1339 pts/1    00:00:00 mysqld
    [root@host57 ~]#
```

6）访问多实例

使用初始化密码登录实例 1

```
[root@host57 ~]# mysql -uroot -p'ly#LryiFE5fT' -S /dir1/mysqld1.sock
 mysql> alter user root@"localhost" identified by "123456"; //修改密码
mysql> exit
Bye
[root@host57 ~]# mysql -uroot -p123456  -S /dir1/mysqld1.sock //新密码登录
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 4
Server version: 5.7.20 MySQL Community Server (GPL)

Copyright (c) 2000, 2017, 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.

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
4 rows in set (0.00 sec)

mysql> create database db1; //创建新库 db1
Query OK, 1 row affected (0.00 sec)

mysql> show databases; //查看已有的库
+--------------------+
| Database           |
+--------------------+
| information_schema |
| db1                |  //db1 库
| mysql              |
| performance_schema |
| sys                |
+--------------------+
5 rows in set (0.00 sec)

mysql> exit  //断开连接
Bye
[root@host56 ~]# ls /dir1  //查看数据库目录文件列表 有 db1 库的文件夹
auto.cnf        ibdata1      ibtmp1       mysqld1.pid          performance_schema
db1             ib_logfile0  mysql        mysqld1.socket       sys
ib_buffer_pool  ib_logfile1  mysqld1.err  mysqld1.socket.lock
[root@host56 ~]#
```

使用初始化密码登录实例 2

```
[root@host57 ~]# mysql -uroot -p'qedTjrZs*8ma' -S /dir2/mysqld2.sock
 mysql> alter user root@"localhost" identified by "654321"; //修改密码
mysql> exit
Bye
```

```
[root@host57 ~]# mysql -uroot -p654321  -S /dir2/mysqld2.sock //新密码登录
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 4
Server version: 5.7.20 MySQL Community Server (GPL)

Copyright (c) 2000, 2017, 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.

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
4 rows in set (0.00 sec)

mysql>
mysql> create database db2;
Query OK, 1 row affected (0.00 sec)

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| db2                |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
5 rows in set (0.00 sec)

mysql> exit
Bye
[root@host56 ~]# ls /dir2
auto.cnf        ib_logfile0  mysqld2.err         performance_schema
db2             ib_logfile1  mysqld2.pid         sys
ib_buffer_pool  ibtmp1       mysqld2.socket
ibdata1         mysql        mysqld2.socket.lock
[root@host56 ~]#
```

7）停止多实例服务

mysqld_multi --user=root --password=密码 stop 实例编号

```
]# netstat -utnlp  | grep  mysqld
tcp6       0      0 :::3307                 :::*                    LISTEN
1250/mysql
tcp6       0      0 :::3308                 :::*                    LISTEN
1451/mysql

]# mysqld_multi  --user=root --password=123456  stop 2

 [root@host56 ~]# netstat -utnlp  | grep  mysqld
tcp6       0      0 :::3307                 :::*                    LISTEN
```

```
1250/mysql

    ]# mysql -uroot   -p123456   -S    /dir2/mysqld2.sock //拒绝连接
    mysql: [Warning] Using a password on the command line interface can be insecure.
    ERROR  2002  (HY000):  Can't  connect  to  local  MySQL  server  through  socket
'/dir2/mysqld2.sock' (2)
```