[Top](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN201904/RDBMS2/DAY02/CASE/01/index.html" \l "page_top_case)

# NSD RDBMS2 DAY02

1. [案例1：实现MySQL读写分离](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN201904/RDBMS2/DAY02/CASE/01/index.html" \l "case1)
2. [案例2：配置MySQL多实例](http://tts.tmooc.cn/ttsPage/LINUX/NSDTN201904/RDBMS2/DAY02/CASE/01/index.html" \l "case2)

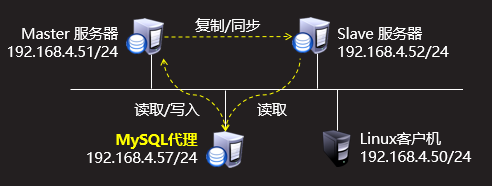
## 1 案例1：实现MySQL读写分离

### 1.1 问题

* 搭建一主一从结构
* 配置maxscale代理服务器
* 测试配置

### 1.2 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

### 1.3 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

1. ]# vim /etc/my.cnf
2. [mysqld]
3. server\_id=51    //指定服务器ID号
4. log-bin=master51        //启用binlog日志，并指定文件名前缀
5. ...
6. [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

1. ]# mysql -uroot -p123456
2. mysql> grant all on \*.\* to 'repluser'@'%' identified by '123456';
3. Query OK, 0 rows affected, 1 warning (0.00 sec)
4. mysql> show master status;
5. +-----------------+----------+--------------+------------------+-------------------+
6. | File | Position | Binlog\_Do\_DB | Binlog\_Ignore\_DB | Executed\_Gtid\_Set |
7. +-----------------+----------+--------------+------------------+-------------------+
8. | master51.000001 | 449 | | | |
9. +-----------------+----------+--------------+------------------+-------------------+
10. 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）

1. ]# mysql -uroot -p123456
2. mysql> change master to master\_host='192.168.4.51',
3. -> master\_user='repluser',
4. -> master\_password='123456',
5. -> master\_log\_file='master51.000001',
6. -> master\_log\_pos=449;
7. Query OK, 0 rows affected, 2 warnings (0.01 sec)
8. mysql> start slave;
9. Query OK, 0 rows affected (0.01 sec)
10. mysql> show slave status\G;
11. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 1. row \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
12. Slave\_IO\_State: Waiting for master to send event
13. Master\_Host: 192.168.4.51
14. Master\_User: repluser
15. Master\_Port: 3306
16. Connect\_Retry: 60
17. Master\_Log\_File: master51.000001
18. Read\_Master\_Log\_Pos: 738
19. Relay\_Log\_File: slave20-relay-bin.000002
20. Relay\_Log\_Pos: 319
21. Relay\_Master\_Log\_File: master51.000001
22. Slave\_IO\_Running: Yes        //IO线程YES
23. Slave\_SQL\_Running: Yes        //SQL线程YES
24. Replicate\_Do\_DB:
25. Replicate\_Ignore\_DB:
26. Replicate\_Do\_Table:
27. Replicate\_Ignore\_Table:
28. Replicate\_Wild\_Do\_Table:
29. Replicate\_Wild\_Ignore\_Table:
30. Last\_Errno: 0
31. Last\_Error:
32. Skip\_Counter: 0
33. Exec\_Master\_Log\_Pos: 738
34. Relay\_Log\_Space: 528
35. Until\_Condition: None
36. Until\_Log\_File:
37. Until\_Log\_Pos: 0
38. Master\_SSL\_Allowed: No
39. Master\_SSL\_CA\_File:
40. Master\_SSL\_CA\_Path:
41. Master\_SSL\_Cert:
42. Master\_SSL\_Cipher:
43. Master\_SSL\_Key:
44. Seconds\_Behind\_Master: 0
45. Master\_SSL\_Verify\_Server\_Cert: No
46. Last\_IO\_Errno: 0
47. Last\_IO\_Error:
48. Last\_SQL\_Errno: 0
49. Last\_SQL\_Error:
50. Replicate\_Ignore\_Server\_Ids:
51. Master\_Server\_Id: 10
52. Master\_UUID: 95ada2c2-bb24-11e8-abdb-525400131c0f
53. Master\_Info\_File: /var/lib/mysql/master.info
54. SQL\_Delay: 0
55. SQL\_Remaining\_Delay: NULL
56. Slave\_SQL\_Running\_State: Slave has read all relay log; waiting for more updates
57. Master\_Retry\_Count: 86400
58. Master\_Bind:
59. Last\_IO\_Error\_Timestamp:
60. Last\_SQL\_Error\_Timestamp:
61. Master\_SSL\_Crl:
62. Master\_SSL\_Crlpath:
63. Retrieved\_Gtid\_Set:
64. Executed\_Gtid\_Set:
65. Auto\_Position: 0
66. Replicate\_Rewrite\_DB:
67. Channel\_Name:
68. Master\_TLS\_Version:
69. 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

1. ]# mysql –uroot –p123456
2. mysql> create database aa;
3. Query OK, 1 row affected (0.00 sec)
4. mysql> show databases;
5. +--------------------+
6. | Database |
7. +--------------------+
8. | information\_schema |
9. | aa |
10. | mysql |
11. | performance\_schema |
12. | sys |
13. +--------------------+
14. 5 rows in set (0.00 sec)

6）从服务器上查看，有aa库

1. mysql> show databases;
2. +--------------------+
3. | Database |
4. +--------------------+
5. | information\_schema |
6. | aa |
7. | mysql |
8. | performance\_schema |
9. | sys |
10. +--------------------+
11. 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

1. ]# rpm -ivh maxscale-2.1.2-1.rhel.7.x86\_64.rpm         //安装maxscale
2. warning: maxscale-2.1.2-1.rhel.7.x86\_64.rpm: Header V4 RSA/SHA1 Signature, key ID 8167ee24: NOKEY
3. Preparing... ################################# [100%]
4. Updating / installing...
5. 1:maxscale-2.1.2-1 ################################# [100%]

2）修改主配置文件

1. ]# vim /etc/maxscale.cnf
2. [maxscale]
3. threads=auto            //运行的线程的数量
4. [server1]            //定义数据库服务器
5. type=server
6. address=192.168.4.51        //主服务器ip
7. port=3306
8. protocol=MySQLBackend
9. [server2]
10. type=server
11. address=192.168.4.52        //从服务器IP
12. port=3306
13. protocol=MySQLBackend
14. [MySQL Monitor]                //定义监控的数据库服务器
15. type=monitor
16. module=mysqlmon
17. servers=server1, server2        //监控的数据库列表，不能写ip
18. user=maxscalemon                    //监控用户
19. passwd=123qqq...A                //密码
20. monitor\_interval=10000
21. #[Read-Only Service]        //不定义只读服务
22. #type=service
23. #router=readconnroute
24. #servers=server1
25. #user=myuser
26. #passwd=mypwd
27. #router\_options=slave
28. [Read-Write Service]            //定义读写分离服务
29. type=service
30. router=readwritesplit
31. servers=server1, server2
32. user=maxscalerouter            //路由用户
33. passwd=123qqq…A                //密码
34. max\_slave\_connections=100%
35. [MaxAdmin Service]        //定义管理服务
36. type=service
37. router=cli
38. #[Read-Only Listener]        //不定义只读服务使用的端口号
39. #type=listener
40. #service=Read-Only Service
41. #protocol=MySQLClient
42. #port=4008
43. [Read-Write Listener]            //定义读写服务使用的端口号
44. type=listener
45. service=Read-Write Service
46. protocol=MySQLClient
47. port=4006
48. [MaxAdmin Listener]        //管理服务使用的端口号
49. type=listener
50. service=MaxAdmin Service
51. protocol=maxscaled
52. socket=default
53. 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

1. mysql> grant replication slave,replication client on \*.\* to maxscalemon@'%' identified by "123qqq…A"; //授权监控用户
2. 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

1. mysql> select user,host from mysql.user where user like “maxscale%”;
2. +----------------+------+
3. | user | host |
4. +----------------+------+
5. | maxscalemon | % |
6. | maxscalerouter | % |
7. +----------------+------+
8. 2 rows in set (0.00 sec)

在代理服务器57主机，测试授权用户

1. ]# yum -y install mariadb //安装提供mysql命令的软件包
2. ]# mysql -h 192.168.4.51 -umaxscalemon -p123qqq…A
3. ]# mysql -h 192.168.4.52 -umaxscalemon -p123qqq…A
4. ]# mysql -h 192.168.4.51 -umaxscalerouter -p123qqq…A
5. ]# mysql -h 192.168.4.52 -umaxscalerouter -p123qqq…A

5）启动服务代理服务

1. ]# maxscale -f /etc/maxscale.cnf
2. ]# ps -C maxscale        //查看进程
3. PID TTY TIME CMD
4. 17930 ? 00:00:00 maxscale
5. ]# netstat -antup | grep :4006 //查看读写分离端口
6. tcp6 0 0 :::4006 :::\* LISTEN 17930/maxscale
7. ]# netstat -antup | grep :4016 //查看管理服务端口
8. 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）

1. ]# maxadmin -uadmin -pmariadb -P4016
2. MaxScale> list servers
3. Servers.
4. -------------------+-----------------+-------+-------------+--------------------
5. Server | Address | Port | Connections | Status
6. -------------------+-----------------+-------+-------------+--------------------
7. server1 | 192.168.4.51 | 3306 | 0 | Master, Running
8. server2 | 192.168.4.52 | 3306 | 0 | Slave, Running
9. -------------------+-----------------+-------+-------------+--------------------

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

1. mysql> create database gamedb;
2. mysql> create table gamedb.a(id int);
3. 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

### 2.1 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

1. ]# yum –y install libaio
2. ]# useradd mysql
3. ]# tar -zxvf mysql-5.7.20-linux-glibc2.12-x86\_64.tar.gz
4. ]# mv mysql-5.7.20-linux-glibc2.12-x86\_64 /usr/local/mysql
5. ]# PATH=/usr/local/mysql/bin:$PATH
6. ]# vim /etc/bashrc
7. export PATH=/usr/local/mysql/bin:$PATH
8. :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

1. ]# vim /etc/my.cnf
2. [mysqld\_multi]        //启用多实例
3. mysqld = /usr/local/mysql/bin/mysqld\_safe        //指定进程文件路径
4. mysqladmin = /usr/local/mysql/bin/mysqladmin    //指定管理命令路径
5. user = root        //指定进程用户
6. [mysqld1]        //实例进程名称
7. port=3307        //端口号
8. datadir=/dir1        //数据库目录 ，要手动创建
9. socket=/dir1/mysqld1.sock        //指定sock文件的路径和名称
10. pid-file=/dir1/mysqld1.pid        //进程pid号文件位置
11. log-error=/dir1/mysqld1.err        //错误日志位置
12. [mysqld2]
13. port=3308
14. datadir=/dir2
15. socket=/dir2/mysqld2.sock
16. pid-file=/dir2/mysqld2.pid
17. log-error=/dir2/mysqld2.err
18. :wq

3）创建数据库目录

1. ]# mkdir /dir2
2. ]# 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

1. [root@host57 ~]# mysqld\_multi start 1 //启动实例1
2. Installing new database in /dir1
3. 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).
4. 2019-06-13T10:46:30.997233Z 0 [Warning] InnoDB: New log files created, LSN=45790
5. 2019-06-13T10:46:31.436904Z 0 [Warning] InnoDB: Creating foreign key constraint system tables.
6. 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.
7. 2019-06-13T10:46:31.605276Z 0 [Warning] Gtid table is not ready to be used. Table 'mysql.gtid\_executed' cannot be opened.
8. 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
9. ]# ls /dir1 //查看数据库目录文件列表
10. auto.cnf ib\_buffer\_pool ibdata1 ib\_logfile0 ib\_logfile1 ibtmp1 mysql mysql3307.log mysql3307.pid mysql3307.sock mysql3307.sock.lock performance\_schema sys
11. ]# mysqld\_multi start 2 //启动实例2
12. Installing new database in /dir1
13. 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).
14. 2019-06-13T10:56:57.199217Z 0 [Warning] InnoDB: New log files created, LSN=45790
15. 2019-06-13T10:56:57.571839Z 0 [Warning] InnoDB: Creating foreign key constraint system tables.
16. 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.
17. 2019-06-13T10:56:57.724096Z 0 [Warning] Gtid table is not ready to be used. Table 'mysql.gtid\_executed' cannot be opened.
18. 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
19. ]# ls /dir1 //查看数据库目录文件列表
20. 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

1. ]# netstat -utnlp | grep :3307
2. tcp6 0 0 :::3307 :::\* LISTEN 1151/mysqld
3. ]# netstat -utnlp | grep :3308
4. tcp6 0 0 :::3308 :::\* LISTEN 1339/mysqld
5. ]# netstat -utnlp | grep mysqld
6. tcp6 0 0 :::3307 :::\* LISTEN 1151/mysqld
7. tcp6 0 0 :::3308 :::\* LISTEN 1339/mysqld
8. # ps -C mysqld
9. PID TTY TIME CMD
10. 1151 pts/1 00:00:00 mysqld
11. 1339 pts/1 00:00:00 mysqld
12. [root@host57 ~]#

6）访问多实例

使用初始化密码登录实例1

1. [root@host57 ~]# mysql -uroot -p'ly#LryiFE5fT' -S /dir1/mysqld1.sock
2. mysql> alter user root@"localhost" identified by "123456";    //修改密码
3. mysql> exit
4. Bye
5. [root@host57 ~]# mysql -uroot -p123456 -S /dir1/mysqld1.sock //新密码登录
6. mysql: [Warning] Using a password on the command line interface can be insecure.
7. Welcome to the MySQL monitor. Commands end with ; or \g.
8. Your MySQL connection id is 4
9. Server version: 5.7.20 MySQL Community Server (GPL)
10. Copyright (c) 2000, 2017, Oracle and/or its affiliates. All rights reserved.
11. Oracle is a registered trademark of Oracle Corporation and/or its
12. affiliates. Other names may be trademarks of their respective
13. owners.
14. Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
15. mysql> show databases;
16. +--------------------+
17. | Database |
18. +--------------------+
19. | information\_schema |
20. | mysql |
21. | performance\_schema |
22. | sys |
23. +--------------------+
24. 4 rows in set (0.00 sec)
25. mysql> create database db1; //创建新库db1
26. Query OK, 1 row affected (0.00 sec)
27. mysql> show databases; //查看已有的库
28. +--------------------+
29. | Database |
30. +--------------------+
31. | information\_schema |
32. | db1 | //db1库
33. | mysql |
34. | performance\_schema |
35. | sys |
36. +--------------------+
37. 5 rows in set (0.00 sec)
38. mysql> exit //断开连接
39. Bye
40. [root@host56 ~]# ls /dir1 //查看数据库目录文件列表 有db1库的文件夹
41. auto.cnf ibdata1 ibtmp1 mysqld1.pid performance\_schema
42. db1 ib\_logfile0 mysql mysqld1.socket sys
43. ib\_buffer\_pool ib\_logfile1 mysqld1.err mysqld1.socket.lock
44. [root@host56 ~]#

使用初始化密码登录实例2

1. [root@host57 ~]# mysql -uroot -p'qedTjrZs\*8ma' -S /dir2/mysqld2.sock
2. mysql> alter user root@"localhost" identified by "654321";    //修改密码
3. mysql> exit
4. Bye
5. [root@host57 ~]# mysql -uroot –p654321 -S /dir2/mysqld2.sock //新密码登录
6. mysql: [Warning] Using a password on the command line interface can be insecure.
7. Welcome to the MySQL monitor. Commands end with ; or \g.
8. Your MySQL connection id is 4
9. Server version: 5.7.20 MySQL Community Server (GPL)
10. Copyright (c) 2000, 2017, Oracle and/or its affiliates. All rights reserved.
11. Oracle is a registered trademark of Oracle Corporation and/or its
12. affiliates. Other names may be trademarks of their respective
13. owners.
14. Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
15. mysql> show databases;
16. +--------------------+
17. | Database |
18. +--------------------+
19. | information\_schema |
20. | mysql |
21. | performance\_schema |
22. | sys |
23. +--------------------+
24. 4 rows in set (0.00 sec)
25. mysql>
26. mysql> create database db2;
27. Query OK, 1 row affected (0.00 sec)
28. mysql> show databases;
29. +--------------------+
30. | Database |
31. +--------------------+
32. | information\_schema |
33. | db2 |
34. | mysql |
35. | performance\_schema |
36. | sys |
37. +--------------------+
38. 5 rows in set (0.00 sec)
39. mysql> exit
40. Bye
41. [root@host56 ~]# ls /dir2
42. auto.cnf ib\_logfile0 mysqld2.err performance\_schema
43. db2 ib\_logfile1 mysqld2.pid sys
44. ib\_buffer\_pool ibtmp1 mysqld2.socket
45. ibdata1 mysql mysqld2.socket.lock
46. [root@host56 ~]#

7）停止多实例服务

mysqld\_multi --user=root --password=密码 stop 实例编号

1. ]# netstat -utnlp | grep mysqld
2. tcp6 0 0 :::3307 :::\* LISTEN 1250/mysql
3. tcp6 0 0 :::3308 :::\* LISTEN 1451/mysql
4. ]# mysqld\_multi --user=root --password=123456 stop 2
5. [root@host56 ~]# netstat -utnlp | grep mysqld
6. tcp6 0 0 :::3307 :::\* LISTEN 1250/mysql
7. ]# mysql -uroot -p123456 -S /dir2/mysqld2.sock //拒绝连接
8. mysql: [Warning] Using a password on the command line interface can be insecure.
9. ERROR 2002 (HY000): Can't connect to local MySQL server through socket '/dir2/mysqld2.sock' (2)