

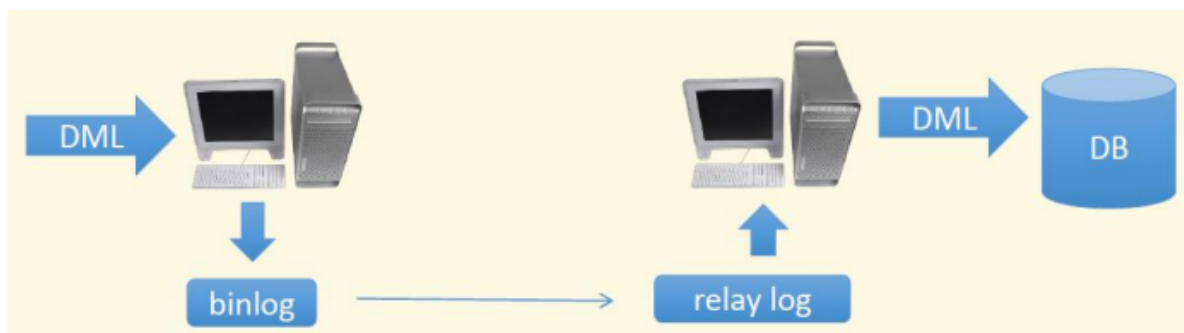
## 1.1 MySQL Replication

### 1.1.1 什么是 Replication

- 1 Replication 可以实现将数据从一台数据库服务器 < master > 复制到一台或多台数据库服务器 < slave >
- 2 默认情况下属于异步复制，无需维持长连接
- 3 通过配置，可以复制所有库或几个库，甚至库中的一些表，是 MySQL 内建的

### 1.1.2 Replication 原理

- 1 master 将数据库的改变写入二进制日志，slave 同步这些二进制日志，并根据这些二进制日志进行数据库操作



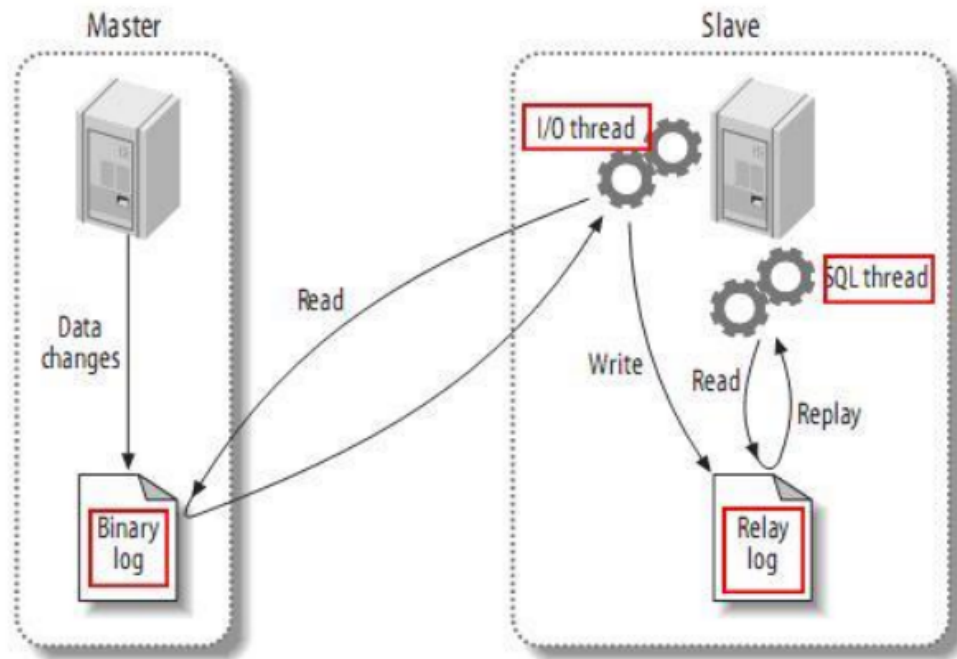
- 1 DML # SQL 操作语句: update, insert, delete
- 2 binlog # 二进制日志
- 3 Relay log # 中继日志

### 1.1.3 Replication 的作用

- 1 1、Fail over: 故障切换
- 2 2、Backup server: 备份服务，无法对 SQL 语句执行产生的故障恢复，有限的备份
- 3 3、High Performance: 高性能，可以多台 slave，实现读写分离

### 1.1.4 Replication 工作原理

- 1 1、master 将改变记录到二进制日志 < binary log > 中，这些记录叫做二进制日志事件，binary log events
- 2 2、slave 将 master 的 binary log events 拷贝到它的中继日志 < relay log >
- 3 3、slave 重做中继日志中的事件，修改 slave 上的数据



#### 1 # MySQL 主从复制过程:

- 2
- 3 1、master 记录二进制日志，在每个事物更新数据完成之前，master 在二进制日志记录这些改变。MySQL 将事务写入二进制日志，即使日志中的语句都是交叉执行的。在事件中写入二进制日志完成后，master 通知存储引擎提交事务
- 4
- 5 2、slave 将 master 的 binary log 拷贝到它自己的中继日志。首先 slave 开始一个工作线程 I/O 线程。I/O 线程在 master 上打开一个普通的连接，然后开始 binlog dump process。binlog dump process 从 master 的二进制日志中读取事件，如果已经执行完 master 产生的所有文件，它会睡眠并等待 master 产生新的事件。I/O 线程将这些事件写入中继日志
- 6
- 7 3、SQL slave thread < SQL 从线程 > 是处理该过程的最后一步。SQL 线程从中继日志读取事件，并重新执行其中的事件而更新 slave 的数据，使其与 master 中的数据一致

### 1.1.5 Replication 方案

- 1 1、One master and Muti slave < 一主多从 >
- 2 一般用来做读写分离，master 写，其它 slave 读，这种架构最大问题 I/O 压力集中在 master 上 < 多台同步影响 I/O >
- 3
- 4 2、M-S-S
- 5 使用一台 slave 作为中继，分担 master 压力，slave 中继需要开启 bin-log，并配置 log-slave-updates
- 6 slave 中继可使用 black-hole 存储引擎，不会把数据存储到磁盘，只记录二进制日志
- 7 3、M-M < 互为主从 >
- 8 很多人误以为这样可以做到 MySQL 负载均衡，实际没什么好处，每个服务器需要做同样的同步更新，破坏了事物的隔离性和数据的一致性
- 9 4、M-M-M
- 10 监控三台机器互为 master
- 11 天生的缺陷：复制延迟，slave 上同步要慢于 master，如果大并发的情况那延迟更严重
- 12 mysql 在 5.6 已经自身可以实现 fail over 故障切换
- 13 5、One slave Muti master < 一从多主 >
- 14 好处：节省成本，将多个 master 数据自动化整合
- 15 缺陷：对库和表数据的修改较多

## 1.2 部署 MySQL 主从同步

### 1.2.1 环境

主机名	IP	系统/MySQL版本	角色
CentOS-7.6	10.0.0.166	CentOS-7.6/5.5.24	Master
CentOS-7.6	10.0.0.167	CentOS-7.6/5.5.24	Slave

```
1  # 两台服务器都执行下列操作
2
3  # 安装 MySQL
4  tar xf mysql-5.7.tar.gz
5  yum -y install ./mysql*.rpm
6
7  # 关闭密码轻度审计插件
8  echo "validate-password=OFF" >> /etc/my.cnf
9  systemctl restart mysqld
10
11 # 修改 root 密码
12 grep "password" /var/log/mysqld.log
13 mysql -uroot -p'M-jPs;qti1:_'
14 set password for root@localhost=password('123');
```

### 1.2.2 配置主数据库服务器

```
1  1、创建需要同步的数据库
2
3  create database HA;
4  use HA;
5  create table t1(id int,name varchar(20));
```

```
1  2、配置 my.cnf
2
3  vim /etc/my.cnf
4  # 在末尾追加以下内容
5  log-bin=mysql-bin-master    # 启用二进制日志
6  server-id=1                 # 本机数据库 ID 标识
7  binlog-do-db=HA             # 可以被从服务器复制的库，二进制需要同步的数据库名
8  binlog-ignore-db=mysql      # 不可以被从服务器复制的库
9
10 systemctl restart mysqld
```

```
1  3、授权
2
3  mysql -uroot -p123
4  grant replication slave on *.* to slave@10.0.0.167 identified by '123';
```

```

1 4、查看状态信息
2
3 show master status;
4 show binlog events\G
5 exit
6 ls /var/lib/mysql

```

```

mysql> show master status;
+-----+-----+-----+-----+-----+
| File                               | Position | Binlog_Do_DB | Binlog_Ignore_DB | Executed_Gtid_Set |
+-----+-----+-----+-----+-----+
| mysql-bin-master.000001           | 447      | HA           | mysql             |                    |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> show binlog events\G
***** 1. row *****
  Log_name: mysql-bin-master.000001
    Pos: 4
  Event_type: Format_desc
  Server_id: 1
End_log_pos: 123
  Info: Server ver: 5.7.25-log, Binlog ver: 4
***** 2. row *****
  Log_name: mysql-bin-master.000001
    Pos: 123
  Event_type: Previous_gtid
  Server_id: 1
End_log_pos: 154
  Info:
***** 3. row *****
  Log_name: mysql-bin-master.000001
    Pos: 154
  Event_type: Anonymous_Gtid
  Server_id: 1
End_log_pos: 219
  Info: SET @@SESSION.GTID_NEXT= 'ANONYMOUS'
***** 4. row *****
  Log_name: mysql-bin-master.000001
    Pos: 219
  Event_type: Query
  Server_id: 1
End_log_pos: 447
  Info: GRANT REPLICATION SLAVE ON *.* TO 'slave'@'10.0.0.167' IDENTIFIED WITH 'mysql_native_password' AS '*23AE809DDACAF96AF0FD78ED04B6A265E05AA257'
4 rows in set (0.00 sec)

```

```

[root@centos-7 ~]# ls /var/lib/mysql
auto.cnf          HA                ibtmp1           mysql.sock.lock  server-key.pem
ca-key.pem        ib_buffer_pool   mysql            performance_schema sys
ca.pem            ibdata1          mysql-bin-master.000001 private_key.pem
client-cert.pem   ib_logfile0      mysql-bin-master.index public_key.pem
client-key.pem    ib_logfile1      mysql.sock       server-cert.pem
[root@centos-7 ~]#

```

```

1 # 复制前要保证同步的数据库一致
2
3 mysqldump -uroot -p123 -B HA > HA.sql
4 scp HA.sql root@10.0.0.167:/root/

```

### 1.2.3 配置从数据库服务器

```

1 1、两台数据库服务器 MySQL 版本要一致
2 show variables like '%version%';

```

```

1 2、测试连接到主服务器是否成功
2 mysql -uslave -p123 -h 10.0.0.166
3 # 只有复制权限是看不到其它库的

```

```

1 3、导入数据库，使两台数据库数据一致
2 mysql -uroot -p123 < HA.sql

```

```

1 4、修改从服务器配置文件
2 vim /etc/my.cnf
3     server-id=2
4 # 从服务器 ID 号，不要和主 ID 相同，如果设置多个从服务器，每个从服务器必须有一个唯一的
   server-id 值，必须与主服务器的以及其它从服务器的不相同。可以认为 server-id 值类似于 IP
   地址：这些 ID 值能唯一识别复制服务器群集中的每个服务器实例。
5 systemctl restart mysqld
6 mysql -uroot -p123
7 change master to
   master_host='10.0.0.166',master_user='slave',master_password='123';
8 start slave;
9 show slave status\G
10     # Slave_IO_Running: 一个负责与 Master 主机的 IO 通信
11     # Slave_SQL_Running: 负责自己的 slave mysql 进程

```

```

mysql> show slave status\G
***** 1. row *****
      Slave_IO_State: Waiting for master to send event
      Master_Host: 10.0.0.166
      Master_User: slave
      Master_Port: 3306
      Connect_Retry: 60
      Master_Log_File: mysql-bin-master.000001
      Read_Master_Log_Pos: 447
      Relay_Log_File: CentOS-7-relay-bin.000002
      Relay_Log_Pos: 674
      Relay_Master_Log_File: mysql-bin-master.000001
      Slave_IO_Running: Yes
      Slave_SQL_Running: Yes
      Replicate_Do_DB:
      Replicate_Ignore_DB:

```

```

1 # 在主服务器查看状态
2 mysql -uroot -p123
3 show processlist\G

```

```

mysql> show processlist\G
***** 1. row *****
      Id: 6
      User: slave
      Host: 10.0.0.167:57150
      db: NULL
      Command: Binlog Dump
      Time: 168
      State: Master has sent all binlog to slave; waiting for more updates
      Info: NULL
***** 2. row *****
      Id: 7
      User: root
      Host: localhost
      db: NULL
      Command: Query
      Time: 0
      State: starting
      Info: show processlist
2 rows in set (0.00 sec)

```

### 1.2.4 插入数据测试同步

```
1 # master 插入数据
2 use HA;
3 insert into t1 values (1,'man');
4
5 # slave 查看
6 select * from HA.t1;
```

### 1.2.5 排错

```
1 # 见学神笔记
```

## 1.3 MySQL 主主双向主从复制

```
1 # 见学神笔记
```

## 1.4 部署 M-S-S 模型

系统	角色	mysql 版本	IP
CentOS-7.6	master	mysql 5.7.24	10.0.0.166
CentOS-7.6	slave 中继	mysql 5.7.24	10.0.0.167
CentOS-7.6	slave	mysql 5.7.24	10.0.0.168

### 1.4.1 部署 master

```
1 # 安装 mysql
2
3 # 上传 mysql-5.7.tar.gz 到本机
4 scp mysql-5.7.tar.gz root@10.0.0.167:/root/
5 scp mysql-5.7.tar.gz root@10.0.0.168:/root/
6 tar xf mysql-5.7.tar.gz
7 yum -y install ./mysql*.rpm
```

```
1 # 关闭密码强度审计插件
2
3 systemctl start mysqld
4 echo "validate-password=OFF" >> /etc/my.cnf
5 systemctl restart mysqld
```

```
1 # 修改用户密码
2
3 grep "password" /var/log/mysqld.log
4 mysql -uroot -p''
5 set password for root@localhost=password('mysql');
```

```

1 # 授权用户
2
3 grant replication slave on *.* to repl@'10.0.0.%' identified by '123';
4 create database HA;
5 use HA;
6 create table t1(id int,name varchar(20));
7 exit
8 mysqldump -uroot -pmysql -B HA > HA.sql

```

```

1 # 修改配置文件
2
3 vim /etc/my.cnf
4 log-bin=mysql-bin-master
5 server-id=1
6 binlog-do-db=HA
7 log-bin=mysql-bin-master
8 binlog-ignore-db=mysql
9 sync-binlog=1      # 执行 N 次写入 binlog 后，与硬盘同步
10 binlog-format=row  # binlog 保存方式，记录哪条数据被修改了
11
12 systemctl restart mysqld

```

```

1 # 数据库备份发个 slave 中继和 slave
2
3 scp HA.sql root@10.0.0.167:/root/
4 scp HA.sql root@10.0.0.168:/root/

```

### 1.4.2 部署 slave 中继

```

1 # 安装 mysql
2
3 tar xf mysql-5.7.tar.gz
4 yum -y install mysql*.rpm

```

```

1 # 关闭密码强度审计插件
2
3 systemctl start mysqld
4 echo "validate-password=OFF" >> /etc/my.cnf
5 systemctl restart mysqld

```

```

1 # 修改用户密码
2
3 grep "password" /var/log/mysqld.log
4 mysql -uroot -p''
5 set password for root@localhost=password('mysql');
6 exit

```

```

1 # 导入数据库
2
3 mysql -uroot -pmysql < HA.sql

```

```
1 # 修改配置文件
2
3 vim /etc/my.cnf
4 server-id=2
5 log-bin=mysql-bin-slave1
6 log-slave-updates=1
7 binlog-format=row
8
9 systemctl restart mysqld
```

```
1 # 授权
2
3 mysql -uroot -pmysql
4 change master to
  master_host='10.0.0.166',master_user='rep1',master_password='123';
5 start slave;
6 show slave status\G
7 grant replication slave on *.* to 'rep1'@'10.0.0.%' identified by '123';
```

### 1.4.3 部署 slave

```
1 # 安装 mysql
2
3 tar xf mysql-5.7.tar.gz
4 yum -y install mysql*.rpm
```

```
1 # 关闭密码强度审计插件
2
3 systemctl start mysqld
4 echo "validate-password=OFF" >> /etc/my.cnf
5 systemctl restart mysqld
```

```
1 # 修改用户密码
2
3 grep "password" /var/log/mysqld.log
4 mysql -uroot -p''
5 set password for root@localhost=password('mysql');
6 exit
```

```
1 # 导入数据库
2
3 mysql -uroot -pmysql < HA.sql
```

```
1 # 修改配置文件
2
3 vim /etc/my.cnf
4 server-id=3
5
6 systemctl restart mysqld
```



```
1 mysql -uroot -pmysql
2 change master to
  master_host='10.0.0.167',master_user='repl',master_password='123';
3 start slave;
4 show slave status \G
```

#### 1.4.4 测试

```
1 # 在 master 上插入数据测试
2
3 mysql -uroot -pmysql
4 use HA;
5 insert into t1 values(1,'AA');
```

```
1 # 在 slave 中继续查看
2
3 select * from HA.t1;
```

```
mysql> select * from HA.t1;
+-----+-----+
| id    | name  |
+-----+-----+
|      1 | AA    |
+-----+-----+
1 row in set (0.00 sec)
```

```
1 # 在 slave 上查看
2
3 select * from HA.t1;
```

```
mysql> select * from HA.t1;
+-----+-----+
| id    | name  |
+-----+-----+
|      1 | AA    |
+-----+-----+
1 row in set (0.01 sec)
```