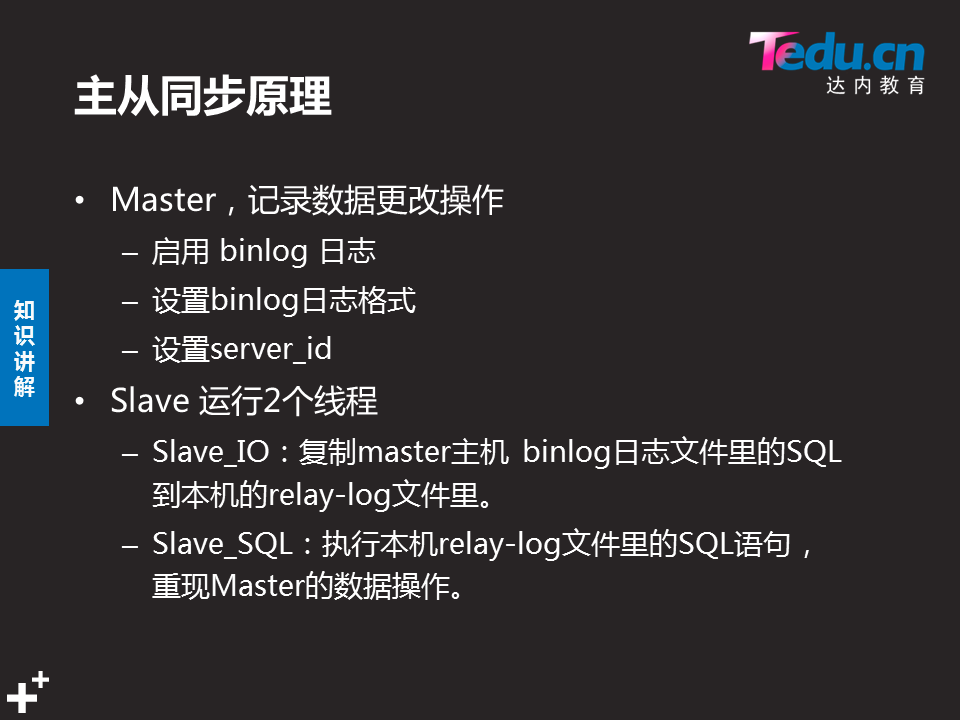
**Mysql主从同步**

配置mysql主从同步：

什么是mysql主从同步？

从数据库向主数据库同步数据的方式就是主从同步

**一、主从同步工作原理：**



怎么配置？

A.主库（master）配置如下：（192.168.4.51上操作）

1. 启用binlog日志文件

[root@host51 ~]# vim /etc/my.cnf

[mysqld]

server\_id=51

log-bin=master51

binlog\_format="mixed"

[root@host51 ~]# systemctl restart mysqld

[root@host51 ~]# systemctl enable mysqld

[root@host51 ~]# ls /var/lib/mysql/master51.\*

1. 用户授权：给从库添加一个连接时使用的用户名

mysql> grant replication slave on \*.\* to repluser@"192.168.4.52" identified by "123qqq...A";

1. 查看正在使用的binlog日志信息

mysql> show master status; #可以查看日志文件信息及偏移量位置

B.从库（slave）配置如下：（192.168.4.52上操作）

mysql> show slave status; #查看当前数据库是否为从库 不是则为空

1. 指定server\_id

[root@host52 ~]# vim /etc/my.cnf

[mysqld]

server\_id=52

[root@host52 ~]# systemctl restart mysqld

1. 测试授权用户

[root@host52 ~]# mysql -h192.168.4.51 -urepluser -p123qqq...A

mysql> show grants; #查看当前用户的权限

1. 管理员本机登录指定主库信息

mysql> change master to

>master\_host="192.168.4.51", #主库

>master\_user="repluser", #从库用户

>master\_password="123qqq...A", #从库用户密码

>master\_log\_file="master51.000001", #指定日志文件信息

>master\_log\_pos=452; #指定偏移量

mysql> show slave status\G;

.......

Slave\_IO\_Running: No

Slave\_SQL\_Running: No

..........

mysql> start slave; #启动配置

1. 查看配置

mysql> show slave status\G;

................

Slave\_IO\_Running: Yes

Slave\_SQL\_Running: Yes

................

mysql> show processlist; #显示当前主机运行的程序列表

C.客户端测试主从同步配置

在master主库服务器上管理员本机登录，添加客户端存储数据时，使用的连接用户（192.168.4.51）

mysql> create database gamedb;

mysql> create table gamedb.t1(id int);

mysql> grant select,insert on gamedb.\* to webadmin@"%" identified by "123qqq...A";

客户端连接主库存储数据（192.168.4.50）

[root@host50 ~]# mysql -h 192.168.4.51 -uwebadmin -p123qqq...A

mysql> show grants;

mysql> insert into gamedb.t1 values(666);

mysql> insert into gamedb.t1 values(777);

mysql> insert into gamedb.t1 values(888);

mysql> insert into gamedb.t1 values(999);

mysql> insert into gamedb.t1 values(111);

在从库管理员root本机登录，也能看到客户端连接主库存储的数据（192.168.4.52）

[root@host52 ~]# mysql -uroot -p123456

mysql> select \* from gamedb.t1;

+------+

| id |

+------+

| 666 |

| 777 |

| 888 |

| 999 |

| 111 |

+------+

5 rows in set (0.00 sec)

客户端连接主库存储数据，客户端也能从从库看到主库一样的数据

C.如何让从库暂时不同步主库的数据

mysql> stop slave;

D.如何把从库还原成独立的数据库服务

[root@host52 mysql]# rm -rf master.info felay-log.info

[root@host52 mysql]# rm -rf host52-relay-bin.index

**二、Mysql主从同步模式：**

主从同步结构模式：

1. **一主一从：**

**B.一主多从：**

主库（192.168.4.51）：

mysql> drop user [repluser@"192.168.4.52";](mailto:repluser@\"192.168.4.52\";) #删除原有授权用户

mysql> drop user [lisi@"192.168.4.53";](mailto:lisi@\"192.168.4.53\";)

mysql> grant replication slave on \*.\* to repluser@"%" identified by "123qqq...A"; #授权从库用户

[root@host51 ~]# mysqldump -uroot -p123456 gamedb > /root/gamedb.sql #备份原仓库文件

[root@host51 ~]# scp /root/gamedb.sql [root@192.168.4.53:/root/](mailto:root@192.168.4.53:/root/) #把备份文件发给从库

mysql> grant select,insert on bbsdb.\* to testuser@"%" identified by "123qqq...A";

#为客户端创建验证用户

从库（192.168.4.53）：

mysql> create database gamedb;

[root@host53 ~]# mysql -uroot -p123456 gamedb < /root/gamedb.sql

测试授权帐号：

[root@host52 ~]# mysql -h192.168.4.51 -urepluser -p123qqq...A

[root@host53 ~]# mysql -h192.168.4.51 -urepluser -p123qqq...A

指定主机id号：

[root@host53 ~]# vim /etc/my.cnf

[mysqld]

server\_id=53

[root@host53 ~]# systemctl restart mysqld

在53上指定主库信息：

mysql> change master to

-> master\_host="192.168.4.51",

-> master\_user="repluser",

-> master\_password="123qqq...A",

-> master\_log\_file="master51.000001",

-> master\_log\_pos=5095;

mysql> start slave; #启动线程服务

同上配置192.168.4.52

**C.主从从结构**

**1.还原host53主机，成为独立的存在**

[root@host53 ~]# systemctl stop mysqld

[root@host53 mysql]# rm relay-log.info master.info

[root@host53 mysql]# rm -rf host53-relay-bin.index

[root@host53 mysql]# systemctl restart mysqld

[root@host53 mysql]# mysql -uroot -p123456

mysql> drop database gamedb;

同上还原52主机，成为独立的存在

**2.以192.168.4.51为主库**

mysql> system vim /etc/my.cnf

[mysqld]

server\_id=51

log-bin=db51

binlog\_format="mixed"

mysql> grant replication slave on \*.\* to yaya@"%" identified by "123qqq...A";

#设置从服务器的权限用户

**3.把192.168.4.52为主库和从库**

配置为192.168.4.53的主库的配置：

[root@host52 mysql]# vim /etc/my.cnf

[mysqld]

server\_id=52

log-bin=master52

binlog\_format="mixed"

log\_slave\_updates #开启极连复制 如果不加此项则当客户端在主库上添加数据时53号主机上无法同步

mysql> grant replication slave on \*.\* to reluser@"%" identified by "123qqq...A";

mysql> show master status;

配置为192.168.4.51的从库的配置：

[root@host52 mysql]# mysql -h192.168.4.51 -uyaya -p123qqq...A #测试帐号

[root@host52 mysql]# mysql -uroot -p123456

mysql> change master to

-> master\_host="192.168.4.51",

-> master\_user="yaya",

-> master\_password="123qqq...A",

-> master\_log\_file="db51.000001",

-> master\_log\_pos=154;

mysql> start slave;

mysql> show slave status\G;

Slave\_IO\_Running: Yes

Slave\_SQL\_Running: Yes

**4.以192.168.4.53配置为192.168.4.52的从库**

[root@host53 mysql]# systemctl stop mysqld

[root@host53 mysql]# vim /etc/my.cnf

[mysqld]

server\_id=53

[root@host53 mysql]# systemctl start mysqld

[root@host53 mysql]# mysql -h192.168.4.52 -ureluser -p123qqq...A #测试授权帐号

[root@host53 mysql]# mysql -uroot -p123456 #用管理员帐号登录

mysql> show slave status\G; #查看本库是否为从库

mysql> change master to #配置主库结构

-> master\_host="192.168.4.52",

-> master\_user="reluser",

-> master\_password="123qqq...A",

-> master\_log\_file="master52.000001",

-> master\_log\_pos=440;

mysql> start slave; #启动配置

mysql> show slave status; #查看线程是否开启

..............

Slave\_IO\_Running: Yes

Slave\_SQL\_Running: Yes

...........

**5.验证：**

在主库上建库建表（192.168.4.51）

mysql> create database bbsdb;

mysql> create table bbsdb.a(id int);

客户端上添加数据（192.168.4.50）：

[root@host50 ~]# mysql -h192.168.4.51 -utestuser -p123qqq...A

mysql> show grants;

mysql> insert into a values(123456);

mysql> insert into a values(654321);

mysql> insert into a values(789456123);

mysql> insert into a values(111111111);

从库上验证（192.168.4.52）（192.168.4.53）

mysql> select \* from bbsdb.a;

Mysql主从同步常用配置选项

1.log\_slave\_updates #开启极连复制 当一个数据库机即做主库又做从库时必须加上此选项

2.Binlog\_do\_db=库名列表 #只允许同步的库

3.Binlog\_ignore\_db=库名列表 #只不允许同步的库

#2、3两种情况不能同时使用，以上三种情况斗都是在主库上配置

1. replicate\_do\_db=库名列表 #只同步的库 用在从库上
2. replicate\_ignore\_db=库名列表 #只不同步的库 用在从库上

主从同步复制模式

启用半同步复制模式：

（192.168.4.54上操作）

Master：

mysql> show variables like 'have\_dynamic\_loading'; #查看是否允许加载模块

查看加载的模块：

mysql> select plugin\_name,plugin\_status

->from information\_schema.plugins

->where plugin\_name like '%semi%';

mysql> install plugin rpl\_semi\_sync\_master #为主库安装模块

-> soname 'semisync\_master.so'; #模块名称

mysql> set global rpl\_semi\_sync\_master\_enabled=1; #启动加载模块

mysql> show variables like 'rpl\_semi\_sync\_%\_enabled'; #查看模块开启情况

Slave：

mysql> install plugin rpl\_semi\_sync\_slave #为从库安装模块

->soname 'semisync\_slave.so'; #模块名称

查看加载的模块：

mysql> select plugin\_name,plugin\_status

->from information\_schema.plugins

->where plugin\_name like '%semi%';

mysql> set global rpl\_semi\_sync\_slave\_enabled=1; #启用从库半同步模块

mysql> show variables like 'rpl\_semi\_sync\_%\_enabled'; #查看模块加载情况

（192.168.4.55上操作）

修改配置文件在配置文件中添加模块

[root@host55 ~]# vim /etc/my.cnf

..........

[mysqld]

plugin-load=”rplsemi\_sync\_master=semisync\_master.so rpl\_semi\_sync\_master\_enabled=1”

plugin-load=”rplsemi\_sync\_master=semisync\_slave.so rpl\_semi\_sync\_slave\_enabled=1”

.....

[root@host55 ~]# systemctl restart mysqld

#$#$#$#$#$#$#$#$#$#$#$#$#$#$#$#$#$#$#$#$#$#$#$#$#$#$#$#$#$#$#$#$#$#

Mysql数据读写分离

把客户端查询数据库的请求和写入数据的请求，分别给不同数据库服务器处理。

目的：减轻主库的并发访问压力，提高从库的硬件利用率。

人肉分离：需要程序员协同完成，规定程序员在存储数据时连接主数据库服务器；查询数据时连接slave数据库服务器。

部署实现数据读写分离服务实现：mysql中间件，实现软件有（mycat、mysql-proxy、maxscale）

配置数据读写分离服务器：

1. 部署mysql主从同步结构（一主一从）