**Morning：**

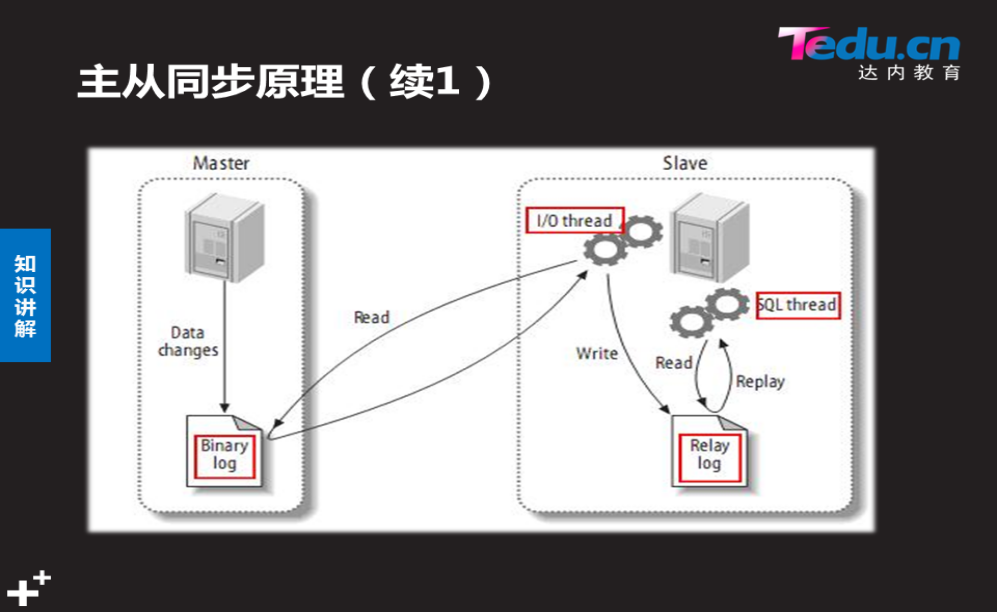
1、主从同步概述

**-What’s主从同步**

多台主机一起提供数据库服务：

可以快速响应客户端连接；拓展存储空间。

**-工作原理**



响应客户端连接请求的服务器作为主库，其他作为从库。

Master：启动binlog日志

Slave\_IO线程：复制主机binlog日志的SQL语句到本机relay-log(中继)。

Slave\_SQL线程：执行本机relay-log文件里的SQL语句，重现主库数据。

Tips：

在配置主从结构之前，必须保证从库与主库数据一致，否则线程会停止工作。

1. 搭建主从同步

**步骤1：确保数据相同**

mysqldump --> scp --> 数据恢复到从库

**步骤2：配置主服务器**

启动binlog日志 --> 重启服务 --> 授权用户 --> 查看master状态

[root@mysql51 ~]# vim /etc/my.cnf

log\_bin=master51

server\_id=51

binlog\_format="mixed"

[root@mysql51 ~]# systemctl restart mysqld

mysql> grant replication slave on \*.\* to

-> repluser@"%" identified by "123456";

#指定授权用户，有同步权限

mysql> show master status;

#记住当前日志文件名、偏移位置

**步骤3：配置从服务器**

指定server\_id --> 指定主库信息 -->启动slave进程 --> 查看slave状态

[root@mysql52 ~]# vim /etc/my.cnf

server\_id=52

[root@mysql52 ~]# systemctl restart mysqld

mysql> change master to

-> master\_host='192.168.4.51',master\_user='repluser',

-> master\_password='123456',

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

-> master\_log\_pos=441;

#分别指定主库ip、主库授权用户名、密码、主库日志文件、偏移量

#任意一项指定错误，都会导致IO/SQL线程无法启动

mysql> start slave;

mysql> show slave status \G;

#查看从库状态信息，可验证是否为从库

Master\_Host: 192.168.4.51

... ...

Relay\_Log\_File: mysql52-relay-bin.000002 #中继日志文件

Relay\_Log\_Pos: 319

Slave\_IO\_Running: Yes

Slave\_SQL\_Running: Yes

... ...

Last\_IO\_Error:

Last\_SQL\_Error:

Tips：

1）指定主库信息时，偏移量也可以指定该binlog日志下的非当前值。

2）Server\_id冲突、主库信息指定错误，都会导致Slave\_SQL无法启动；

3）当执行中继日志的SQL命令时报错，Slave\_SQL线程立即停止；

4）排除错误后，重新启动start slave。

**步骤4：测试配置**

1）192.168.4.51主数据库服务器建表、授权客户端用户yaya100。

mysql> create database db1;

mysql> create table db1.a (id int);

mysql> grant select,insert on db1.\*

-> to yaya100@"%" identified by "123456";

2）192.168.4.50客户端插入数据

[root@client50 ~]# mysql -uyaya100 -p123456 -h'192.168.4.51'

mysql> insert into db1.a values(111);

3）192.168.4.51/192.168.4.52实现数据同步

mysql> select \* from db1.a;

+-------+

| id |

+-------+

| 111 |

+-------+

**-如何移除主从关系？**

从服务器相关文件：

|  |  |
| --- | --- |
| 文件名(/var/lib/mysql/) | 说明 |
| master.info | 主库信息（ip、授权用户、密码、端口、binlog文件、偏移量） |
| relay-log.info | 中继日志信息：  记录当前中继日志名、偏移量  主库的binlog日志名，偏移量 |
| 主机名-relay-bin.xxxxxx | 中继日志，默认保留最新2个  与binlog格式一致 |
| 主机名-relay-bin.index | 记录中继日志名的索引文件 |

从服务器恢复为独立数据库：

[root@mysql52 mysql]# rm -rf master.info

[root@mysql52 mysql]# rm -rf mysql52-relay-bin.\*

[root@mysql52 mysql]# rm -rf relay-log.info

#删除从服务器相关文件

[root@mysql52 mysql]# systemctl restart mysqld

[root@mysql52 mysql]# mysql -uroot -p123456 -e "show slave status\G";

**Afternoon：**

1、主从同步常用配置参数

**-主库配置参数**

|  |  |
| --- | --- |
| 选项 | 用途 |
| binlog\_do\_db=db1,db2... | 设置Master对哪些库记日志 |
| binlog\_ignore\_db=db1,db2... | 设置Master对哪些库不记日志 |

[root@mysql51 ~]# vim /etc/my.cnf

binlog\_do\_db=...

或binlog\_ignore\_db=...

[root@mysql51 ~]# systemctl restart mysqld

mysql> show master status;

Tips：

1）在master服务器中配置，对所有slave服务器都生效。

2）一般情况binlog\_do\_db与binlog\_ignore\_db不同时配置。

3）生产环境下，一般不限制。

**-从库配置参数**

|  |  |
| --- | --- |
| 选项 | 用途 |
| log\_slave\_updates | 允许链式复制，相当于把中继日志的SQL命令写入本机binlog日志。  (配置在既是主库、又是从库的服务器) |
| relay\_log=xxx | 指定中继日志文件名 |
| replicate\_do\_db=db1,db2... | 仅复制指定库(省略时复制所有库) |
| replicate\_ignore\_db=db1,db2... | 不复制指定库 |

[root@mysql52 ~]# vim /etc/my.cnf

[root@mysql52 ~]# systemctl restart mysqld

mysql> show slave status\G;

Tips：

1）在slave服务器中配置，只对slave本机有效。

2）一般情况replicate\_do\_db与replicate\_ignore\_db不同时配置。

2、主从同步结构模式

**-结构模式分类**

|  |  |  |
| --- | --- | --- |
| 主从结构 | 拓扑 | 备注 |
| 一主一从 | 主 --> 从 |  |
| 一主多从 | 从 <-- 主 --> 从 |  |
| 主从从(链式复制) | 主 --> 从 --> 从 |  |
| 互为主从 | 主 <--> 主 | 不可同时被访问（写锁冲突）  常用于高可用集群 |

**-配置主从从结构**

1. 把主机51配置成主库

2）把主机52配置为master & 51slave

[root@mysql52 ~]# vim /etc/my.cnf #启动binlog日志

log\_slave\_updates

log\_bin=master52

server\_id=52

binlog\_format="mixed"

[root@mysql52 ~]# systemctl restart mysqld

mysql> grant replication slave on \*.\* #授权同步用户

-> to repluser@"%"

-> identified by '123456';

mysql> show master status;

mysql> show slave status\G;

3）把主机53配置为52slave

4）客户端50作验证

Tips：上述步骤1）、3）、4）参考Morning ->主从同步案例

3、主从同步复制模式

服务器如何从主库复制数据到从库？

**-复制模式分类**

|  |  |
| --- | --- |
| 复制模式 | 概念&功能 |
| 异步复制(默认) | 主库执行完一次事务后，立即将结果返回给客户端，并不关心从库是否已接收并处理。  客户响应速度快，无法保证从库的数据同步。 |
| 全同步复制(一般不开) | 当主库执行完一次事务，且所有从库都执行了该事务后才返回给客户端。  客户端响应速度相当慢。 |
| 半同步复制(生产环境) | 主库在执行完一次事务后，等待至少一个从库接收并写到relay log中才返回给客户端。 |

**-半同步复制的配置**

注意：主库、从库都需要配置。

1）临时配置(立刻生效)

mysql> show variables like ‘have\_dynamic\_loading’

#查看是否允许动态加载模块

mysql> INSTALL PLUGIN rpl\_semi\_sync\_master #变量名称

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

mysql> INSTALL PLUGIN rpl\_semi\_sync\_slave

-> SONAME 'semisync\_slave.so';

#加载主库、从库的半同步功能模块

mysql> select PLUGIN\_NAME,PLUGIN\_STATUS from

-> INFORMATION\_SCHEMA.PLUGINS where

-> PLUGIN\_NAME like "%semi%";

#查看模块是否已安装

mysql> SET GLOBAL rpl\_semi\_sync\_master\_enabled=1;

mysql> SET GLOBAL rpl\_semi\_sync\_slave\_enabled=1;

mysql> show variables like 'rpl\_semi\_sync\_%\_enabled';

#查看半同步复制功能是否启动

2）永久配置

[root@mysql51 ~]# vim /etc/my.cnf

plugin-load=rpl\_semi\_sync\_master=semisync\_master.so;rpl\_semi\_sync\_slave=semisync\_slave.so

rpl\_semi\_sync\_master\_enabled=1

rpl\_semi\_sync\_slave\_enabled=1

#同时安装主、从的半同步复制模式