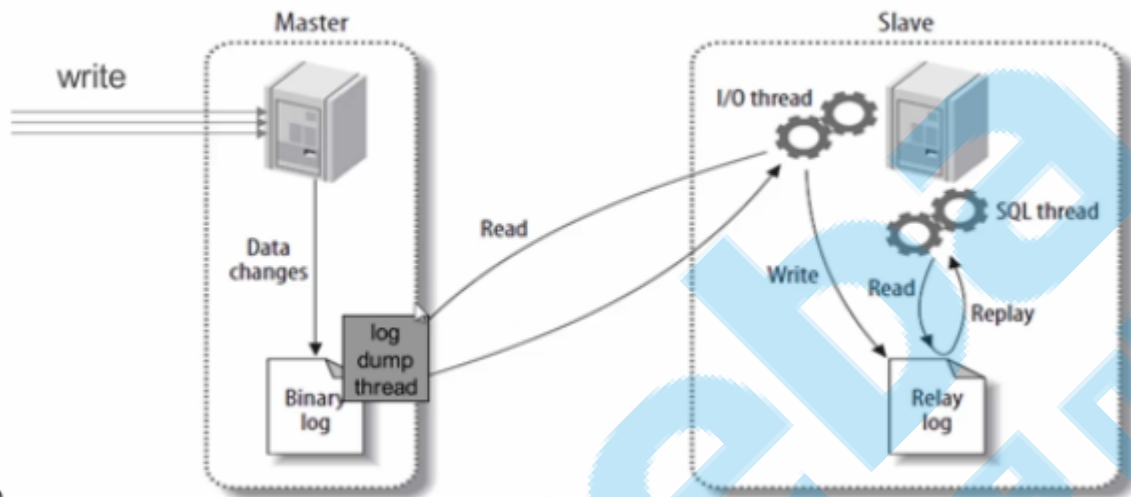


主从复制原理

主对外工作，从对内备份。



binlog介绍和relay日志

查看bin log和relay log日志：

主从复制实践

关闭主从机器的防火墙

主服务器配置

第一步：修改my.conf文件

在[mysqld]段下添加：

```
#启用二进制日志
log-bin=mysql-bin
#服务器唯一ID，一般取IP最后一段
server-id=133
```

第二步：重启mysql服务

第三步：主机给从机授备份权限

注意：先要登录到MySQL命令客户端

第四步：刷新权限

第五步：查询master的状态

从服务器配置

第一步：修改my.conf文件

```
[mysqld]
server-id=135
#slave要复制的库(可选)
replicate_do_db=kkb2
```

第二步：删除UUID文件

第三步：重启并登录到MySQL进行配置从服务器

第四步：启动从服务器复制功能

第五步：检查从服务器复制功能状态

```
mysql> show slave status \G;

.....(省略部分)
Slave_IO_Running: Yes //此状态必须YES
Slave_SQL_Running: Yes //此状态必须YES
.....(省略部分)
```

集群搭建之读写分离

为什么要有读写分离集群？主从集群的问题：只有主对外工作，从不对外工作。主既要负责写操作，也要负责读操作。对于主从集群来说，只是保证了数据的安全备份。主：负责读、写 从：负责读

MySQL-Proxy安装

- 下载

```
wget https://downloads.mysql.com/archives/get/file/mysql-proxy-0.8.5-linux-el6-x86-64bit.tar.gz
```

- 解压缩

```
tar -xf mysql-proxy-0.8.5-linux-el6-x86-64bit.tar.gz -C /kkb
```

MySQL-Proxy配置

- 创建mysql-proxy.cnf文件

```
[mysql-proxy]
user=root
admin-username=root
admin-password=root
proxy-address=192.168.10.137:4040
proxy-backend-addresses=192.168.10.135:3306
proxy-read-only-backend-addresses=192.168.10.136:3306
proxy-lua-script=/root/mysql-proxy/share/doc/mysql-proxy/rw-splitting.lua
log-file=/root/mysql-proxy/logs/mysql-proxy.log
log-level=debug
keepalive=true
daemon=true
```

- 修改mysql-proxy.cnf文件的权限

```
chmod 660 mysql-proxy.cnf    #可读写
```

- 修改rw-splitting.lua脚本

```
37 -- connection pool
38 if not proxy.global.config.rwsplit then
39     proxy.global.config.rwsplit = {
40         min_idle_connections = 1,
41         max_idle_connections = 2,
42     }
43     is_debug = false
44 end
45 end
```

MySQL-Proxy启动域测试

- 启动命令

```
./mysql-proxy --defaults-file=mysql-proxy.cnf配置文件的地址
```

注意事项:

如果没有配置profile文件的环境变量,则需要去拥有mysql-proxy命令的目录通过./mysql-proxy进行启动。

- 在其他客户端,通过mysql命令去连接MySQL Proxy机器

```
mysql -uroot -proot -h192.168.10.137 -P4040
```

注: 关闭防火墙

分库分表介绍

使用背景

- 当【表的数量】达到了几百上千张表时，众多的业务模块都访问这个数据库，压力会比较大，考虑对其进行分库。
- 当【表的数据】达到了几千万级别，在做很多操作都比较吃力,所以，考虑对其进行分库或者分表

数据切分（sharding）方案

数据的切分（Sharding）根据其切分规则的类型，可以分为两种切分模式：

- **垂直切分**：按照业务模块进行切分，将不同模块的表切分到不同的数据库中。
- **水平切分**：将一张大表按照一定的切分规则，按照行切分成不同的表或者切分到不同的库中。

水平切分规则

常用的切分规则有以下几种：

- **按照ID取模**：对ID进行取模，余数决定该行数据切分到哪个表或者库中
- **按照日期**：按照年月日，将数据切分到不同的表或者库中
- **按照范围**：可以对某一列按照范围进行切分，不同的范围切分到不同的表或者数据库中。
-

分库分表需要解决的问题

分布式事务问题

分布式主键ID问题

跨库join问题

跨库count、order by、group by问题

分库分表实现技术

- 阿里的TDDL、Cobar
- 基于阿里Cobar开发的Mycat
- 当当网的sharding-jdbc