

DBA进阶

NSD RDBMS2

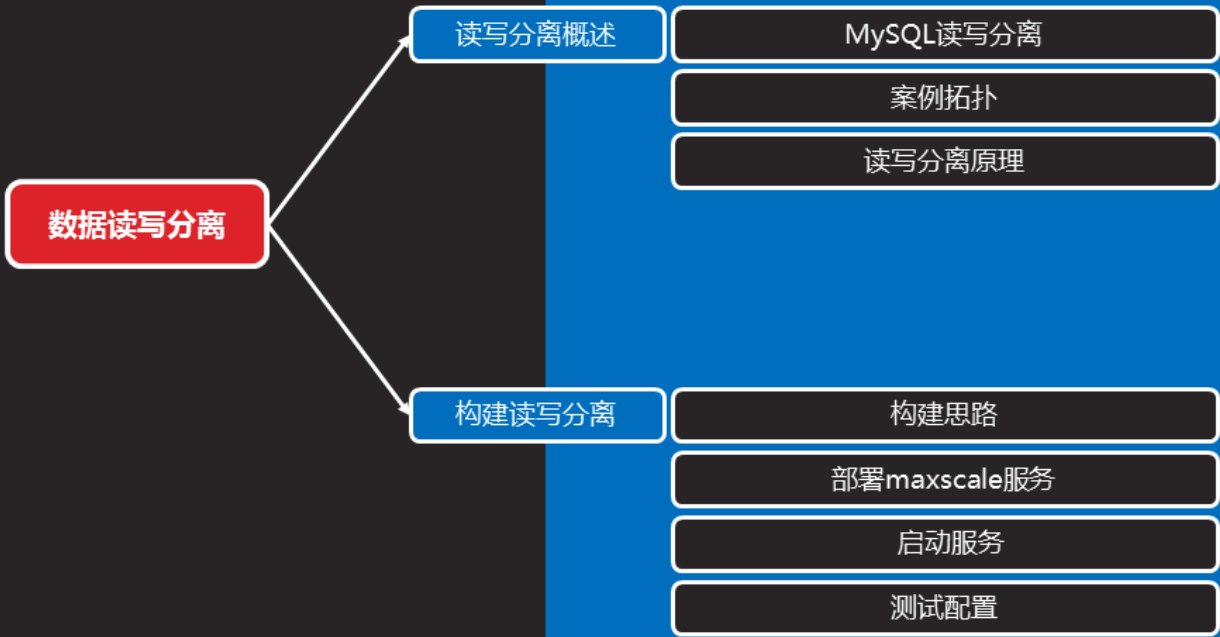
DAY02

内容

上午	09:00 ~ 09:30	作业讲解和回顾
	09:30 ~ 10:20	数据读写分离
	10:30 ~ 11:20	
	11:30 ~ 12:00	
下午	14:00 ~ 14:50	多实例服务
	15:00 ~ 15:50	
	16:10 ~ 17:00	
	17:10 ~ 18:00	总结和答疑



数据读写分离



读写分离概述

MySQL读写分离

- 主从复制的应用局限性？
- 如何分离MySQL读、写流量？
- 在客户端区分，还是在服务器端区分？

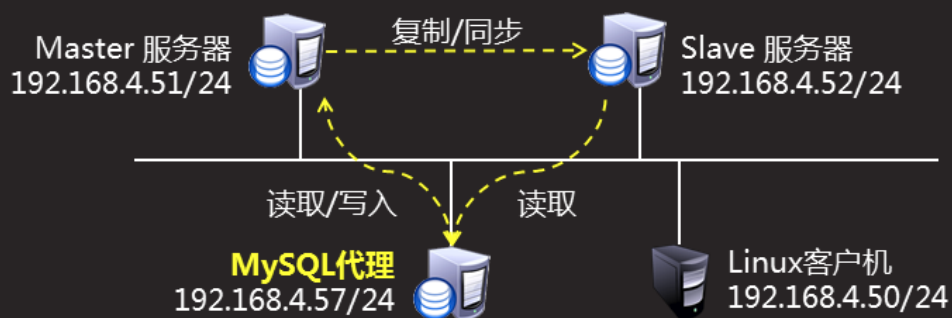
知识讲解



案例拓扑

- 添加一个MySQL代理
 - 为客户端提供统一的数据库访问接口

知识讲解



读写分离的原理

- 由MySQL代理面向客户端提供服务
 - 收到SQL写请求时，交给master服务器处理
 - 收到SQL读请求时，交给slave服务器处理

知识讲解



构建读写分离

构建思路

知识讲解

1. 部署MySQL一主一从结构
 - 主服务器 192.168.4.51
 - 从服务器 192.168.4.52
2. 部署MySQL代理服务器
 - 装包、修改配置文件、启动服务
3. 测试配置
 - 客户端连接代理服务访问数据



部署maxscale服务

知识讲解

- MaxScale代理软件
 - 由 MySQL 的兄弟公司 MariaDB 开发
 - 下载地址
<https://downloads.mariadb.com/files/MaxScale>
- ```
[root@pxysvr ~]# rpm -ivh maxscale-2.1.2-1.rhel.7.x86_64.rpm
...
```



## 部署maxscale服务（续1）

知识讲解

- 修改配置 /etc/maxscale.cnf

```
[server1] //定义数据库服务器主机名
type=server
address=192.168.4.10 //master主机ip地址
port=3306
protocol=MySQLBackend

[server2] //定义数据库服务器
type=server
address=192.168.4.20 //slave主机ip地址
port=3306
protocol=MySQLBackend
```



## 部署maxscale服务（续2）

知识讲解

- 修改配置文件

```
[MySQL Monitor] //定义要监视的数据库节点
type=monitor
module=mysqlmon
servers=server1,server2 //主、从数据库的主机名
user=scalemon //用户名
passwd=111111 //密码
monitor_interval=10000

[Read-Write Service] //定义读写分离的数据库节点
type=service
router=readwritesplit
servers=server1,server2 //主、从数据库的主机名
user=maxscale //用户名
passwd=111111 //密码
max_slave_connections=100%
```



## 部署maxscale服务（续3）

知识讲解

- 在主、从数据库服务器创建授权用户

```
mysql> grant replication slave, replication client on *.* to
scalemon@'%' identified by '123qqq...A'; //创建监控用户

mysql> grant select on mysql.* to maxscale@ '%' identified by
'123qqq...A'; //创建路由用户
```



## 启动服务

- 主要操作：
  - 启动服务、查看端口、停止服务

知识讲解

```
]# maxscale -f /etc/maxscale.cnf
```

```
]# netstat -utnlp | grep maxscale
```

```
]# kill -9 19081
```



## 测试配置

- 在代理服务器本机访问管理
  - ]#maxadmin -uadmin -pmariadb -P端口
- 客户端连接代理访问数据
  - ]#mysql -h服务器地址 -P端口 -u用户名 -p密码

知识讲解





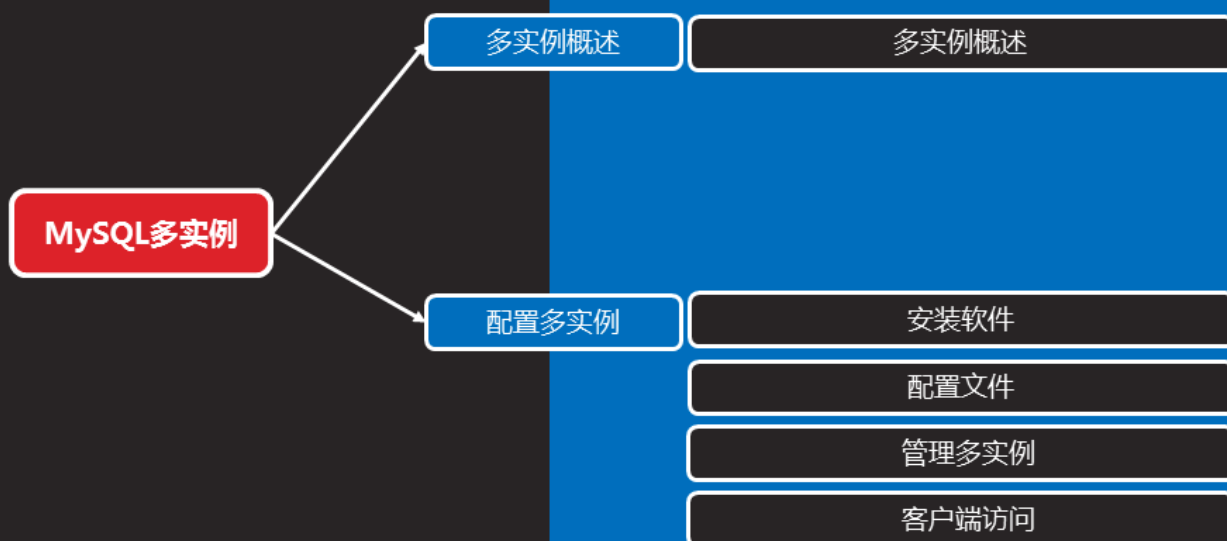
# 案例1：实现MySQL读写分离

课堂练习

1. 搭建一主一从结构
2. 配置maxscale代理服务器
3. 测试配置



## MySQL多实例



# 多实例概述

## 多实例概述

- 什么是多实例
  - 在一台物理主机上运行多个数据库服务
- 为什么要使用多实例
  - 节约运维成本
  - 提高硬件利用率



# 配置多实例

## 安装软件

- 具体操作
  - 解压软件
  - 修改目录名

```
tar -zxvf mysql-5.7.20-linux-glibc2.12-x86_64.tar.gz
```

```
mv mysql-5.7.20-linux-glibc2.12-x86_64 /usr/local/mysql
```

知识讲解



## 配置文件

知识讲解

- 主配置文件/etc/my.cnf
  - 每个实例要有独立的数据库目录和监听端口号
  - 每个实例要有独立的实例名称和独立的sock文件

```
[mysqld_multi] //启用多实例
mysql = /usr/local/mysql/bin/mysql_safe //指定进程文件路径
mysqladmin = /usr/local/mysql/bin/mysqladmin //指定管理命令路径
user = root //指定进程用户
```

```
[mysqldX] //实例进程名称 ,X表示实例编号,如 [mysql2]
port = 3307 //端口号
datadir = /data3307 //数据库目录 , 要手动创建
socket = /data3307/mysql.sock //指定sock文件的路径和名称
pid-file = /data3307/mysql.pid //进程pid号文件位置
log-error = /data3307/mysql.err //错误日志位置
```



## 管理多实例

- 启动服务
- 停止服务

知识讲解

```
]# /usr/local/mysql/bin/mysql_multi start 实例编号 //启动服务
```

```
]# /usr/local/mysql/bin/mysql_multi --user=root --password=密码 stop 实例编号 //停止服务
```



# 客户端访问

知识讲解

- 本机连接
  - 使用初始密码连接
  - 修改本机登陆密码
  - 连接实例

```
]#/usr/local/mysql/bin/mysql -uroot -p初始密码 -S sock文件
```

```
mysql> alter user root@ "localhost" identified by '新密码';
```



## 案例2：配置MySQL多实例

课堂练习

1. 在主机192.168.4.56上，配置第1个MySQL实例
  - 实例名称mysqld1、端口3307
  - 数据库目录/data3307、pid文件mysql1.pid
  - 错误日志mysql1.err
2. 在主机192.168.4.56上，配置第2个MySQL实例
  - 实例名称mysqld2、端口3308
  - 数据库目录/data3308、pid文件mysql2.pid
  - 错误日志mysql2.err



## 总结和答疑

总结和答疑

MySQL读写分离

问题现象

故障分析及排除

**Tedu.cn**  
达内教育

# MySQL读写分离

## 问题现象

知识讲解

- 客户端连接mysql代理服务失败
  - 报错： `ERROR 2003 (HY000): Can't connect to MySQL server ....`

```
[root@room9pc00 ~]# mysql -h172.40.50.132 -ujerry -p123
Warning: Using a password on the command line interface can
be insecure.
ERROR 2003 (HY000): Can't connect to MySQL server on
'172.40.50.132' (111)
[root@room9pc00 ~]#
```



## 故障分析及排除

知识讲解

- 原因分析
  - 连接使用的用户名或密码错误
  - mysql-proxy服务没有启动
- 解决办法
  - 查看授权用户是否存在
  - 查看mysql-proxy是否运行

```
mysql> select user,host from mysql.user; //查看授权用户是否存在
ps aux | grep "mysql-proxy" //查看mysql-proxy是否运行
```



