

DBA进阶

NSD DBA2

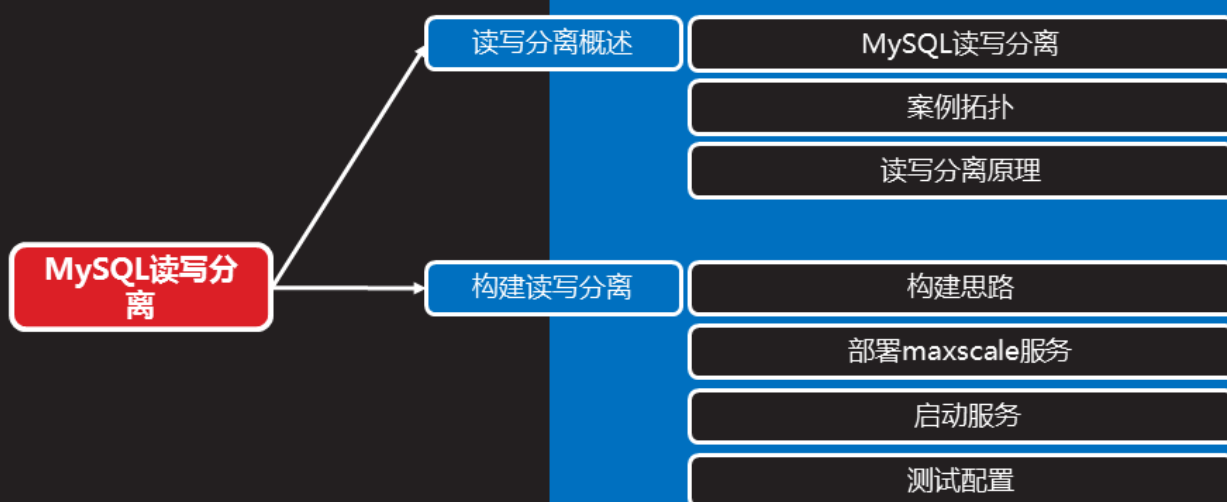
DAY02

内容

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



MySQL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？

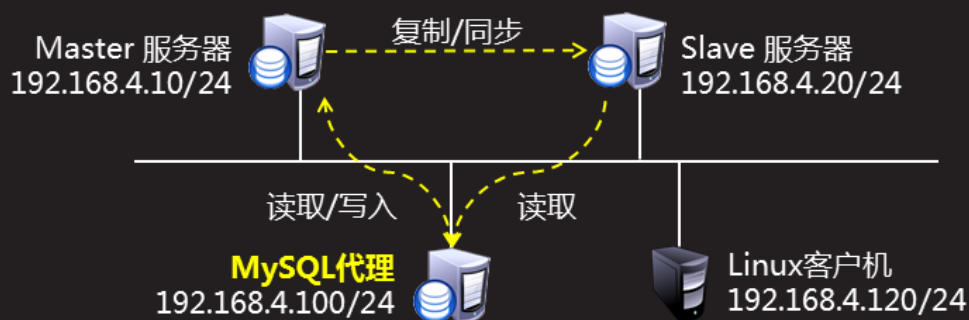
知识讲解



案例拓扑

- 添加一个MySQL代理
 - 为客户端提供统一的数据库接口

知识讲解



读写分离的原理

- 多台MySQL服务器
 - 分别提供读、写服务，均衡流量
 - 通过主从复制保持数据一致性
- 由MySQL代理面向客户端
 - 收到SQL写请求时，交给服务器A处理
 - 收到SQL读请求时，交给服务器B处理
 - 具体区分策略由服务设置

知识讲解



构建读写分离

构建思路

知识讲解

1. 已搭建好MySQL主从复制
 - 基于上一个实验的结果
 - 其中Slave为只读
2. 添加一台MySQL代理服务器
 - 部署/启用 maxscale
3. 客户端通过代理主机访问MySQL数据库
 - 访问代理服务器



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3）

- 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 '111111'; //创建监控用户
```

```
mysql> grant select on mysql.* to maxscale@ '%' identified by
'111111'; //创建路由用户
```

```
mysql> grant all on *.* to student@'%' identified by '111111';
//创建访问数据用户
```

## 启动服务

- 主要操作：
  - 启动服务、查看端口、停止服务

知识讲解

```
[root@bogon ~]# maxscale --config=/etc/maxscale.cnf
```

```
[root@bogon ~]# netstat -utnlp | grep maxscale
tcp 0 0 192.168.1.110:58960 192.168.1.101:3306
ESTABLISHED 19081/maxscale
tcp 0 0 192.168.1.110:43508 192.168.1.111:3306
ESTABLISHED 19081/maxscale
tcp6 0 0 :::4006 :::* LISTEN
19081/maxscale
```

```
[root@bogon ~]# kill -9 19081
```



## 测试配置

- 在maxscale本机连接管理端口
  - maxadmin -uadmin -pmariadb -P端口
- 访问maxscale代理
  - mysql -h服务器地址 -P端口 -u用户名 -p密码

知识讲解

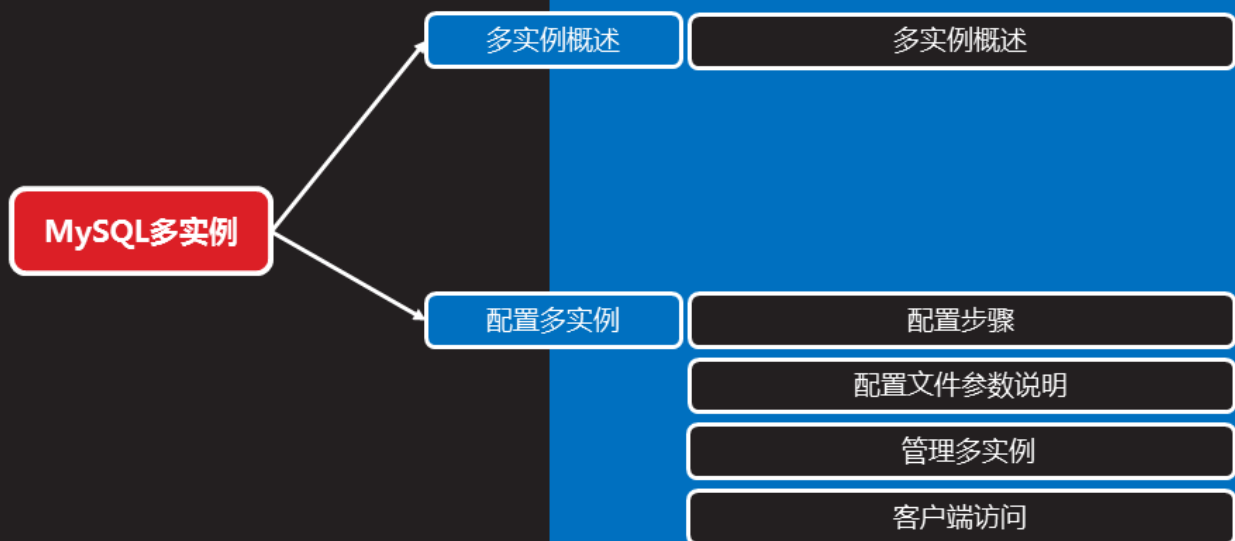
```
[root@bogon ~]# mysql -h192.168.4.100 -P4006 -ustudent -p111111
```

```
MySQL [(none)]> select @@hostname; //查看当前主机名
+-----+
| @@hostname |
+-----+
| slave111 |
+-----+
```





## MySQL多实例



# 多实例概述

## 多实例概述

知识讲解

- 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

## 配置步骤

- 配置步骤
  - 1) 安装支持多实例服务的软件包
  - 2) 修改主配置文件
  - 3) 初始化授权库
  - 4) 启动服务
  - 5) 客户端访问测试





## 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

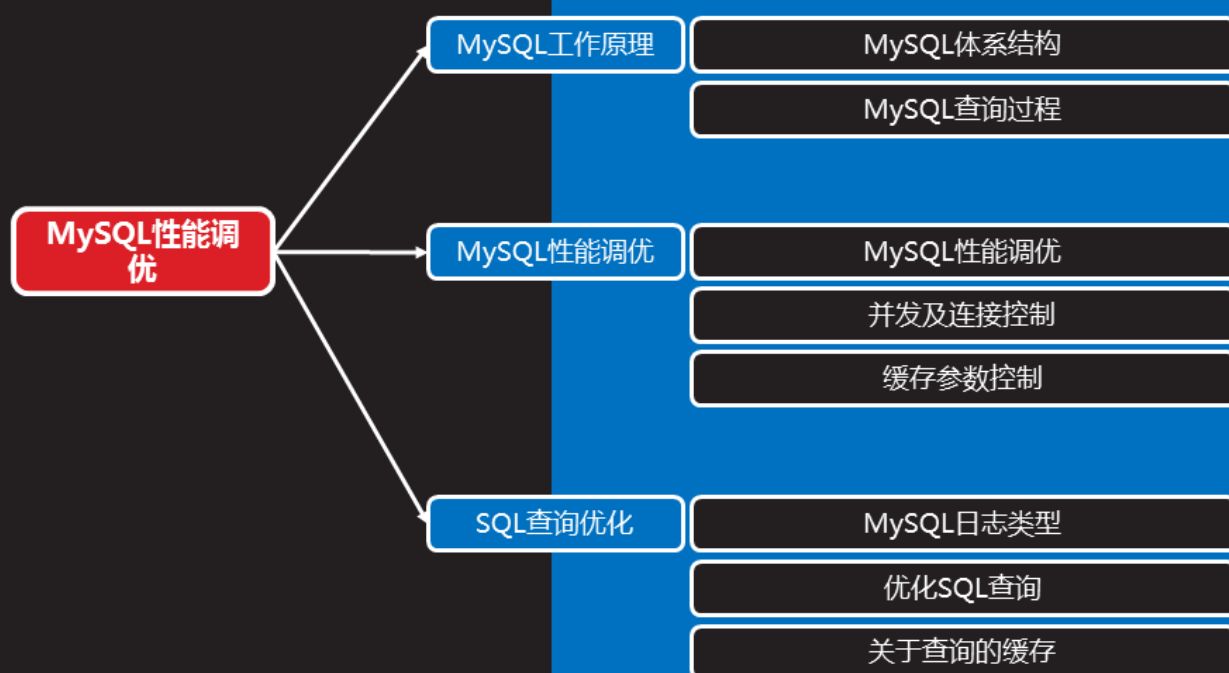
- 实例名称mysql1、端口3307
- 数据库目录/data3307、pid文件mysql1.pid
- 错误日志mysql1.err

2. 在主机192.168.4.56上，配置第2个MySQL实例

- 实例名称mysql2、端口3308
- 数据库目录/data3308、pid文件mysql2.pid
- 错误日志mysql2.err

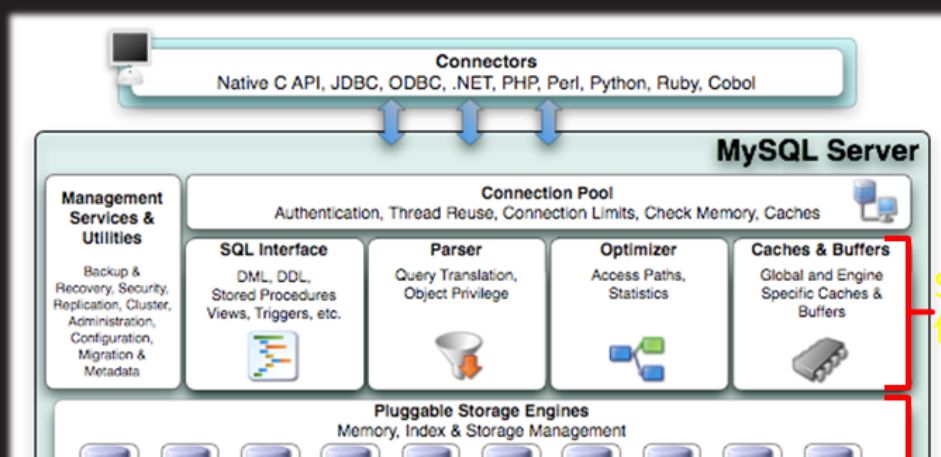


## MySQL性能调优



# MySQL工作原理

## MySQL体系结构

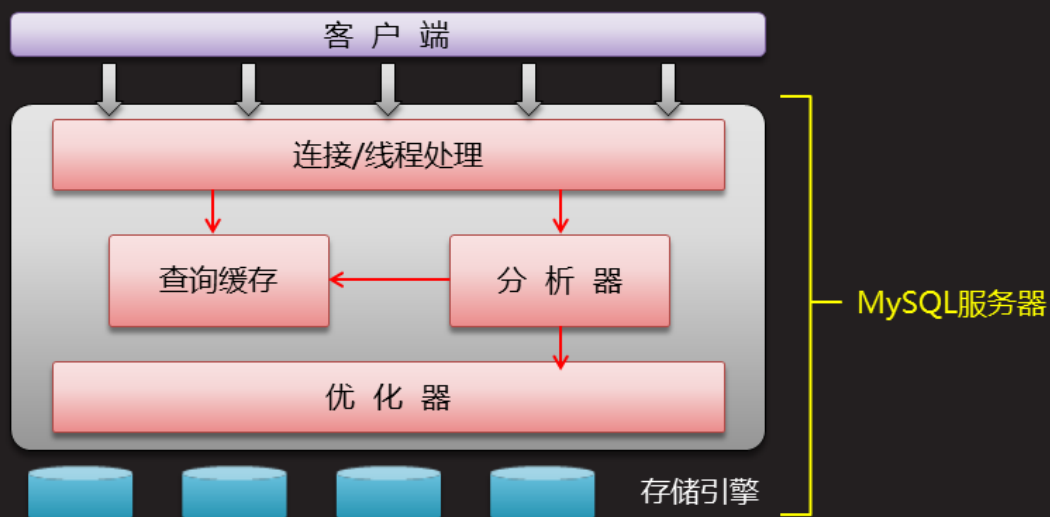


SQL接口/解析器/  
优化器/缓存

# MySQL执行流程

- MySQL执行流程

知识讲解



# MySQL性能调优

知识讲解

- 提高MySQL系统的性能、响应速度
  - 替换有问题的硬件（CPU/磁盘/内存等）
  - 服务程序的运行参数调整
  - 对SQL查询进行优化



## 并发及连接控制

知识讲解

- 连接数、连接超时

| 选项              | 含义                           |
|-----------------|------------------------------|
| max_connections | 允许的最大并发连接数                   |
| connect_timeout | 等待连接超时，默认10秒，仅登录时有效          |
| wait_timeout    | 等待关闭连接的不活动超时秒数，默认28800秒（8小时） |





## 并发及连接控制（续1）

知识讲解

- 查看当前已使用的连接数

```
mysql> FLUSH STATUS;
Query OK, 0 rows affected (0.05 sec)

mysql> SHOW GLOBAL STATUS LIKE "max_used_connections";
+-----+-----+
| Variable_name | Value |
+-----+-----+
| Max_used_connections | 5 |
+-----+-----+
```

- 查看默认的最大连接数

```
mysql> SHOW VARIABLES LIKE "max_connections";
+-----+-----+
| Variable_name | Value |
+-----+-----+
| max_connections | 151 |
+-----+-----+
```

理想比率  $\leq 85\%$ 

## 缓存参数控制

- 缓冲区、线程数量、开表数量

知识讲解

| 选项                | 含义                  |
|-------------------|---------------------|
| key_buffer-size   | 用于MyISAM引擎的关键索引缓存大小 |
| sort_buffer_size  | 为每个要排序的线程分配此大小的缓存空间 |
| read_buffer_size  | 为顺序读取表记录保留的缓存大小     |
| thread_cache_size | 允许保存在缓存中被重用的线程数量    |
| table_open_cache  | 为所有线程缓存的打开的表的数量     |



## 缓存参数控制（续2）

- sort\_buffer\_size=256K
  - 增大此值可提高ORDER和GROUP的速度

知识讲解

```
mysql> SHOW VARIABLES LIKE "sort_buffer_size";
```

| Variable_name    | Value  |
|------------------|--------|
| sort_buffer_size | 262144 |



## 缓存参数控制（续3）

- 查看表记录读取缓存
  - 此缓存值影响SQL查询的响应速度

知识讲解

```
mysql> SHOW VARIABLES LIKE "read_%_size";
```

| Variable_name        | Value  |
|----------------------|--------|
| read_buffer_size     | 131072 |
| read_rnd_buffer_size | 262144 |



## 缓存参数控制（续4）

- 查看可重用线程数

```
mysql> SHOW VARIABLES LIKE "thread_%_size";
```

| Variable_name     | Value |
|-------------------|-------|
| thread_cache_size | 9     |

- 查看当前的线程重用状态

```
mysql> SHOW GLOBAL STATUS LIKE "threads_%";
```

| Variable_name     | Value |
|-------------------|-------|
| Threads_cached    | 1     |
| Threads_connected | 2     |
| Threads_created   | 3     |
| Threads_running   | 2     |



# MySQL日志类型

- 常用日志种类及选项

| 类 型   | 用 途                | 配 置                                                        |
|-------|--------------------|------------------------------------------------------------|
| 错误日志  | 记录启动/运行/停止过程中的错误消息 | log-error[=name]                                           |
| 查询日志  | 记录客户端连接和查询操作       | general-log<br>general-log-file=                           |
| 慢查询日志 | 记录耗时较长或不使用索引的查询操作  | slow-query-log<br>slow-query-log-file=<br>long-query-time= |

知识讲解

