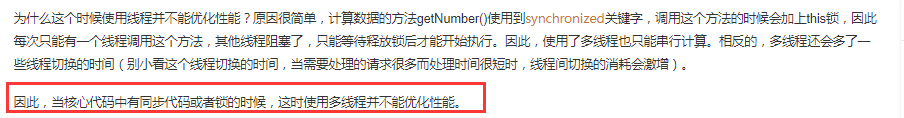
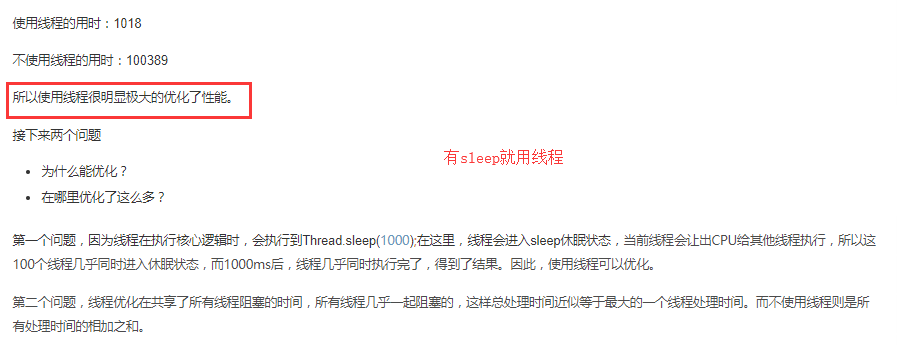
# 理解性能优化

## 容量评估，平均QPS,高峰QPS，单机QPS

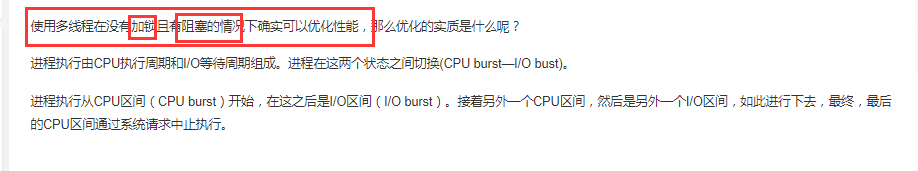


### 使用线程一定能优化吗





### 多线程优化的本质

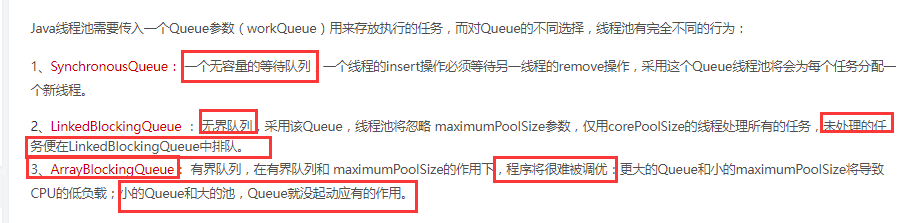


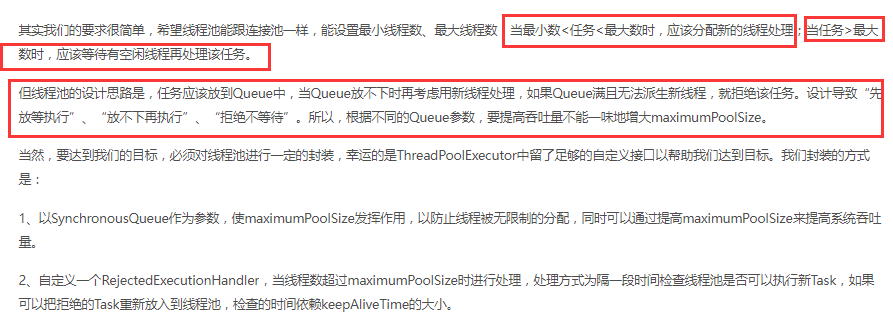
## CDN网络加速原理

<https://blog.csdn.net/apricotCandy/article/details/88718498>

## 连接池的性能优化

### 1.线程池（解决用户响应时间长的问题）

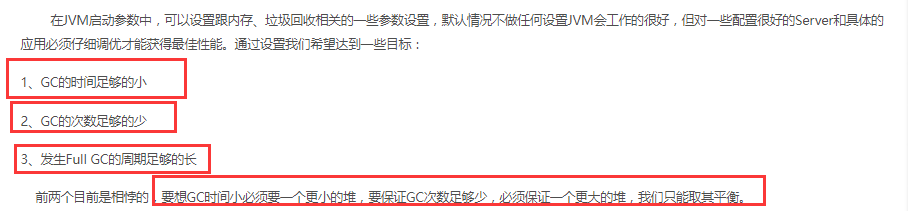


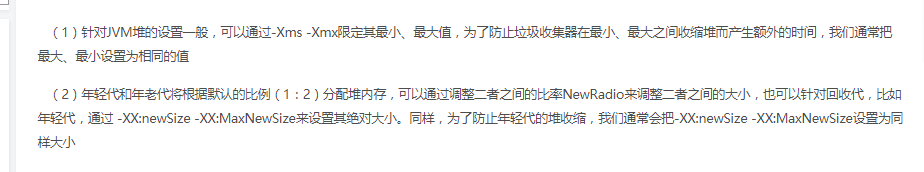


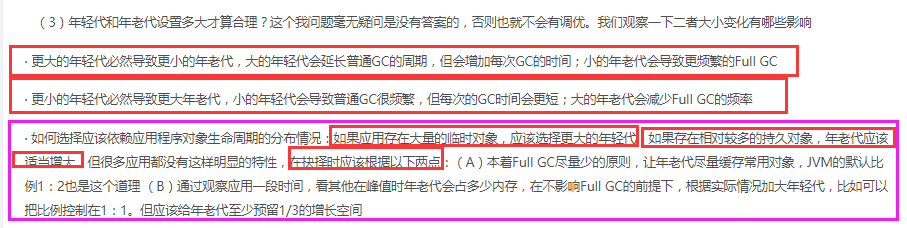
### 2.连接池（数据库连接池BasicDataSource）



### 3JVM启动参数：调整各代的内存比例和垃圾回收算法，提高吞吐量







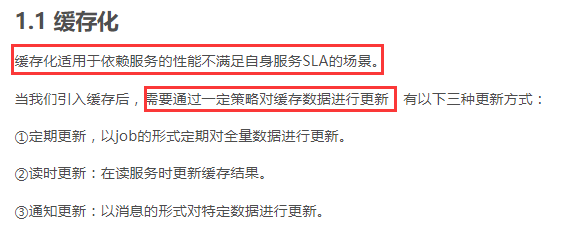


### 4程序算法：改进程序逻辑算法提高性能

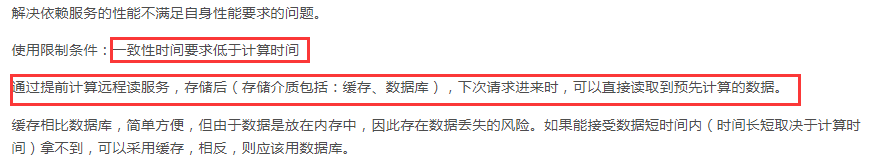
## 性能优化方法论

### 依赖远程读服务优化

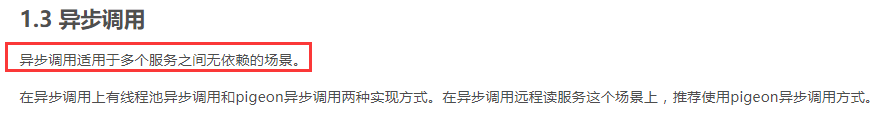
#### 1缓存化



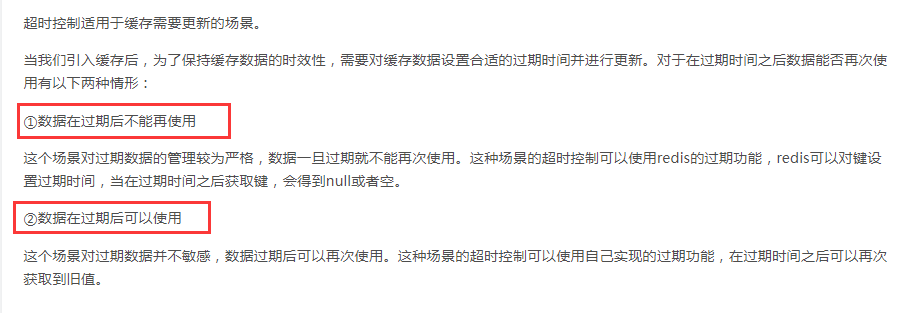
#### 2预计算



#### 3异步调用



#### 4超时控制



### 高并发写场景

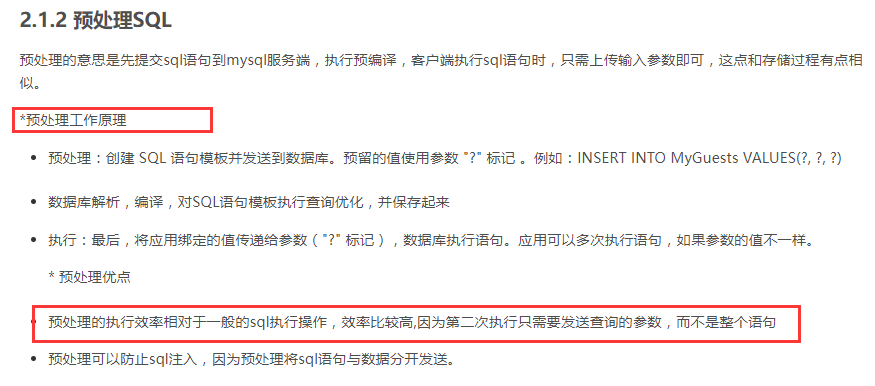
#### 1合并写

多条SQL写入场景，可采用合并写SQL的方式来提高SQL操作性能。具体方法如下：

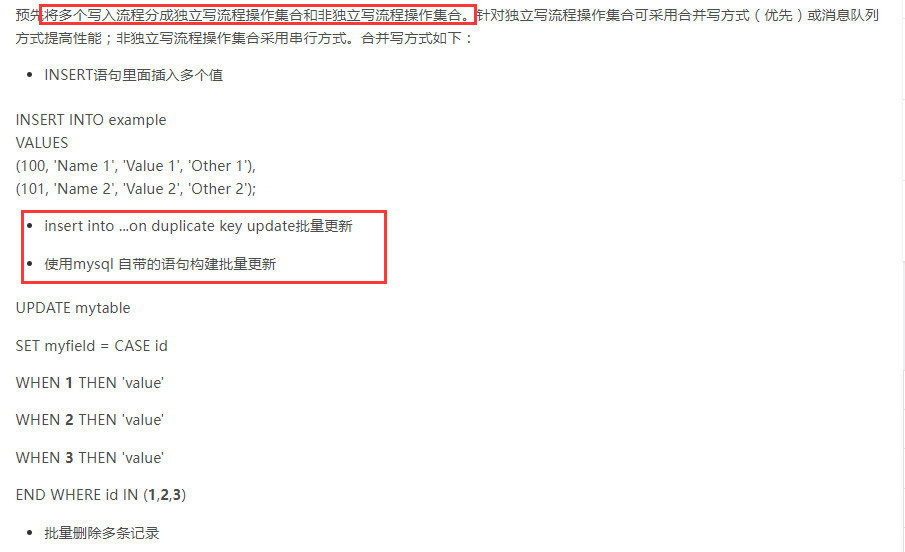
##### 1.1消息队列



##### 1.2预处理SQL



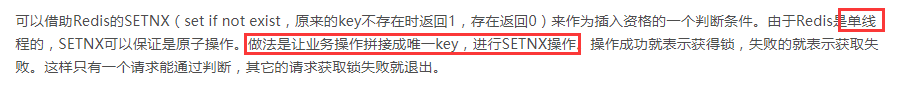
##### 1.3长流程写入操作



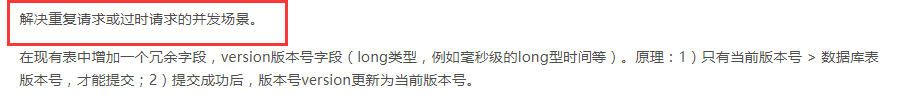
#### 2比较更新

高并发写场景，可采用比较更新策略（CAS），过滤不符合条件的SQL，保证数据一致性的同时，也提高了性能，避免死锁带来的死等待。具体方法如下：

##### 2.1基于redis



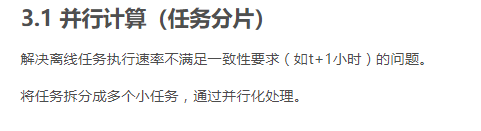
##### 2.2基于版本号



##### 2.3基于状态机条件更新



### 离线任务计算场景



# MySQL调优

理解MySQL底层原理B+Tree机制

索引优化揭秘

SQL语句优化

SQL执行机制详解

# 测试理论

深度解析TDD测试驱动开发

全链路压测经验

Nginx根据ip进行灰度发布A/BTest

真实流量监测工具tcpcopy

单元测试minUnit

# linux调优

内核代码组织架构

系统IO参数调优配置

网络协议栈参数配置

内核代码编译与内核更换

# GCC调优

快速编译工程Makefile编写

编译机制，编译原理