莫锦波

**华东师范大学 计算机科学与软件工程学院**

性能测试报告

Meethere场馆预订系统

目录

[1 测试结果评价与建议 4](#_Toc501373243)

[1.1 测试结果 4](#_Toc501373244)

[1.2 系统性能评价 5](#_Toc501373245)

[1.2.1 交易响应时间 5](#_Toc501373246)

[1.2.2 业务处理能力 5](#_Toc501373247)

[1.2.3 稳定性/健壮性 5](#_Toc501373248)

[1.3 系统性能建议 5](#_Toc501373249)

[2 性能问题解决分析 5](#_Toc501373250)

[2.1 待解决问题 5](#_Toc501373251)

[2.1.1 严重性能问题 5](#_Toc501373252)

[2.1.2 响应时间问题 5](#_Toc501373253)

[2.2 已解决问题 5](#_Toc501373254)

[2.2.1 严重性能问题 5](#_Toc501373255)

[2.2.2 响应时间问题 5](#_Toc501373256)

[3 独立/混合场景结果 5](#_Toc501373257)

[3.1 测试结果描述 5](#_Toc501373258)

[3.2 详细测试结果 5](#_Toc501373259)

[4 性能缺陷统计分析 7](#_Toc501373290)

[4.1 缺陷严重程度 7](#_Toc501373291)

[4.2 缺陷类型分析 7](#_Toc501373292)

[4.3 缺陷原因分析 7](#_Toc501373293)

[5 测试环境分析 7](#_Toc501373294)

[5.1 系统架构设计 7](#_Toc501373295)

[5.2 测试环境配置 7](#_Toc501373296)

[5.3 环境差异分析 7](#_Toc501373297)

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 场景类型 | 场景名 | 场景步骤 |
| 1 | 独立场景 | 用户登录 | 1.用户登录 |
| 2 | 独立场景 | 查看新闻 | 1.进入新闻页面查看新闻列表 |
| 3 | 独立场景 | 查看订单 | 1.进入我的订单页面查看订单 |
| 4 | 独立场景 | 搜索场馆 | 1.进入场馆页面搜索场馆 |
| 5 | 混合场景 | 发表评论 | 1. 进入新闻页面 2. 点击某一个新闻 3. 查看新闻与评论 4. 发表评论 |
| 6 | 混合场景 | 取消/评价订单 | 1. 进入我的的订单页面查看订单 2. 取消或评价订单 |
| 7 | 混合场景 | 创建订单 | 1. 进入场馆页面搜索场馆 2. 选择场馆预订 3. 进入我的订单界面查看订单 |

# 测试结果评价与建议

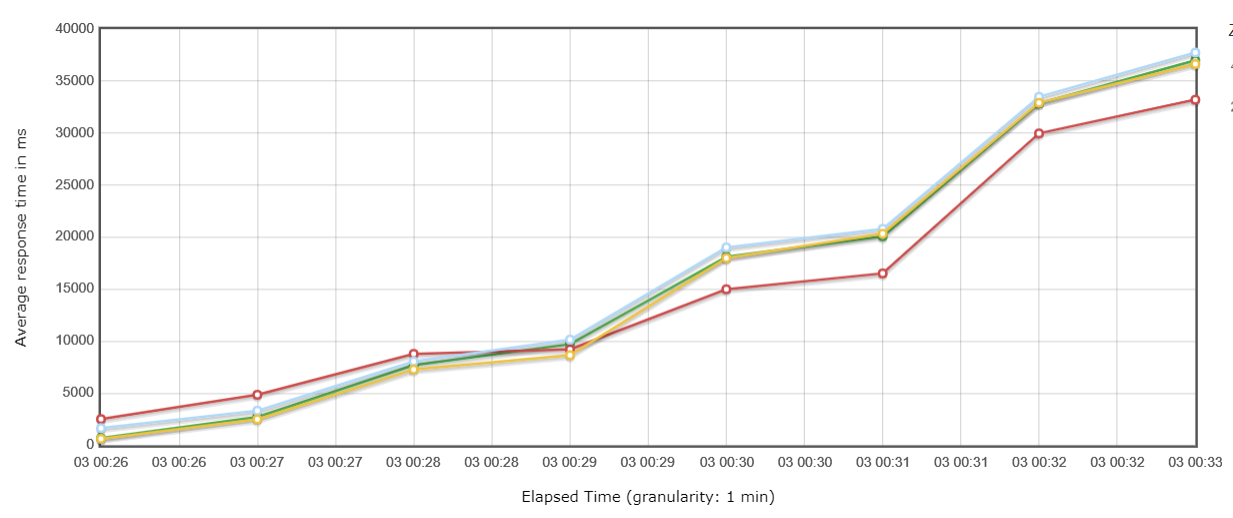
## 测试结果

|  |  |
| --- | --- |
| 测试开始日期 | 2019年12月29日 |
| 测试结束日期 | 2019年12月31日 |
| 提交缺陷数 | 1 |
| 延期/未解决缺陷数 | 0 |
| 拒绝缺陷数 | 0 |
| 总测试场景 | 4个独立场景+3个混合场景 |
| 执行场景数 | 8 |
| 场景执行率 | 100% |
| 缺陷率（%） | 0 |
| 投入人数 | 1 |
| 测试生产率 | 无 |
| 系统上线建议 | 系统响应时间太长，建议升级服务器带宽 |

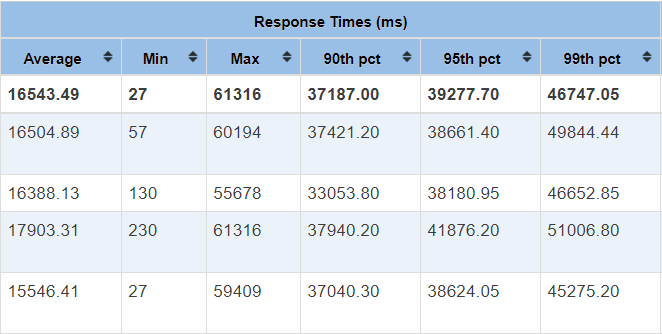
## 系统性能评价

### 交易响应时间

我们来分析测试场景7 创建订单的混合场景的压力测试（20/50/100/200），可以看到随着并发数增加，响应时间也上升。

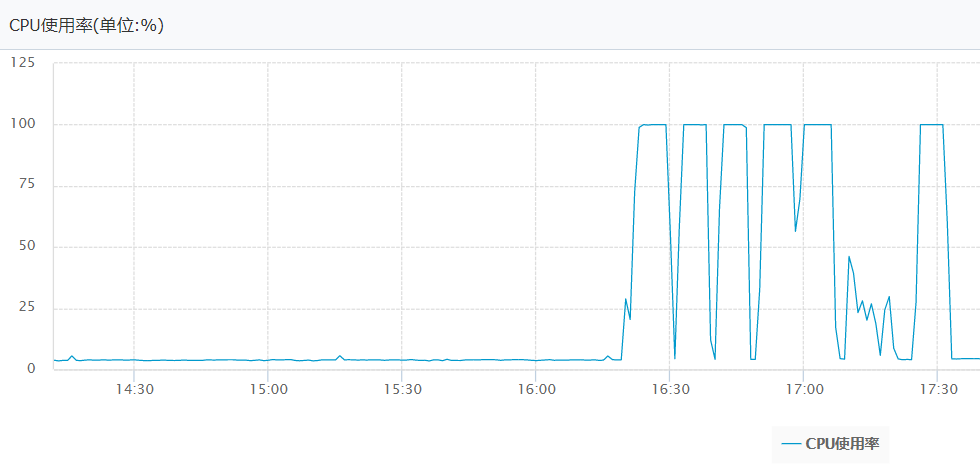


测试5的流逝时间-响应时间图

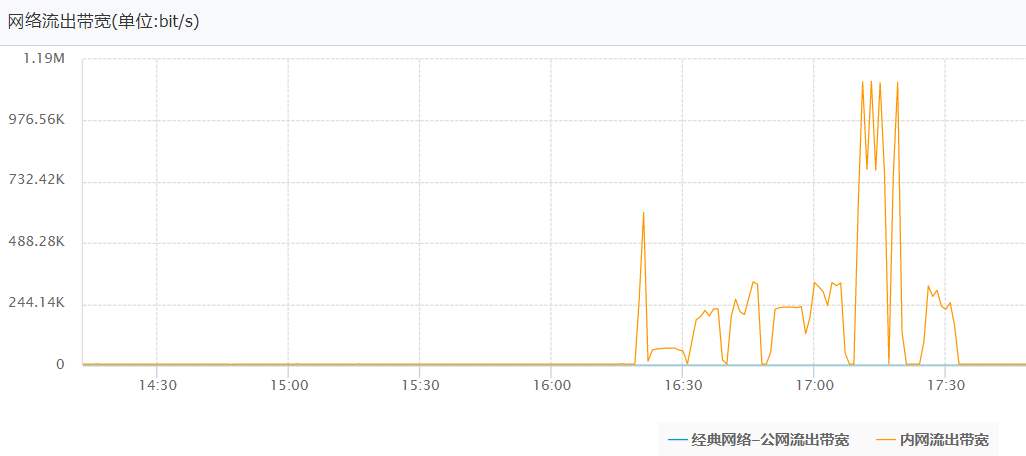


测试5的响应时间表格

平均响应时间居然达到了夸张的16s，我们来查看服务器资源使用率来看看为什么这么慢



服务器CPU使用率

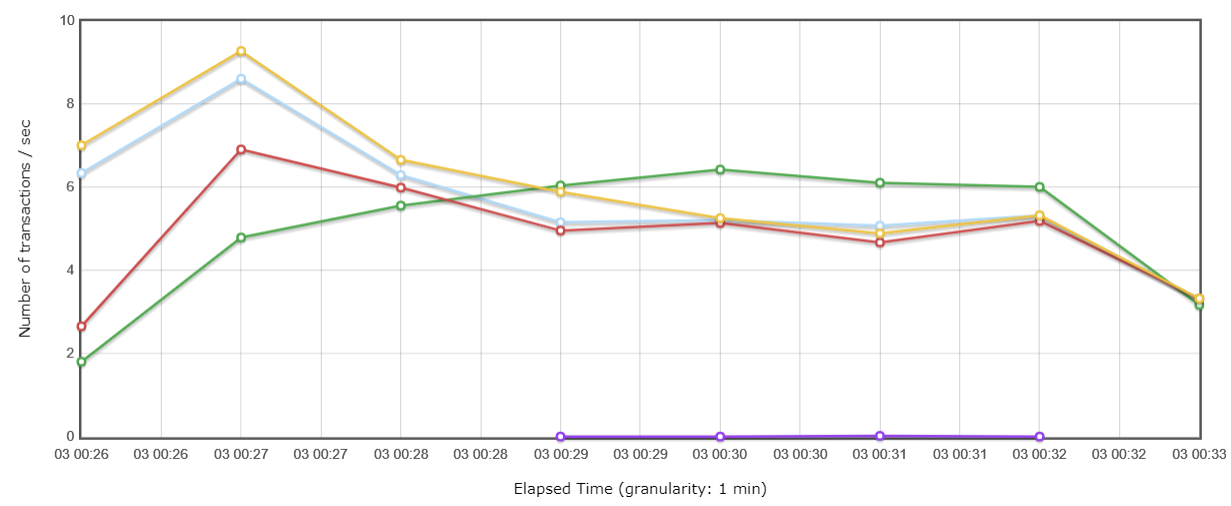


上行带宽使用量

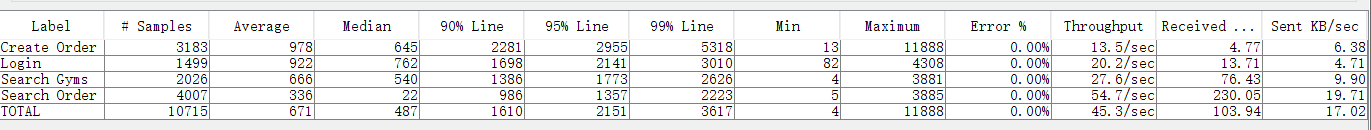
我们可以看到服务器CPU和带宽(1mbps)都已经打满，已经运行到了瓶颈。推测响应时间慢的原因是这个。

### 业务处理能力

并发线程数达到100时tps最大，此后tps随并发数增长而稳定在24-26左右。推测瓶颈在24-26。

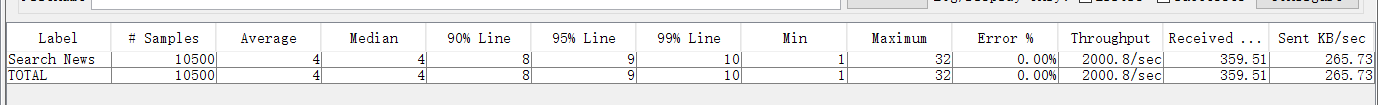


为什么速度这么慢呢？我们在本地机器上测试看看

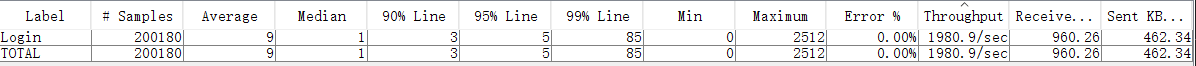


可以看到吞吐量还是几十上下，也不快。并且响应时间在600ms左右

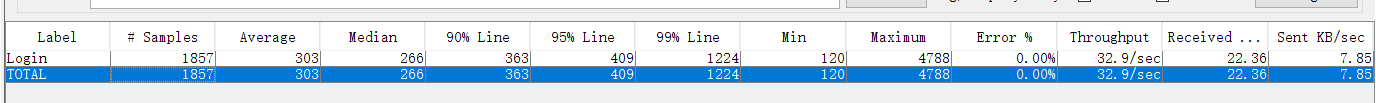
我们尝试了另一个项目的一个搜索新闻的接口（写了两套后台，另一个后台是我写的，最终提交的是这一个项目）发现吞吐量能达到2000+，有时稳定在3000甚至接近4000。并且平均响应时间在4ms左右远快于600ms。



再来看看登录接口（未使用缓存），发现其吞吐量能达到2000，平均响应时间9ms



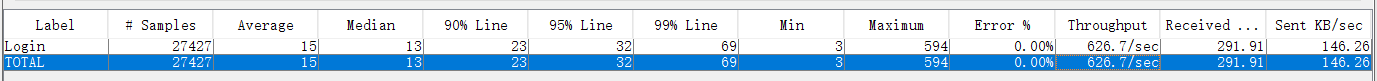
单独测试偏慢的那个项目的登录接口，结果如下，吞吐量33，响应时间3ms



这两个项目使用的是同一个本地数据库，同一个本地缓存。速度差异十分之大。

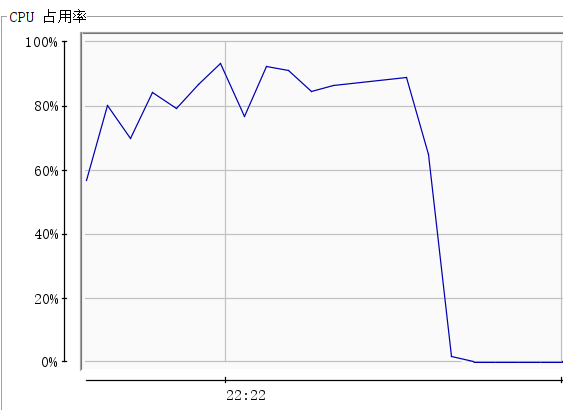
由于快的项目session存储在JVM中，慢的存储在redis中。

我们将快的项目session也存储在redis中，得到以下性能测试数据，吞吐量630，响应时间15ms

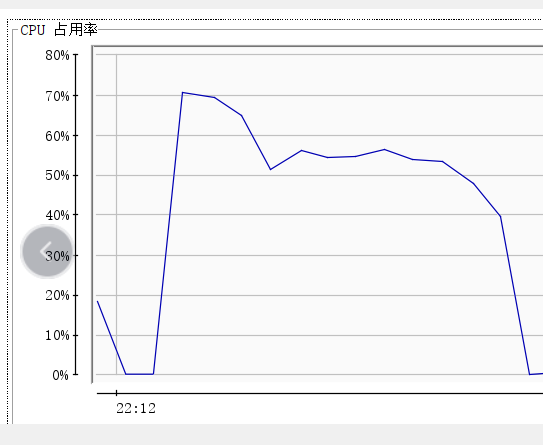


推测redis不是慢的原因。

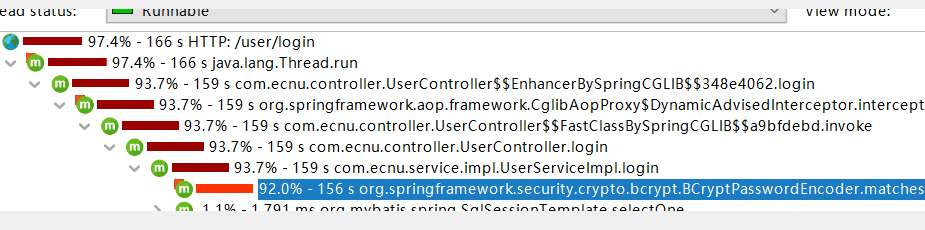
怀疑是程序的原因，使用JConsole监控运行程序，可以看到慢的项目一直接近90%左右的CPU使用率。



快的项目的CPU使用率最高才达到70%，稳定在50多左右



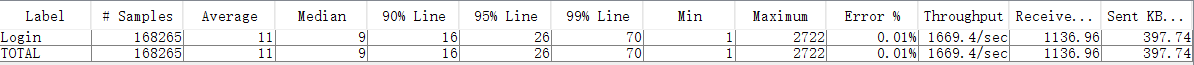
**CPU使用率并不正常，**使用**JProfiler**进一步分析



发现97%的cpu消耗在登录接口的BCryptPasswordEncoder.matches上。

查询资料后得知，BCryptPasswordEncoder加密速度十分的慢，在100ms-700ms。性能凶手找到。

将BCryptPasswordEncoder改成明文对比后的性能情况如下



可以看到从吞吐量33，响应时间300ms飙升到了1669吞吐量，响应时间11ms。性能问题解决

### 稳定性/健壮性

系统错误率接近0（大部分错误是用户会话过期），非常稳定

## 系统性能建议

提高服务器带宽

# 性能问题解决分析

## 待解决问题

### 严重性能问题

* 上行带宽过小
* Bcrypt加密过久，但是为了安全性并未替换。考虑性能可能会替换成md5加密

### 响应时间问题

* 上行带宽过小

## 已解决问题

### 严重性能问题

无，需要更好的服务器

### 响应时间问题

无，需要更好的服务器

# 独立/混合场景结果

## 测试结果描述

独立场景使用如下加压方式

|  |  |  |
| --- | --- | --- |
| **并发数** | **加压时间** | **加压方式** |
| 1 | 30s | 直接加压 |
| 10 | 1分钟 | 1个用户/1秒 |
| 20 | 1分钟 | 1个用户/1秒 |
| 50 | 2分钟 | 1个用户/1秒 |
| 100 | 2分钟 | 1个用户/1秒 |
| 200 | 1分钟 | 5个用户/1秒 |

混合场景使用以下加压方式

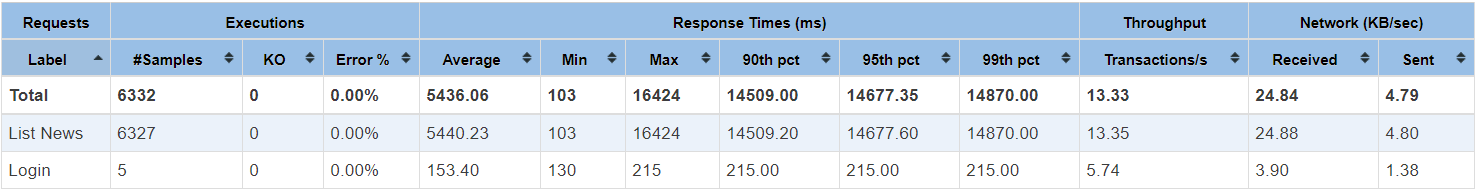
|  |  |  |
| --- | --- | --- |
| **并发数** | **加压时间** | **加压方式** |
| 20 | 1分钟 | 1个用户/1秒 |
| 50 | 2分钟 | 1个用户/1秒 |
| 100 | 2分钟 | 1个用户/1秒 |
| 200 | 1分钟 | 5个用户/1秒 |

对于独立与混合场景响应时间都远超过1s。。

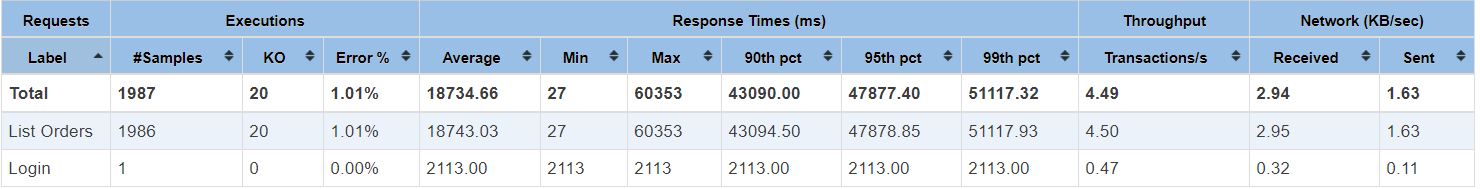
## 详细测试结果

## 

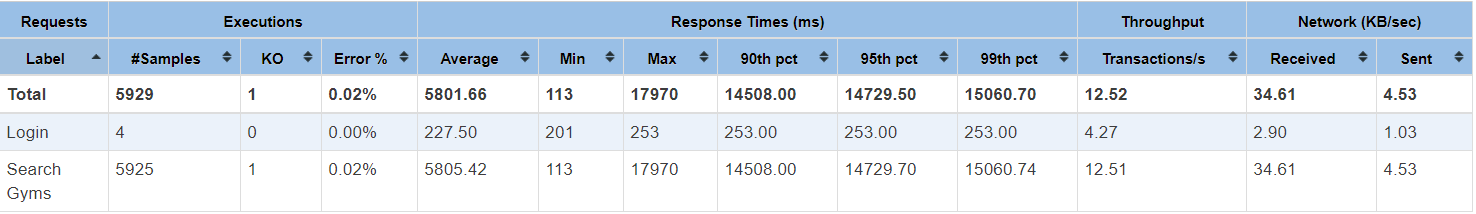
测试1



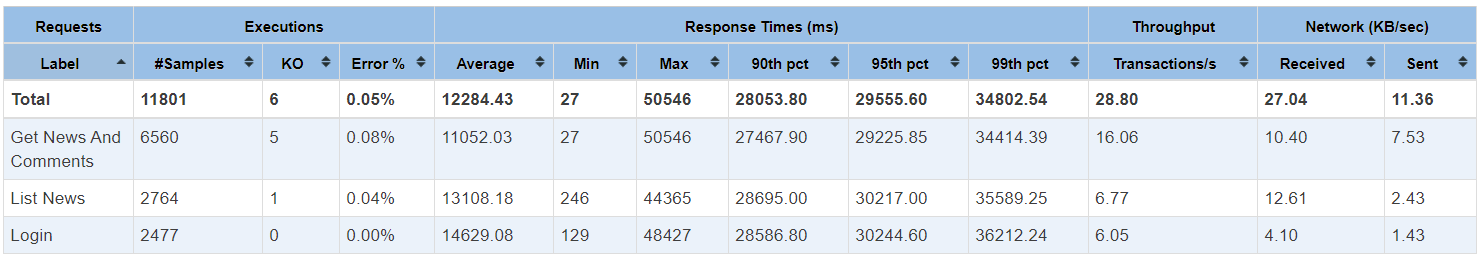
测试2



测试3



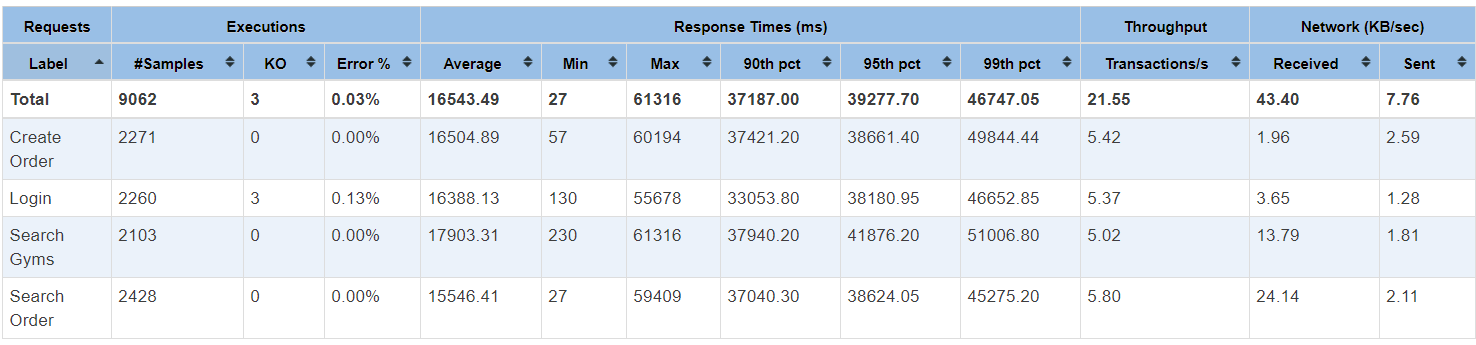
测试4



测试5

## 

测试6



测试7

# 性能缺陷统计分析

代码中有不明原因性能缺陷

# 测试环境分析

## 系统架构设计

前端React配合Nginx请求转发，后端Spring Boot，数据库MySQL，数据访问Spring Data Jpa

## 测试环境配置

硬件环境

|  |  |
| --- | --- |
| **环境名** | **环境信息** |
| 阿里云ECS云服务器 | 1核CPU 2G RAM 40G磁盘 |
| 网络设备 | 1Mbps带宽，平均RTT 12ms |
| 客户访问设备 | — |

软件环境

|  |  |
| --- | --- |
| **软件名** | **版本** |
| CentOS | 7.2 Linux内核版本 4.9 |
| Redis | 5.0 |
| Mysql | 8.0 |
| Jdk | 11 |
| Nginx | 1.1.17 |

## 环境差异分析

无