# 压测环境



\*下述所有压测脚本均是基于内网IP进行的

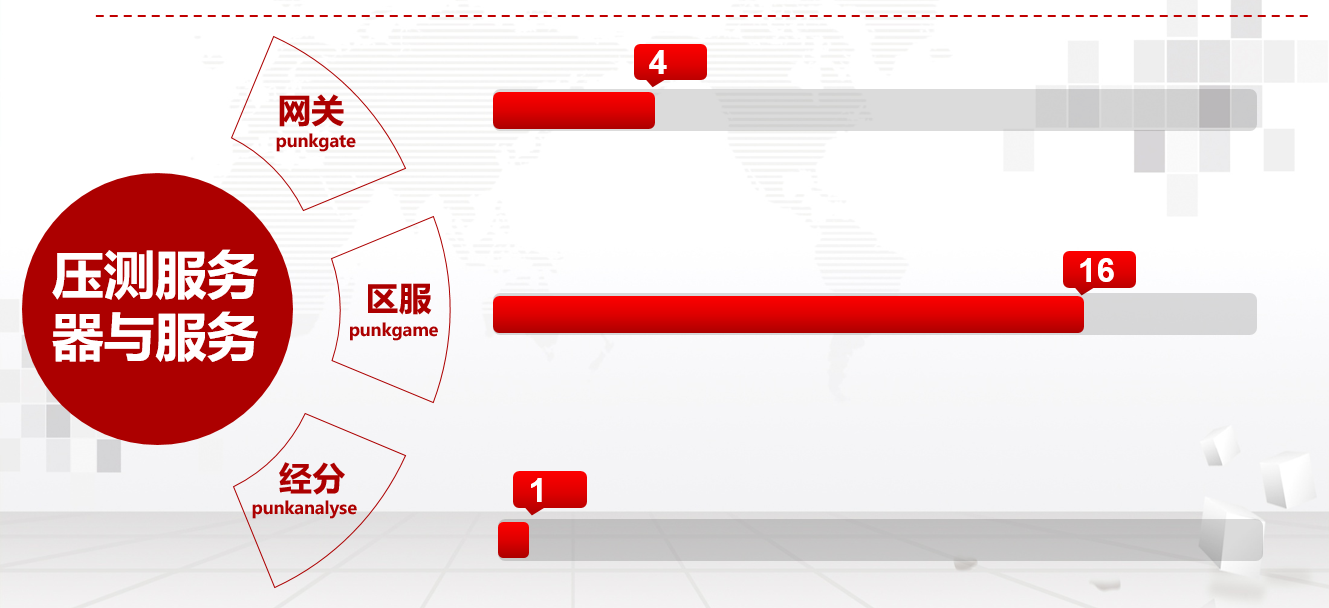
|  |  |
| --- | --- |
| **内网** | **外网** |
|  |  |

## 服务器环境

|  |  |  |
| --- | --- | --- |
| **CPU** | **内存** | **硬盘** |
| 4核 | 8G | 标准 |

同一台服务器上面运行着三个服务，分别是经分（punkanalyse）、网关（punkgate）和区服（punkgame）





## 客户端环境

|  |  |  |
| --- | --- | --- |
| **CPU** | **内存** | **硬盘** |
| 2核 | 4G | 标准 |

## 网络环境

|  |  |  |  |
| --- | --- | --- | --- |
| **序列号** | **服务器** | **IP** | **带宽** |
| 1 | 压测服务器 | 内网IP：  外网IP： | 1G以上 |
| 2 | 压测客户端 | 内网IP：  外网IP： | 1G以上 |
| 3 | 数据库 |  | 1G以上 |

# 压测说明

## 测试场景

|  |  |  |  |
| --- | --- | --- | --- |
| **测试项编号** | **测试场景** | **测试业务点** | **测试过程描述** |
| 1 | 创建角色与登录 | http协议的/login接口与websocket的"cmd":"connect" | 模拟并发用户在线综合场景：注册与登录 |
| 2 | 竞技场战斗 | http协议的/login接口与websocket的"cmd":"arenaBattle" | 模拟并发用户在线综合场景：竞技场排阵与战斗 |
| 3 | 创建联盟 | http协议的/login接口与websocket的"cmd":" leagueJoin" | 模拟并发用户在线综合场景：创建联盟与加入联盟 |
| 4 | 卡牌升级 | http协议的/login接口与websocket的"cmd":" cardLevelUp " | 模拟并发用户在线综合场景：消耗经验书与卡牌升级 |

## 测试方法

本次测试，使用如下方法进行压测

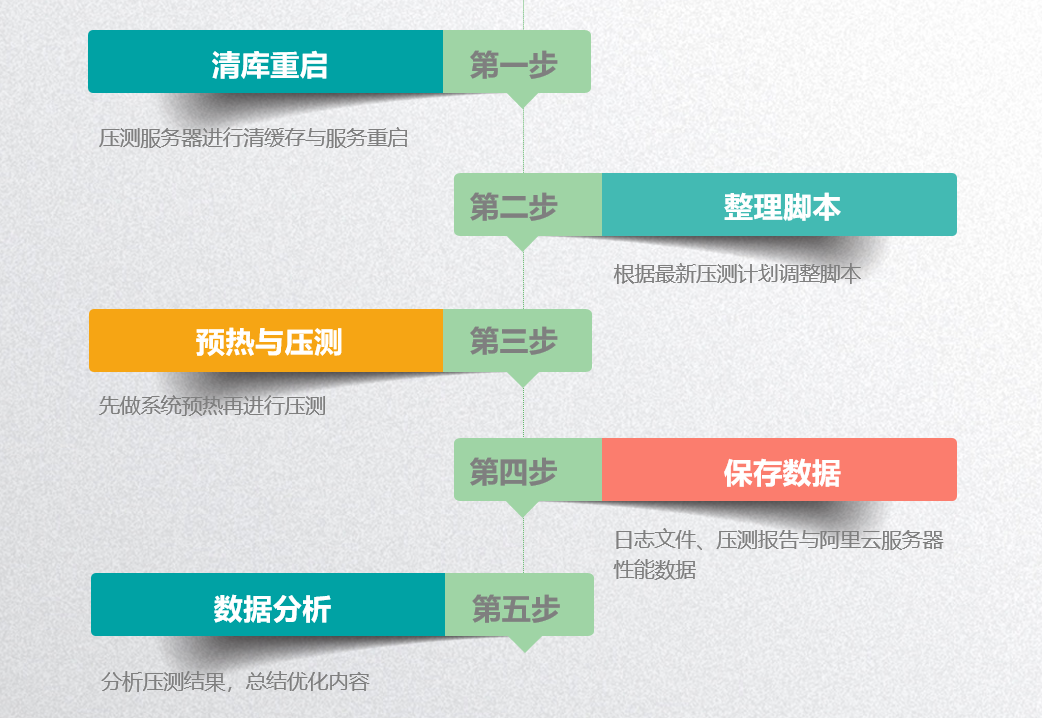
|  |  |  |  |
| --- | --- | --- | --- |
| **测试项编号** | **测试场景** | **线程数** | **循环数** |
| 1 | 创建角色与登录 | 100、200、300、400 | 100 |
| 2 | 竞技场战斗 | 100、200、300、400 | 100 |
| 3 | 创建联盟 | 100 | 100 |
| 4 | 卡牌升级 | 100、200、300 | 100 |

如创建角色与登录模块压测参数为100线程，10000个测试样本，使用指令如下：

sh jmeter -n -t RegistAndLogin.jmx -l RegistAndLogin.jtl -e -o /home/develop/jmeter/bin/report -JthreadNum=100 -JloopNum=100 -JrampupTime=2

其中，生成jtl测试结果数据文件与位于report目录的html测试报告，-JthreadNum代表线程数，-JloopNum代表循环次数，JrampupTime代表完成建线程时间。

另外，每次压测都会进行下图的流程，其中第一步清库不是必须的，要根据服务cpu情况、数据库RDS的cpu情况与实际redis的数据量去判定是不是要做这一步，还有第三步的预热这一步操作尽量每一次压测都要进行，防止服务器cpu使用率超过90%，最后，总结问题与提出优化建议时，使用Arthas等软件进行辅助发现性能问题。



# 测试结果分析

## 测试结果

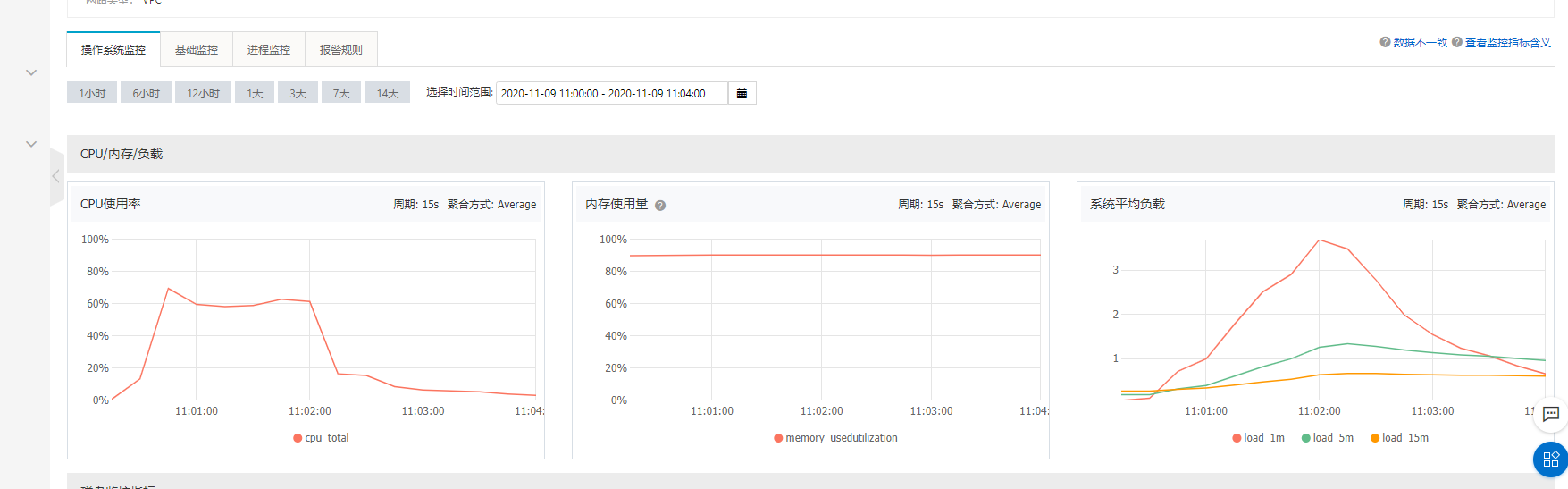
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **场景设定** | | **总体情况** | | | | **服务器资源** | |
| **检测场景** | **检测业务点** | **并发用户/TPS** | **并发成功率(%)** | **平均响应时间/接口最大平均响应时间（秒）** | **成功率（%）** | **CPU利用率（平均%/最高%）** | **内存使用情况** |
| 创建角色与登录 | 连接、注册、登录 | 100/394.45 | 100% | 0.004/0.011 | 99.76% | 58% | 90% |
| 创建角色与登录 | 连接、注册、登录 | 200/527.61 | 100% | 0.005/0.014 | 99.75% | 65% | 88% |
| 创建角色与登录 | 连接、注册、登录 | 300/473.77 | 100% | 0.013/0.021 | 99.72% | 80% | 90% |
| 创建角色与登录 | 连接、注册、登录 | 400/635.94 | 100% | 0.006/0.018 | 99.74% | 82% | 75% |
| 竞技场战斗 | 排阵、战斗 | 100/332.61 | 100% | 0.085/1.067 | 99.48% | 30% | 67% |
| 竞技场战斗 | 排阵、战斗 | 200/356.47 | 100% | 0.119/2.196 | 99.64% | 35% | 72 |
| 竞技场战斗 | 排阵、战斗 | 300/435.24 | 100% | 0.091/1.117 | 99.46% | 43% | 78% |
| 竞技场战斗 | 排阵、战斗 | 400/364.91 | 100% | 0.088/1.159 | 99.51% | 55% | 82% |
| 创建联盟 | 充值vip、创建联盟、加入联盟 | 100/226.24 | 100% | 0.143/0.286 | 97.77% | 30% | 72% |
| 卡牌升级 | 购买经验书、卡牌升级 | 100/499.14 | 100% | 0.032/0.054 | 99.48% | 30% | 88% |
| 卡牌升级 | 购买经验书、卡牌升级 | 200/849.7 | 100% | 0.034/0.056 | 99.46% | 50% | 90% |
| 卡牌升级 | 购买经验书、卡牌升级 | 300/791.4 | 100% | 0.038/0.064 | 99.38% | 51% | 91% |

测试指令与Jmeter测试报告对照表

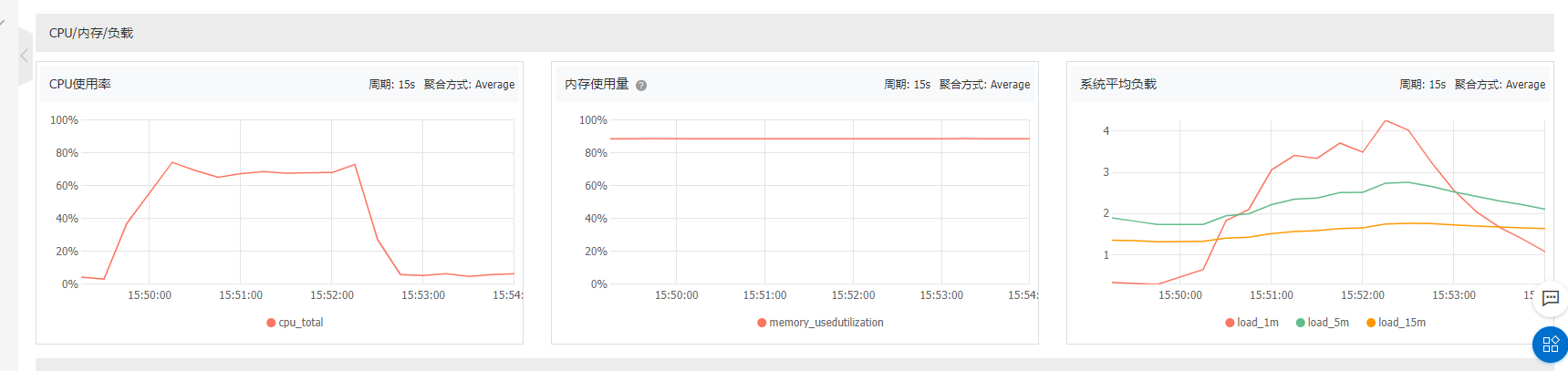
|  |  |  |
| --- | --- | --- |
| **压测任务** | **压测指令** | **Jmeter测试报告** |
| 创建角色与登录/100线程 | sh jmeter -n -t INRegistAndLogin.jmx -l INRegistAndLogin.jtl -e -o /home/develop/jmeter/bin/report -JthreadNum=100 -JloopNum=100 -JrampupTime=100 | report（20） |
| 创建角色与登录/200线程 | sh jmeter -n -t INRegistAndLogin.jmx -l INRegistAndLogin.jtl -e -o /home/develop/jmeter/bin/report -JthreadNum=200 -JloopNum=100 -JrampupTime=150 | report（21） |
| 创建角色与登录/300线程 | sh jmeter -n -t INRegistAndLogin.jmx -l INRegistAndLogin.jtl -e -o /home/develop/jmeter/bin/report -JthreadNum=300 -JloopNum=100 -JrampupTime=250 | report（22） |
| 创建角色与登录/400线程 | sh jmeter -n -t INRegistAndLogin.jmx -l INRegistAndLogin.jtl -e -o /home/develop/jmeter/bin/report -JthreadNum=400 -JloopNum=100 -JrampupTime=250 | report（23） |
| 竞技场战斗/100线程 | sh jmeter -n -t newAccountScript/INArenaBattle.jmx -l INArenaBattle.jtl -e -o /home/develop/jmeter/bin/report -JthreadNum=100 -JloopNum=50 -JrampupTime=100 | report（9） |
| 竞技场战斗/200线程 | sh jmeter -n -t newAccountScript/INArenaBattle.jmx -l INArenaBattle.jtl -e -o /home/develop/jmeter/bin/report -JthreadNum=200 -JloopNum=25 -JrampupTime=100 | report（10） |
| 竞技场战斗/300线程 | sh jmeter -n -t newAccountScript/INArenaBattle.jmx -l INArenaBattle.jtl -e -o /home/develop/jmeter/bin/report -JthreadNum=300 -JloopNum=25 -JrampupTime=100 | report（11） |
| 竞技场战斗/400线程 | sh jmeter -n -t newAccountScript/INArenaBattle.jmx -l INArenaBattle.jtl -e -o /home/develop/jmeter/bin/report -JthreadNum=600 -JloopNum=25 -JrampupTime=100 | report（12） |
| 创建联盟/100线程 | sh jmeter -n -t newAccountScript/INLeagueCreate.jmx -l INLeagueCreate.jtl -e -o /home/develop/jmeter/bin/report -JthreadNum=100 -JloopNum=100 -JrampupTime=200 | report（15） |
| 卡牌升级/100线程 | sh jmeter -n -t newAccountScript/FNCardLevelUp.jmx -l FNCardLevelUp.jtl -e -o /home/develop/jmeter/bin/report -JthreadNum=100 -JloopNum=100 -JrampupTime=200 | report（17） |
| 卡牌升级/200线程 | sh jmeter -n -t newAccountScript/FNCardLevelUp.jmx -l FNCardLevelUp.jtl -e -o /home/develop/jmeter/bin/report -JthreadNum=200 -JloopNum=100 -JrampupTime=200 | report（18） |
| 卡牌升级/300线程 | sh jmeter -n -t newAccountScript/FNCardLevelUp.jmx -l FNCardLevelUp.jtl -e -o /home/develop/jmeter/bin/report -JthreadNum=300 -JloopNum=100 -JrampupTime=200 | report（25） |

## 服务端资源情况

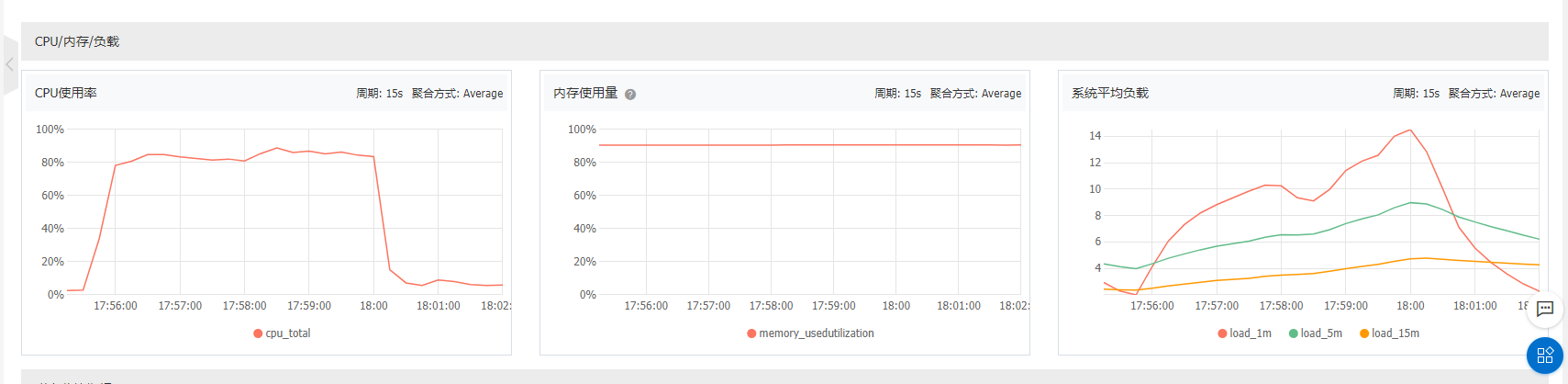
* 创建角色与登录/100线程



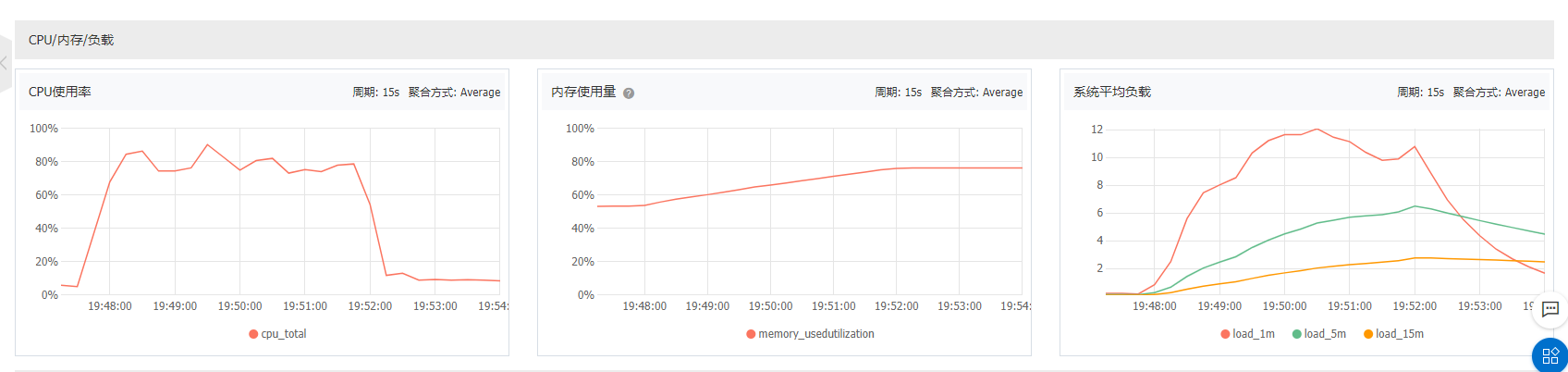
* 创建角色与登录/200线程



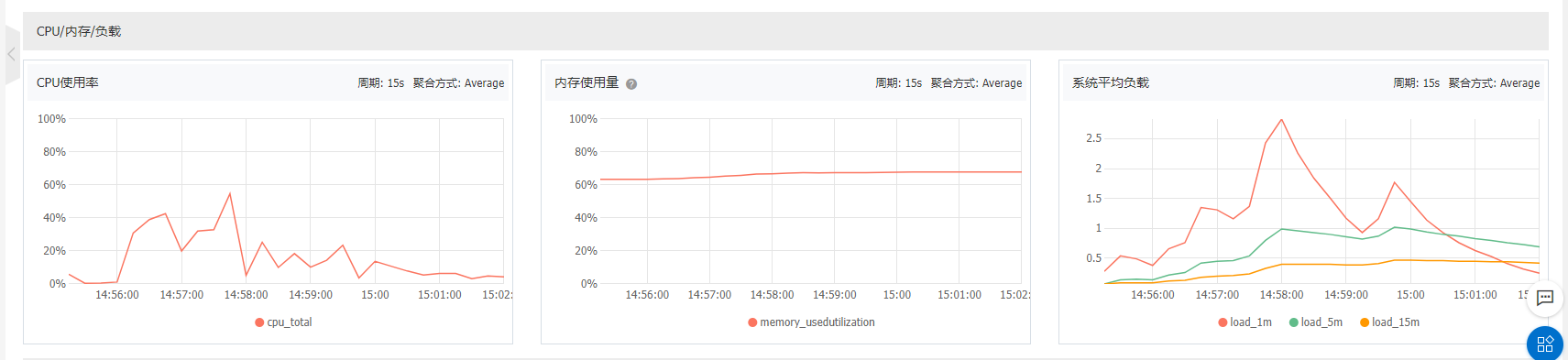
* 创建角色与登录/300线程



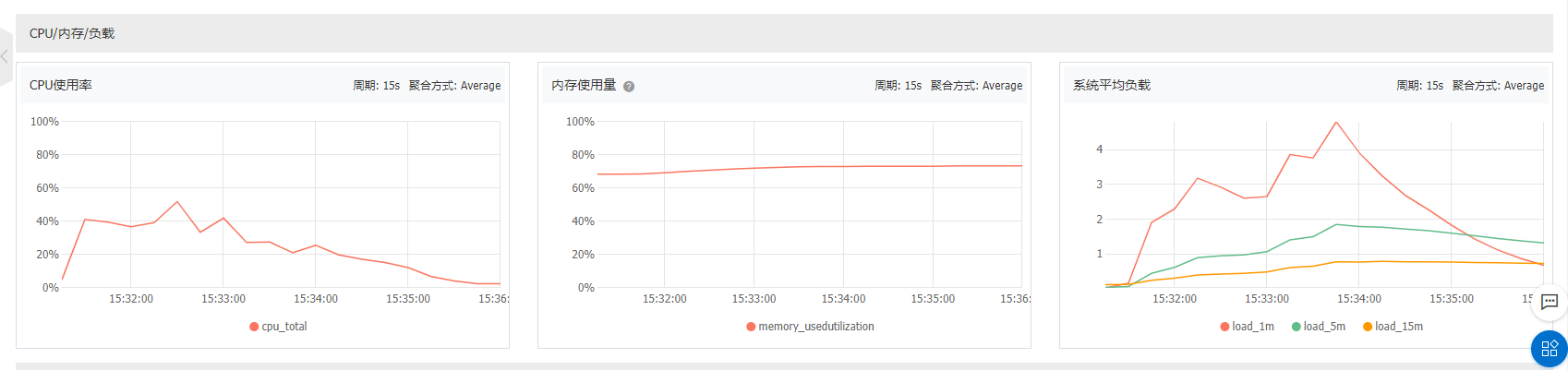
* 创建角色与登录/400线程



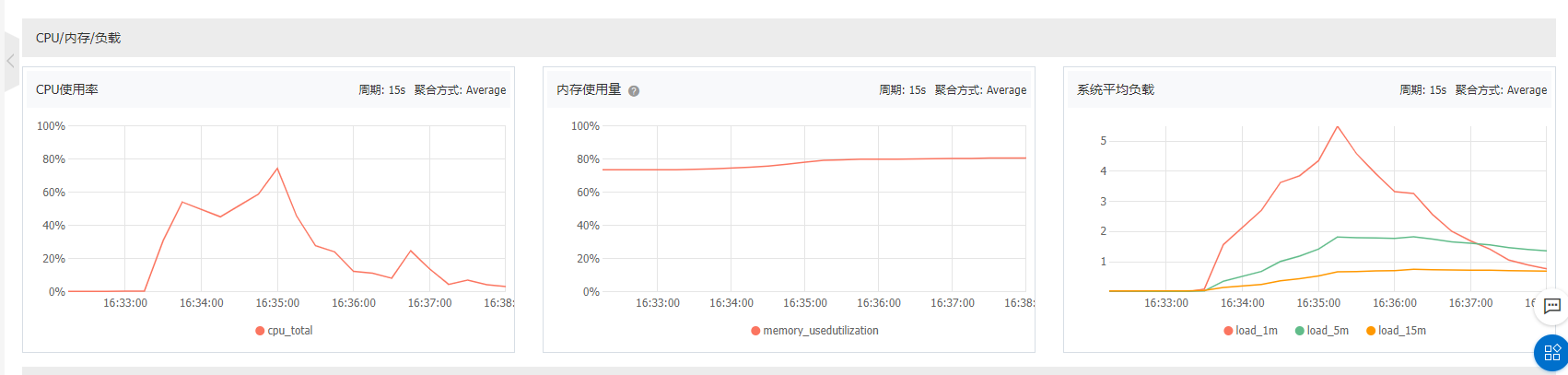
* 竞技场战斗/100线程



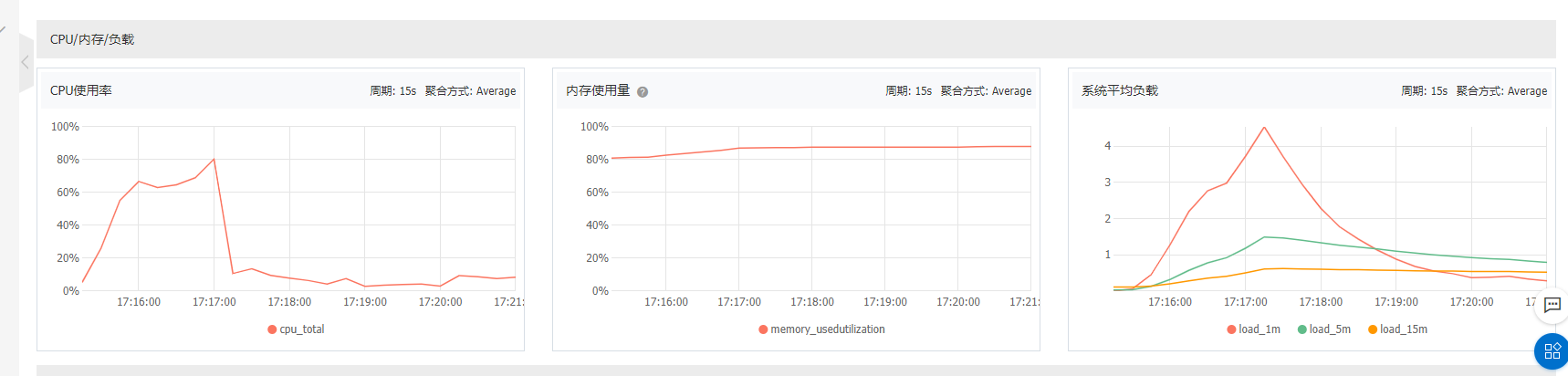
* 竞技场战斗/200线程



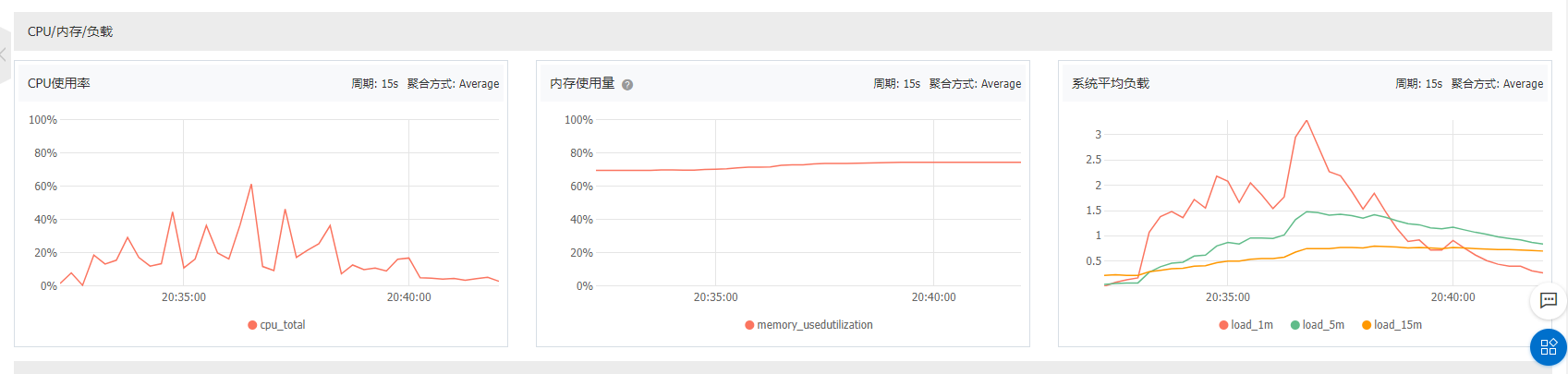
* 竞技场战斗/300线程



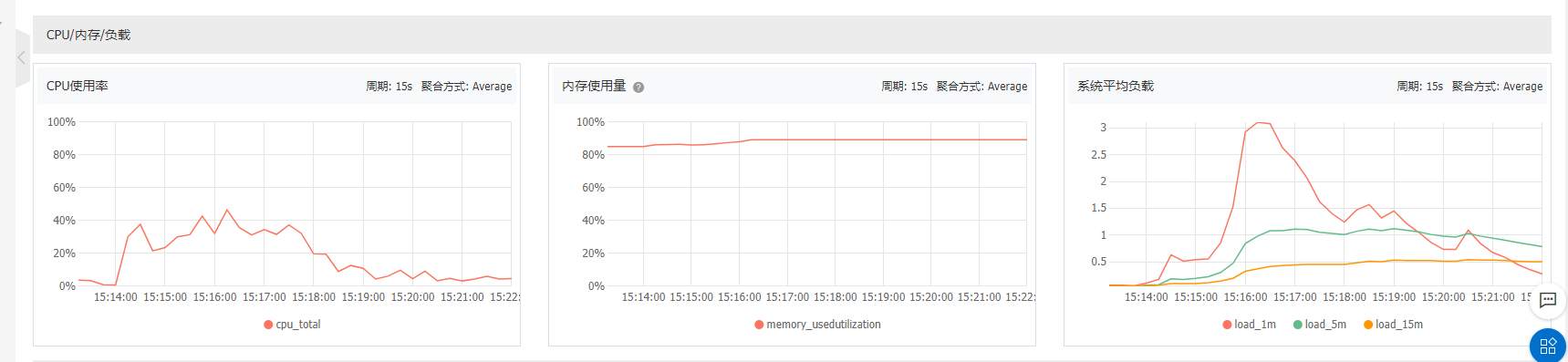
* 竞技场战斗/400线程



* 创建联盟/100线程



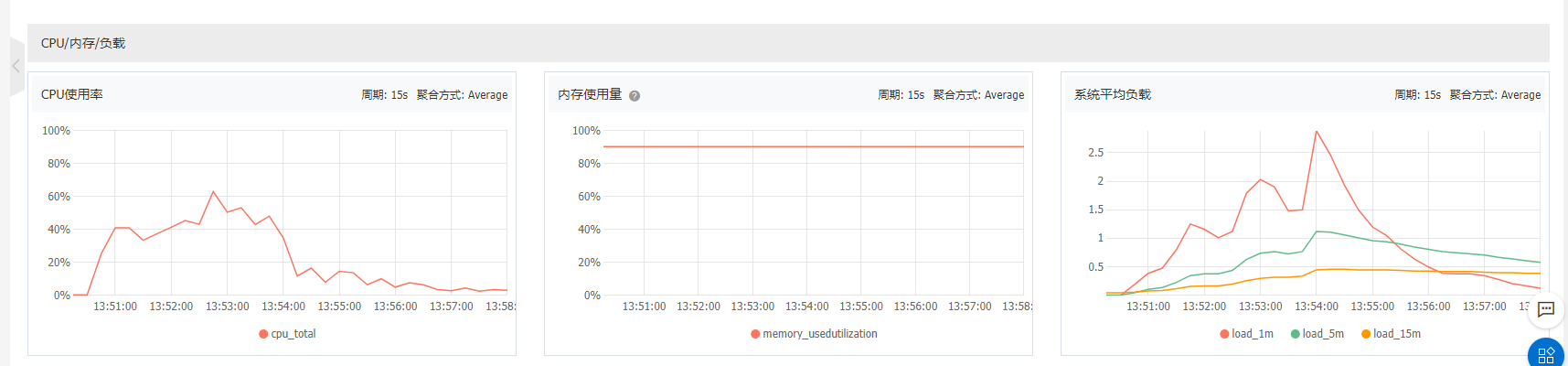
* 卡牌升级/100线程



* 卡牌升级/200线程



* 卡牌升级/300线程



## 客户端资源情况

根据全部测试结果分析，压测客户端暂不存在cpu超过80%等性能瓶颈问题，结论为客户端不需要扩容就可以满足全部测试，所以本次客户端资源只展示其中一次压测数据，如下

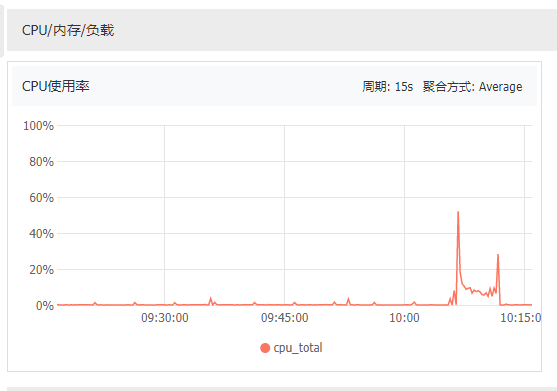


图3.1.1

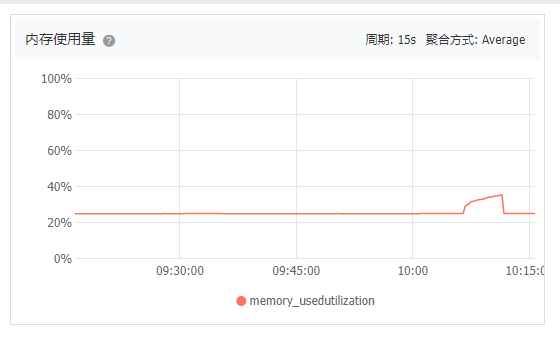


图3.1.2

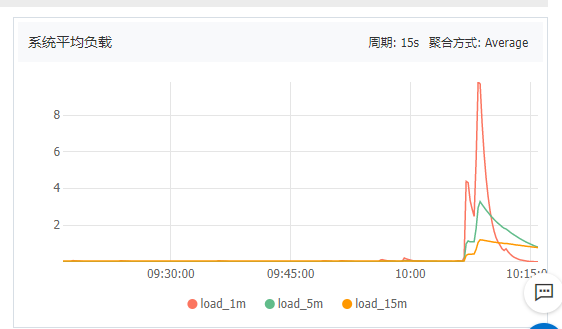


图3.1.3



图3.1.4

## 数据库资源情况

由于本次压测是大部分基于新用户的，全部数据先写入redis，等压测完成后才同步数据库，所以数据库方面不是瓶颈，只列其中某一次数据库的情况供参考，如下

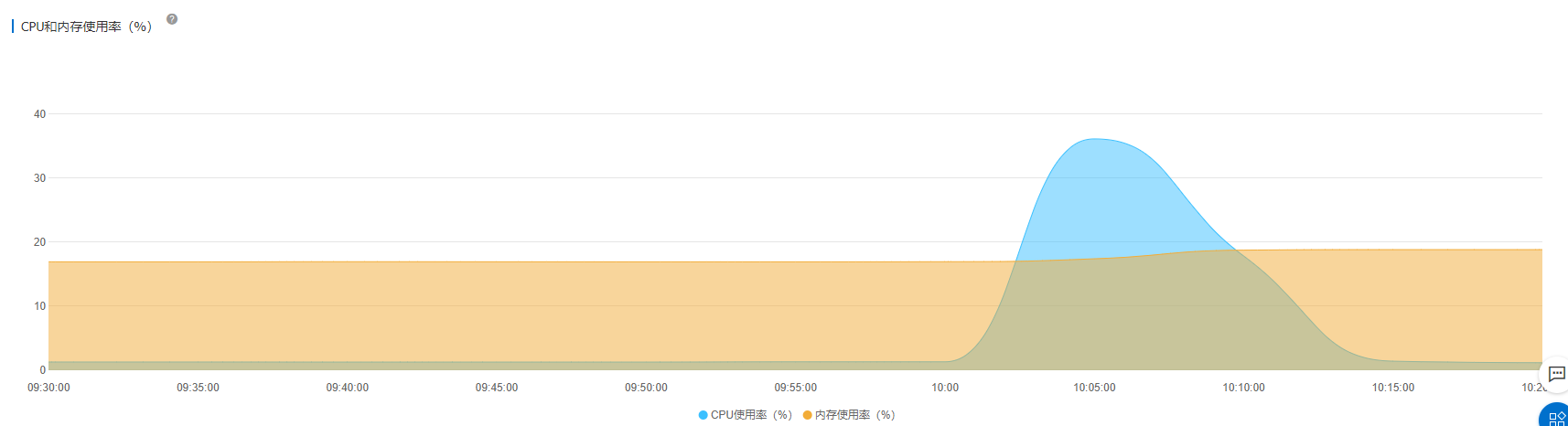


图3.3.1

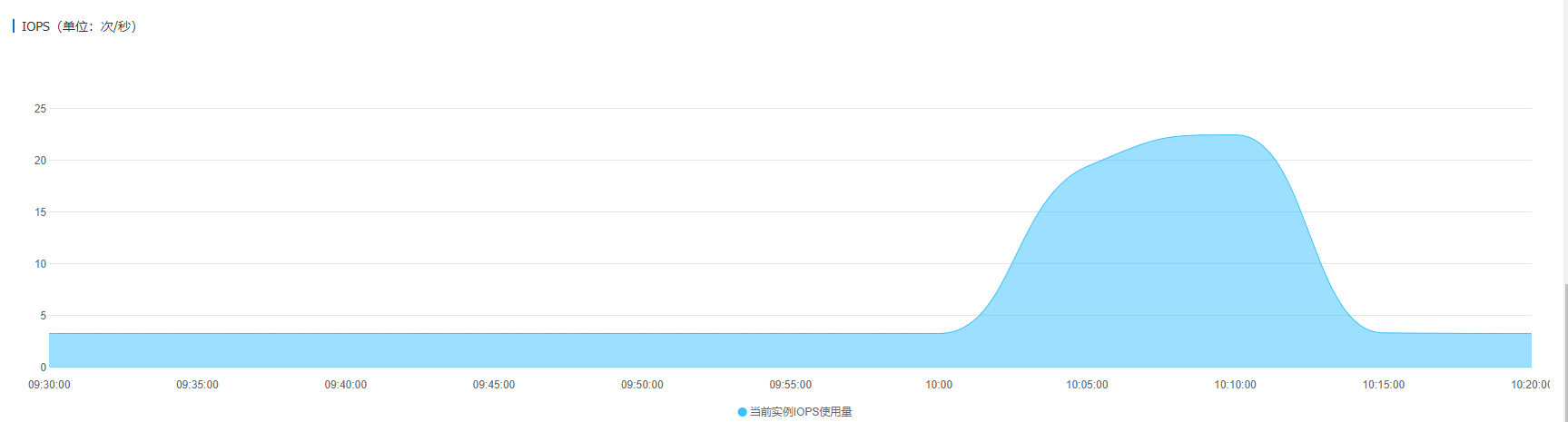


图3.3.2

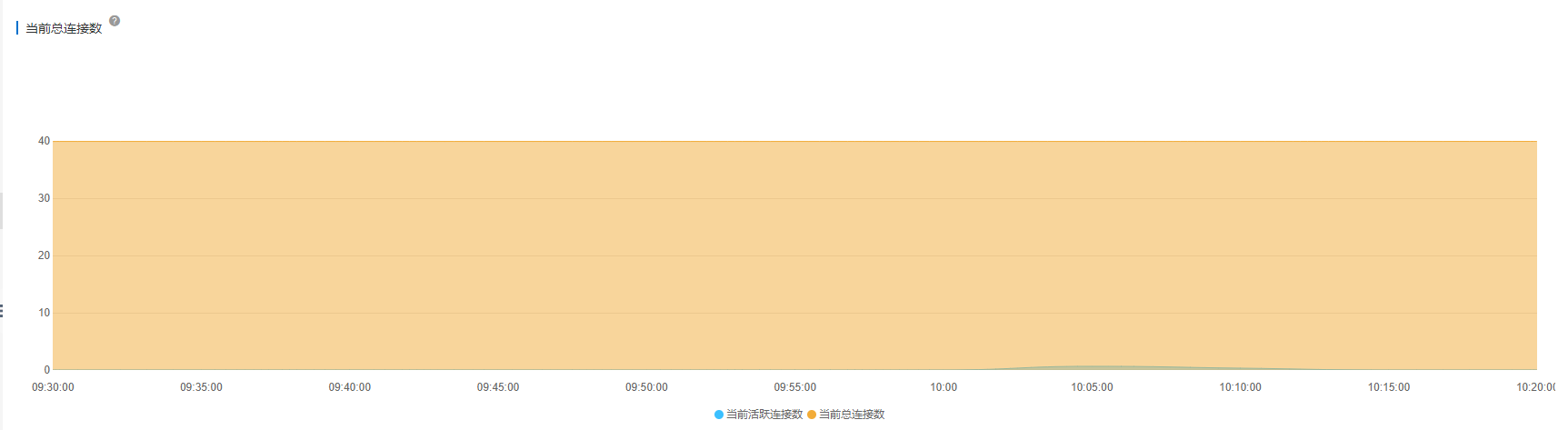


图3.3.3

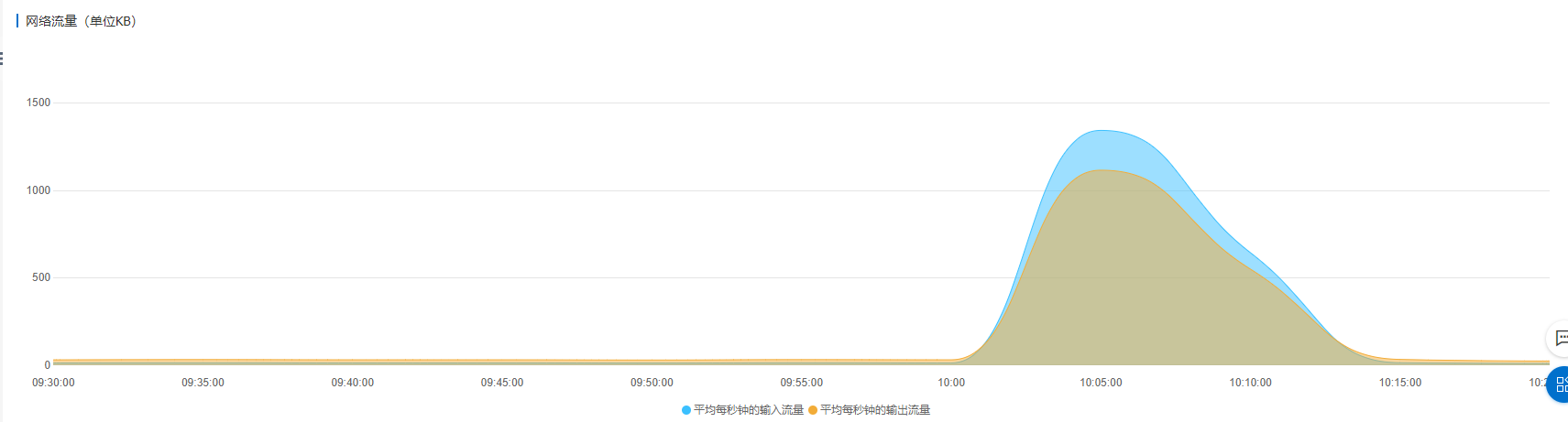


图3.3.4

## 测试结果分析

下面是分析过程：

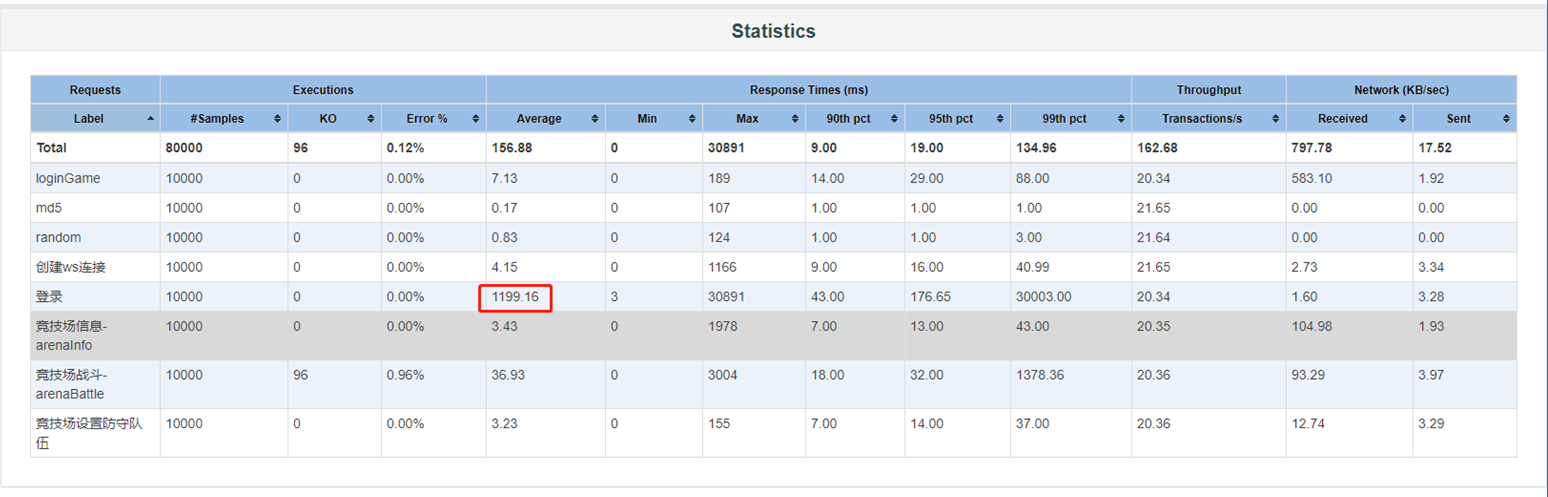


图4

* 从图4可以看到，老用户竞技场压测结果里登录时间超过1秒，后续无法达到压测预期效果，经过开发同事分析与排查，发现是老用户账号数据的缓存命中率极低，造成大量压测都是Redis查不到数据后再去查询数据库的，加上数据库查询数据较慢，综合来说，这是严重级别的性能问题

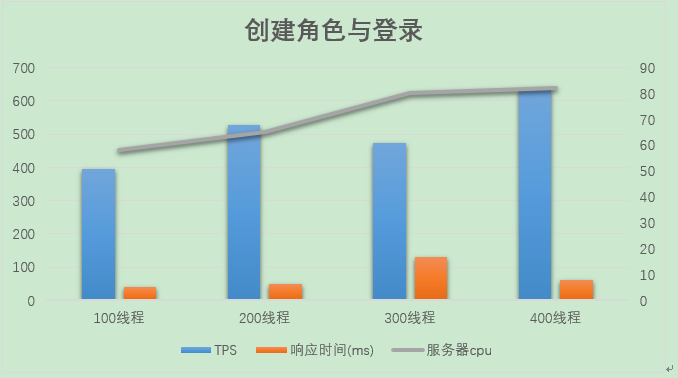


图1

* 从图1分析可以分析出登录与创角模块的最大TPS约为640，并发数量为400，再加线程数量意义不大，因为服务器cpu使用率已超过80%，另外后续压测产生的错误基本都是由于用户角色名称非法（错误码30008），错误数量占压测样本总数比例均不超过0.5%，可以视为正常情况

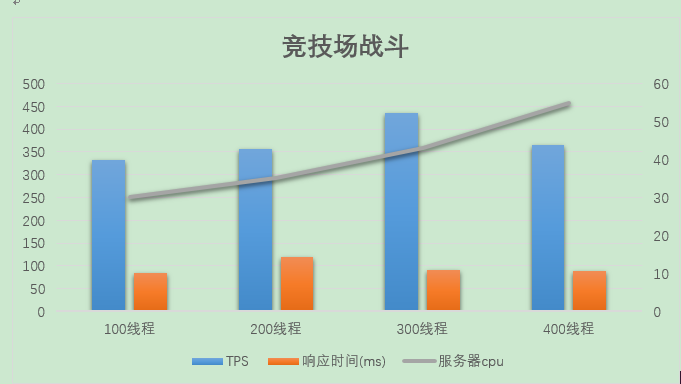


图2

* 从图2分析可以分析出300个并发要比400个并发的TPS数据要高，由于S项目是按照服务器进行业务逻辑等运算，而客户端只是负责渲染与极少的运算逻辑，所以进行相应的竞技场战斗的优化，目标是400线程情况下，TPS超过500，并且服务器cpu使用率低于80%，这是一般的性能问题

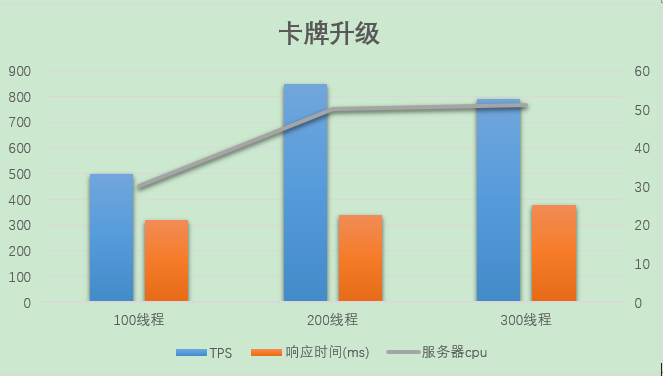
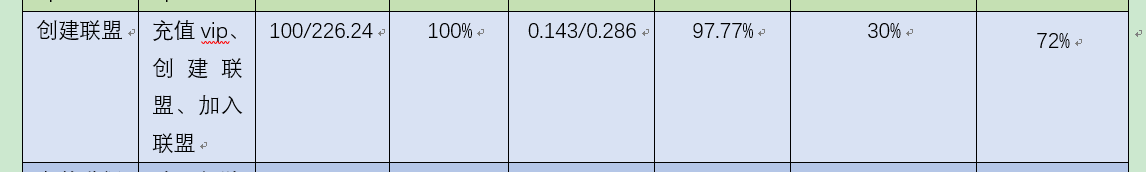


图3

* 从图3可以看到，卡牌升级时200线程与300线程综合情况基本一致，存在一定的优化空间，再根据实际情况来看，如玩家点击经验书，然后会有一个动画显示经验书转换英雄卡牌的经验增加，暂定为一个建议性的优化点



* 从【创建联盟】压测结果来看，暂时来说还是比较合理的，因为当初设想是进行模拟充值成为VIP，才能创建联盟的，由于暂时充值VIP只是一个GM指令，还有每个区服实际联盟数量估算是在10000下以内，所以本次只是初步判断一下100并发用户时，TPS达到226，后续等待真正的充值VIP接口再加大线程进行压测



图5

* 从服务器环境那章节可以看到，三大服务均部署在同一台机子，区服、网关与经分系统的内存分配比例为16：4：1，经过大量压测后，并且根据铁血王师的服务器架构来看，建议使用图5的服务器设计来做，可以应对更大量的游戏玩家

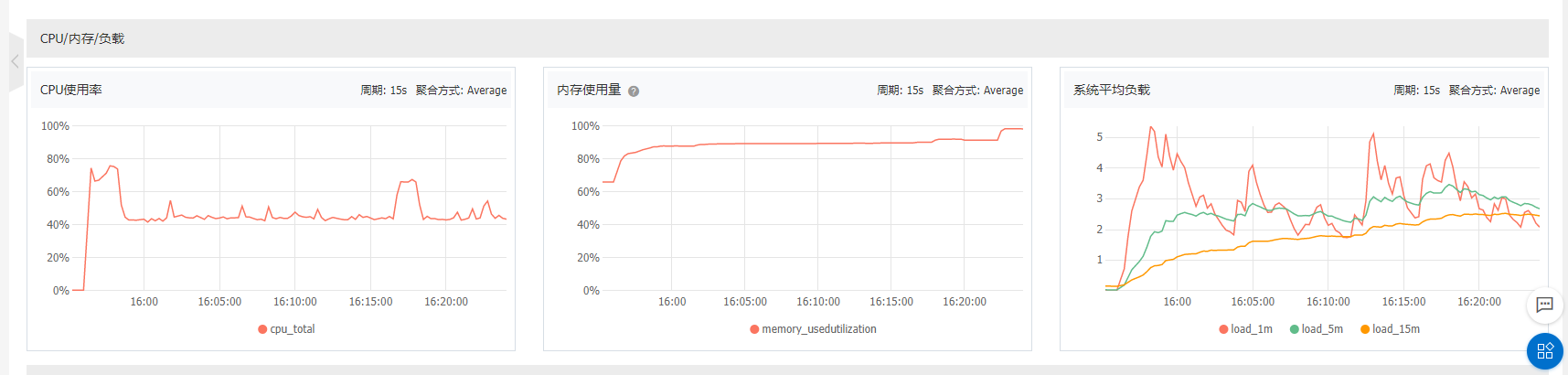


图6

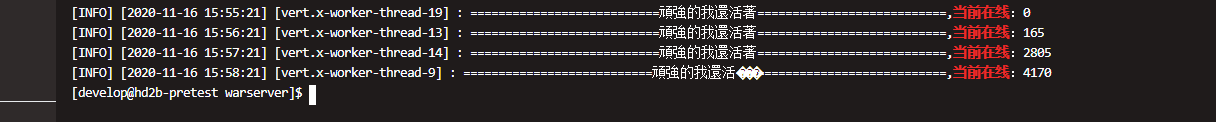


图7

* 两台Jmeter压测客户端每台2500线程进行测试用户在线，其中如图7结果，可以估算服务器可以达到超过5000用户在线的要求，因为压测客户端只有两台数量不足情况下，估算理论峰值应该是8000用户在线。

测试指令：sh jmeter -n -t INRegistAndLoginAndOnline.jmx -l INRegistAndLoginAndOnline.jtl -e -o /home/develop/jmeter/bin/report -JthreadNum=3000 -JloopNum=1 -JrampupTime=50

针对上述问题的情况，建议做如下优化措施：

* **优化数据库查询速度，数据库优化如检查数据库索引、 慢SQL与甚至重建索引等，代码检查是否存在jdbc的DAO层优化**
* **优化战斗类型的代码逻辑**
* **调整缓存容量，尽量保证老用户在某时间段内的数据存放在缓存**

另外，压试完毕后，CPU 使用率较快下降，空闲内存上涨较为缓慢，同时系统还是可以正常处理所有请求，如下示例：

|  |  |  |  |
| --- | --- | --- | --- |
| **服务** | **开始时间** | **结束时间** | **持续时间** |
| 压测服务器 | 17:42:45 | 17:44:45 | 2分钟 |
| 消化队列 | 17:44:45 | 18:04:45 | 20分钟 |
| 压测数据库 | 17:45:00 | 18:20:00 | 35分钟 |

当压测完成后，第一阶段为消化队列，如200并发循环100次后，消化队列单纯种要处理20分钟，消化完成后再写进行批量写入数据库，这阶段暂时没有出现cpu100%的情况，基本都是15%~25%cpu左右。

此结果说明，服务器硬件与系统性能优良，暂时认定可以承受来自系统处理的业务场景设计。

# 压测总结

从压测结果看，共有 1 个严重性问题、1 个一般性问题、1 个建议性问题，共计 3 个问题。

另外经过多轮的压测，S项目优化了如下内容：

1. 使用阿里云的redis，把数据放到缓存，响应时间由1秒缩短到80ms
2. 调整jvm与磁力数等系统参数，对注册账号、websocket创建、登录与新建用户角色得到优化，尤其是新建用户角色接口，从原来的1秒响应时间，降到60ms
3. 删除部分的反射代码，整体CPU使用率从使用反射时的接近100%降至不使用反射时的80%，CPU使用率下降20%
4. 优化创建联盟数据处理，先作公共数据定期保存，后续再放数据库，TPS从140多上升到220，性能得到一定的提升

# 附录

## Jmeter脚本

## Jmeter压测报告zip