**接口性能测试**

## 测试目的

针对entry\_task中自己用python语言实现的接口进行性能测试， 并通过对测试结果的分析，得到对接口性能的评估；

## 测试工具

使用jmeter进行测试。

## 测试环境

接口服务和jmeter 5.0部署在MacBook Pro本地环境

## 测试对象

## <http://127.0.0.1:8080/shopee/test?a=2&b=shm>

## 测试场景

安装启动Jmeter，使用非 GUI 模式运行 JMeter 对接口进行性能测试，并生成测试报告（见文件夹report里面的html文件）

测试时长为10分钟

测试并发线程分别为1，5，10

## 测试结果分析

### 1、jmeter测试聚合报告数据如下：

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Threads** | **Average** | **Min** | **Max** | **Error %** | **Throughput** | **Received KB/sec** | **Sent KB/sec** |
| 1 | 0.63 | 0 | 5773 | 0.00% | 1521.43 | 312.01 | 206.52 |
| 5 | 2.33 | 0 | 1737 | 0.00% | 2113 | 433.43 | 286.89 |
| 10 | 13.41 | 0 | 76751 | 0.07% | 722 | 149.74 | 97.93 |

### 2、表头说明：

*Label - 如图，在不勾选"Include group name in label?"复选框的情况下，为请求取样器的名称，否则为“请求取样器所在线程组:请求取样器名称”*

*# Samples - 用同一个请求取样器，发送请求的数量(注意：该值是不断累计的)。比如，10个线程数设置为10，迭代10次，那么每运行一次测试，该值就增加10\*10=100*

*Average - 默认情况下是单个Request的平均响应时间，当使用了Transaction Controller 时，也可以以Transaction为单位显示平均响应时间*

*Min - 针对同一请求取样器，请求样本的最小响应时间*

*Max - 针对同一请求取样器，请求样本的最大响应时间*

*Error % - 出现错误的请求样本的百分比*

*Throughput - 吞吐量以“requests/second、requests /minute、requests /hour”来衡量。 时间单位已经被选取为second，所以，显示速率至少是1.0，即每秒1个请求。 当吞吐量被保存到CVS文件时，采用的是requests/second，所以30.0 requests/second 在CVS中被保存为0.5*

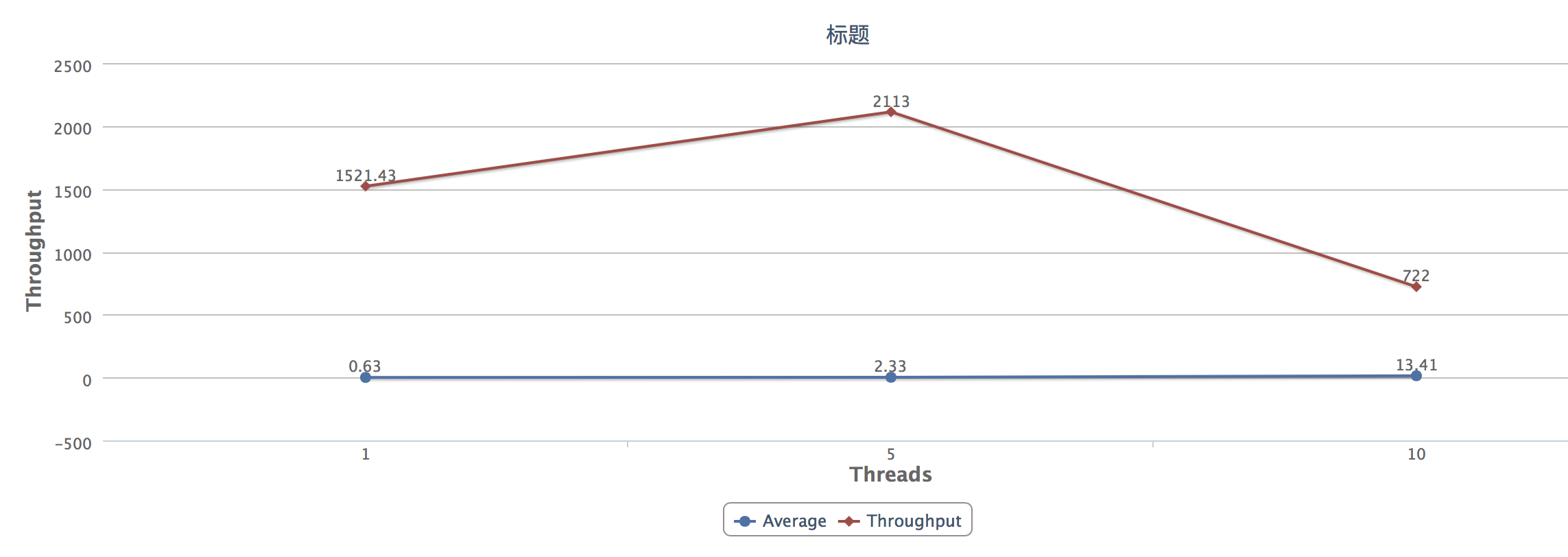
*Kb/sec - 以Kilobytes/seond来衡量的吞吐量*

#### *注意：*

*1>.无特别说明，以上时间的单位均为ms*

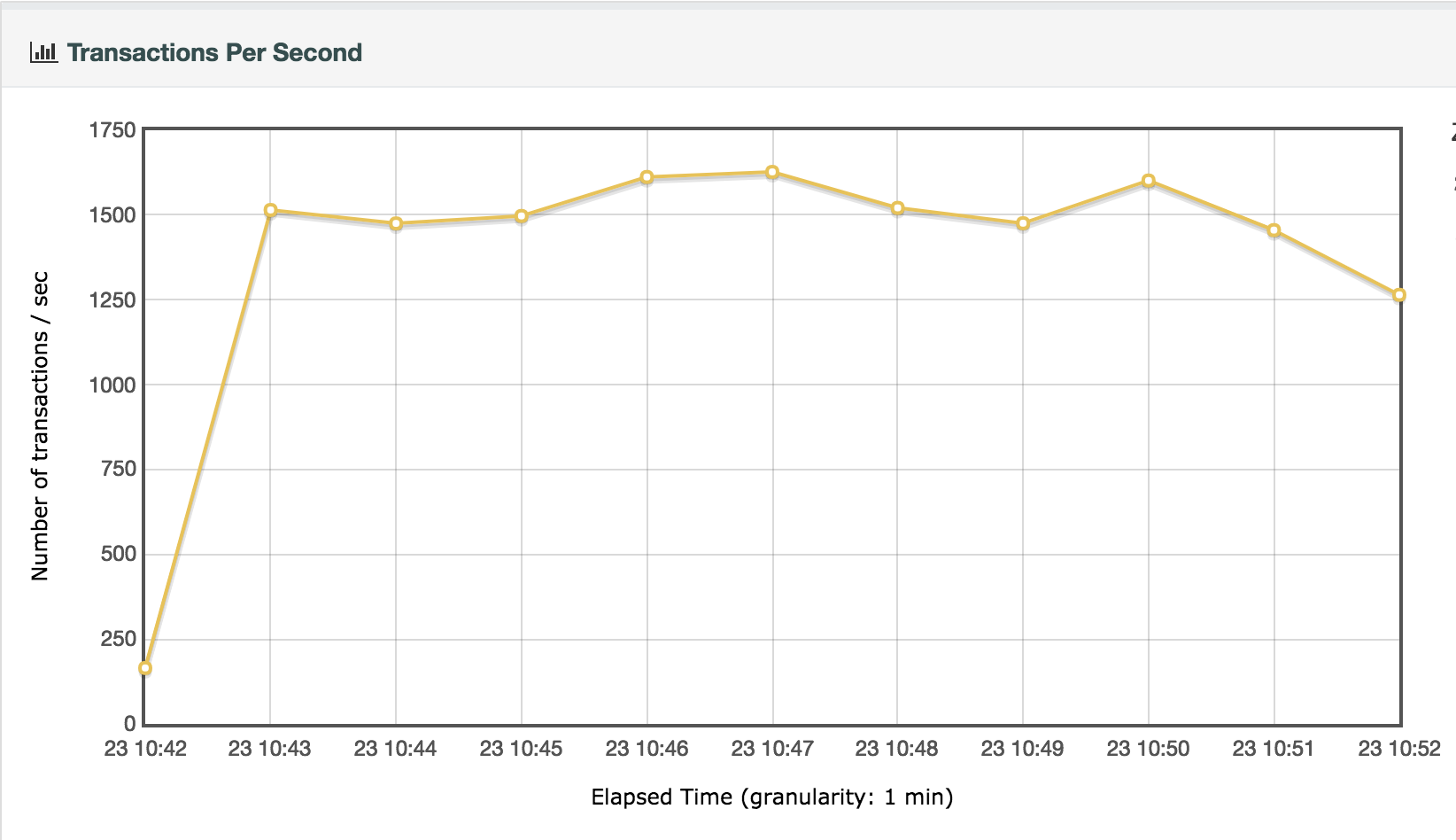
*2>.请求响应时间指的是从client端发出请求到得到响应的整个时间*

### 3、随着并发数增加，总体的Throughput，Average与之关系图如下：

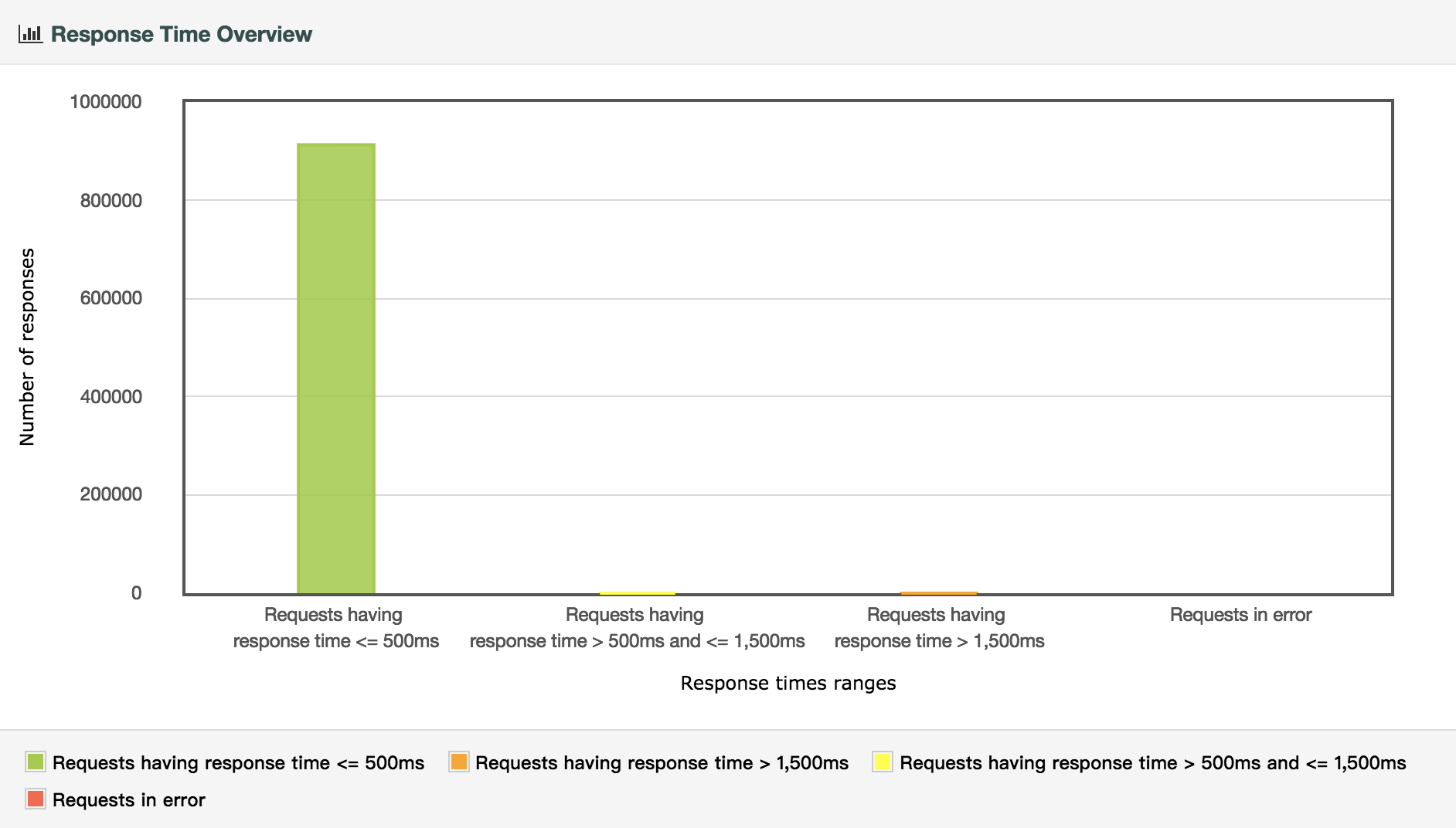


#### 3-1、当线程并发数为1时，

TPS随着时间的变化曲线如下：

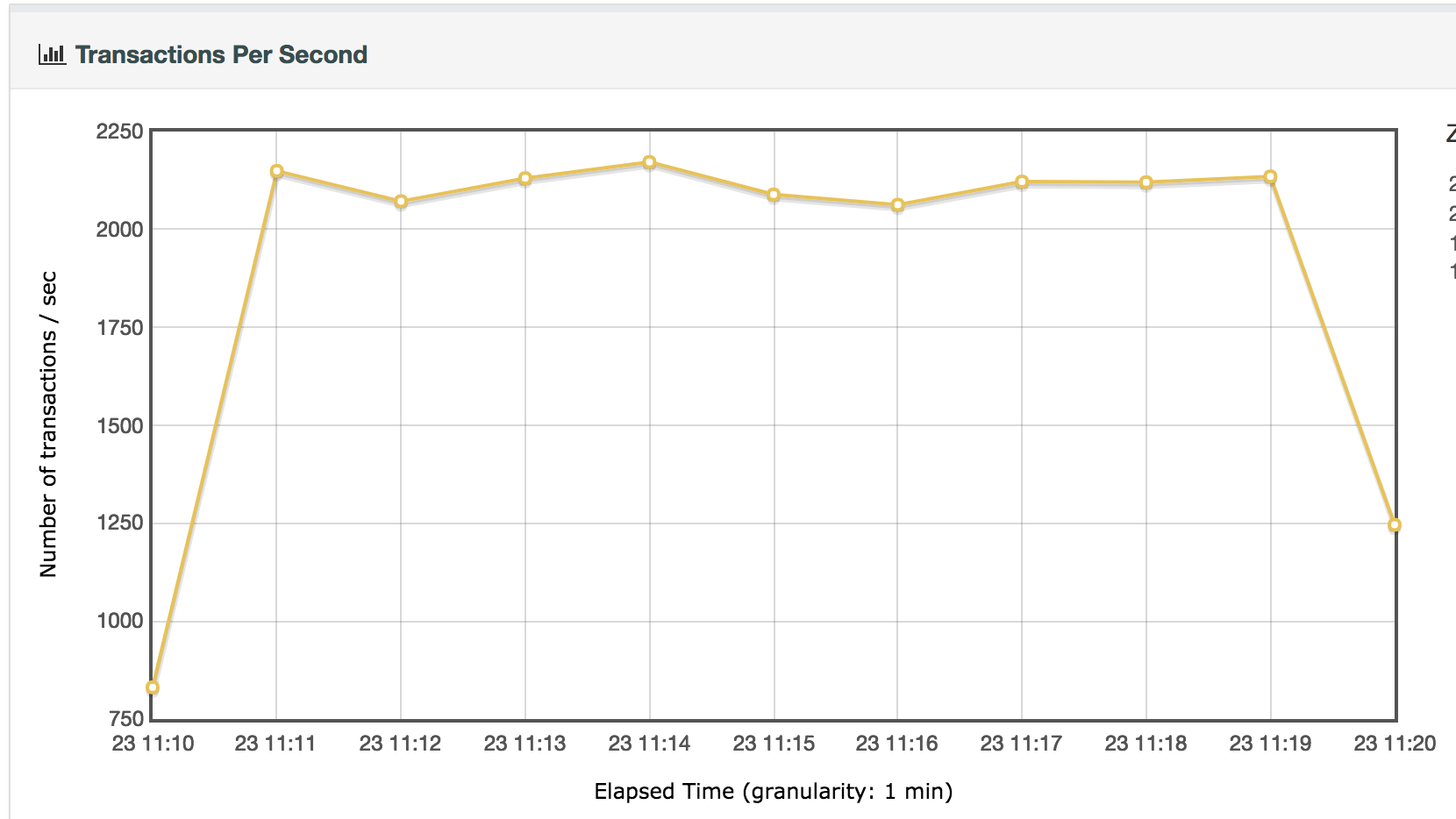


响应时间分布如下：

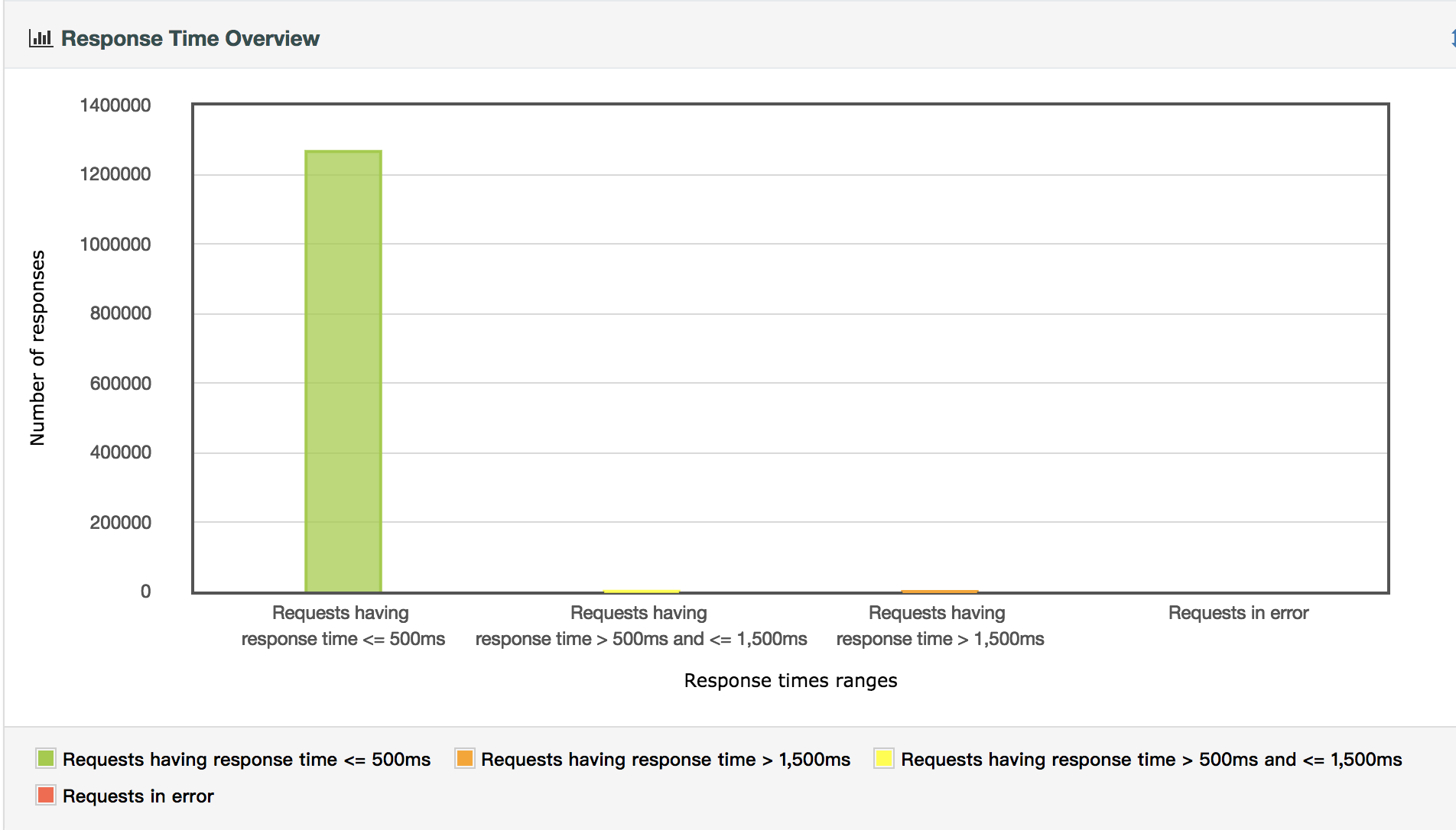


#### 3-2、当线程并发数为5时，

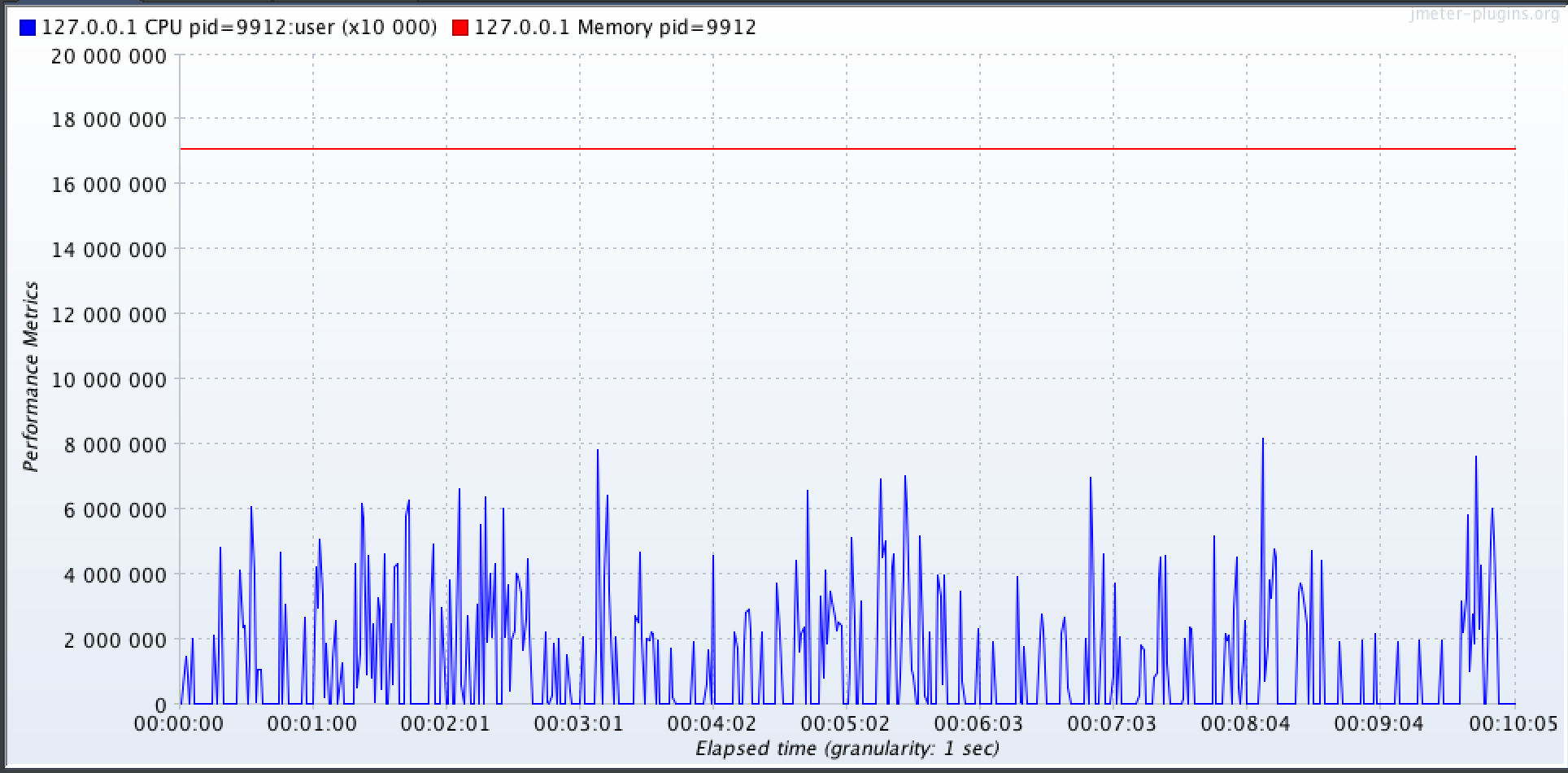
TPS随着时间的变化曲线如下：



响应时间分布如下：



CPU和内存随着时间对变化



#### 3-3、当线程并发数为10时，

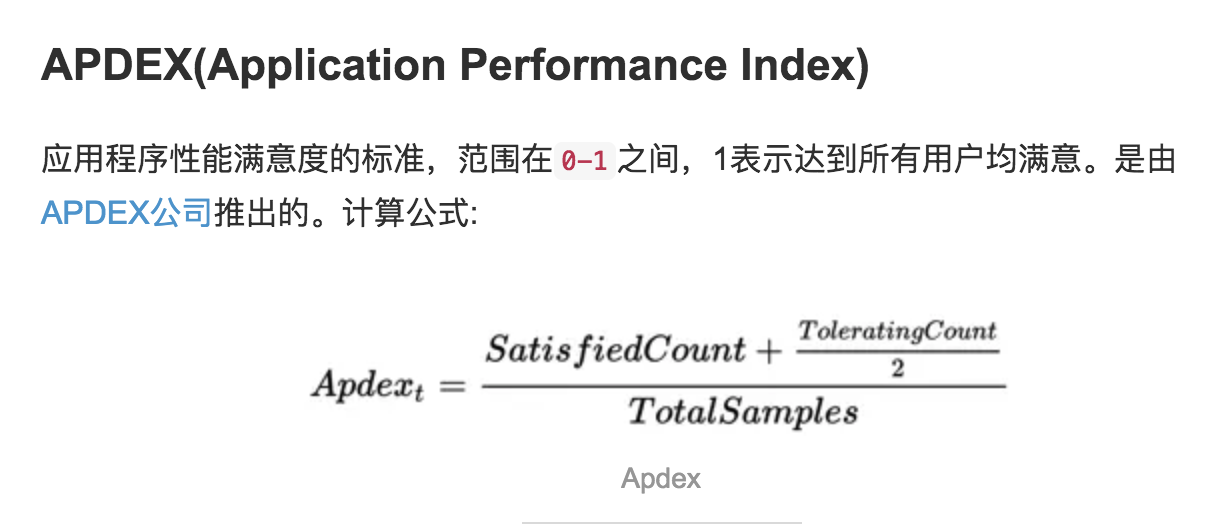
## 

## 测试总结

1、根据测试报告（/reports/threadis1&threadis5中的html文件）结果分析可知：

1. 当线程并发数不大于5时：

随着并发数增加，此接口的TPS呈增长趋势，最大达到2100 左右，响应时间也呈增大趋势，错误率为0%，满意度（APDEX）为1



且，CPU最大不超过80%，更多时候保持在60%以下，内存状态持续稳定，比较理想

1. 当线程并发数大于5时：

随着并发数增加，此接口的TPS呈迅速下降趋势，当线程并发数为10时，TPS降低为720左右，响应时间也快速呈增大趋势，错误率开始大于0%，满意度（APDEX）开始小于1，说明此接口性能在并发为5左右时已经达到瓶颈

综上所述，此接口的性能状况如下：

TPS最高约为2100，此时的响应时间基本都小于500ms，CPU最大时未超过80%，内存稳定。

当然，此性能测试的结果不是非常准确，可能受网络带宽，Python语言本身特点或者单机负载能力有限，压测脚本或配置等问题的影响，专业化的测试结果可能需要更多考虑。

PS：由于本次测试对象是python实现的比较简单的http服务器，性能数据仅供参考