**接口性能测试**

## 测试对象

针对entry\_task中自己用python语言实现的接口进行性能测试

## 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

## 测试场景

安装启动Jmeter，对接口分别以1、5、10、20、50、80、100、150、200线程并发调用，每次循环100次；然后统计性能数据

### 测试结果分析

1、jmeter测试聚合报告数据如下：

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Threads** | **Sample** | **Average** | **Min** | **Max** | **Error%** | **Throughput** | **Std.dev.** | **Received  KB/sec** | **Sent  KB/sec** | **Avg.Bytes** |
| 1 | 100 | 0 | 0 | 2 | 0.00% | 877.2 /sec | 0.46 | 235.72 | 92.03 | 210 |
| 5 | 500 | 1 | 0 | 4 | 0.00% | 525.8 /sec | 0.44 | 107.82 | 71.4 | 210 |
| 10 | 1000 | 1 | 0 | 4 | 0.00% | 974.8 /sec | 0.61 | 199.88 | 132.3 | 210 |
| 20 | 2000 | 4 | 0 | 1040 | 0.15% | 971.3 /sec | 39.97 | 201.31 | 131.3 | 213.3 |
| 50 | 5000 | 15 | 0 | 2147 | 0.24% | 1541.3 /sec | 130.86 | 323.97 | 208.72 | 215.2 |
| 80 | 8000 | 20 | 0 | 3147 | 0.53% | 1880 /sec | 166.51 | 406.55 | 253 | 221 |
| 100 | 10000 | 23 | 0 | 4196 | 0.71% | 1889.3 /sec | 192.81 | 415.94 | 254 | 225.4 |
| 150 | 15000 | 81 | 0 | 8101 | 2.75% | 1109.4 /sec | 420.72 | 292.45 | 146 | 269 |
| 200 | 20000 | 397 | 0 | 67895 | 5.04% | 226.4 /sec | 3298.36 | 70.77 | 29.18 | 320.2 |

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

Median - 中位数。表示响应时间本不大于该时间值的请求样本数占总数的50%

90% Line - 表示响应时间不大于该时间值的请求样本数占总数的90%

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，Throughput，Average，Error%与之关系图如下：

1）、线程数和吞吐量关系

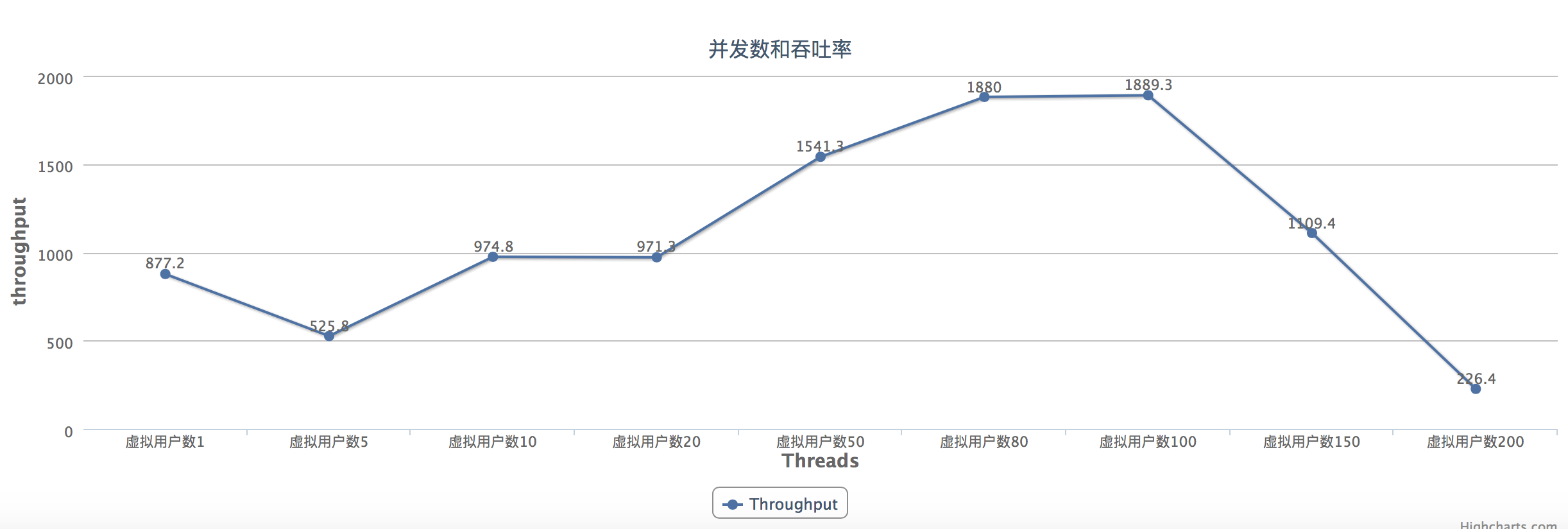


图1

2）、线程数和平均响应时间关系

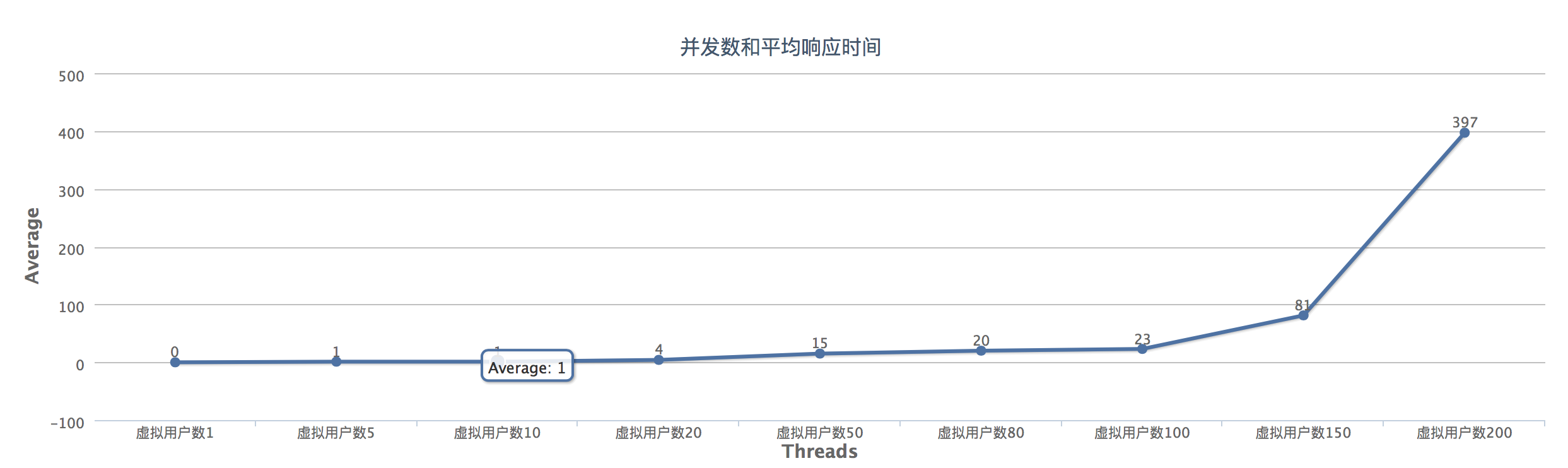


图2

3）、线程数与错误率的关系

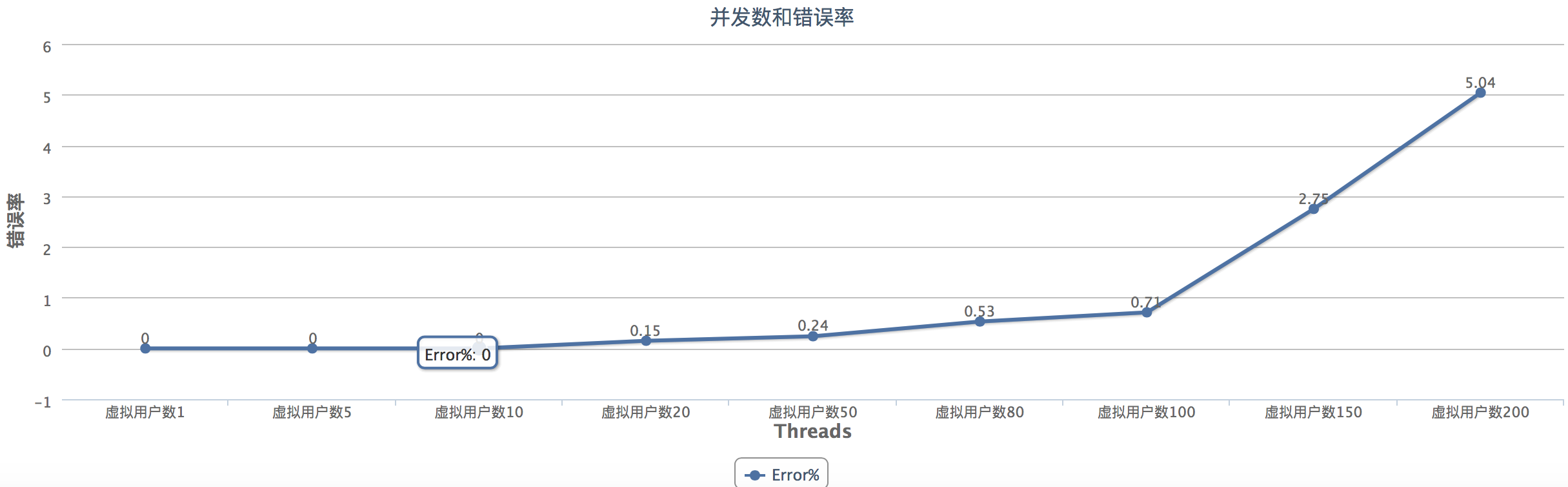


图3

## 测试总结

综上所述，

1 根据聚合报告以及关系图形结果分析

1. 每间隔一秒钟并发的线程数越多，吞吐量先减后增，当线程数在80～100时，吞吐量达到较大值，约为1850/秒左右 （图1）
2. 每间隔一秒钟并发的线程数越多，平均响应时间值呈现增长趋势，在100线程数时开始迅速增长（图2）
3. 由图3可知，当线程数增加到100以后，错误率逐渐开始大于1%，并呈快速增长趋势 （--作为一般惯例，0.5-1.0％在一般应用程序/网站中是可接受的，但如果其任务关键/一些非常重要的应用程序，则错误率具有高优先级）

当然，此性能测试的结果不是非常准确，可能受网络带宽，jemter的本身占用内存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