**春秋航空电商APP**

**性能测试报告**

**（V1.0）**

二〇一八年五月

文档信息

|  |  |  |  |
| --- | --- | --- | --- |
| **文档名称** | 春秋航空电商秒杀性能测试报告 | | |
| **电子文档** | 性能测试报告 | | |
| **文件状态** | □**草 稿** ■ **正式发布** □ **正在修改** | | |
| **编 写 人** | **彭志辉** | **日 期** | 2018年 05月 23 日 |
| **校 对 人** |  | **日 期** |  |
| **审 核 人** |  | **日 期** |  |
| **批 准 人** |  | **日 期** |  |

变更记录

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **变更序号** | **变更原因** | **变更页码** | **变更前版本号** | **变更后版本号** | **更改人** | **批准人** | **生效日期** | **备 注** |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |

目 录

[1. 测试概述 1](#_Toc514837560)

[1.1. 测试目的 1](#_Toc514837561)

[1.2. 名词解译 1](#_Toc514837562)

[1.3. 测试工具描述 1](#_Toc514837563)

[1.4. 测试人员 1](#_Toc514837564)

[2. 测试环境描述 1](#_Toc514837565)

[2.1. 测试环境网络环境 1](#_Toc514837566)

[2.2. 服务器软硬件环境 1](#_Toc514837567)

[3. 性能测试目标需求 1](#_Toc514837568)

[3.1. 性能测试方案 1](#_Toc514837569)

[3.2. 性能测试目标 1](#_Toc514837570)

[4. 性能测试场景测试 1](#_Toc514837571)

[4.1 测试场景: 12个线程的请求 1](#_Toc514837572)

[4.2 测试场景: 20个线程的请求 1](#_Toc514837573)

[4.3测试场景：100个线程的请求 1](#_Toc514837574)

[4.4测试场景：200个线程的请求 1](#_Toc514837575)

[5. 压力测试总结 1](#_Toc514837576)

# 测试概述

## 测试目的

本次测试的目的是对电商短链接项目进行性能评估，根据单节点服务器的测试结果，对项目正式使用进行数据量化评估，以预期所需服务器数量和配置。

性能测试指标及目标如下：

1. 在线上单节点服务器进行压力测试，获得并发测试后的性能指标——PV量、并发用户数、每秒处理事务数TPS、事务响应时间、事务通过率等；
2. 监控服务器资源利用率——记录被测系统在不同应用场景执行中的CPU、内存利用率，记录网络流量；
3. 对于出现的error、及影响系统平稳运行的性能瓶颈，记录问题，查看服务器日志，逐步定位问题区域，协同开发人员进行问题排查及调优。

## 名词解译

* **系统的响应能力：**

即在各种负载压力情况下，系统的响应时间，也就是从客户端交易发起，到服务器端交易应答返回所需要的时间，包括网络传输时间和服务器处理时间。

* **并发用户数：**

用户通过客户端（如Web浏览器）与应用程序进行交互。用户对应用服务器发送请求，这是用户的会话处于活动状态，即为“在线用户”；当多个类似的用户请求同时访问应用服务器，这就是“并发用户”。

* **应用系统的吞吐率：**

即应用系统在单位时间内完成的交易量，也就是在单位时间内，应用系统针对不同的负载压力，所能完成的交易数量。

* **应用系统的负载能力：**

即系统所能容忍的最大用户数量，也就是在正常的响应时间中，系统能够支持的最多的客户端的数量。

* **TPS：**

即每秒系统能够处理的交易或事务的数量。它是衡量系统处理能力的重要指标。

## 测试工具描述

Apache JMeter，是一种预测系统行为和性能的负载测试工具。通过以模拟上千万用户实施并发负载及实时性能监测的方式来确认和查找问题，Apache JMeter能够对整个企业架构进行测试。通过使用 Apache JMeter，企业能最大限度地缩短测试时间，优化性能和加速应用系统的发布周期。 Apache JMeter是一种适用于各种体系架构的自动负载测试工具，它能预测系统行为并优化系统性能。

本次性能测试工具使用的性能测试软件Apache JMeter3.1，使用Http协议编写测试脚本，模拟使用者发送请求，**通过不同的航线随机2个月内的日期进行航班搜索，价格日历的查询**，进行并发性能测试。

在测试过程中，由Apache JMeter的管理平台Master调用4台测试压力机上的Slave，发起各种场景下的交易请求，并跟踪记录服务器端的运行情况和返回给客户端的运行结果。

## 测试人员

* 压力测试： 王会勇 彭志辉
* 服务器监控： Grafana
* 测试时间： 2018-05-23 19:30

# 测试环境描述

## 测试环境网络环境

生产线上环境配置，单台应用压力测试，数据库保持不变。

1. 应用服务器

10.16.17.21 master

10.16.17.23 slave

## 服务器软硬件环境

1. 客户端Apache JMeter环境描述：
2. Apache JMeter：

负载机： windows server系统的机器，四核，内存6G，硬盘大于20G空间。

IP地址： 10.16.20.231，10.16.20.232，10.16.20.233，10.16.20.233 共4台负载机。

Master机器：10.16.20.230，

# 性能测试目标需求

## 性能测试方案

1. 使用线上机器10.16.17.21加端口31118走内网的方式访问ecold服务；

2.1从12个线程10秒内启动超时时间设置10秒，模拟不同航线不同日期的航班搜索，价格日历。观察TPS、响应时间、内存、CPU利用率等指标；

2.2从20个线程10秒内启动超时时间设置10秒，模拟不同航线不同日期的航班搜索，价格日历。观察TPS、响应时间、内存、CPU利用率等指标

2.3从100个线程10秒内启动超时时间设置30秒，模拟不同航线不同日期的航班搜索，价格日历。观察TPS、响应时间、内存、CPU利用率等指标

2.4从200个线程30秒内启动超时时间设置为60秒，模拟不同航线不同日期的航班搜索，价格日历。观察TPS、响应时间、内存、CPU利用率等指标

## 性能测试目标

1. 在大促销高峰时期，每秒处理的请求量不低于90。
2. 请求成功率为99.99%。
3. 资源使用指标：核心系统各服务器CPU（**80%以下**）、内存占用率（**80%以下**）、数据库服务器（**80%以下**）；Apache JMeter压力机器CPU使用率、内存占用率都不能高于80%。

# 性能测试场景测试

在生产线环境，并发测试采用初始1用户进行测试，然后按照比例递增（1用户）至200，请求数据参数化，用户每次发送的参数不同，当做新用户进行访问。

## 4.1 测试场景: 12个线程的请求

**压力测试：**主要测试单台机器在12个并发下的处理能力。

**数据流向：**EC →GDS。

**测试结果：** 从12个线程10秒内启动超时时间设置10秒，整个测试过程持续5分钟，共请求次数2230次。其中超时的报错占总请求数的2.11%。Cpu持续上线最高达到3.6%，出口宽带最高在1.091M，入口带宽最高在304.3Kb

| **Label** | **#Samples** | **KO** | **Error %** | **Average response time** | **90th pct** | **95th pct** | **99th pct** | **Throughput** | **Received KB/sec** | **Sent KB/sec** | **Min** | **Max** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Total** | **2230** | **47** | **2.11%** | **1619.21** | **5322.40** | **7742.40** | **10008.00** | **7.29** | **67.50** | **7.40** | **66** | **10025** |
| 价格日历 | 1109 | 0 | 0.00% | 242.21 | 93.00 | 1896.50 | 2983.20 | 3.71 | 52.66 | 3.61 | 66 | 4423 |
| 航班查询 | 1121 | 47 | 4.19% | 2981.47 | 7714.60 | 9496.90 | 10011.00 | 3.66 | 16.02 | 3.87 | 716 | 10025 |

## 4.2 测试场景: 20个线程的请求

**压力测试：**主要测试单台机器在20个并发下的处理能力。。

**数据流向：**EC → GDS。

**测试结果：**在20个线程10秒内启动超时时间设置10秒，整个测试过程持续10分钟，共请求次数22489.其中超时的报错占总请求数的1.07%。Cpu最高达到10.02%。出口带宽最高在3.162M，入口带宽最高在951Kb

| **Label** | **#Samples** | **KO** | **Error %** | **Average response time** | **90th pct** | **95th pct** | **99th pct** | **Throughput** | **Received KB/sec** | **Sent KB/sec** | **Min** | **Max** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Total** | **22489** | **240** | **1.07%** | **1062.30** | **2596.80** | **3654.95** | **10003.00** | **18.70** | **174.56** | **19.19** | **65** | **10016** |
| 价格日历 | 11235 | 1 | 0.01% | 93.62 | 83.00 | 85.00 | 147.92 | 9.36 | 132.67 | 9.11 | 65 | 10014 |
| 航班查询 | 11254 | 239 | 2.12% | 2029.34 | 3530.50 | 5066.50 | 10007.00 | 9.36 | 42.17 | 10.09 | 707 | 10016 |

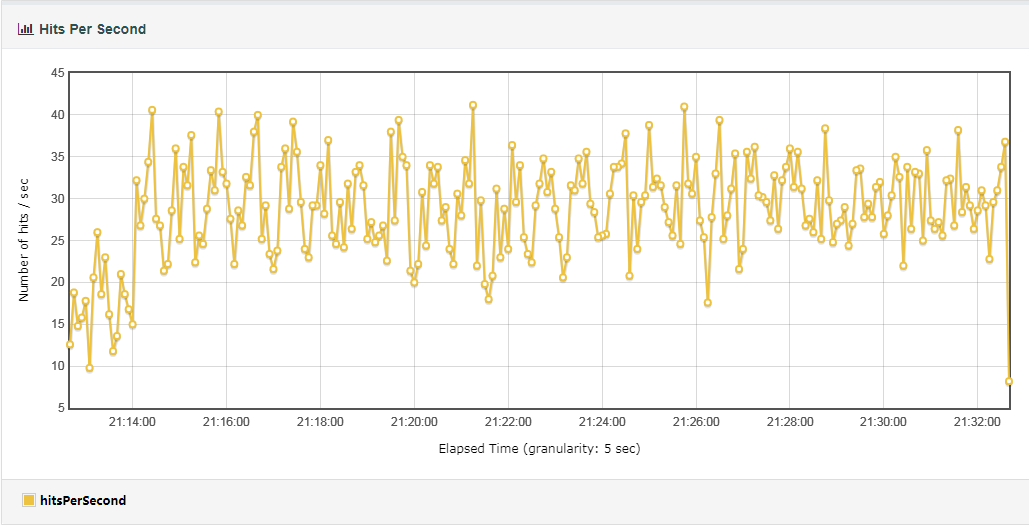
## 4.3测试场景：100个线程的请求

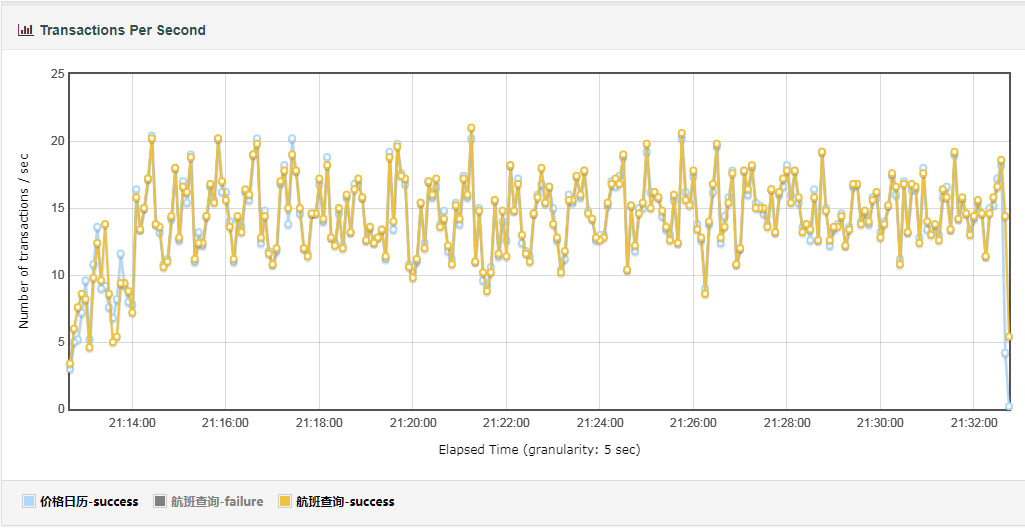
**压力测试：**主要测试单台机器在100个并发下的处理能力。。

**数据流向：**EC → GDS。

**测试结果：**在100个线程10秒内启动超时时间设置30秒，整个测试过程持续20分钟，共请求次数34463.其中超时的报错占总请求数的0.001%。90%的接口请求时间为8秒。Cpu最高达到12.78%。出口带宽最高在3.872M，入口带宽最高在1.227M

| **Label** | **#Samples** | **KO** | **Error %** | **Average response time** | **90th pct** | **95th pct** | **99th pct** | **Throughput** | **Received KB/sec** | **Sent KB/sec** | **Min** | **Max** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Total** | **34463** | **1** | **0.00%** | **2779.44** | **8006.00** | **8695.95** | **16495.94** | **28.55** | **268.17** | **29.64** | **68** | **24573** |
| 价格日历 | 17192 | 0 | 0.00% | 186.83 | 84.00 | 85.00 | 6895.14 | 14.27 | 202.28 | 13.89 | 68 | 13672 |
| 航班查询 | 17271 | 1 | 0.01% | 5360.18 | 8917.00 | 12660.60 | 18305.96 | 14.31 | 66.28 | 15.77 | 797 | 24573 |





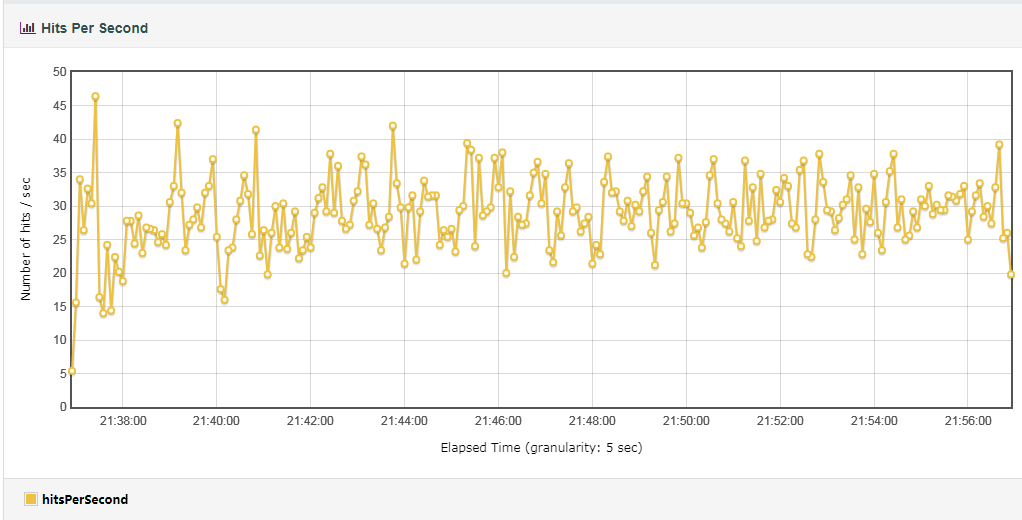
## 4.4测试场景：200个线程的请求

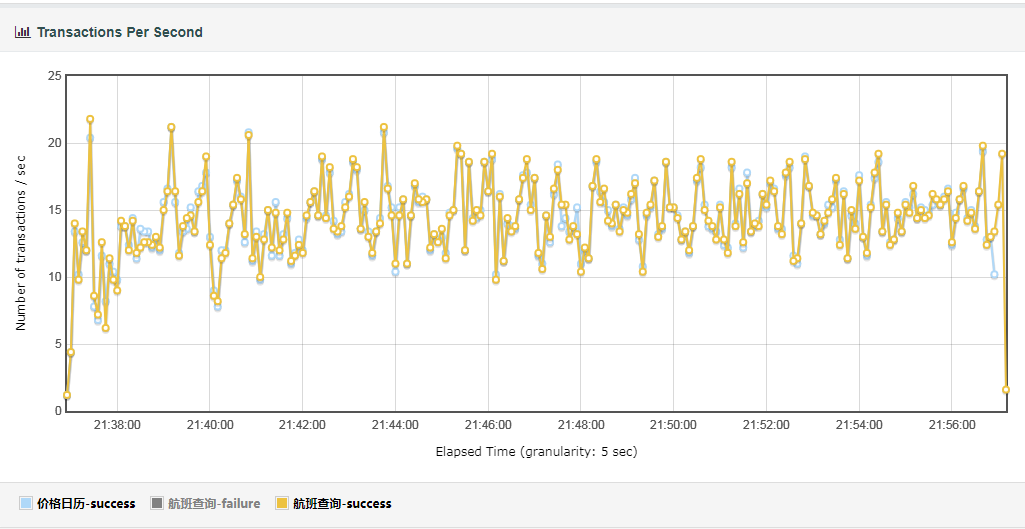
**压力测试：**主要测试单台机器在200个并发下的处理能力。。

**数据流向：**EC → GDS。

**测试结果：**在200个线程30秒内启动超时时间设置60秒，整个测试过程持续20分钟，共请求次数34775.其中超时的报错占总请求数的0.03%。90%的接口请求时间为20秒。Cpu最高达到12.40%。出口带宽最高在3.980M，入口带宽最高在1.283M

| **Label** | **#Samples** | **KO** | **Error %** | **Average response time** | **90th pct** | **95th pct** | **99th pct** | **Throughput** | **Received KB/sec** | **Sent KB/sec** | **Min** | **Max** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Total** | **34775** | **10** | **0.03%** | **6854.40** | **20726.00** | **21926.95** | **44263.64** | **28.64** | **267.75** | **29.72** | **65** | **60013** |
| 价格日历 | 17289 | 0 | 0.00% | 224.02 | 83.00 | 85.00 | 149.10 | 14.41 | 204.10 | 14.02 | 65 | 23625 |
| 航班查询 | 17486 | 10 | 0.06% | 13410.09 | 22901.00 | 34178.65 | 50924.26 | 14.40 | 66.04 | 15.86 | 797 | 60013 |





# 压力测试总结

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 线程 | 持续时间 | 总请求数 | 设置的超时时间 | 90%请求时间 | 超时报错率 | cpu | 内存 | 出口带宽 | 入口带宽 |
| 12个线程 | 5分钟 | 2230 | 10秒 | 5.3s | 2.11% | 3.60% | 持续上升 | 1.091M | 304.3Kb |
| 20个线程 | 10分钟 | 22489 | 10秒 | 2.5s | 1.07% | 10.02% | 持续上升 | 3.162M | 951Kb |
| 100个线程 | 20分钟 | 34463 | 30秒 | 8s | 0.00% | 12.78% | 持续上升 | 3.872M | 1.227M |
| 200个线程 | 20分钟 | 34775 | 60秒 | 20.7s | 0.03% | 12.40% | 持续上升 | 3.980M | 1.283M |

综上所有的测试结果:

* + - 1. 日历的接口耗时远远小于航班查询的接口耗时。
      2. 12个线程的QPS峰值为15.6，20个线程的QPS峰值为31.2。100个线程QPS峰值在41.2,200个线程QPS峰值在46.4。对比100个线程和200个线程的QPS都在30上下徘徊。两个的QPS并没有多大区别。说明服务器的处理速度并没有上升
      3. 12个线程的TPS峰值在8,20个线程的TPS峰值在15.6。100个线程TPS峰值在21和200个线程TPS峰值在22，两个TPS都是稳定在14
      4. 从2.3可以看出 线程越大耗时越多，服务器的处理速度还是一样，cpu压力无法上去。如果用户量大，查询航班的接口耗时会在60s往上，用户体验不好
      5. 在压测过程中内存是持续上升的情况没有出现内存被回收
      6. 压力测试结束后master机器10.16.17.21，salve机器10.16.17.23内存消耗只剩下100M。如果用户量大这个机器的内存消耗会耗尽，影响到这个机器上面其他的系统的运行
      7. 与目前线上的EC系统查询航班价格日历接口2s内可以出结果进行耗时对比。这个系统需要进一步处理速度的优化才能上线

**由于服务器接口响应时间长，压力上不去。故不能通过测试**