版本号：V1.0

**SecSpace性能**

**测试计划草案**

刘大东

2018年6月13日

**修订记录**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 修订日期 | 版本号 | 变更记录 | 修订描述 | 修订人 |
| 2018.06.13 | V1.0 |  | 初稿 | 刘大东 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

目录

[一、 测试目的 4](#_Toc516674068)

[二、 测试环境准备 4](#_Toc516674069)

[三、 测试计划 4](#_Toc516674070)

[四、 测试场景 4](#_Toc516674071)

[4.1 场景一：用户登录 5](#_Toc516674072)

[4.2 场景二：解锁 5](#_Toc516674073)

[4.3 场景三：策略拉取 5](#_Toc516674074)

[4.4 场景四：检查更新 5](#_Toc516674075)

[4.5 场景五：应用商店获取应用列表 6](#_Toc516674076)

[4.6 场景六：我的文档获取文档列表 6](#_Toc516674077)

[五、 测试实施 6](#_Toc516674078)

[5.1 测试准备 6](#_Toc516674079)

[5.2 实施步骤 7](#_Toc516674080)

[六、 风险备忘 7](#_Toc516674081)

## 测试目的

本次性能测试目的在于摸清服务器对客服端请求的性能表现，确认目前系统负载的极限及性能瓶颈。

## 测试环境准备

2.1 硬件/系统配置

a) 服务端配置

nginx服务器一台

b) 客户端配置

主控测试机一台:

系统： win7 旗舰版 64位

CPU: Intel(R) Core(TM) i5-6400 CPU @ 2.70GHz

RAM: 8.00GB

打压机若干（如有需要）

2.2 程序配置

准备可以方便快捷批量修改数据库测试样本数据的服务端脚本或接口，用于定量准备测试数据。

## 测试计划

|  |  |  |
| --- | --- | --- |
| 日期 | 任务 | 负责人 |
| 06.13 | 讨论筛选被测业务，确认性能测试场景。 | 张俊、郭振相、林强、  聂新宇、刘大东 |
| 06.14 | 准备服务器样本数据定量控制脚本  （用于测试样本数据准备） | 张俊 |
| 06.14 | 分场景梳理对应接口，说明访问顺序  （写脚本需要） | 郭振相、林强 |
| 06.14-06.28 | 每三天完成一个分组测试任务，包括脚本编写、打压部署、瓶颈定位及分组简报 | 刘大东 |
| 06.29-06.31 | 编写测试报告 | 刘大东 |

## 测试场景

预估实际使用时**访问频率较高**、**请求数据量较大**或者**需要查询大数据库**的接口，筛选性能测试场景（预计筛选出5-7个代表性场景），测试脚本将基于场景编写，覆盖相应接口。测试进度也将以场景为最小单元进行控制，预计每三天完成一个场景的性能测试。

### 4.1 场景一：用户登录

场景特征：

访问频率高，存在瞬时峰值。

样本数据：

Server端在用户数据库准备2000条虚拟注册用户。

测试方法：

1. 模拟用户操作过程，组成打压请求脚本
2. 使用Loadrunner向服务器打压
3. 每60分钟增加100个/秒，一直增加到300并发并持续60分钟 （假设2000用户全部在线，根据并发取在线用户10%的经验，预计并发常态为200左右，保守向上取300并发，作为登录场景测试的最高并发）
4. 重点关注同时并发用户数，TPS，响应时间，server端的CPU和内存占用情况
5. 分析对比得到的性能结果数据，辨别高消耗接口，尝试找出性能瓶颈。

### 4.2 场景二：解锁

场景特征：

访问频率特高、请求数据量较大。

测试方法：

1. 模拟用户操作过程，组成打压请求脚本
2. 使用Loadrunner向服务器打压
3. 每60分钟增加100个/秒，一直增加到500/秒并持续60分钟（解锁操作频率非常高，适当增大测试并发）
4. 重点关注同时并发用户数，TPS，响应时间，server端的CPU和内存占用情况
5. 分析对比得到的性能结果数据，尝试找出性能瓶颈。

### 4.3 场景三：策略拉取

场景特征：

一旦进行策略拉取操作，会存在瞬时超高并发。

测试方法：

1. 模拟用户操作过程，组成打压请求脚本
2. 使用Loadrunner向服务器打压
3. 每60分钟增加100个/秒，一直增加到2000/秒并持续60分钟（假如所有用户都在线，服务端进行策略下发操作时，就有可能产生瞬时超高并发 ）
4. 重点关注同时并发用户数，TPS，响应时间，server端的CPU和内存占用情况
5. 分析对比得到的性能结果数据，尝试找出性能瓶颈。

### 4.4 场景四：检查更新

场景特征：

访问频率很高。

测试方法：

1. 模拟用户操作过程，组成打压请求脚本
2. 使用Loadrunner向服务器打压
3. 每60分钟增加100个/秒，一直增加到400/秒并持续60分钟（检查更新频率很高，适当增大测试并发 ）
4. 重点关注同时并发用户数，TPS，响应时间，server端的CPU和内存占用情况
5. 分析对比得到的性能结果数据，尝试找出性能瓶颈。

### 4.5 场景五：应用商店获取应用列表

场景特征：

数据查询量很大。

测试方法：

1. 模拟用户操作过程，组成打压请求脚本
2. 使用Loadrunner向服务器打压
3. 每60分钟增加100个/秒，一直增加到400/秒并持续60分钟
4. 重点关注同时并发用户数，TPS，响应时间，server端的CPU和内存占用情况
5. 分析对比得到的性能结果数据，尝试找出性能瓶颈。

### 4.6 场景六：我的文档获取文档列表

场景特征：

数据查询量很大。

测试方法：

1. 模拟用户操作过程，组成打压请求脚本
2. 使用Loadrunner向服务器打压
3. 每60分钟增加100个/秒，一直增加到400/秒并持续60分钟
4. 重点关注同时并发用户数，TPS，响应时间，server端的CPU和内存占用情况
5. 分析对比得到的性能结果数据，尝试找出性能瓶颈。

## 测试实施

### 5.1 测试准备

服务端：

隔离服务器、数据库准备、做必要的接口调整（如有需要）。

客户端：

模拟真实的用户操作编写脚本、打压机部署。

### 5.2 实施步骤

a) 了解服务端接口逻辑，编写脚本模拟实际请求，覆盖业务涉及的特定接口;

b) 使用Loadrunner导入脚本在不同并发下打压服务器，采集分析各项性能参数；

c）整理分析测试报告，定位瓶颈。

## 风险备忘

1. 随着对系统性能的深入了解，可能会不断调整测试计划，量化测试参数。
2. 随着负载的不断增加，可能会导致系统不稳定，以致部分测试操作失效，时间消耗不完全可控。