莫锦波10175101120

摘要

meethere场馆预约系统的性能测试计划

性能测试计划

Meethere

目录

[1 项目概要介绍 3](#_Toc501354155)

[1.1 项目简介 3](#_Toc501354156)

[1.2 项目成员 3](#_Toc501354157)

[1.3 测试范围 3](#_Toc501354158)

[1.3.1 范围内 3](#_Toc501354159)

[1.3.2 范围外 3](#_Toc501354160)

[1.4 前提假设 3](#_Toc501354161)

[1.5 测试目标 3](#_Toc501354162)

[2 性能测试策略 3](#_Toc501354163)

[2.1 性能测试模型 3](#_Toc501354164)

[2.2 性能测试场景 4](#_Toc501354165)

[2.3 重点测试策略 4](#_Toc501354166)

[2.3.1 重点测试原则 4](#_Toc501354167)

[2.3.2 重点测试交易 5](#_Toc501354168)

[3 测试案例设计 5](#_Toc501354169)

[3.1 生产压力分析 5](#_Toc501354170)

[3.2 场景通过标准 5](#_Toc501354171)

[3.3 测试场景设计 5](#_Toc501354172)

[3.3.1 独立场景 5](#_Toc501354173)

[3.3.2 混合场景 5](#_Toc501354174)

[4 测试实施安排 8](#_Toc501354181)

[4.1 测试进度 8](#_Toc501354182)

[4.2 测试流程 8](#_Toc501354183)

[4.3 测试报告需求 8](#_Toc501354184)

[4.4 性能缺陷管理 8](#_Toc501354185)

[5 性能测试标准 8](#_Toc501354186)

[5.1 启动标准 8](#_Toc501354187)

[5.2 中止标准 8](#_Toc501354188)

[5.3 通过标准 9](#_Toc501354189)

[6 测试环境规划 9](#_Toc501354190)

[6.1 部署环境 9](#_Toc501354191)

[6.2 执行环境 9](#_Toc501354192)

[7 测试风险分析 9](#_Toc501354193)

[8 角色与职责 9](#_Toc501354194)

表目录

[Table 1 性能测试类型 6](#_Toc501358463)

[Table 2 场景通过标准 8](#_Toc501358464)

[Table 3 独立场景设计 8](#_Toc501358465)

[Table 4 混合场景设计 9](#_Toc501358466)

# 项目概要介绍

## 项目简介

MeetHere——针对场馆预约与管理的Web电子商务平台，基于SpringBoot与React框架开发。

项目部署在阿里云ECS（CentOs 7.2 1核2G内存），带宽1mbps RTT 12ms

## 项目成员

彭锋、钟天景、兰于权、莫锦波

## 测试范围

### 范围内

### 普通用户

* 登录
* 查看新闻
* 查看新闻与评论
* 发表评论
* 创建订单
* 搜索场馆
* 查看订单

管理员

* + 订单管理（根据不同维度查询所有订单）

### 范围外

* 新闻与场馆的写操作

## 前提假设

单元测试、集成测试、静态代码分析、系统测试已达到要求

## 测试目标

1) 系统在单步操作响应时间方面，能满足用户当前及未来1到3年的发展需求；

2) 系统的批量作业运行稳定，处理能力能满足用户当前及未来1到3年的发展需求；

3) 在业务处理能力方面，能满足当前及未来1到3年的业务增长需求；

4) 发现并解决宕机，内存泄漏等严重问题，使系统具备良好的稳定性、健壮性；

5) 系统在资源使用方面比较合理，各项资源平均利用率在30%左右。

# 性能测试策略

与其它测试类型一样，性能测试周期分为：性能测试需求分析、性能测试设计、性能测试实现和性能测试报告四个存在迭代的阶段。其中，性能测试需求和性能测试设计由测试人员手工完成，而性能测试实现需借助性能测试工具达成，通过对性能测试报告的分析判定系统是否存在性能缺陷。

## 性能测试模型

性能测试模型定义了性能测试涉及的各个待测方面，是性能测试设计的指南。



Figure 1 性能测试模型

根据Figure 1所示的性能测试模型，可以设计如Table 1所示的性能测试类型

Table 1性能测试类型

|  |  |  |  |
| --- | --- | --- | --- |
| **测试种类** | **测试方法** | **测试内容** | **场景类型** |
| 性能测试 | 以系统最大并发用户数的1-2倍作为上限对关键交易进行梯度加压测试，在压力时间内通过的交易量应接近峰值时段的交易量，甚至超过系统全天的交易量 | 核心业务，需求分析中发现的容易出现性能问题的业务 | 独立场景 |
| 压力测试 | 以系统预期最大并发用户数的1-2倍作为上限对常用/关键交易进行混合场景梯度加压测试，在压力时间内通过的交易量应接近峰值时段的交易量，甚至超过系统全天的交易量 | 存在一定关联关系的常用/核心业务 | 混合场景 |

## 性能测试场景

根据项目实际，确定本次性能测试场景，例如可包含以下场景的性能测试：

* 独立场景
* 混合场景

## 重点测试策略

### 重点测试原则

为了全面评估系统性能，本次重点测试策略安排如下：

1. 对关键交易开展全面的独立场景测试；
2. 对关键交易开展全面的峰值、容量测试；
3. 多渠道发起以关键/常用交易为主的疲劳测试；
4. 在集成性能测试方面，对交易量大、容易出现性能瓶颈的模块，重点进行测试并安排专人在系统发布前一直跟进开发组的需求变更以便进行及时的回归性能测试。
5. 在渠道性能测试方面，对渠道场景按照业务量及重要性来划分优先级，按照优先级先后来执行，保证系统上线后关键业务的稳定性。

### 重点测试交易

* 订单创建
* 订单审核

# 测试案例设计

## 生产压力分析

生产环境下读多写少，读压力测试较多。

## 场景通过标准

Table 2 场景通过标准

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **场景类型** | **单步操作响应时间** | **加压时间** | **操作账户数据** | **处理业务笔数** | **事务成功率** | **各后台利用率** | **并发/在线上限** | **在线思考时间设置** |
| 独立场景 | <1秒 | 30分钟 | >1万 | >5000 | 100% | <50% | 100 | 无 |
| 混合场景 | <1秒 | 1小时 | >1万 | >10万 | 100% | <80% | 200 | 无 |
| 峰值场景 | <2秒 | 1小时 | >1万 | >10万 | 100% | <80% | 200 | 无 |
| 容量场景 | <3秒 | 4小时 | >1万 | >50万 | 100% | <80% | 300 | 无 |

## 测试场景设计

### 独立场景

对于关键场景，将分别对其采用渐进式加压的方式来进行独立场景测试。每个场景测试多组并发、并发数从1逐步增加到200。对于具体的场景，测试几组并发依据案例执行。案例需要按照业务量设计。对于使用频度较低的场景，可以采用独立/混合场景的方式进行测试，即在背景压力下以小并发的方式来进行长时间疲劳测试。

Table 3 独立场景设计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **并发数** | **响应时间要求** | **TPS要求** | **加压时间** | **加压方式** |
| 1 | <1秒 | N/A | 3分钟 | 直接加压 |
| 10 | <1秒 | N/A | 10分钟 | 1个用户/5秒 |
| 20 | <1秒 | N/A | 10分钟 | 1个用户/5秒 |
| 50 | <1秒 | >100笔/秒 | 20分钟 | 1个用户/5秒 |
| 100 | <1秒 | >100笔/秒 | 20分钟 | 1个用户/5秒 |
| 200 | <1秒 | >100笔/秒 | 10分钟 | 1个用户/1秒 |

### 混合场景

对于相关联的关键交易，对其采用渐进式加压的方式来进行混合场景测试，以测试这些关联交易是否存在性能问题。每个场景测试多组并发，并发数从20逐步增加到200。对于具体的场景，测试几组并发依据案例来执行（案例主要依据业务量来设计）。对于使用频度较低的场景，可以采用独立/混合场景的方式进行测试，即在背景压力下以小并发的方式来进行长时间疲劳测试。

Table 4 混合场景设计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **并发数** | **响应时间要求** | **TPS要求** | **加压时间** | **加压方式** |
| 20 | <1秒 | N/A | 10分钟 | 1个用户/5秒 |
| 50 | <1秒 | >100笔/秒 | 20分钟 | 1个用户/5秒 |
| 100 | <1秒 | >100笔/秒 | 20分钟 | 1个用户/5秒 |
| 200 | <1秒 | >100笔/秒 | 10分钟 | 1个用户/1秒 |

# 测试实施安排

## 测试进度

|  |  |  |  |
| --- | --- | --- | --- |
| **测试任务** | **开始时间** | **结束时间** | **接受者** |
| 了解项目常见场景与性能需求 | 2019-12-28 | 2019-12-28 | 测试人员、开发人员、项目经理 |
| 在服务器上搭建测试环境与编写测试脚本 | 2019-12-28 | 2019-12-28 | 测试人员 |
| 运行测试脚本生成测试数据 | 2019-12-29 | 2019-12-30 | 测试人员 |
| 编写测试报告 | 2019-12-31 | 2020-1-1 | 测试人员 |

## 测试流程

1. 了解项目常见场景与性能需求，如该业务场景下的压力与用户希望的响应时间
2. 搭建测试环境与编写测试脚本：
   * + 1. 在云服务器上使用docker搭建测试环境
       2. Jmeter测试脚本的编写为手工编写
3. 运行测试脚本并生成测试报告
   * + 1. 使用jmeter生成html测试报告

## 测试报告需求

Table 9 测试报告需求

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **报告类型** | **响应时间要求** | **报告者** | **接受者** | **报告内容** |
| 消息 | 每日 | 测试负责人 | 开发人员，项目经理 | 每日结果总结报告 |
| 文档 | 各测试阶段结束 | 测试负责人 | 测试人员，项目经理 | 性能测试报告 |

## 性能缺陷管理

测试过程采用给开发人员与项目经理发送消息来进行缺陷管理

# 性能测试标准

## 启动标准

1. 测试环境满足计划需求
2. 基准参数配置完成校验
3. 关键交易通过冒烟测试

## 中止标准

1. 测试环境或关键系统不可用
2. 测试环境距生产标准差距太大
3. 缺陷周转周期不符合规定的时间
4. 出现宕机、不响应等严重的性能问题
5. 系统的交易成功率低于95%

## 通过标准

系统上线至少满足下面标准：

1. 系统无宕机、不响应类的严重性能问题
2. 系统响应时间80%达到系统的期望值
3. 系统的业务吞吐量达到预期目标，即当前生产需求的3倍以上
4. 柜台类系统要求通过7\*12小时以上的疲劳强度测试
5. 电子渠道类系统要求通过7\*24小时以上的疲劳强度测试

# 测试环境规划

## 部署环境

硬件环境

|  |  |
| --- | --- |
| **环境名** | **环境信息** |
| 阿里云ECS云服务器 | 1核CPU 2G RAM 40G磁盘 |
| 网络设备 | 1Mbps带宽，平均RTT 12ms |
| 客户访问设备 | — |

软件环境

|  |  |
| --- | --- |
| **软件名** | **版本** |
| CentOS | 7.2 Linux内核版本 4.9 |
| Redis | 5.0 |
| Mysql | 8.0 |
| Jdk | 11 |
| Nginx | 1.1.17 |

## 执行环境

与部署环境相同

# 测试风险分析

|  |  |
| --- | --- |
| **风险名** | **应对措施** |
| 测试新增时，插入数据太多，造成后续数据访问缓慢 | 定时删除新增的测试数据 |

# 角色与职责

|  |  |  |
| --- | --- | --- |
| **角色** | **姓名** | **职责** |
| 性能测试人员 | 莫锦波 | 1. 性能测试需求分析 2. 编写测试计划 3. 开发测试脚本 4. 编写测试报告 |