作者 彭钧涛 李尚真

**华东师范大学 软件工程学院**

Meethere场地预约与管理系统性能测试计划

目录

[1 项目概要介绍 4](#_Toc28870759)

[1.1 项目简介 4](#_Toc28870760)

[1.2 项目成员 4](#_Toc28870761)

[1.3 测试范围 4](#_Toc28870762)

[1.3.1 范围内 4](#_Toc28870763)

[1.3.2 范围外 4](#_Toc28870764)

[1.4 前提假设 4](#_Toc28870765)

[1.5 测试目标 4](#_Toc28870766)

[2 性能测试策略 5](#_Toc28870767)

[2.1 性能测试模型 5](#_Toc28870768)

[2.2 性能测试场景 6](#_Toc28870769)

[3 测试案例设计 6](#_Toc28870770)

[3.1 生产压力分析 6](#_Toc28870771)

[3.2 场景通过标准 6](#_Toc28870772)

[3.3 测试场景设计 7](#_Toc28870773)

[3.3.1 独立场景 7](#_Toc28870774)

[3.3.1 8](#_Toc28870781)

[3.3.1.1 注册 8](#_Toc28870782)

[3.3.1.2 登录 8](#_Toc28870783)

[3.3.1.3 评论留言 9](#_Toc28870784)

[3.3.1.4 浏览新闻 9](#_Toc28870785)

[3.3.1.5 预约场馆 9](#_Toc28870786)

[3.3.1.6 修改密码 10](#_Toc28870787)

[3.3.2 混合场景 10](#_Toc28870788)

[4 测试实施安排 11](#_Toc28870789)

[4.1 测试进度 11](#_Toc28870790)

[4.2 测试流程 12](#_Toc28870791)

[流程分为需求分析、设计、实现和测试报告四个阶段。 12](#_Toc28870792)

[4.2.1 性能测试需求分析与设计： 12](#_Toc28870793)

[4.2.1.1 熟悉被测试系统，定义性能测试要求，例如并发用户的数量、典型业务流程和所需响应时间。 12](#_Toc28870794)

[4.2.2 性能测试实现： 12](#_Toc28870795)

[4.2.2.1 创建Locust脚本。 12](#_Toc28870796)

[4.2.3 生成性能测试报告： 12](#_Toc28870797)

[4.2.3.1 使用 Locust和Microsoft Office Excel创建图和报告并评估性能。 12](#_Toc28870798)

[4.3 测试报告需求 12](#_Toc28870799)

[4.4 性能缺陷管理 12](#_Toc28870800)

[5 性能测试标准 12](#_Toc28870801)

[5.1 启动标准 12](#_Toc28870802)

[5.2 中止标准 13](#_Toc28870803)

[5.3 通过标准 13](#_Toc28870804)

[6 测试环境规划 13](#_Toc28870805)

[6.1 部署环境 13](#_Toc28870806)

[6.2 执行环境 14](#_Toc28870807)

[7 测试风险分析 14](#_Toc28870808)

[8 角色与职责 15](#_Toc28870809)

表目录

[Table 1性能测试类型 6](#_Toc28870829)

[Table 2 场景通过标准 6](#_Toc28870830)

[Table 3 独立场景设计 7](#_Toc28870831)

[Table 4 混合场景设计 11](#_Toc28870832)

[Table 5 测试报告需求 12](#_Toc28870833)

# 项目概要介绍

## 项目简介

MeetHere是一个场馆预约与管理的Web电子商务网站，主要功能包括：

1) 普通用户：注册、登录、个人信息管理、查看场馆介绍和预约信息、场馆预约、场馆预约订单管理、查看新闻、留言管理（发布、浏览、删除、修改）。

2) 管理员：用户管理、场馆信息管理（场馆介绍、场馆位置、场馆租金、场馆空闲时间...）、预约订单审核、预约订单统计（按照场馆、时间等对预约订单进行统计）、新闻动态管理（发布、增、删、改）、留言审核

3) 场馆预约订单：已预定场馆信息的查看、取消、信息修改

## 项目成员

彭钧涛 李尚真 郭源杰

## 测试范围

### 范围内

用户注册、用户登录、用户发表评论、用户浏览新闻、用户预约场馆、用户修改密码

### 范围外

管理员增删改查用户、新闻、场地、记录

## 前提假设

被测系统完成功能性系统测试，达到预计要求

## 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；

# 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 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

# 测试案例设计

## 生产压力分析

## 场景通过标准

Table 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(/s)** | **处理业务笔数(/s)** | **事务成功率** | **并发/在线上限** | **在线思考时间设置** |
| 独立场景 | <1秒 | 10分钟 | >500 | >500 | 100% | 200 | 无 |
| 混合场景 | <1秒 | 30分钟 | >500 | >500 | 100% | 200 | 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 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 | 5分钟 | 1个用户/5秒 |
| 20（异常流） | <1秒 | >40笔/秒 | 5分钟 | 1个用户/5秒 |
| 50 | <1秒 | >100笔/秒 | 10分钟 | 1个用户/5秒 |
| 100 | <1秒 | >200笔/秒 | 10分钟 | 1个用户/5秒 |
| 200（正常流） | <1秒 | >500笔/秒 | 10分钟 | 1个用户/1秒 |



### 注册

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 场景步骤 | 步骤描述 | 输入 | 预期结果 |
| 1 | 场景1:成功注册 | 1、进入注册页面  2、输入用户名  3、输入密码  4、输入姓名  5、输入电话  6、选择性别  7、点击“注册” | 用户名 username  姓名 name  密码 1234567890  性别 MALE  电话 13000000000 | 跳转回首页，自动填充用户密码 |
| 2、 | 场景2：用户名已存在 | 1、进入注册页面  2、输入用户名  3、输入密码  4、输入姓名  5、输入电话  6、选择性别  7、点击“注册” | 用户名 username | 网页提示该用户名已存在“” |

### 登录

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 场景步骤 | 步骤描述 | 输入 | 预期结果 |
| 1、 | 场景1:成功登录 | 1、进入登录页面  2、输入用户名  3、输入密码  4、点击“登录” | 用户名 username  密码 1234567890 | 跳转至登录后空白页 |
| 2、 | 场景4：用户名或密码错误 | 1、进入登录页面  2、输入用户名  3、输入密码  4、点击“登录” | 用户名 username  密码 wrongpassword | 网页提示“用户名或密码错误” |

### 评论留言

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 场景步骤 | 步骤描述 | 输入 | 预期结果 |
| 1、 | 场景1:成功评论 | 1、进入评论留言页面  2、点击“我要留言”  3、留言内容  4、点击“保存” | 留言 这是一条留言。 | 跳转至评论浏览页 |

### 浏览新闻

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 场景步骤 | 步骤描述 | 输入 | 预期结果 |
| 1、 | 场景1:浏览新闻 | 1、进入浏览新闻  2、点击新闻 | N/A | 进入新闻页面 |

### 预约场馆

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 场景步骤 | 步骤描述 | 输入 | 预期结果 |
| 1、 | 场景1:成功预约 | 1、进入场馆页面  2、选择场馆  3、输入开始日期  4、输入天数  5、点击“预约” | 开始日期2020-01-02  天数 2 | 跳转至场馆页面 |

### 修改密码

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 场景步骤 | 步骤描述 | 输入 | 预期结果 |
| 1、 | 场景4：原密码错误 | 1、进入修改密码页面  2、输入原密码  3、输入新密码  4、重复新密码  5、点击“提交” | 原密码 wrongpassword | 网页提示“旧密码错误“ |

### 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。

具体场景内的操作流程包括：

1、用户注册—>用户登录—>用户退出

2、用户登录—>用户点击新闻浏览—>用户查看新闻内容—>用户退出

3、用户登录—>用户点击评论浏览—>用户发表评论—>用户退出

4、用户登录—>用户点击场馆列表—>用户输入信息—>用户预约—>用户退出

5、用户登录—>用户点击预约记录—>用户退出

对于操作流程1，2，3，4，6，该流程在系统中出现率非常高，采用从0到200逐渐加压的方式。以得到该场景通过测试的成功率。而对于5，8等操作流程，用户使用频率一般，因此采用20到100的加压方式。

Table 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（异常） | <1秒 | N/A | 10分钟 | 1个用户/5秒 |
| 50 | <1秒 | >100笔/秒 | 10分钟 | 1个用户/5秒 |
| 100 | <1秒 | >100笔/秒 | 15分钟 | 1个用户/5秒 |
| 200（正常） | <1秒 | >100笔/秒 | 30分钟 | 1个用户/1秒 |

# 测试实施安排

## 测试进度

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **编号** | **任务** | **开始日期** | **结束日期** | **参与人员** |
| 1 | 熟悉被测试系统，确定典型事务 | 2019.12.30 | 2019.12.30 | 测试人员  开发人员  项目经理 |
| 2 | 搭建测试环境，  录制典型事务的脚本，增强脚本 | 2019.12.30 | 2019.12.30 | 测试人员 |
| 3 | 执行测试并收集相关数据 | 2019.12.30 | 2019.12.30 | 测试人员 |
| 4 | 数据分析，编写测试报告 | 2019.12.30 | 2020.01.01 | 测试人员 |

## 测试流程

### 流程分为需求分析、设计、实现和测试报告四个阶段。

### 性能测试需求分析与设计：

### 熟悉被测试系统，定义性能测试要求，例如并发用户的数量、典型业务流程和所需响应时间。

### 性能测试实现：

### 创建Locust脚本。

### 生成性能测试报告：

### 使用 Locust和Microsoft Office Excel创建图和报告并评估性能。

## 测试报告需求

Table 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** |
| 邮件 | 每日 | 各渠道测试组负责人 | 测试中心，项目经理 | 每日结果总结报告 |
| 邮件 | 每日 | 各渠道测试组负责人 | 测试中心，项目经理 | 每日测试结果记录 |
| 邮件 | 每日 | 性能测试负责人 | 测试中心，项目经理 | 周报 |
| 文档 | 各测试阶段结束 | 各渠道测试组负责人 | 测试中心，项目经理 | 性能测试报告 |

## 性能缺陷管理

测试过程采用Github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

# 测试环境规划

## 部署环境

|  |  |  |  |
| --- | --- | --- | --- |
| **软件环境（相关软件、操作系统等）** | | | |
| **名称** | **版本** | **数量** | **获得途径** |
| Windows | 10 | 3 | 学校正版Win10 |
| MySQL | 8.0 | 3 | 开源版 |
| Chrome浏览器 | 79 | 3 | 网络 |
| **硬件环境（网络、设备等）** | | | |
| **名称** | **版本** | **数量** | **获得途径** |
| Web服务器 | N/A | N/A | N/A |
| 数据库服务器 | N/A | N/A | N/A |
| 前台客户机 | N/A | N/A | N/A |

## 执行环境

|  |  |  |  |
| --- | --- | --- | --- |
| **软件环境（相关软件、操作系统等）** | | | |
| **名称** | **版本** | **数量** | **获得途径** |
| Windows | 10 | 3 | 学校正版Win10 |
| MySQL | 8.0 | 3 | 开源版 |
| Chrome浏览器 | 79 | 3 | 网络 |
| **硬件环境（网络、设备等）** | | | |
| **名称** | **版本** | **数量** | **获得途径** |
| Web服务器 | N/A | N/A | N/A |
| 数据库服务器 | N/A | N/A | N/A |
| 前台客户机 | N/A | N/A | N/A |

# 测试风险分析

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **风险因素** | **可能结果** | **可能发生时间** | **风险**  **级别** | **应对措施** |
| 工具缺陷 | 测试工具和监控工具无法全部支持信贷业务系统的测试和监控 | 随时 | 中 | 评估被测系统，分析所有需求。  通过其它工具实现对需求的支持程度。 |
| 测试数据的准备备份及恢复无法正常完成 | 测试过程中数据用尽或不满足测试需求，将导致测试无法实施。 | 测试执行时 | 高 | 运维方配合完成数据的准备、备份和恢复 |
| 测试环境有其他用户连接进行操作，服务器产生性能缺陷 | * 1. 测试方获得最大负载压力与实际最大负载有差距   2. 服务器出现性能缺陷的现象，运维方定位性能缺陷模块并非真正性能缺陷的模块 | 测试执行时 | 高 | 测试方进行负载测试时，保证测试环境无其他连接和用户操作 |
| 测试服务器访问状态不稳定 | 测试准备和测试执行中断，测试计划时间延后 | 随时 | 高 | 保证测试期间测试环境访问畅通 |

# 角色与职责

|  |  |  |  |
| --- | --- | --- | --- |
| **角色** | **姓名** | **具体职责** | **技能要求** |
| 性能测试经理 | 彭钧涛 | 性能测试项目管理和协调  性能测试需求分析  测试计划和测试报告编写 | 1. 组织协调能力 2. 熟悉性能测试架构 3. 熟悉应用架构 |
| 性能测试设计人员 | 李尚真 | 编制测试案例，开发测试脚本，设计测试场景，执行性能测试 | 1. 熟悉性能测试体系 2. 熟悉性能测试方法和工具 3. 脚本编写和调试能力 |
| 测试环境和测试数据准备人员 | 李尚真 | 准备性能测试环境  准备基础数据和测试数据 | 1. Locust部署能力 2. 熟悉核心系统、数据库 |
| 性能调优人员 | 李尚真 | 核心主机系统优化 | 1. 熟悉数据库优化 2. 熟悉应用架构 |