狄亦凡

**华东师范大学 软件工程学院**

性能测试计划

目录

[1 项目概要介绍 4](#_Toc1138677957)

[1.1 项目简介 4](#_Toc1533644882)

[1.2 项目成员 4](#_Toc1870800480)

[1.3 测试范围 4](#_Toc1235591633)

[1.3.1 范围内 4](#_Toc421709341)

[1.3.2 范围外 4](#_Toc972859087)

[1.4 前提假设 4](#_Toc2049670598)

[1.5 测试目标 4](#_Toc1028559059)

[2 性能测试策略 5](#_Toc1896229910)

[2.1 性能测试模型 5](#_Toc1278775890)

[2.2 性能测试场景 6](#_Toc370044054)

[2.3 重点测试策略 6](#_Toc217773866)

[2.3.1 重点测试原则 6](#_Toc813231374)

[2.3.2 重点测试交易 6](#_Toc1393773310)

[3 测试案例设计 6](#_Toc396399694)

[3.1 生产压力分析 6](#_Toc795384064)

[3.2 场景通过标准 7](#_Toc2081744720)

[3.3 测试场景设计 7](#_Toc1079932116)

[3.3.1 独立场景 7](#_Toc2034772815)

[具体的独立场景测试用例如下所示： 7](#_Toc1897106877)

[3.3.2 混合场景 9](#_Toc985574730)

[3.3.3 压力测试 10](#_Toc1013117799)

[4 测试实施安排 10](#_Toc73010730)

[4.1 测试进度 10](#_Toc878176673)

[4.2 测试流程 10](#_Toc2007720927)

[4.3 测试报告需求 11](#_Toc355074778)

[4.4 性能缺陷管理 11](#_Toc2032222480)

[5 性能测试标准 11](#_Toc1983299472)

[5.1 启动标准 11](#_Toc73057170)

[5.2 中止标准 11](#_Toc1658693753)

[5.3 通过标准 12](#_Toc1180684964)

[6 测试环境规划 12](#_Toc1023291668)

[6.1 部署环境 12](#_Toc1414018900)

[6.2 执行环境 12](#_Toc1361614598)

[7 测试风险分析 13](#_Toc1070806154)

[8 角色与职责 13](#_Toc1126068418)

# 项目概要介绍

## 项目简介

Meethere是一个场馆预约和管理系统，主要面向的使用者为普通用户和系统管理员。

普通用户可以通过注册账号，在进行登录操作后进行查看场馆列表、查看新闻、场馆预约、留言以及个人信息管理等操作。

管理员通过管理员账号登录后，可以审核用户预约信息、用户管理、留言管理、新闻更新及管理、场馆添加及管理等。

本项目可以用于大型企业或公共场所的官方网站，便于对其内部场地和用户的管理。

## 项目成员

陶传贵、狄亦凡、黄皓冬

## 测试范围

### 范围内

Meethere场馆预约管理系统的前端业务，包括用户注册、登录、场馆预约、查看个人信息、查看新闻、发布留言、查看场馆信息等前台业务功能。

### 范围外

由于管理员人数有限，因此管理员登录该网站进行的各种管理操作将不在本次性能测试的范围之内。

## 前提假设

被测系统完成功能测试，测试报告显示达到要求。

## 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。

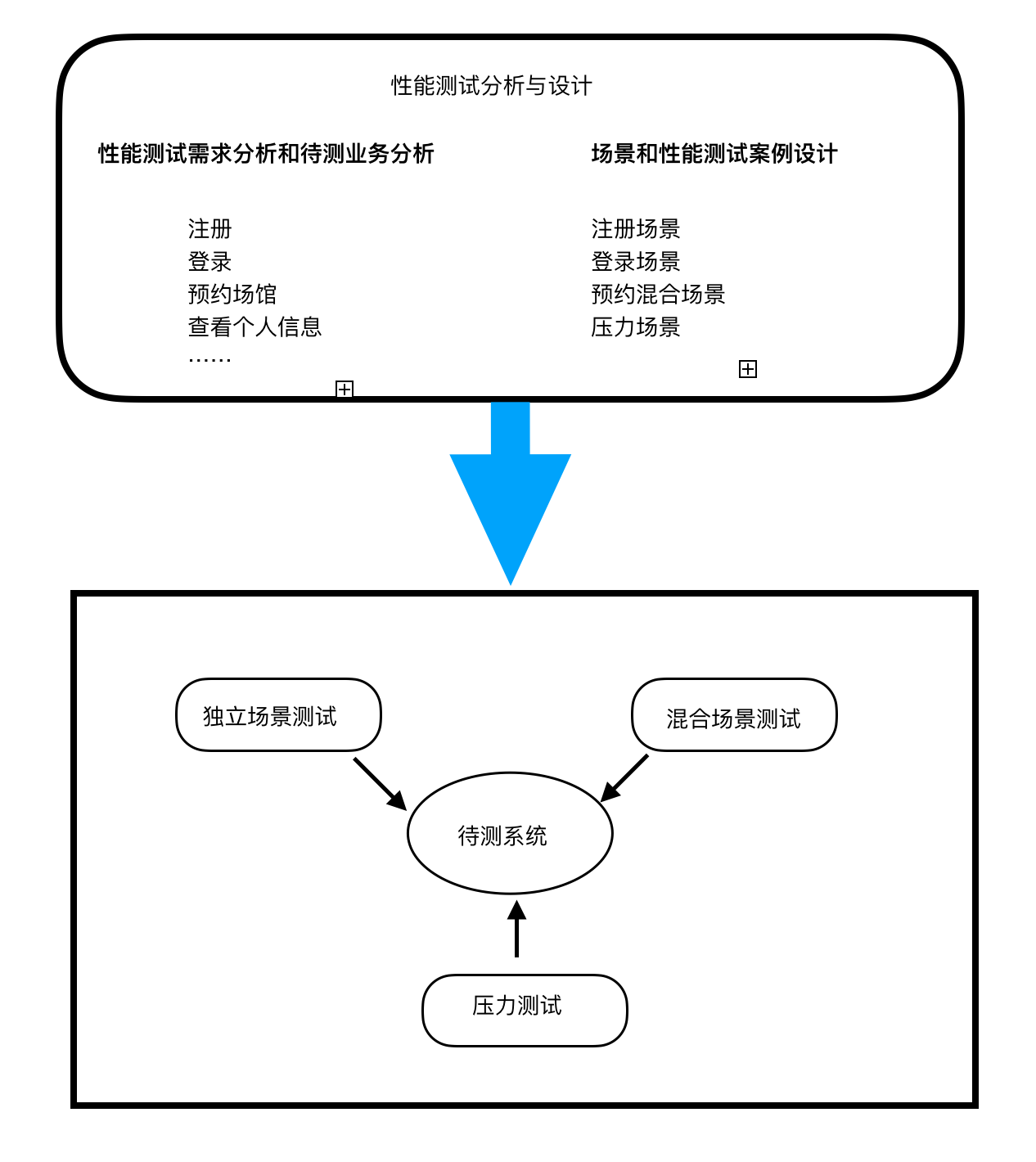


Figure 1 性能测试模型

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10倍作为上限对常用/关键预约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预约量，甚至超过系统全天的预约量 | 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

压力测试

## 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. 对关键操作开展全面的独立场景测试；
2. 对关键操作开展全面的峰值测试；

3）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。

1. 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

# 测试案例设计

## 生产压力分析

对于本系统的开发

## 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 | >100 | >100 | 100% | <50% | 100 | 无 |
| 混合场景 | <1秒 | 1小时 | >200 | >200 | 100% | <80% | 100 | 5秒 |
| 压力测试 | <2秒 | 10分钟 | >2000 | >2000 | 95% | <90% | 2000 | 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50。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 |

### 具体的独立场景测试用例如下所示：

1. 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、输入邮箱  5、输入手机号码  6、点击“注册并登录” | 用户名:user密码:123456  邮箱名:example@qq.com  电话:13711111111 | 弹出注册成功框 |
| 2、 | 场景2:邮箱名不符合规定格式 | 1、进入注册页面  2、输入邮箱名  3、输入密码  4、再次输入密码  5、输入验证码  6、点击“注册” | 用户名:user密码:123456  邮箱名:qq.com  电话:13711111111 | 页面显示格式异常提示 |
| 3、 | 场景3:电话号码不符合规定格式 | 1、进入注册页面  2、输入邮箱名  3、输入密码  4、再次输入密码  5、输入验证码  6、点击“注册” | 用户名:user密码:123456  邮箱名:example@qq.com  电话:12345 | 页面显示格式异常提示 |

1. 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:user  密码:123456 | 弹出登录成功框 |
| 2、 | 场景2:用户名或密码错误 | 1、进入登录页面  2、输入用户名名  3、输入密码  4、点击“登录” | 用户名:wrong  密码:1111 | 弹出用户名或密码错误提示框 |

1. 测试

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用例编号 | 场景步骤 | 步骤描述 | 输入 | 预期结果 |
| 1、 | 场景1:预约场馆 | 1、进入预约页面  2、选择场地名称  3、选择开始时间  4、提交 | 场地名称：篮球场  开始时间：yyyy-MM-DD hh:mm:ss  结束时间：yyyy-MM-DD hh:mm:ss | 弹出预约成功框 |
| 2、 | 场景2:用户浏览场馆信息 | 1、进入主页  2、选择场馆信息 | 无 | 页面加载正常 |
| 3、 | 场景3：用户浏览新闻 | 1、进入主页  2、选择新闻  3、查看新闻详情 | 暂无 | 页面加载正常 |
| 4、 | 场景4：用户发布留言 | 1、进入留言页面  2、输入留言  3、点击发布 | Message:${message} | 弹出发布成功提示 |
| 5、 | 场景5:用户查看个人信息 | 1. 进入主页 2. 点击用户名 3. 查看个人信息 | 暂无 | 页面显示正常 |

### 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。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->用户登出
2. 用户登录->查看个人信息->用户登出
3. 用户登录->查看场馆信息->用户登出
4. 用户登录->查看新闻->用户登出
5. 用户登录->用户发布留言->点击提交->用户登出
6. 用户登录->用户填写预约信息->点击提交->用户登出
7. 用户登录->用户点击个人信息->用户查看预约信息->用户登出
8. 用户登录->用户点击个人信息->用户修改个人信息->用户点击更新->用户登出

其中，操作1、6为该系统的核心操作，因此选择从20到100逐渐加压的方式，以得到该场景通过测试的成功率；而操作8属于不频繁操作，因此采用从20到50逐渐加压方式；其余操作为一般操作，采用从20到75逐渐加压的方式进行。

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/秒 | 20分钟 | 1个用户/5秒 |
| 100 | <1秒 | >100/秒 | 20分钟 | 1个用户/5秒 |

### 压力测试

对于进一步探索本系统的最高承载能力，我们选择对本系统中的核心功能，即预约场地进行压力测试。我们期望系统能承受在10秒内并发的2000个线程的预约请求，并且能在2秒之内作出响应。对于如此高的并发数，期望系统的成功处理率能在95%以上。

# 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 | 2019.12.25 | 2018.12.26 | 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 | 2019.12.27 | 2019.12.28 | 测试人员 |
| 3 | 执行测试并收集相关数据 | 2019.12.28 | 2019.12.31 | 测试人员 |
| 4 | 数据分析，编写测试报告 | 2020.1.1 | 2020.1.1 | 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：

创建虚拟用户脚本：将最终用户活动捕获到自动脚本中。

定义场景：使用JMeter设置负载测试环境。

运行场景：通过JMeter驱动、管理和监控负载测试。

生成性能测试报告：

使用 JMeter命令行导出性能测试报告创建图和报告并评估性能。

## 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** |
| 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Quality Center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. 系统的预约成功率低于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

测试环境要求尽量和真实环境相同，至少要求服务器配置和网络带宽和拓扑结构应该相似。主要内容：服务器数量和配置，操作系统和数据库版本，软硬件部署等。

|  |  |  |  |
| --- | --- | --- | --- |
| **软件环境（相关软件、操作系统等）** | | | |
| **名称** | **版本** | **数量** | **获得途径** |
| MAC OSX | 10.15.2 | 1 |  |
| mysql | 8.0.18 |  |  |
| FireFox浏览器 | 最新版本 |  |  |
| **硬件环境（网络、设备等）** | | | |
| **名称** | **版本** | **数量** | **获得途径** |
| Web服务器 |  |  |  |
| 数据库服务器 |  |  |  |
| 前台客户机 |  |  |  |

## 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** |
| MAC OSX | 10.15.2 |  |  |
| mysql | 8.0.18 |  |  |
| FireFox浏览器 | 最新版本 |  |  |
| JMeter | 最新版本 |  |  |
| **硬件环境（网络、设备等）** | | | |
| **名称** | **版本** | **数量** | **获得途径** |
| Web服务器 |  |  |  |
| 数据库服务器 |  |  |  |
| 前台客户机 |  |  |  |

# 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MeetHere场馆预约管理系统的测试和监控 | 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 | 狄亦凡 | 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 | 狄亦凡 | 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 | 狄亦凡 | 准备性能测试环境  准备基础数据和测试数据 | 1. JMeter部署能力 2. 熟悉核心系统、数据库 |
| 性能调优人员 | 狄亦凡 | 核心主机系统优化 | 1. 熟悉数据库优化 2. 熟悉应用架构 |