



# 性能测试计划

MeetHere 系统

## 摘要

报告将从 项目简介 / 性能测试策略 / 测试案例设计 / 性能测试标准  
等多个方面详细介绍 MeetHere 项目的性能测试计划。

陈俊杰 项慕凡

华东师范大学 软件工程学院

目录

- 1. 项目概要介绍.....3
  - 1.1. 项目简介 .....3
  - 1.2. 项目成员 .....3
  - 1.3. 测试范围 .....3
    - 1.3.1. 范围内 .....3
    - 1.3.2. 范围外 .....3
  - 1.4. 前提假设 .....3
  - 1.5. 测试目标 .....3
- 2. 性能测试策略.....4
  - 2.1. 性能测试模型 .....4
  - 2.2. 性能测试场景 .....5
  - 2.3. 重点测试策略 .....5
    - 2.3.1. 重点测试原则 .....5
    - 2.3.2. 重点测试交易 .....6
- 3. 测试案例设计.....6
  - 3.1. 生产压力分析 .....6
  - 3.2. 场景通过标准 .....6
  - 3.3. 测试场景设计 .....7
    - 3.3.1. 独立场景 .....7
    - 3.3.2. 混合场景 .....7
- 4. 测试实施安排.....8
  - 4.1. 测试进度 .....8
  - 4.2. 测试流程 .....9
  - 4.3. 测试报告需求 .....9
  - 4.4. 性能缺陷管理 .....9
- 5. 性能测试标准.....9
  - 5.1. 启动标准 .....9
  - 5.2. 中止标准 .....9
  - 5.3. 通过标准 .....9
- 6. 测试环境规划.....10
  - 6.1. 部署环境 .....10

6.2.执行环境 .....	10
7. 测试风险分析.....	11
8. 角色与职责 .....	11

## 1. 项目概要介绍

### 1.1. 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

### 1.2. 项目成员

童翰文 10175101152 陈俊杰 10175101148

项慕凡 10175101151 徐滔锴 10175101147

### 1.3. 测试范围

#### 1.3.1. 范围内

由于本次项目中的 MeetHere 前端基于 Vue 开发、后端基于 SpringBoot 开发，因此本次性能测试主要包括：

- ① 利用 JMeter 对不同的功能接口进行独立场景测试，包括“登录”、“浏览新闻”、“浏览场馆”、“预约场馆”、“订单管理”、“个人信息管理”等；
- ② 利用 JMeter 对整个系统进行混合场景测试；
- ③ 根据 JMeter 的性能测试分析报告对系统性能进行总结分析。

#### 1.3.2. 范围外

未填写

### 1.4. 前提假设

### 1.5. 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。

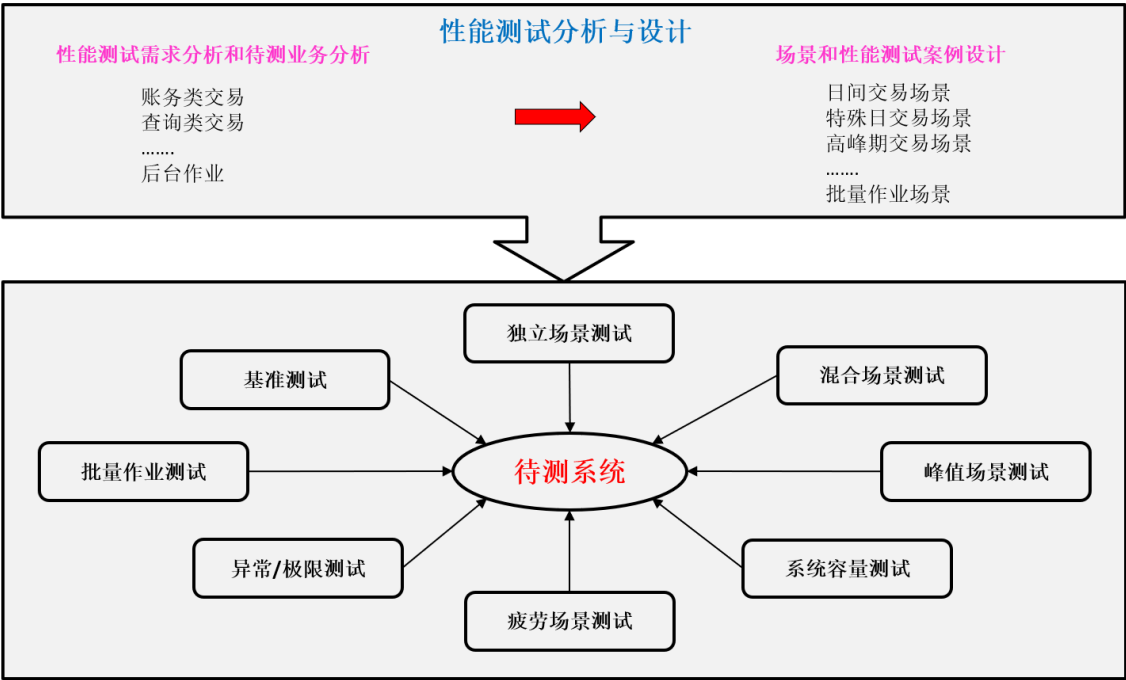
## 2. 性能测试策略

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。

### 2.1. 性能测试模型

性能测试模型定义了性能测试涉及各个待测方面，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
峰值测试	按照系统峰值时预期最大并发用户数的2-3倍对峰值时段涉及的核心交易进行的混合场景梯度加压测试，在测试时间内通过的交易数量应接近或超过系统全天的交易量	生产上峰值场景中的核心业务	峰值场景
容量测试	通过阶段性不断增加系统并发用户的梯度加压方式，测试出系统对核心交易的最大并发处理能力以及系统的最大业务处理能力，测试时最大并发用户数为系统预期最大并发用户数的3-5倍	针对核心交易	容量场景
疲劳测试	以系统预期最大并发用户数的1-2倍作为上限对关键交易进行持续加压的稳定性与健壮性测试，在压力时间内通过的交易量应为全天交易量3倍以上	针对核心交易	疲劳场景
批量测试	采用手工/自动调用的方式对后台作业进行测试	针对核心批量作业	批处理场景
极限测试	以超过系统处理能力的压力对系统进行不断加压的极限状态测试，测试系统在极限压力下处理能力是否稳定以及是否出现宕机等严重性能问题	针对核心交易	极限场景
异常测试	测试在正常生产压力下的各种异常场景，例如功能上有互斥关系或者有锁机制的场景、网络闪断、数据库主机切到备机、数据库执行备份操作等场景，重点检查异常场景发生时是否会导致系统出现响应异常等各类性能问题	针对核心交易	异常场景

## 2.2. 性能测试场景

根据 MeetHere 项目的实际需求，确定本次性能测试场景为：

- 独立场景
- 混合场景

## 2.3. 重点测试策略

### 2.3.1. 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，按照优先级先后执行，保证系统上线后关键业务的稳定性。

2.3.2. 重点测试交易

- 用户登录
- 浏览场馆
- 场馆预约
- 订单管理
- 个人信息管理

3. 测试案例设计

3.1. 生产压力分析

MeetHere 项目的生产压力主要来源于对核心交易“浏览场馆”和“预约场馆”的并发访问，当并发用户数达到一定数量时，可能会导致信息加载过慢，系统卡顿等问题。因此我们拟通过独立场景和混合场景的性能测试来探索系统性能能否满足未来1-3年的运行要求，以及如果不足，系统性能的瓶颈可能会出现在哪些地方。

3.2. 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
疲劳场景	<1秒	12小时	>1万	>100万	100%	<50%	100	无

### 3.3. 测试场景设计

#### 3.3.1. 独立场景

对于独立场景测试，我们计划选取以下关键场景进行独立测试。每个场景测试多组并发，并发数从1逐步增加到200。

Table 3-1 独立场景设计

场景编号	场景描述	测试步骤
1	用户登录 MeetHere 系统	1. 用户进入登录页面 2. 根据包含用户名密码的指定 .csv 文件，多线程并发登录
2	用户浏览新闻界面	1. 用户登录进入 MeetHere 系统 2. 用户获取新闻信息，浏览新闻页面
3	用户浏览场馆信息界面（名称/地点/价格/描述/评论等）	1. 用户登录进入 MeetHere 系统 2. 用户进入场馆信息界面，浏览场馆 3. 用户进入场馆详情页，查看场馆名称/地点/价格/描述/评论等
4	用户新增场馆预约订单	1. 用户登录进入 MeetHere 系统 2. 用户进入场馆信息界面，浏览场馆 3. 用户进入场馆详情页，新增预约订单
5	用户个人信息管理	1. 用户登录进入 MeetHere 系统 2. 用户进入个人信息界面，修改个人信息

Table 3-2 独立场景要求设计

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

#### 3.3.2. 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 4-1 混合场景设计

场景编号	场景描述	测试步骤
1	用户登录、浏览新闻、退出	用户登录 -> 用户浏览新闻界面 -> 用户退出
2	用户浏览场馆并评论	用户登录 -> 用户浏览场馆信息 -> 用户查看场馆相关评论 -> 用户发送评论 -> 用户退出
3	用户浏览场馆并预约	用户登录 -> 用户浏览场馆信息 -> 用户预约场馆 -> 用户退出
4	用户查看个人信息并修改	用户登录 -> 用户查看个人信息 -> 用户修改个人信息并提交 -> 用户退出
5	用户浏览场馆并评论，随后删除自己的评论。之后用户查看个人信息并修改。	用户登录 -> 用户浏览场馆信息 -> 用户查看场馆评论 -> 用户发送评论 -> 用户删除评论 -> 用户查看个人信息 -> 用户退出
7	用户浏览场馆并预约，随后浏览场馆评论，查看个人信息。	用户登录 -> 用户浏览场馆信息 -> 用户预约场馆 -> 用户浏览评论 -> 用户查看个人信息 -> 用户退出

Table 4-2 混合场景要求设计

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

## 4. 测试实施安排

### 4.1. 测试进度

时间	测试安排	操作者
2019-12-26 至 2019-12-27	学习性能测试基本知识，了解性能测试工具 熟悉被测系统，确定典型事物	陈俊杰 项慕凡
2019-12-28 至 2019-12-29	依据测试计划执行测试，收集相关数据	陈俊杰 项慕凡
2019-12-30 至 2019-12-31	测试结果分析	陈俊杰 项慕凡
2019-01-01	完善性能测试的相关报告	陈俊杰 项慕凡

## 4.2. 测试流程

- 针对 MeetHere 项目进行特定的性能测试需求分析
- 制定性能测试计划
- 执行性能测试（录制测试脚本），利用工具生成性能测试报告
- 对性能测试的结果进行分析并对系统进行评估
- 向开发人员反馈系统测试结果

## 4.3. 测试报告需求

Table 5 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

## 4.4.性能缺陷管理

测试过程采用 Quality Center 进行缺陷管理

## 5. 性能测试标准

### 5.1. 启动标准

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

### 5.2. 中止标准

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

### 5.3. 通过标准

系统上线至少满足下面标准：

- 系统无宕机、不响应类的严重性能问题

- 系统响应时间80%达到系统的期望值
- 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
- 柜台类系统要求通过7\*12小时以上的疲劳强度测试
- 电子渠道类系统要求通过7\*24小时以上的疲劳强度测试

## 6. 测试环境规划

### 6.1. 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。

Table 6 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
CentOS	7.6	-	-
MySQL	8.0.18	-	-
Nginx	-	-	-
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	CentOS 7.6 64位 单核 2G 1Mbps / MySQL 8.0.18 Redis	1	购买

### 6.2. 执行环境

- APACHE JMeter 5.2.1
- Chrome 79.0.3945.88
- macOS Catalina 10.15.2 (19C57)
- 3.1 GHz 双核Intel Core i5
- 16 GB 2133 MHz LPDDR3

## 7. 测试风险分析

Table 7 测试风险

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	<ul style="list-style-type: none"><li>测试方获得最大负载压力与实际最大负载有差距</li><li>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</li></ul>	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

## 8. 角色与职责

Table 8 角色与指责

角色	姓名	具体职责	技能要求
性能测试经理	陈俊杰 项慕凡	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	陈俊杰 项慕凡	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	陈俊杰 项慕凡	准备性能测试环境 准备基础数据和测试数据	1. 服务器部署能力 2. 熟悉核心系统、数据库
性能调优人员	陈俊杰 项慕凡	核心主机系统优化	1. 熟悉数据库优化 2. 熟悉应用架构