|  |  |  |
| --- | --- | --- |
| 文件状态：  [ ] 草稿  [√] 正式发布  [ ] 正在修改 | 文件标识： | PT\_Plan |
| 当前版本： | 1.0 |
| 作 者： | 李尚真 |
| 完成日期： | 2020-04-12 |

ACHIEVEIT软件项目管理平台  
性能测试计划

版 本 历 史

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本/状态 | 作者 | 参与者 | 起止日期 | 备注 |
| v0.1 | 李尚真 | 李尚真 | 2020/03/09~2020/3/15 | 性能测试计划初稿 |
| V0.2 | 李尚真 | 李尚真 | 2020/03/16~2020/03/22 | 在初稿基础上对生产压力进行量化 |
| V0.3 | 李尚真 | 李尚真 | 2020/03/23~2020/03/29 | 调整了场景通过标准 |
| V0.4 | 李尚真 | 李尚真 | 2020/03/30~2020/04/05 |  |
| V1.0 | 李尚真 | 李尚真 | 2020/04/06~2020/04/12 | 发布正式版 |

目录

[1 项目概要介绍 1](#_Toc1744)

[1.1 项目简介 1](#_Toc1518)

[1.2 项目成员 1](#_Toc10433)

[1.3 测试范围 1](#_Toc8053)

[1.3.1 范围内 1](#_Toc7878)

[1.3.2 范围外 1](#_Toc13241)

[1.4 前提假设 2](#_Toc17562)

[1.5 测试目标 2](#_Toc18809)

[2 性能测试策略 2](#_Toc24646)

[2.1 性能测试模型 2](#_Toc2003)

[2.2 性能测试场景 4](#_Toc30910)

[2.3 重点测试策略 4](#_Toc11507)

[2.3.1 重点测试原则 4](#_Toc615)

[2.3.2 重点测试交易 4](#_Toc22279)

[3 测试案例设计 5](#_Toc6302)

[3.1 生产压力分析 5](#_Toc6685)

[3.2 场景通过标准 5](#_Toc29893)

[3.3 测试场景设计 6](#_Toc29072)

[3.3.1 独立场景 6](#_Toc5723)

[3.3.2 混合场景 6](#_Toc21031)

[4 测试实施安排 6](#_Toc27943)

[4.1 测试进度 6](#_Toc2603)

[4.2 测试流程 7](#_Toc7560)

[4.3 测试报告需求 7](#_Toc18149)

[4.4 性能缺陷管理 7](#_Toc30815)

[5 性能测试标准 7](#_Toc4976)

[5.1 启动标准 7](#_Toc19249)

[5.2 中止标准 7](#_Toc709)

[5.3 通过标准 7](#_Toc4722)

[6 测试环境规划 8](#_Toc1889)

[6.1 部署环境 8](#_Toc10012)

[6.2 执行环境 8](#_Toc19715)

[7 测试风险分析 8](#_Toc2553)

[8 角色与职责 9](#_Toc19780)

[表 1 性能测试类型 3](#_Toc14170)

[表 2 场景通过标准 5](#_Toc2915)

[表 3 独立场景设计 6](#_Toc8981)

[表 4 混合场景设计 6](#_Toc20837)

[表 5 测试报告需求 7](#_Toc516)

[表 6 服务器配置 8](#_Toc10360)

[表 7 客户机配置 8](#_Toc28896)

[表 8 成员职责 9](#_Toc1990)

[图 1 性能测试模型 3](#_Toc8763)

# 项目概要介绍

## 项目简介

本系统名为“软件项目管理平台AchieveIt”，是为了满足客户四月科技有限公司对软件项目管理平台的高度自定义需求而研发的软件项目管理平台。

四月科技有限公司主要为国内外客户开发软件系统，每年需要开展数十个项目。随着公司业务日益增多，由于项目数据的不规范、不完整、不一致而造成的各种错误、混乱已经为公司带来了不断上升的成本损失。为了能够统一管理公司各个项目的信息和数据、提高管理效率、挖掘潜在价值、支持合理决策，公司希望采用一个软件项目管理平台来改善目前的状态。但是，经过调研，四月科技有限公司发现市面上大多数开源软件只具备项目管理某方面的功能，与公司的诉求有较大差距，因此，决定自行研发符合其需求的平台，并命名为：软件项目管理平台 AchieveIt。

## 项目成员

项目成员由华东师范大学软件工程学院的学生团队组成，包括：徐雯蕾、宁静、杨政达、高尚、王浩和李尚真六位同学。

徐雯蕾同学作为项目经理，是项目的主要责任人。

宁静同学负责前端架构设计和功能开发；杨政达同学负责前端功能开发；高尚同学负责数据库和后端架构设计以及后端功能开发；王浩同学负责后端功能开发；李尚真同学负责系统和性能测试的执行和缺陷报告。

## 测试范围

### 范围内

本系统的最终成果将是一个软件项目管理平台，该软件项目管理平台是一个基于网页开发的在线项目管理平台。因此，性能测试的范围将集中在平台内部所需要处理的业务部分。

### 范围外

由于本系统服务于四月科技有限公司，因此将不可避免地与四月科技有限公司的其他内部系统产生一定的交互，但是其他系统的性能不在本系统的可控范围内，所以将不对其他系统进行性能测试。

## 前提假设

对本系统的性能测试将基于以下前提实施：

1. 和本系统产生交互的其它系统总是会对本系统所发出的请求给出合理的响应，无论本系统发出的请求正常与否；
2. 在单一用户的状态下，本系统可以正常完成所有业务流程，包括对异常流的处理；
3. 本系统运行所依赖的软件和硬件始终能保持正常运行。

## 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。

图 1性能测试模型

根据图 1所示的性能测试模型，可以设计如表 1所示的性能测试类型

表 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 |

## 性能测试场景

根据项目实际，确定本次性能测试场景包含以下场景的性能测试：

1. 独立场景
2. 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. 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。
3. 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

1. 登录
2. 查询项目信息
3. 管理项目工时信息

# 测试案例设计

## 生产压力分析

根据四月科技有限公司所提需求的描述，公司每年将开展数十个项目，按照一般公司人力资源配备和业务流程，可以合理地推测这些项目将被近乎均匀地分散在一年内，因此“创建项目”的场景在同一时刻同时执行的可能性很低，对服务器产生的压力也很有限。

在“更新项目”这一类的操作中，由于每个项目都有其独立于其他项目的周期，而每个项目被创建的时间被近乎均匀地分散在一年内，因此可以做出合理估计：对于“管理项目人员信息”、“管理项目人员权限”、“管理项目功能列表”、“管理项目状态”、“管理项目缺陷信息”、“管理项目设备信息”这7类随项目进度进行操作并且操作者为不特定的少数具有较高权限的人的功能，集中在同一时刻同时进行操作的可能性极低，在生产环境中对于服务器的压力可以忽略不计。

由于所有公司员工都是平台用户，因此对于所有平台用户均可操作或操作频率较高的3类功能（“登录”、“查询项目信息”、“管理项目工时信息”），预计将对服务器的负载产生显著的影响。

预期公司人数为1000人，因此本系统支持的在线用户数需要达到1000人。在每日申报工时的时段内，所有公司员工都将在某个时间点登录系统进行工时申报，本系统应当支持200人的并发用户数。对于申报工时这一行为，涉及三个阶段，第一个阶段为用户登录系统并进入工时申报页面，第二个阶段为填写完工时信息并提交，第三个阶段为刷新工时页面，这三个阶段分别向服务器请求3次、1次、1次。因此，本系统最少需要支持200\*(3+1+1)=1000次/秒的请求量。

## 场景通过标准

表 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** | **并发/在线上限** | **在线思考时间设置** |
| 独立场景 | <3秒 | 30分钟 | 1000 | >5000 | 95% | 200 | 无 |
| 混合场景 | <3秒 | 1小时 | 1000 | >1万 | 95% | 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。

表 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 | N/A | 20分钟 | 1个用户/5秒 |
| 100 | <2秒 | >100笔/秒 | 20分钟 | 1个用户/5秒 |
| 200 | <3秒 | >100笔/秒 | 30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。

表 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 | <2秒 | >100笔/秒 | 20分钟 | 1个用户/5秒 |
| 200 | <3秒 | >100笔/秒 | 1小时 | 1个用户/1秒 |

# 测试实施安排

## 测试进度

待系统开发基本完成后，使用脚本进行测试，当天完成，每次有新的代码合并入系统时，都执行一次测试。

## 测试流程

首先进行独立场景测试，在独立场景测试全部完成并且无性能问题后，再执行混合场景测试。

## 测试报告需求

表 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 |

## 性能缺陷管理

测试过程采用人工管理方式

# 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

表 6服务器配置

|  |  |
| --- | --- |
| CPU | Intel(R) Xeon(R) E5-2682 v4 (Broadwell) @ 2.5 GHz |
| 内存 | 2 GB DDR4 |
| 硬盘 | 40 GB（2120 IOPS） |
| 网络带宽（出） | 1 Mbps |
| 网络带宽（入） | 500 Mbps |
| 操作系统 | CentOS 7.7 x64 |

## 执行环境

表 7客户机配置

|  |  |
| --- | --- |
| CPU | Intel(R) Core(TM) i7-8750H @ 2.2 GHz |
| 内存 | 32 GB DDR4 |
| 硬盘 | 256GB SSD + 1TB HDD |
| 网络带宽（出） | 30 Mbps |
| 网络带宽（入） | 300 Mbps |
| 操作系统 | Windows 10 家庭中文版 1909 |

# 测试风险分析

由于各种各样的原因，本性能测试的服务器端配置难以模拟真实生产环境，且服务器和客户机之间通过公网连接，带宽有限，因此，测试得出的数据仅能作为当本系统运行在相同及更高规格服务器上的性能表现，不能对运行在更低规格服务器上的性能表现作任何保证。

# 角色与职责

表 8成员职责

|  |  |
| --- | --- |
| 徐雯蕾 | 项目经理、配置管理 |
| 宁静 | 架构师、前端开发 |
| 杨政达 | 前端开发 |
| 高尚 | 架构师、后端开发 |
| 王浩 | 后端开发 |
| 李尚真 | 测试 |