



# ACHIEVEIT 性能测试计划

文件状态：	文件标识：	
<input checked="" type="checkbox"/> 草稿	当前版本：	V1.0
<input type="checkbox"/> 正式发布	作 者：	G03
<input type="checkbox"/> 正在修改	完成日期：	2020-4-13

## 版本历史

版本/状态	作者	参与者	起止日期	备注
V0.1	G03 陈弈君	苏美澄 叶姝晴 曹威杰 赵宁 陶明沔 陈弈君	2020-3-9 至 2020-3-15	
V0.2	G03 陈弈君	苏美澄 叶姝晴 曹威杰 赵宁 陶明沔 陈弈君	2020-3-16 至 2020-3- 23	
V0.3	G03 陈弈君	苏美澄 叶姝晴 曹威杰 赵宁 陶明沔 陈弈君	2020-3-23 至 2020-3- 30	
V0.4	G03 陈弈君	苏美澄 叶姝晴 曹威杰 赵宁 陶明沔 陈弈君	2020-3-31 至 2020-4-5	
V1.0	G03 苏美成	苏美澄 叶姝晴 曹威杰 赵宁 陶明沔 陈弈君	2020-4-6 至 2020-4-13	

## 目录

1	项目概要介绍 .....	4
1.1	项目简介 .....	4
1.2	项目成员 .....	4
1.3	测试范围 .....	4
1.3.1	范围内 .....	4
1.3.2	范围外 .....	4
1.4	前提假设 .....	5
1.5	测试目标 .....	5
2	性能测试策略 .....	5
2.1	性能测试模型 .....	5
2.2	性能测试场景 .....	7
2.3	重点测试策略 .....	7
2.3.1	重点测试原则 .....	7
2.3.2	重点测试交易 .....	8
3	测试案例设计 .....	8
3.1	生产压力分析 .....	8
3.2	场景通过标准 .....	8
3.3	测试场景设计 .....	8
3.3.1	独立场景 .....	8
3.3.2	混合场景 .....	9
3.3.3	峰值场景 .....	9
3.3.4	容量场景 .....	10
3.3.5	疲劳场景 .....	11
4	测试实施安排 .....	11
4.1	测试进度 .....	11
4.2	测试流程 .....	11
4.3	测试报告需求 .....	12
4.4	性能缺陷管理 .....	12
5	性能测试标准 .....	12
5.1	启动标准 .....	12
5.2	中止标准 .....	12
5.3	通过标准 .....	13
6	测试环境规划 .....	13

6.1	部署环境.....	13
6.2	执行环境.....	13
7	测试风险分析 .....	14
8	角色与职责.....	14

## 1 项目概要介绍

### 1.1 项目简介

该项目是一个为了能够帮助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，支持合理决策的项目管理系统。

### 1.2 项目成员

责任人：G03 小组

项目经理：陈弈君

小组成员：苏美澄、叶姝晴、曹威杰、赵宁、陶明沔、陈弈君

### 1.3 测试范围

#### 1.3.1 范围内

1. 用户鉴权
2. 项目新建
3. 项目查看与编辑
4. 项目检索
5. 项目成员管理功能
6. 项目工时管理功能
7. 项目权限管理功能
8. 项目设备管理功能
9. 项目风险管理
10. 项目缺陷管理
11. 项目功能管理
12. 项目状态管理

#### 1.3.2 范围外

1. 项目需求的导出与导入
2. 项目归档模块
3. 用户信息的导入

## 4. 消息模块

### 1.4 前提假设

功能测试完成，缺陷问题报告单已解决。

### 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 80%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。

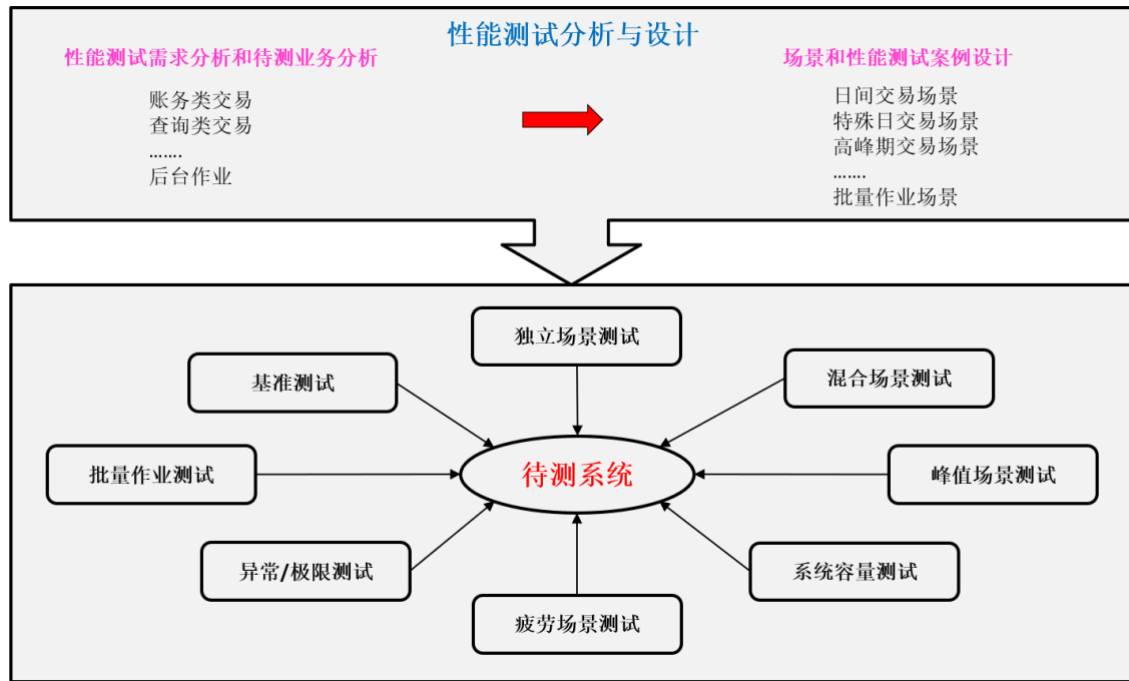


Figure 1 性能测试模型

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访问量应接近峰值时段的访问量，甚至超过系统全天的访问量	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访问量应接近峰值时段的访问量，甚至超过系统全天的访问量	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访问量	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	针对核心操作	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	针对核心操作	疲劳场景

	测试，在压力时间内通过的访问量应为全天访问量 3 倍以上		
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	针对核心操作	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	针对核心操作	异常场景

## 2.2 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

峰值场景

容量场景

疲劳场景

## 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，对访问量大、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

## 3 测试案例设计

### 3.1 生产压力分析

对于本系统的开发基于 1Core/2GB-RAM/1Mbps 的云服务器开发, 上下行速率又有很大限制。因此产品性能将会受到一定影响。本次性能测试主要目的是为系统寻找出性能基准, 以便日后的拓展。同时测试在极限情况下系统的应对能力。

### 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	>300	>500	100%	<70%	100	无
混合场景	<2 秒	1 小时	>400	>750	100%	<80%	150	无
峰值场景	<5 秒	2 小时	>800	>1000	95%	<95%	200	无
容量场景	<10 秒	2 小时	>1000	>2000	95%	<95%	300	无
疲劳场景	<5 秒	2 小时	>400	>5000	90%	<95%	150	无

### 3.3 测试场景设计

编号	测试种类	场景类型	优先级	场景名称	脚本名称	测试步骤	并发用户数	加压方式			预估时间(分钟)
								加载方式	持续时间	退出方式	
1	性能测试	独立场景	高	用户登录	用户登录	1. 进入主页 2. 输入账号密码 3. 点击登录	1/20/50/100	1个/1秒	1/5/5/10分钟	1个/5秒	39
2	性能测试	独立场景	高	项目新建	项目新建	1. 进入新建项目页 2. 提交项目信息	1/20/50/100	1个/1秒	1/5/5/10分钟	1个/5秒	39
3	性能测试	独立场景	高	项目编辑	项目编辑	1. 进入项目管理页 2. 修改一个项目的信息	1/20/50/100	1个/1秒	1/5/5/10分钟	1个/5秒	39
4	性能测试	独立场景	高	项目检索	项目检索	1. 进入项目检索页 2. 输入一个项目名称进行检索	1/20/50/100	1个/1秒	1/5/5/10分钟	1个/5秒	39
5	压力测试	混合场景	高	项目成员管理相关业务	查看所有项目成员 新增项目成员 删除项目成员	1. 进入项目成员管理页 2. 查看某一项目所有成员 1. 进入项目成员管理页 2. 添加某一项目成员 1. 进入项目成员管理页 2. 删除某一项目成员	20/50/100/150	1个/1秒	5/10/10/20分钟	1个/5秒	77
6	压力测试	混合场景	高	项目工时管理相关业务	项目成员提交工时单 项目经理审核工时单	1. 进入项目工时管理页 2. 对某一项目提交工时单 1. 进入项目工时管理页 2. 查看某一成员工时信息	20/50/100/150	1个/1秒	5/10/10/20分钟	1个/5秒	77
7	峰值测试	峰值场景	高	项目工时管理相关业务	项目成员提交工时单	1. 进入项目工时管理页 2. 查看某一成员工时信息	140/200	1个/1秒	20/30分钟	1个/5秒	84
8	峰值测试	峰值场景	高	项目设备管理相关业务	项目成员租借新设备 项目成员归还设备	1. 进入项目设备管理页 2. 提交新设备信息 1. 进入项目设备管理页 2. 归还设备	140/200	1个/1秒	20/30分钟	1个/5秒	84
9	容量测试	容量场景	高	项目工时管理相关业务	项目成员提交工时单	1. 进入项目工时管理页 2. 查看某一成员工时信息	50/100/150/200	1个/1秒	10/10/20/20分钟	1个/10秒	152
10	疲劳测试	疲劳场景	高	项目工时管理相关业务	项目成员提交工时单	1. 进入项目工时管理页 2. 对某一项目提交工时单	140/200	1个/1秒	30/60分钟	1个/10秒	147

#### 3.3.1 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 100。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	1 分钟	1 个用户/1 秒
20	<1 秒	N/A	5 分钟	1 个用户/1 秒
50	<1 秒	>50 笔/秒	10 分钟	1 个用户/1 秒
100	<1 秒	>100 笔/秒	10 分钟	1 个用户/1 秒

### 3.3.2 混合场景

对于相关联的关键操作，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 150。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/1 秒
50	<1 秒	>100 笔/秒	20 分钟	1 个用户/1 秒
100	<2 秒	>200 笔/秒	20 分钟	1 个用户/1 秒
150	<2 秒	>300 笔/秒	20 分钟	1 个用户/1 秒

### 3.3.3 峰值场景

通过分析生成数据，多数系统日间各个峰值共有两个时段，上午 9:00-12:00 的 3 个小时；下午 13:00-17:00 点的 4 个小时，因此，峰值最长持续时间 4-5 小时。考虑到系统的扩展性及实际情况，峰值测试共测试三组：第一组测试正常峰值压力的处理能力；第二组测试 2 倍、4 倍峰值压力下的系统处理能力，以满足未来的业务发展需求，测试时各组中的交易按照业务比例同时加载。

第一组：先用生成峰值时压力的 70% 加压 1 小时作为基准压力，之后采用当前生产上的实际峰值压力并发进行加压。

Table 5 峰值场景设计 1

并发数	响应时间要求	TPS 要求	加压时间	加压方式
生产压力的 70%	<2 秒	>100 笔/秒	30 分钟	1 个用户/1 秒
生产压力	<3 秒	>100 笔/秒	30 分钟	1 个用户/1 秒

第二、三组：分别采用生产压力的 2 到 4 倍来进行加压，第二组达标后再进行后面的测试。

Table 6 峰值测试设计 2

并发数	响应时间要求	TPS 要求	加压时间	加压方式
生产压力的 2 倍	<5 秒	>100 笔/秒	30 分钟	1 个用户/5 秒
生产压力的 4 倍	<10 秒	>100 笔/秒	30 分钟	1 个用户/5 秒

对于性能较好的系统，可以直接加压到最大并发数，持续加压 5 小时来测试系统的峰值处理能力。

### 3.3.4 容量场景

容量测试从生产压力作为基准压力来加压，逐步加压到生产压力的 5-10 倍，在这个过程中得出如下结果：

- 1) 最佳处理能力/最佳并发数：最佳处理能力是指不断增大压力的过程中，在业务处理能力与交易响应时间等性能指标满足用户性能需求的前提下，系统所能达到的最大处理能力，系统初始达到最佳处理能力时对应的并发数可以称为最佳并发数。可以看出，在满足用户性能需求这一原则下，最佳处理能力往往不是最大处理能力，因此，对于在很小压力下，业务处理能力或响应时间已经不满足用户性能需求的系统，不讨论最佳处理能力与最佳并发数两个性能指标。
- 2) 最大处理能力/最大并发数：最大处理能力是指系统系统所能支撑的最大业务吞吐量，可以用“万笔/小时”或“笔/秒”等来度量。对于性能较好的系统，最大并发数是指响应时间和业务吞吐量等性能指标满足用户性能需求的前提下，系统所能支持的最大并发数。可以看出在满足用户性能需求这一原则下，系统在加压过程中可能先达到最大处理能力，如果这是各项指标仍然满足性能需求，则需要继续加压才能测出最大并发数，即将后继加压过程中业务处理能力或交易响应时间等性能指标开始不满足用户需求前的并发数作为最大并发数。可以看出，对于性能较好的系统最大处理能力和最大并发数往往不存在对应关系。对于性能较差的系统，最大并发数习惯上定义为业务吞吐量达到最大值时所对应的并发数，还有一种做法是将加压过程中响应时间开始不满足用户需求前的并发数定义为最大并发数，实际上，对于响应时间与业务吞吐量等关键性能指标不能满足用户性能需求的系统，讨论最大处理能力和最大并发数没有实际意义，这一点与性能较差时不讨论最佳处理能力与最佳并发数这两个性能指标的出发点是一致的。对于此类性能较差的系统，首先需要做的是对系统进行全面优化，然后再进一步测试系统的最大处理能力/最大并发数。
- 3) 系统容量上限：系统交易成功率<99%时的并发数
- 4) 系统上限：系统变得极其缓慢、交易大量失败甚至停止服务时的并发数。对于各个渠道，定义为交易成功率<90%时的并发数。

Table 7 容量测试场景设计

并发数	响应时间要求	TPS 要求	加压时间	加压方式
50	<1 秒	>100 笔/秒	20 分钟	1 个用户/10 秒
100	<1 秒	>100 笔/秒	20 分钟	1 个用户/10 秒
150	<1 秒	>100 笔/秒	20 分钟	1 个用户/10 秒
200+	<2 秒	>100 笔/秒	每增加 100 个用户后续持续加压 10 到 20 分钟	1 个用户/5 秒
300+	<5 秒	>100 笔/秒		1 个用户/5 秒

### 3.3.5 疲劳场景

联机与批处理每天先后轮流执行，连续测试一周。9:00-21:00 执行联机交易，21:00-次日 9:00 执行 EOD。测试过程可以根据实际情况进行调换。

联机交易场景采用正常压力持续加压，加压分为 3 个时段进行。这个过程的目的是为模拟全行一天 24 小时的业务。

Table 8 疲劳测试场景设计

并发数	响应时间要求	TPS 要求	加压时间	加压方式
生产压力*70%	<1 秒	>100 笔/秒	30 分钟	1 个用户/10 秒
生产压力	<1 秒	>100 笔/秒	1 小时	1 个用户/10 秒
生产压力*70%	<1 秒	>100 笔/秒	30 分钟	1 个用户/10 秒

## 4 测试实施安排

### 4.1 测试进度

Table 9 测试进度安排

编号	任务	开始日期	结束日期	参与人员
1	熟悉被测系统， 确定典型事务	2020.4.10	2020.4.11	测试人员 项目经理
2	搭建测试环境， 录制典型事务的脚本， 增强脚本	2020.4.12	2020.4.15	测试人员
3	执行测试并收集相关数据	2020.4.16	2020.4.17	测试人员
4	数据分析，编写测试报告	2020.4.18	2020.4.19	测试人员

### 4.2 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。

- 1) 性能测试需求分析与设计:
  - i. 熟悉被测系统, 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。
- 2) 性能测试实现:
  - i. 创建虚拟用户脚本: 将最终用户活动捕获到自动脚本中。
  - ii. 定义场景: 使用 Jmeter 设置负载测试环境。
  - iii. 运行场景: 通过 Jmeter 驱动、管理和监控负载测试。
- 3) 生成性能测试报告:
  - i. 使用 Jmeter 创建图和报告并评估性能。

### 4.3 测试报告需求

Table 10 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
Issue	遇到缺陷后	测试负责人	开发人员, 项目经理	测试分支, 缺陷内容
文档	各测试阶段结束	测试负责人	测试, 项目经理	性能测试报告

### 4.4 性能缺陷管理

Github Issue 进行缺陷记录管理, 以及修复情况

## 5 性能测试标准

### 5.1 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) 关键操作通过冒烟测试

### 5.2 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) 系统的操作成功率低于 95%

## 5.3 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

## 6 测试环境规划

### 6.1 部署环境

测试环境即真实部署环境，采用阿里云轻量级服务器。具体配置为 1 核 2GB 内存，50GB 的存储空间以及 5MBps 的上下行速度。

Table 11 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	1	开源软件下载
MySQL	5.7	1	开源软件下载
Chrome/Firefox	Latest	1	开源软件下载
硬件环境（网络、设备等）			
名称	版本	数量	获得途径
腾讯云轻量级云服务器	/		购买

### 6.2 执行环境

Table 12 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	1	开源软件下载
MySQL	5.7	1	开源软件下载
Chrome/Firefox	Latest	1	开源软件下载
硬件环境（网络、设备等）			
名称	版本	数量	获得途径
腾讯云轻量级云服务器	/		购买

## 7 测试风险分析

Table 13 风险分析

风险因素	可能结果	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项目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

## 8 角色与职责

Table 14 角色与职责

角色	姓名	具体职责	技能要求
性能测试经理	G03	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	G03	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