



性能测试计划

摘要

该计划主要描述 Timeline 的性能测试计划，具体包括项目介绍、测试策略、项目计划、测试标准等内容。

Timeline 开发组

华东师范大学 计算机科学与工程

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5
 - 1.3.1 范围内5
 - 1.3.2 范围外5
 - 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6
 - 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7
- 3 测试案例设计.....7
 - 3.1 生产压力分析7
 - 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8
 - 3.3.3 峰值场景9
 - 3.3.4 容量场景9
 - 3.3.5 疲劳场景10
 - 3.3.6 批处理场景 **Error! Bookmark not defined.**
 - 3.3.7 极限测试 11
 - 3.3.8 异常场景 **Error! Bookmark not defined.**
- 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 11
 - 4.4 性能缺陷管理 11
- 5 性能测试标准..... 11
 - 5.1 启动标准 11
 - 5.2 中止标准 12

5.3	通过标准	12
6	测试环境规划.....	12
6.1	部署环境	12
6.2	执行环境	12
7	测试风险分析.....	12
8	角色与职责.....	Error! Bookmark not defined.

图目录

Figure 1 Timeline 设计模板4

Figure 2 性能测试模型6

表目录

Table 1 性能测试类型6

Table 2 场景通过标准8

Table 3 独立场景设计8

Table 4 混合场景设计9

Table 5 峰值场景设计 19

Table 6 峰值测试设计 29

Table 7 容量测试场景设计10

Table 8 疲劳测试场景设计10

Table 9 测试报告需求11

1 项目概要介绍

1.1 项目简介

Timeline 是社交软件中按时间显示动态功能的简化版，需要桌面应用和 Web 应用两种形式，界面框架如下图描述。

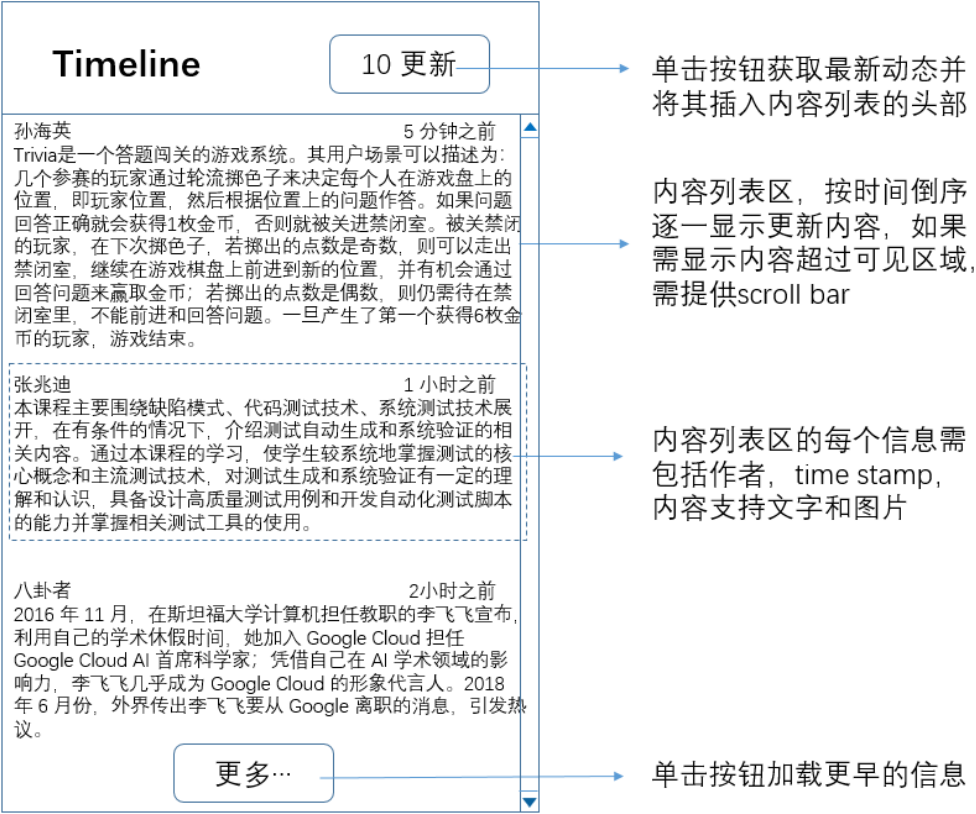


Figure 1 Timeline 设计模板

为简化问题，Timeline 显示的动态信息对登录系统的所有用户一致，即任何用户发布更新删除的内容对所有人有效且发布内容只支持文字和图片，也无需复杂的用户管理功能，只需实现基本的登录注册即可。

1.2 项目成员

Timeline 项目包括四位项目成员。其中，开发人员两位，由张宇宸（网页版本）、毕冉（桌面版本）完成；测试人员两位，由符嘉铭（系统测试）、吴文翰（性能测试）完成。

1.3 测试范围

1.3.1 范围内

测试包括桌面版本测试和网页版本测试两部分，其中这两部分的测试将集中于登录、验证、刷新等相关功能。性能测试的主体在于对这些功能进行，并对不同场景下的不同场景进行分析。以上都是本次测试的性能测试范围。

1.3.2 范围外

项目的真实环境下运行，具体包括部署到未知的服务器，以及在每个人的计算机上的个性化检测报告并不包括在本次性能测试计划中。性能测试将利用正常的、通用的计算机结构对该程序进行测试。测试结果不一定真实反应所有计算机的真实情况，每台电脑都有对应的性能差异。

1.4 前提假设

该系统性能测试将假设用户的操作是正常的、可复现的。用户自身利用脚本机进行大量的生成与伪造用户数、进程数不在本次性能测试的常规环境之中。以这些目标为目标的测试，及其带来的程序问题、系统问题、性能问题。由于其不满足前提假设，因而不在于测试范围内。

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发展需求；
- 3) 在业务处理能力方面，能满足当前业务增长需求；
- 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。下页图展示了性能测试的相关分析与设计。

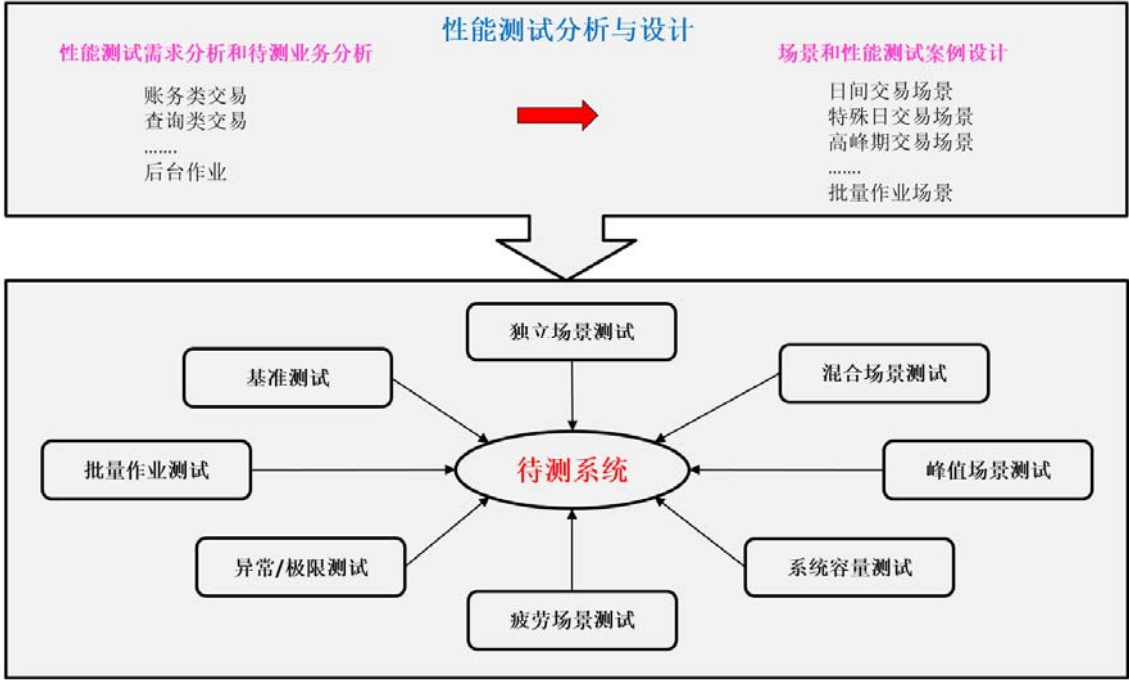


Figure 2 性能测试模型

根据 Figure 2 所示的性能测试模型，可以设计相关性能测试类型。如下表所示：

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	针对核心交易	疲劳场景

	全天交易量 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

根据项目实际，确定本次性能测试场景，性能测试将包括以下场景：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、常用操作开展全面的独立场景测试；
- 2) 对关键、常用操作开展全面的峰值、容量测试；
- 3) 多渠道发起以关键、常用操作为主的疲劳测试；
- 4) 在集成性能测试方面，对常用功能或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 重点测试内容

重点测试内容包括登录测试、刷新测试、发布测试等。应确保这类测试的通过和正常运行。

3 测试案例设计

3.1 生产压力分析

该项目通常用于用户新建并查阅相关信息。单日访问数不定，最大可以达到 20000 次左右。用户在相同时刻产生的操作数（即真实并发数）峰值大约可在

200 次左右，平均约 20 次左右。

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%	50	无
混合场景	<1 秒	1 小时	>1 万	>10 万	100%	<80%	50	无
峰值场景	<2 秒	1 小时	>1 万	>10 万	100%	<80%	50	无
容量场景	<3 秒	4 小时	>1 万	>50 万	100%	<80%	50	无
疲劳场景	<1 秒	12 小时	>1 万	>100 万	100%	<50%	50	无

3.3 测试场景设计

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 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 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 秒钟	直接加压
10	<1 秒	N/A	5 秒钟	2 个用户/1 秒
20	<1 秒	N/A	10 秒钟	4 个用户/2 秒
50	<1 秒	N/A	20 秒钟	5 个用户/2 秒
100	<1 秒	N/A	20 秒钟	10 个用户/2 秒
200	<1 秒	N/A	20 秒钟	20 个用户/2 秒
500	<1 秒	N/A	20 秒钟	50 个用户/2 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 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	<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

3.3.3 峰值场景

通过相关同类产品的分析，我们可以得到以下结论：对于以社交等为导向的软件，用户高峰往往出现在晚间，即在家休息时的使用。由于该产品没有移动端，因此上班时间用户数量不大。我们考虑晚上 7:00-10:00 作为高峰阶段。

以下我们将设计不同阶段的峰值场景，来满足设计需求。

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%	<1 秒	>100 次/秒	1 小时	1 个用户/5 秒
生产压力	<1 秒	>100 次/秒	4 小时	1 个用户/5 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	<1 秒	>100 次/秒	5 小时	1 个用户/5 秒
生产压力的 4 倍	<1 秒	>100 次/秒	5 小时	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	加压时间	加压方式
50	<1 秒	20 分钟	1 个用户/10 秒
100	<1 秒	20 分钟	1 个用户/10 秒
150	<1 秒	20 分钟	1 个用户/10 秒
200+	<1 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+	<1 秒		1 个用户/5 秒
500+	<3 秒		1 个用户/2 秒
1000+	<5 秒		1 个用户/1 秒
2000+	<10 秒		1 个用户/1 秒

注：以上“响应时间要求”各个渠道根据实际情况进行要求

3.3.5 疲劳场景

通过长时间的不断运行，以达到疲劳场景的效果。具体的场景设计如下：

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	20 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

3.3.6 极限测试

极限测试将对系统边界进行测试，本次性能测试将探究强并发情况下的测试情况，并由此分析系统的并发界限。具体计划如下：

并发数	响应动作	TPS 要求	加压时间	加压方式
10	输出浏览信息页面	>10 次/秒	2 分钟	逐步加压
100	输出浏览信息页面	>100 次/秒	2 分钟	逐步加压
500	输出浏览信息页面	>500 次/秒	2 分钟	逐步加压
1000	输出浏览信息页面	>1000 次/秒	2 分钟	逐步加压

4 测试实施安排

4.1 测试进度

测试将在开发后进行。对项目依次进行相关性能测试。

4.2 测试流程

分步进行，对每种性能测试进行检测。

4.3 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

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

- 1) 测试环境满足计划需求
- 2) 基准参数配置完成校验
- 3) 关键交易通过性能测试

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) 系统的交互成功率低于 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

6 测试环境规划

6.1 部署环境

系统在本机进行部署。

6.2 执行环境

依据不同需求执行，今后将部署到相应服务器上。具体服务器的性能要求视实际需求与业务量决定。

7 测试风险分析

性能测试将直接与数据库交互，有可能导致数据库崩溃情况（由于数据量过大）。同时，对本机的性能测试，如果出现了性能泄露，会导致测试机器运行较为缓慢，甚至出现蓝屏等问题。因此，正确的在合适机器上进行性能测试有助于测试风险的降低。通常，测试控制器与测试机器应当分开完成，但受制于项目部署环境，无法做到这一点。