<HONEYCOMB>

测试报告

版本 <1.0>

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| 2018.9.8 | 0.1 | 添加部分基础用例的测试报告 | 曹金坤 |
| 2018.9.8 | 0.2 | 添加部分测试用例 | 曹金坤，谢添翼 |
| 2018.9.8 | 0.5 | 添加性能测试的部分 | 朱锦昊 |
| 2018.9.8 | 1.0 | 完善报告的其他内容 | 曹金坤，朱锦昊，马轲，谢添翼 |

[1. 简介 4](#_Toc524289433)

[1.1 目的 4](#_Toc524289434)

[1.2 范围 4](#_Toc524289435)

[1.3 定义、首字母缩写词和缩略语 4](#_Toc524289436)

[1.4 参考资料 4](#_Toc524289437)

[2. 测试概要 4](#_Toc524289438)

[3. 测试环境 4](#_Toc524289439)

[4. 测试结果及分析 4](#_Toc524289440)

[4.1 需求覆盖率及缺陷分布 4](#_Toc524289441)

[4.2 缺陷严重程度 5](#_Toc524289442)

[5. 缺陷清单 5](#_Toc524289443)

[5.1 功能性缺陷 5](#_Toc524289444)

[5.2 非功能性缺陷 6](#_Toc524289445)

[6. 测试结论与建议 6](#_Toc524289446)

测试报告

# 简介

## 目的

说明针对HoneyComb项目的测试用例和其他测试需求形成的结果。

## 范围

本文档定义的内容，涉及到的范围包括Honeycomb项目包含的所有源代码、文档和媒体文件。

## 定义、首字母缩写词和缩略语

SSH: Spring, Struts, Hibernate形成的后端框架

## 参考资料

《HoneyComb项目立项建议书》*,2018.6.20,Honeycomb小组*

《HoneyComb需求规约》*,2018.6.20,Honeycomb小组*

《HoneyComb测试用例》,*2018.9.6,HoneyComb小组*

# 测试概要

测试时间：2018.9.8 – 2018.9.9

测试人员：曹金坤、谢添翼、马轲、朱锦昊

测试方法：

1. 前端测试：基于使用用例的基础上设置测试用例进行手动测试
2. 后端测试：基于自动化测试程序，对主要的后端功能函数进行测试
3. 性能测试: 基于loadrunner进行模拟的压力和并发测试；
4. 部署测试：手动进行

# 测试环境

测试环境：

1. Macbook pro 2015 early，8G内存
2. Windows10, i7-7代标压，8G内存
3. Ubuntu 14.04, 2G内存

# 测试结果及分析

## 需求覆盖率及缺陷分布

*[需求覆盖率是指经过测试的需求/功能和软件需求规约中所有需求/功能的比值，通常情况下要达到100%的目标。需给出功能测试及非功能测试的需求覆盖情况。对于功能测试，需给出各个主要功能模块的需求覆盖率。如果需求覆盖率未达到100%可在备注项中说明原因及未测试内容。缺陷率是指本项缺陷占总缺陷数的百分比。请尽量使用图、表进行描述。]*

**表1 需求覆盖率及缺陷分布**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **测试内容** | | **测试**  **用例数** | **需求**  **覆盖率** | **缺陷数** | **缺陷率** | **备注** |
| **功能项** | **功能项1** |  |  |  |  |  |
| **功能项2** |  |  |  |  |  |
| **……** |  |  |  |  |  |
| **功能项小计** |  |  |  |  |  |
| **非功**  **能项** | **性能** |  |  |  |  |  |
| **可靠性** |  |  |  |  |  |
| **…** |  |  |  |  |  |
|  | **非功能项小计** |  |  |  |  |  |
|  | **总计** |  |  |  |  |  |

## 缺陷严重程度

*[描述缺陷按严重程度分布情况，包括不同严重程度缺陷个数及占缺陷总数百分比等。尽量使用图、表表示。]*

**表2 缺陷严重程度**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **严重程度** | **致命** | **严重** | **一般** | **细微** | **总数** |
| **缺陷个数** |  |  |  |  |  |
| **占缺陷百分比** |  |  |  |  |  |

# 缺陷清单

## 功能性缺陷

*[简要阐明功能缺陷汇总情况，并给出具体功能缺陷列表。]*

**表3 功能性缺陷列表**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **序**  **号** | **缺陷**  **编号** | **严重程度** | **功能模块** | **缺陷标题** | **缺陷描述** | **测试用例编号** |
| 1 |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |
| 3 |  |  |  |  |  |  |

表中相关项说明：

1. 缺陷编号：为每个缺陷赋予一个唯一的编号，可以通过此编号对缺陷进行跟踪。例如：Bug001。
2. 缺陷严重程度：缺陷可以根据严重程度分为以下几种情况。
3. 致命（fatal）：致命的错误，测试执行直接导致系统死机、蓝屏、挂起、或是程序非法退出；系统的主要功能或需求没有实现。
4. 严重（critical）：严重错误，系统的次要功能点或需求点没有实现；数据丢失或损坏。执行软件主要功能的测试用例导致系统出错，程序无法正常继续执行；程序执行过于缓慢或是占用过大的系统资源。
5. 一般（major）：不太严重的错误，这样的缺陷虽然不影响系统的基本使用，但没有很好地实现功能，没有达到预期的效果。如次要功能丧失，界面错误，打印内容、格式错误，提示信息不太正确，或用户界面太差，简单的输入限制未放在前台进行控制，删除操作未给出提示，操作时间长等。
6. 细微（minor）：一些小问题，对功能几乎没有影响，产品及属性仍可使用。如软件的实际执行过程与需求有较小的差异；程序的提示信息描述容易使用户产生混淆；有个别错别字、 文字排列不整齐；辅助说明描述不清楚，显示格式不规范，长时间操作未给用户进度提示，提示窗口文字未采用行业术语，可输入区域和只读区域没有明显的区分标志，系统处理未优化等。
7. 功能模块：所测试并出现该缺陷的功能模块名称。
8. 缺陷标题：描述缺陷的标题。
9. 缺陷描述：对缺陷的详细描述。描述缺陷的重现步骤和问题所在，描述需清晰明了，使研发人员可以按照描述将缺陷重现出来。必要时可以放上相应的截图，以便于研发人员理解问题所在。

## 非功能性缺陷

*[简要阐明非功能缺陷汇总情况，并给出具体非功能缺陷列表。(根据所测系统的实际情况，可选择性地对系统进行必要的非功能性测试，包括系统的性能、可靠性、易用性、安全性、兼容性等。)请参考功能缺陷列表给出相关缺陷清单。]*

**表4 非功能性缺陷列表**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **序**  **号** | **缺陷**  **编号** | **严重程度** | **测试类型** | **缺陷标题** | **缺陷描述** | **测试用例编号** |
| 1 |  |  | 性能 |  |  |  |
| 2 |  |  | 性能 |  |  |  |
| … |  |  | … |  |  |  |
| n |  |  | 可靠性 |  |  |  |

对于系统的并发性能，基于Loadrunner，小组进行了系统的分析，具体的测试结果见附件中的”performance\_analysis”部分，在此处简要进行分析说明：

1. 如图1所示，在50个并发用户的规模下，Honeycomb系统对功能性事务，新用户生成和结束任务的用户退出事务都保持了100%的完成成功率，体现了这个并发规模下，项目性能的可靠性。其中HTTP\_200对应的操作是影响性能的主要瓶颈；
2. 如图2和图3所示，随着并发的用户数量的增加，工程中网络通信的规模成比例上升，且增长的速度较为稳定，符合良好运行情况下的预期；由于loadrunner测试时的模拟用户操作均基于独立的事务分配，所以图4的事务规模随时间变化示意图也表现了这一点；
3. 如图5所示，单个事务的平均响应时间随时间的变化呈明显的周期性，考虑到事务的增加是随时间均匀上升的，这表明在处理事务的过程中可能存在一定程度的阻塞，或者队列操作性能的不均匀。这可能是未来性能优化的一个切入点。

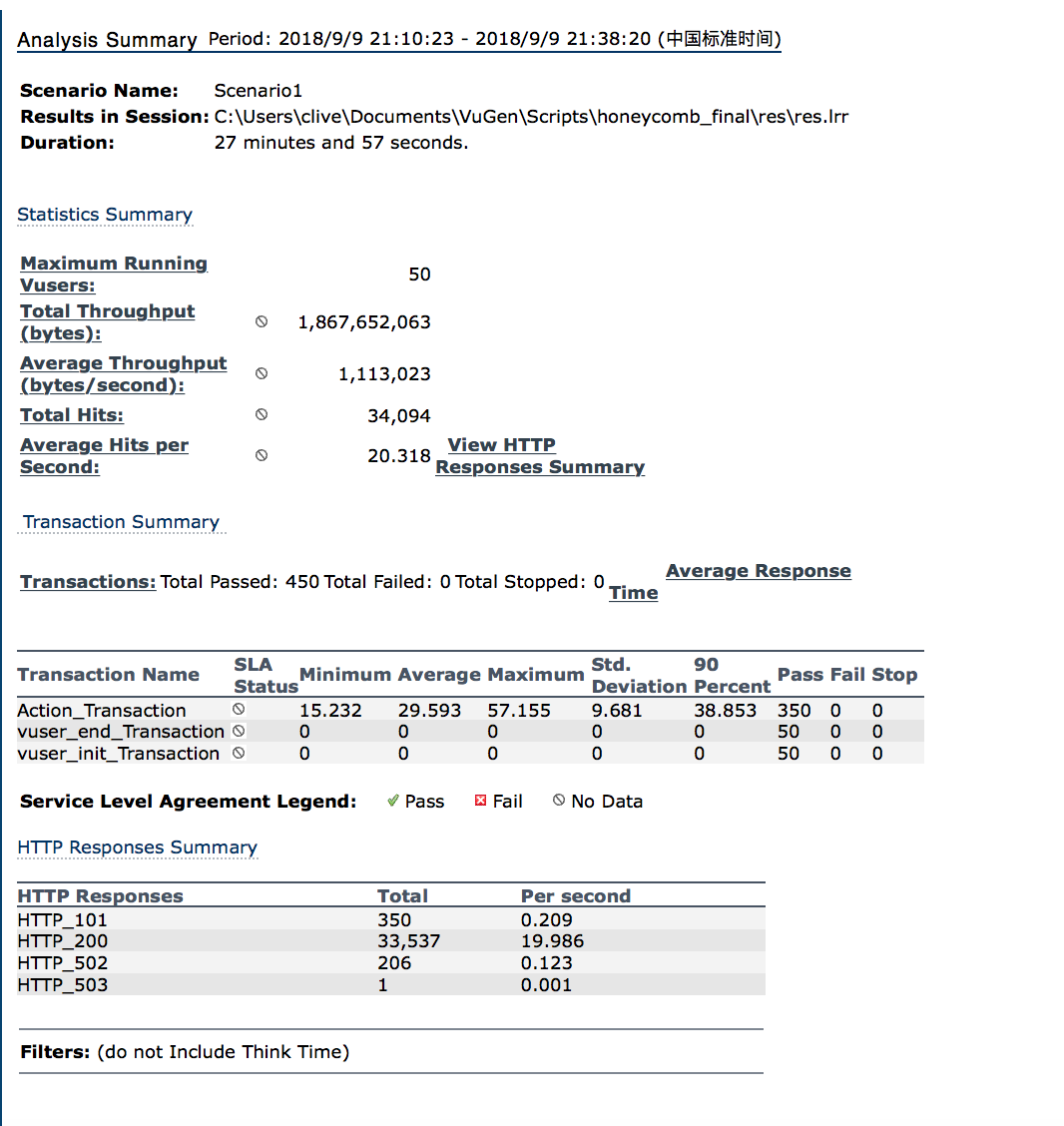


图1.性能测试总览



图2.模拟用户的添加时刻

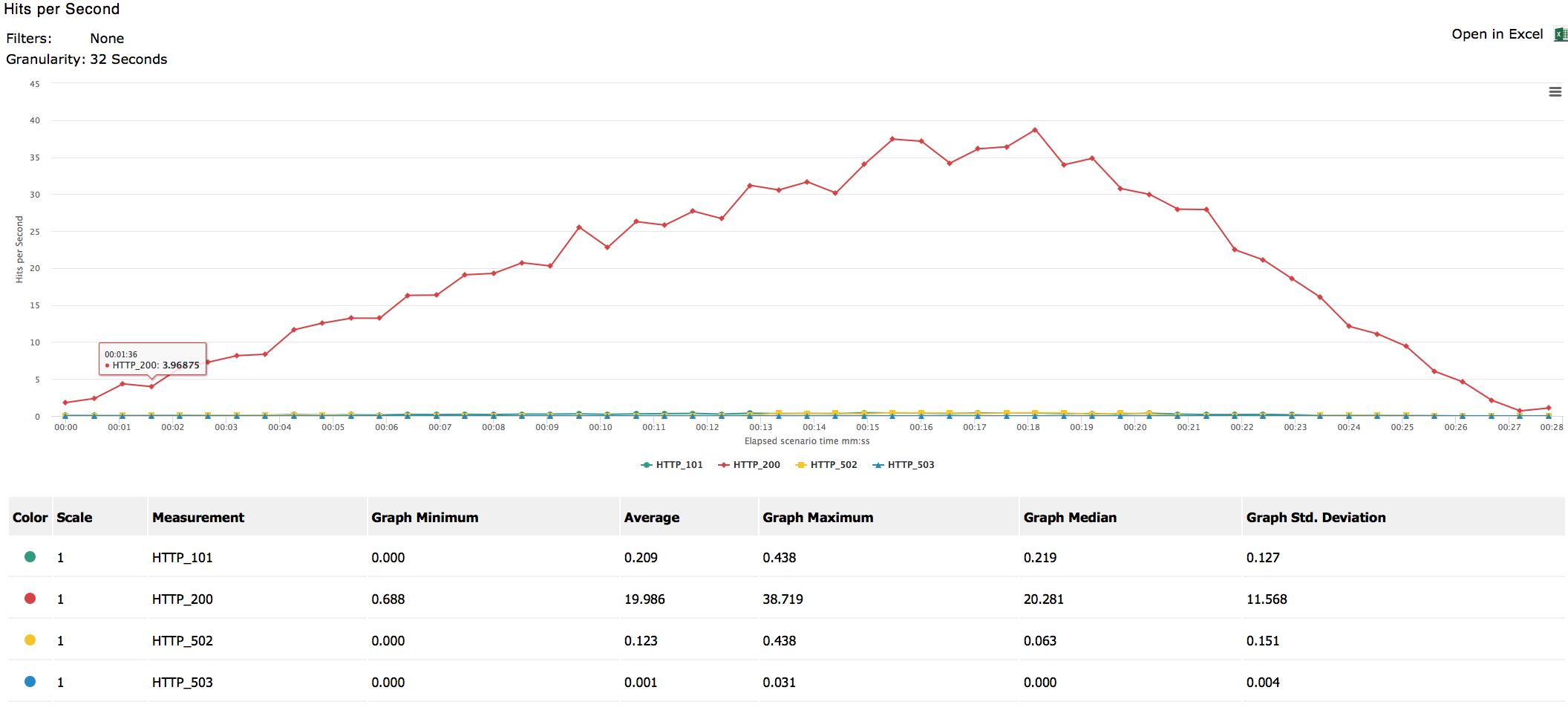


图3. 每秒的信息交互统计

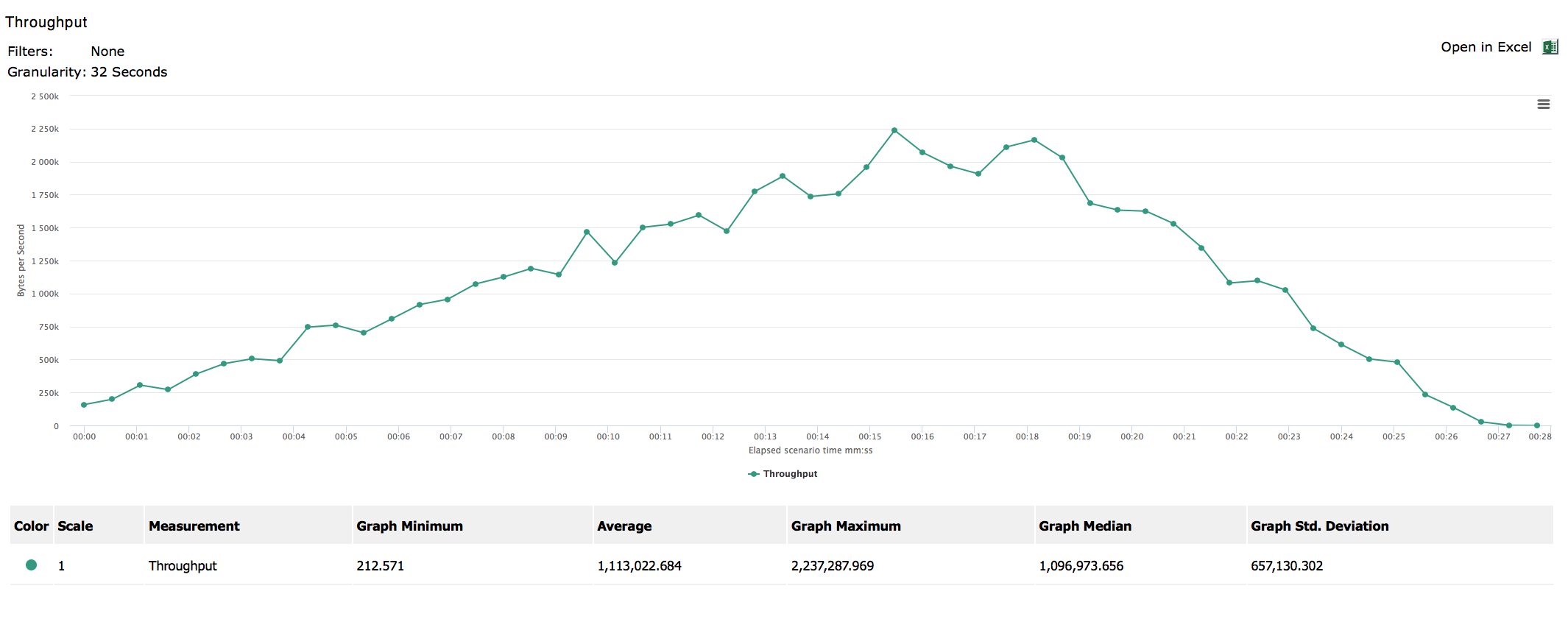


图4.事务运行规模随时间变化

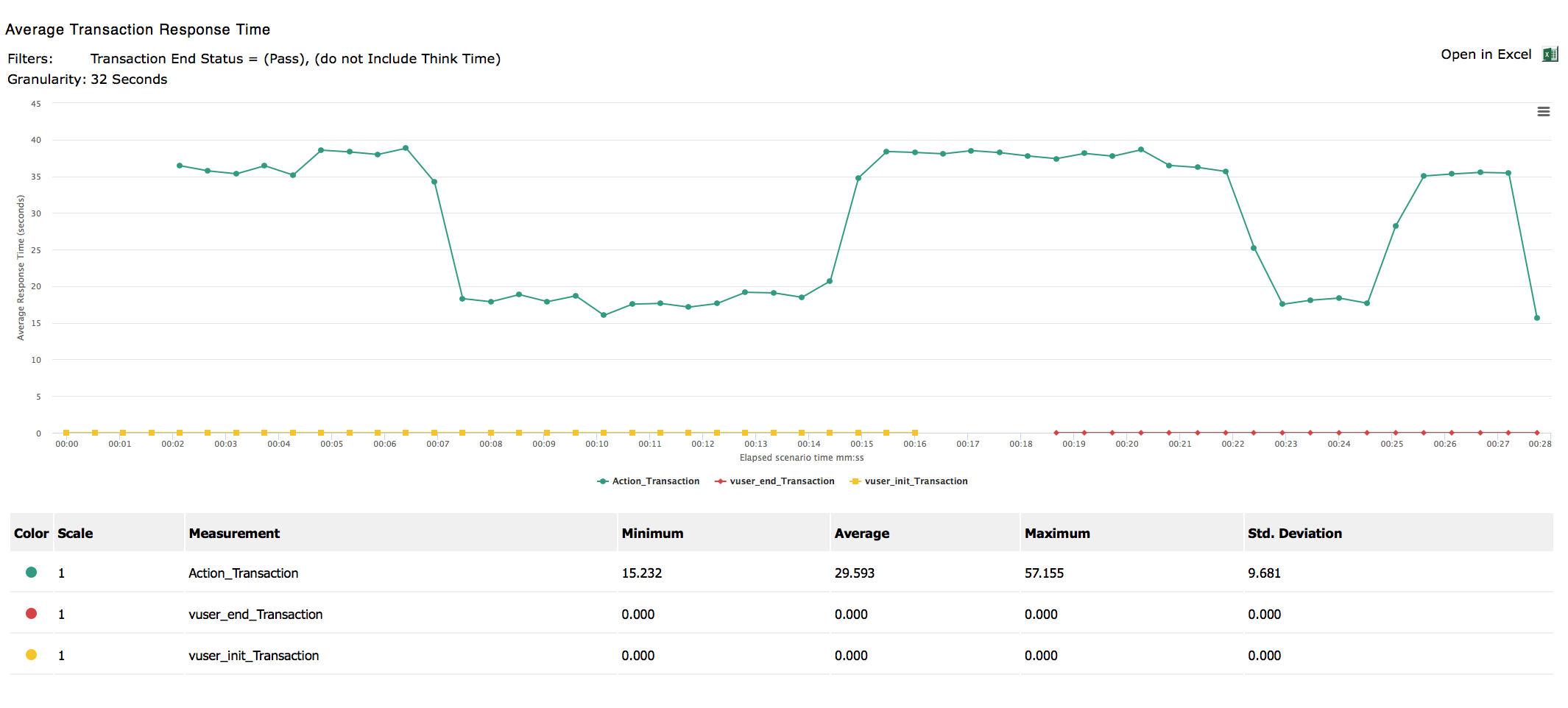


图5.事务平均响应时间随时间变化示意图

# 测试结论与建议

经过前述的测试工作，我们得出了以下的主要测试结论：

1. 工程的功能性需求得到了完整的满足，基本符合立项时的计划，站内私信功能之后因为需求变更没有实现，相关说明已经在项目的总结报告中得到了体现；
2. 少数的功能性测试中出现了一些漏洞，部分测试用例的设计也不够完善，测试过程可以进一步完善；
3. 对于总共40W条的论文数据，在搜索和论文可视化时，工程体现出了可以察觉的延迟，有进一步的优化空间，但是基本满足了功能性的需求；
4. 对于错误的输入、前置条件等负测试用例，测试过程体现了工程基本的健壮性，但是反馈的错误信息有进一步优化的余地，比如对于已注册用户输错密码和登录的用户未注册，前端给出了同样的提示，有时会让用户迷惑；
5. 工程对于大量并发的支持性较弱，在loadrunner的压力测试中体现出来了较大的不足，可以通过进一步的代码优化乃至于深层次的重构和架构设计来解决；
6. 在远端服务器上的工程部署实现了从0到1的效果，但是实际使用体验不好，特别是宕机后的再恢复以及设计大量数据操作的用户操作；
7. 工程前端元素较为统一，基本基于bootstrap / md-bootstrap的设计规范开发，但是对于一些移动设备的适配还不够完美。