

测试报告

# “智能花园”网站性能测试项目 测试报告

版本 <1.2>

李嘉昊、李晗东、程浩、吴正雨

< “智能花园” 网站>	版本: 1.0
测试报告	Date: 27/07/2018
TestRep_1.4	

## 修订历史记录

日期	版本	说明	作者
2018/07/27	1.0	撰写网页性能测试报告	程浩

< “智能花园” 网站>	版本: 1.0
测试报告	Date: 27/07/2018
TestRep_1.4	

# 目录

1.	简介	4
1.1	目的	4
1.2	范围	4
1.3	参考资料	4
1.4	概述	4
2.	Loadrunner 的 component 使用及功能	4
3.	测试用例及测试结果分析	5
3.1	测试用例	5
3.2	测试结果分析	6
4.	性能瓶颈与调优方案	8
5.	附录	10

< “智能花园” 网站>	版本: 1.0
测试报告	Date: 27/07/2018
TestRep_1.4	

## 1. 简介

本测试用例报告主要针对“智能花园”网站（本地 localhost 启动）的性能测试用例设计以及分析。

“智能花园”是一个在线的花园管理及智能浇灌网站，用户可以通过邮箱注册后，对自己的花园、传感器等进行管理，并且能够当前湿度以及温度进行智能浇灌的网站。

### 1.1 目的

编写测试用例报告，对各个功能进行用例设计，测试软件的性能是否符合设计要求，发现软件开发中存在的各种问题，便于改正。

### 1.2 范围

本文档适用于对 Web 项目“智能花园”（本地 localhost 启动）网站的性能测试工作。

### 1.3 参考资料

《软件测试——一个软件工艺师的方法》、《智能花园立项建议书》、《智能花园需求规约》

### 1.4 概述

本文档主要对网站的基本功能进行并发、吞吐、响应的测试。过程中需要接受测试的网站内容有：

- 用户的登陆功能
- 花园创建、删除、修改
- 传感器、喷头、摄像头的创建、删除、修改
- 随机数据生成，折线图、热力图、坐标图的绘制
- 喷头根据随机数据进行自动浇灌

## 2. Loadrunner 的 component 使用及功能

LoadRunner 是一种预测系统行为和性能的负载测试工具。Loadrunner 的测试流程包括：

1. 规划测试：确定测试要求如并发用户数量、典型业务场景流程，指定测试计划，设计用例。
2. 生成 Vuser 脚本、并点击 DesignStudio 自动关联：使用 Virtual User Generator 录制、编辑和完善测试脚本
3. 定义场景：使用 LoadRunner Controller 设置测试场景，设置用户数 50（初始为 10 个，每 15s 增加 10 个直至 500）、持续（重复）运行 5min、最后每 15s 退出 5 个用户直至为 0
4. 运行场景：使用 LoadRunner Controller 驱动、管理并监控场景的运行。
5. 分析结果：使用 LoadRunner Analysis 生成报告和图表并评估性能。

< “智能花园” 网站>	版本: 1.0
测试报告	Date: 27/07/2018
TestRep_1.4	

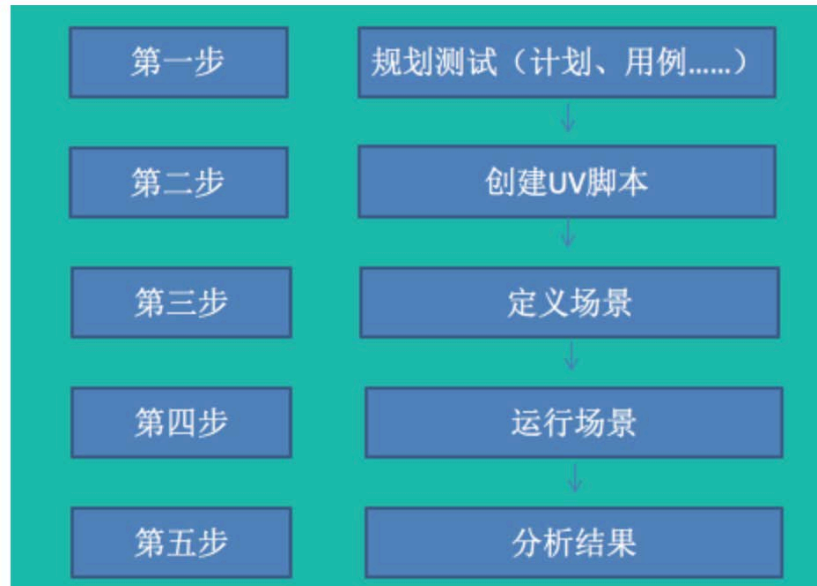


图 1. 性能测试的实施规划

LoadRunner 提供了 3 大主要功能模块，既可以作为独立的工具完成各自的功能，又可以作为 LoadRunner 的一部分彼此衔接，与其他模块共同完成软件性能的整体测试，这 3 大模块分别是：

1. Virtual User Generator
2. LoadRunner Controller
3. LoadRunner Analysis

其中，Virtual User Generator 用于录制性能测试脚本，LoadRunner Controller 用于创建、运行和监控场景，LoadRunner Analysis 用于分析性能测试结果。

### 3. 测试用例及测试结果分析

#### 3.1 测试用例

< “智能花园” 网站>	版本: 1.0
测试报告	Date: 27/07/2018
TestRep_1.4	

表 1. 性能测试的用例说明

方法:	<ul style="list-style-type: none"> <li>进入“智能花园”登录首页</li> <li>输入账号密码后登陆</li> <li>进行花园的创建（长宽分别为 1000，可进行随机输入）</li> <li>修改花园信息（长宽修改为 1001，可随机输入），其他名字、花园位置等可随机输入</li> <li>进入花园信息界面，各自创建一个温度、湿度传感器、监控器（坐标随机，范围在该花园内）</li> <li>修改传感器、监控器信息，并删除一个监控器、温度、湿度传感器，并再次创建已进行下一步测试</li> <li>进入喷头界面，创建、修改、删除喷头</li> <li>进入热力图页面，点击数据生成，并点击当前花园查看热力图变化</li> <li>点击折线图页面，选择特定的传感器点击生成数据</li> <li>点击该花园的喷头界面，将所有喷头设置为开，喷头开关可随随机数据的变化而变化</li> <li>点击“Log out”退出，回到登陆界面</li> </ul>
Action 迭代	在录制好的脚本 runtime settings 处设置迭代次数，本次测试迭代 50 次
关联	使用上述用例，使用录制后关联，通过脚本回放和扫描关联，系统尝试找到录制与执行时服务器的差异部分，找到需要关联的数据，并进行关联。
自定义场景	<p>通过 Controller 定义，本次定义的设置:</p> <ul style="list-style-type: none"> <li>同时初始化所有 Vuser (50)</li> <li>同时初始时只有 10 个 Vuser，每 10s 增加是个 Vuser 直至为 50</li> <li>所有 Vuser 按照指定时间 (5min) 重复运行脚本</li> <li>释放策略 (每隔 15s 释放 5 个 Vuser)</li> </ul>
结果分析	使用 Analysis 进行结果分析

### 3.2 测试结果分析

性能测试的指标主要包括吞吐量、事务响应时间与负载、事务响应时间（百分比）、事务响应时间（分布）。我们用 LoadRunner Analysis 导出了实际运行结果的图表，并加以分析。

< “智能花园” 网站>	版本: 1.0
测试报告	Date: 27/07/2018
TestRep_1.4	

### ● Throughput (吞吐率)

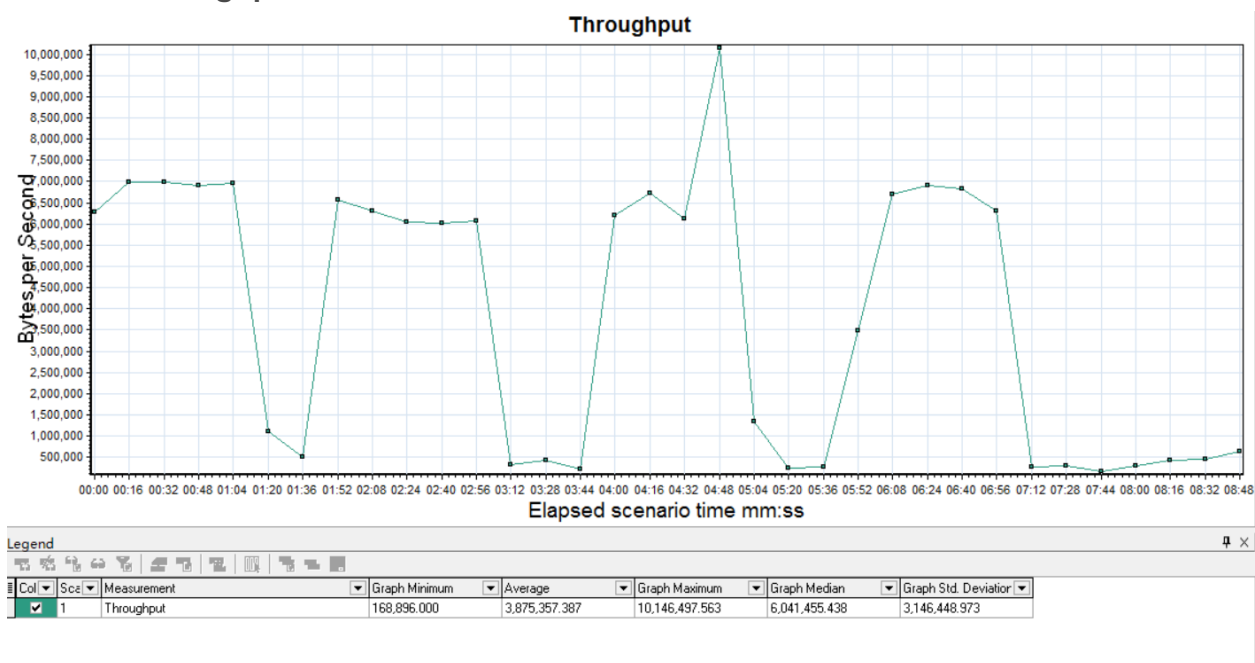


图 2. 吞吐率的测试结果示意图

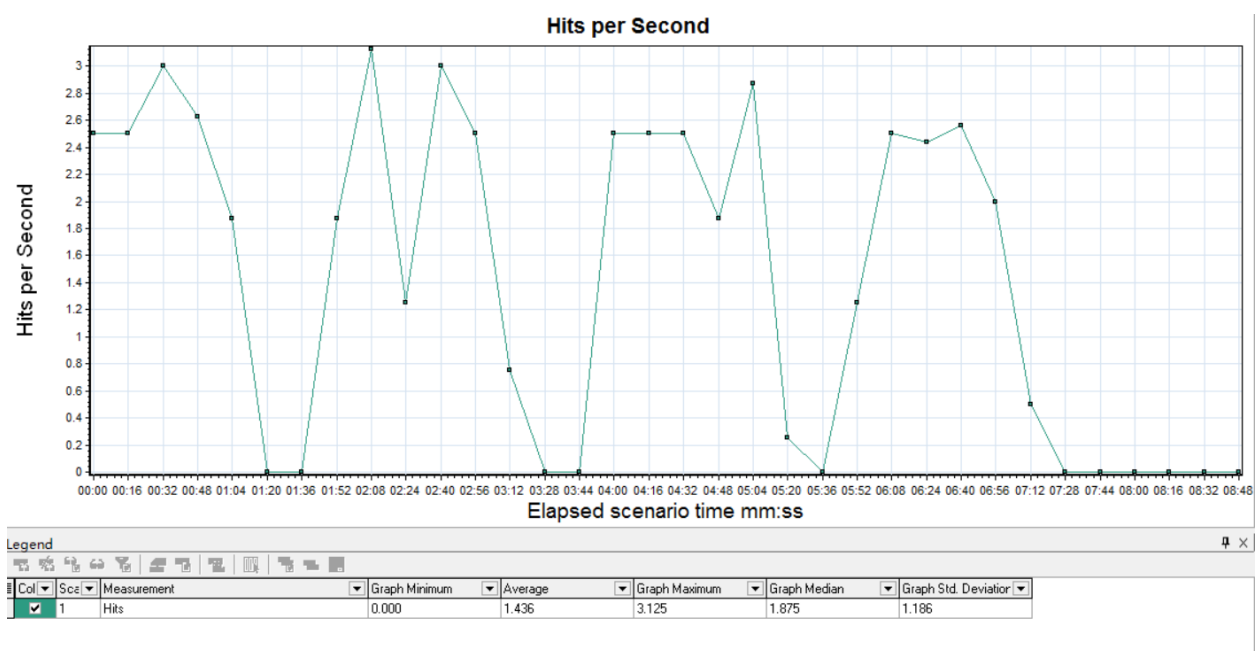


图 3. Hits/Second 的测试结果示意图

“吞吐率”显示的是场景运行过程中服务器的每秒的吞吐量。

“吞吐率”图是每秒服务器处理的 HTTP 申请数。“点击率”图是客户端每秒从服务器获得的总数据量。

< “智能花园” 网站>	版本: 1.0
测试报告	Date: 27/07/2018
TestRep_1.4	

从图中可以看出，吞吐率与点击率的图像十分相似，一共出现了 3 个峰值。每个峰值出现的时间基本吻合，表示这些时间点用户有大规模统一操作，比如作业提交 deadline 的前一晚，就可能出现图中所示的峰值。吞吐率和点击率也与虚拟用户数有关，用户数量的激增和剧减会产生大量的登录和登出操作，都会使吞吐率和点击率的曲线产生波动。

#### ● Transaction Response Time Under Load（事务响应时间与负载）

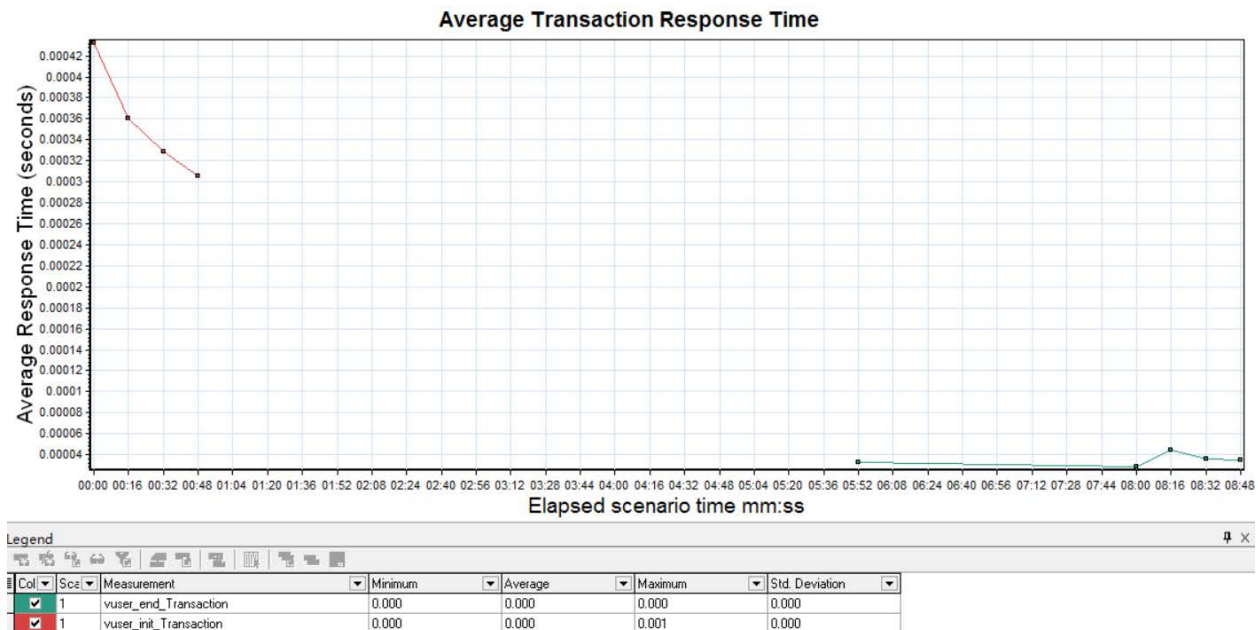


图 4. 事务响应时间和负载的测试结果示意图

“事务响应时间与负载”是“正在运行的虚拟用户”图和“平均响应事务时间”图的组合，通过它可以看出在任一时间点事务响应时间与用户数目的关系。从图中可以看出，该图像随着时间的推移有所波动，出现了一个峰值，而该峰值与吞吐率最大峰值出现的时间完全一致。整个图像的变化趋势也与吞吐率的变化趋势相吻合。

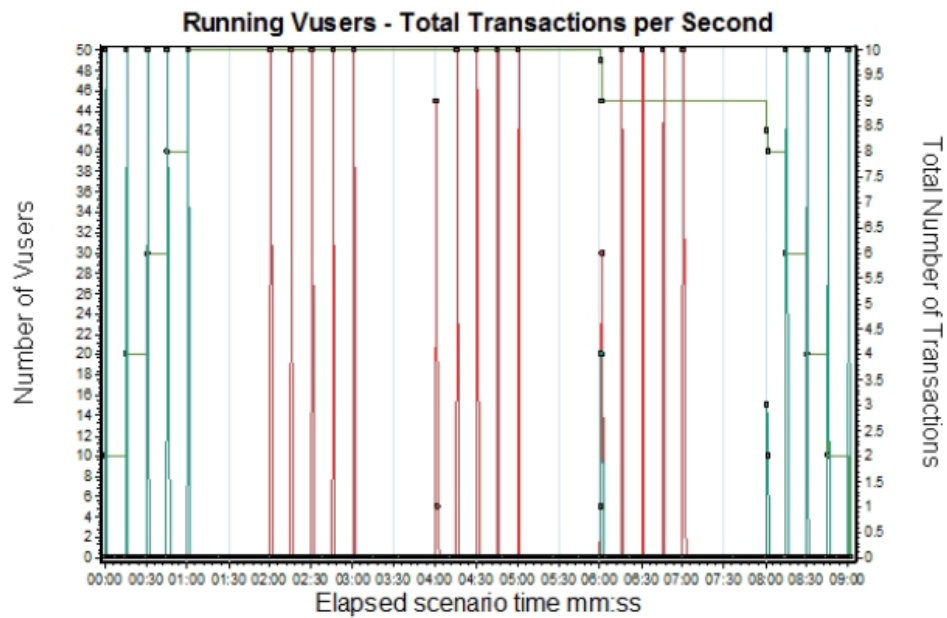
## 4. 性能瓶颈与调优方案

### 4.1 性能瓶颈

我们用 Loadrunner Analysis 模块导出了反应性能瓶颈的图表，如下图所示。图表统计了所有成功和失败的事务，并计算出了事务成功率为 33.9%，一共有 195 个 error。这其中的原因是我们进行录制时只能根据特定的 SensorID、GardenID 等特定的 ID 进行操作，而且这些 ID 创建时在数据库中是特定的，所以操作会出错误，但是通过测试后发现，我们的“智能花园”网站在有多个用户进行不停操作时的性能表现较好，效应时间较低、吞吐量也表现较好。



< “智能花园” 网站>	版本: 1.0
测试报告	Date: 27/07/2018
TestRep_1.4	



Color	Graph	Scale	Measurement	Graph's Minimum	Graph's Average	Graph's Maximum	Graph's Median	Graph's Std. Deviation
	Running Vusers	1	Run	0.000	25.733	50.000	30.000	17.808
	Total Transactions per Second	1	Fail	0.000	0.358	10.000	0.000	1.807
	Total Transactions per Second	1	Pass	0.000	0.184	10.000	0.000	1.295

Measurement	Value
Run Name	res.lrr
Weighted Average of Transaction Response Time	0
Total Passed Transactions	100
Total Failed Transactions	195
Transactions Success Rate, %	33.9
Total Errors per Second	0.4
Total Errors	195

图 5. 性能瓶颈的分析示意图

## 4.2 调优方案

针对测试过程中暴露的“智能花园”网站在性能方面的不足和不完善之处，我们小组经过分析和讨论，给出了以下的性能调优方案：

### 1.减少 http 请求

< “智能花园” 网站>	版本: 1.0
测试报告	Date: 27/07/2018
TestRep_1.4	

HTTP 协议是无状态的应用层协议，意味着每次 HTTP 请求都需要简历通信链路，进行数据传输，而在服务器端，每个 HTTP 都需要启动独立的线程去处理，这些通信和服务的开销都很昂贵，减少 HTTP 请求的数目可有效提高访问性能。减少 HTTP 请求的主要手段是合并 CSS，合并 JavaScript，合并图片。将浏览器一次访问需要的 JavaScript,CSS 合并成一个文件，这样浏览器就只需要一次请求。多张图片合并成一张，如果每张图片都有不同的超链接，可通过 CSS 偏移响应鼠标点击操作，构造不同的 URL。

## 2. 异步操作

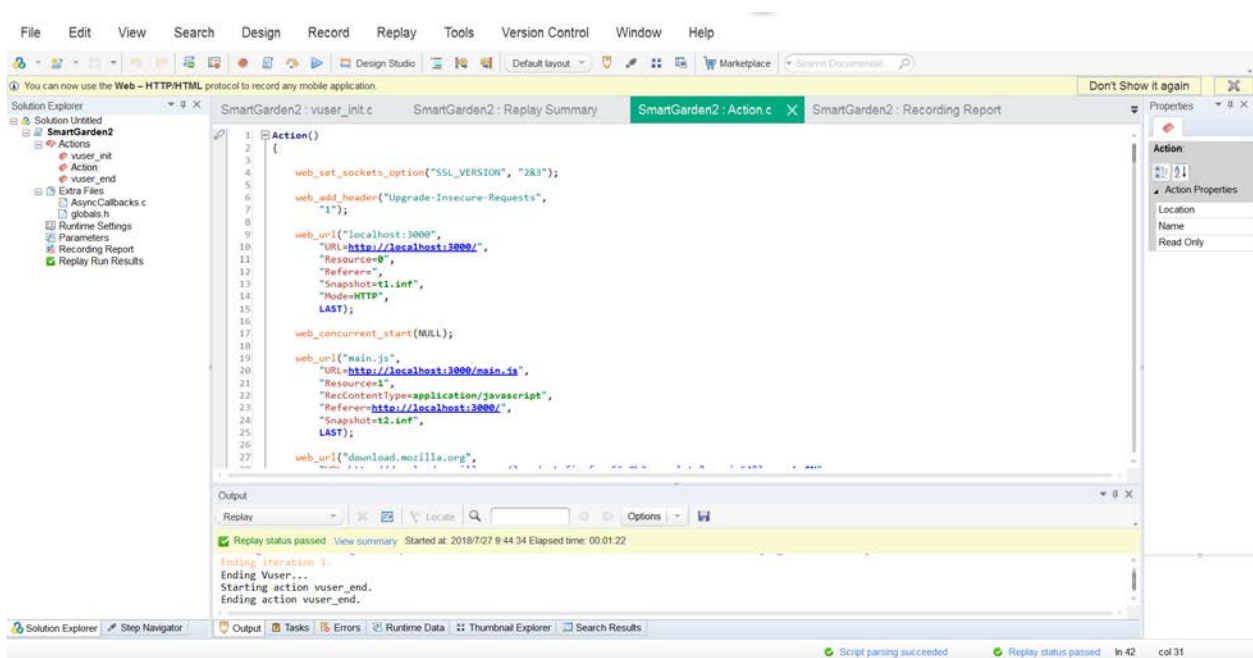
使用消息队列将调用异步化，可改善网站的扩展性。事实上，使用消息队列还可以改善网站系统的性能。在不使用消息队列的情况下，用户的请求数据直接写入数据库，在高并发的情况下，会对数据库造成巨大的压力，同事也是的响应延迟加剧。在使用消息队列后，用户请求的数据发送给消息队列后立即返回，再由消息队列的消费者进程从消息队列中获取数据，异步写入数据库。由于消息队列服务器处理速度远快于数据库，因此用户的相应延迟可得到有效改善。此外消息队列还具有很好的削峰作用，即通过异步处理，将短时间高并发产生的事务消息存储在消息队列中，从而削平高峰期的并发事务。

## 3. 减少 LocalStorage 传输

一方面，LocalStorage 包含在每次请求和响应中，太大的 LocalStorage 会严重影响数据传输，因此哪些数据需要写入 LocalStorage 需要慎重考虑，尽量减少 LocalStorage 中传输的数据量。另一方面，对于某些静态资源的访问，如 CSS,JS 等，发送 LocalStorage 没有意义，可以考虑静态资源使用独立域名访问，避免请求静态资源时发送 LocalStorage，减少 LocalStorage 传输的次数。

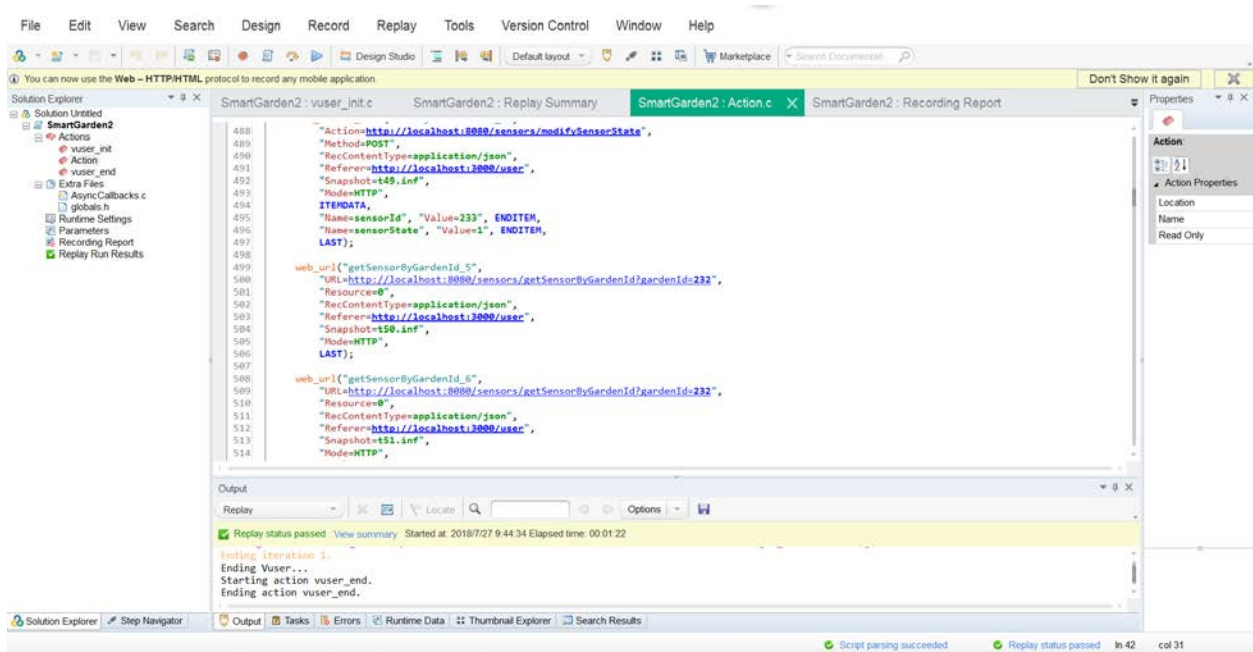
## 5. 附录

### 5.1 程序部分的 GUI 截图

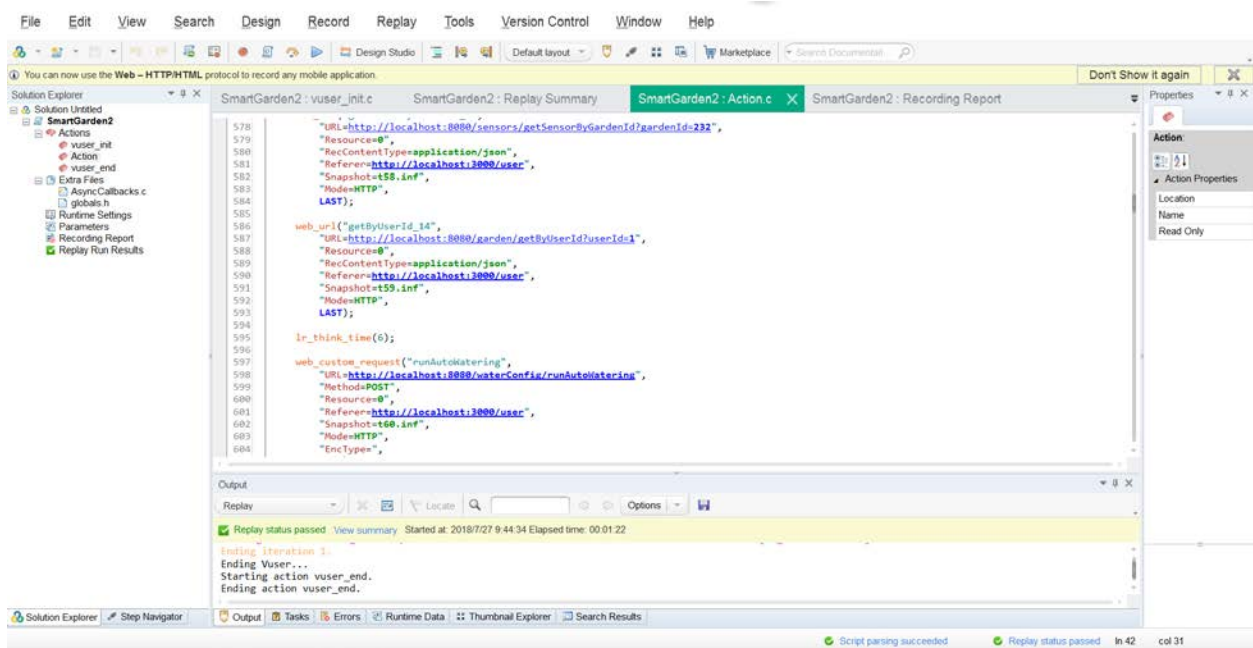


附图 1 代码片段-1

< “智能花园” 网站>	版本: 1.0
测试报告	Date: 27/07/2018
TestRep_1.4	

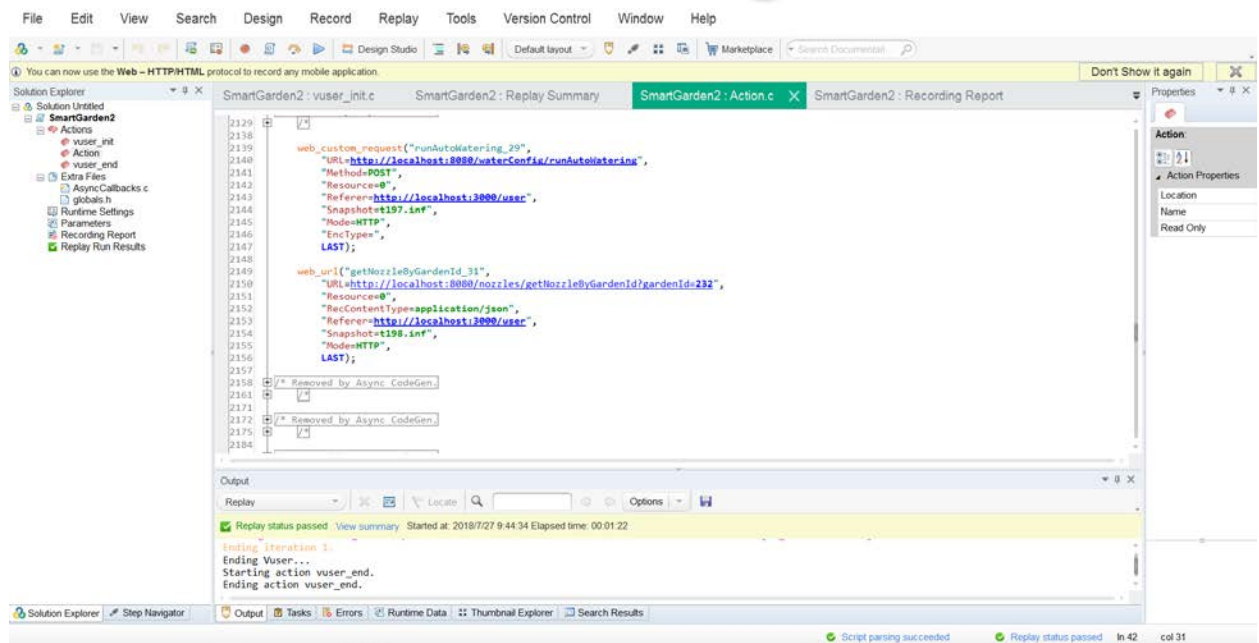


附图 2 代码片段-2



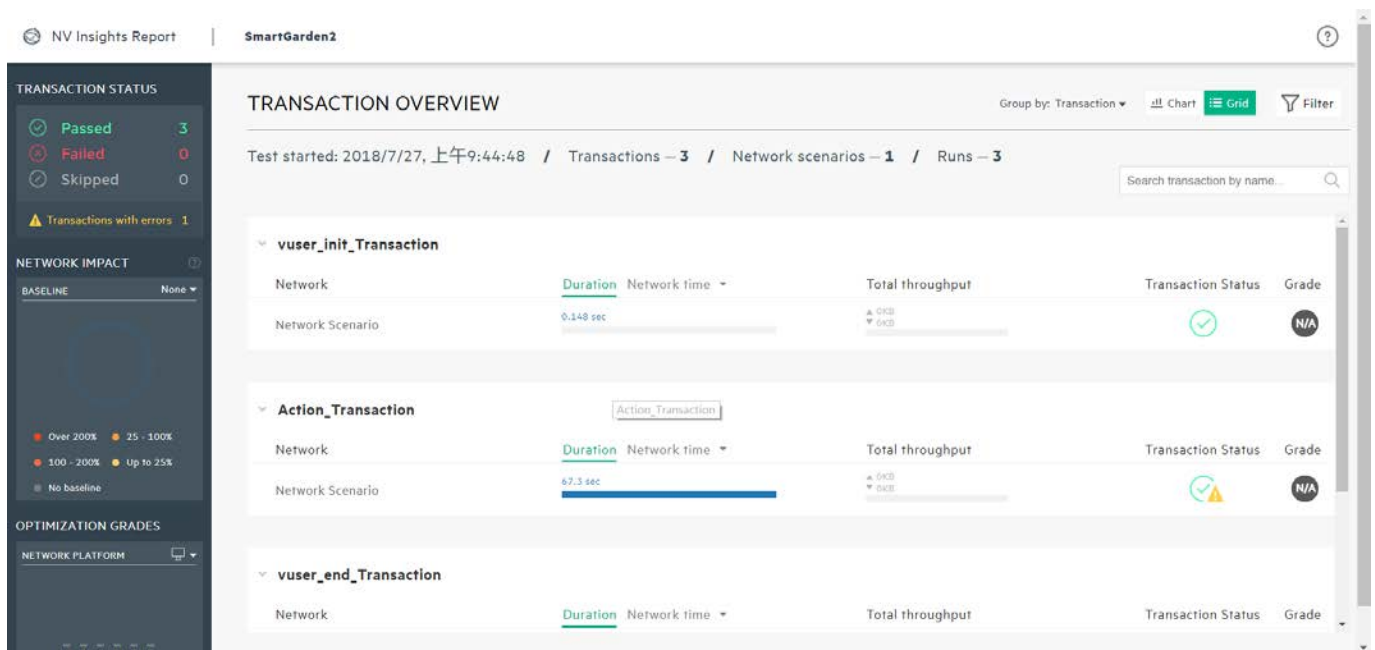
附图 3 代码片段-3

< “智能花园” 网站>	版本: 1.0
测试报告	Date: 27/07/2018
TestRep_1.4	



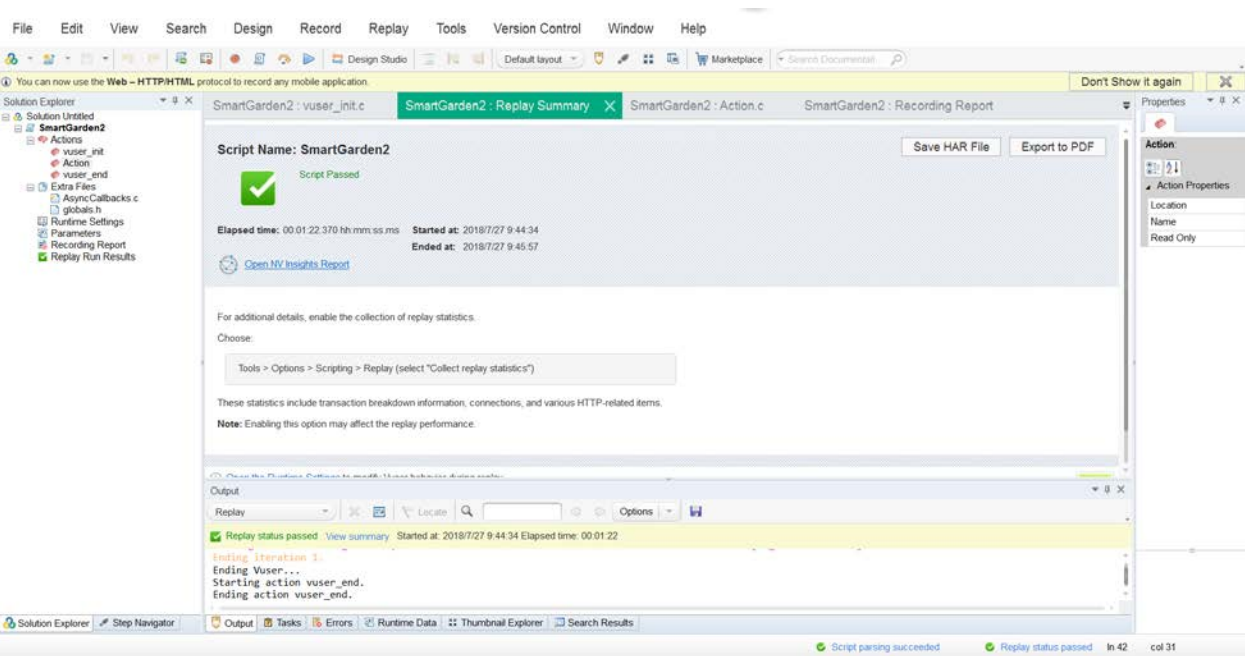
附图 4 代码片段-4

## 5.2 参数化的 GUI 界面截图



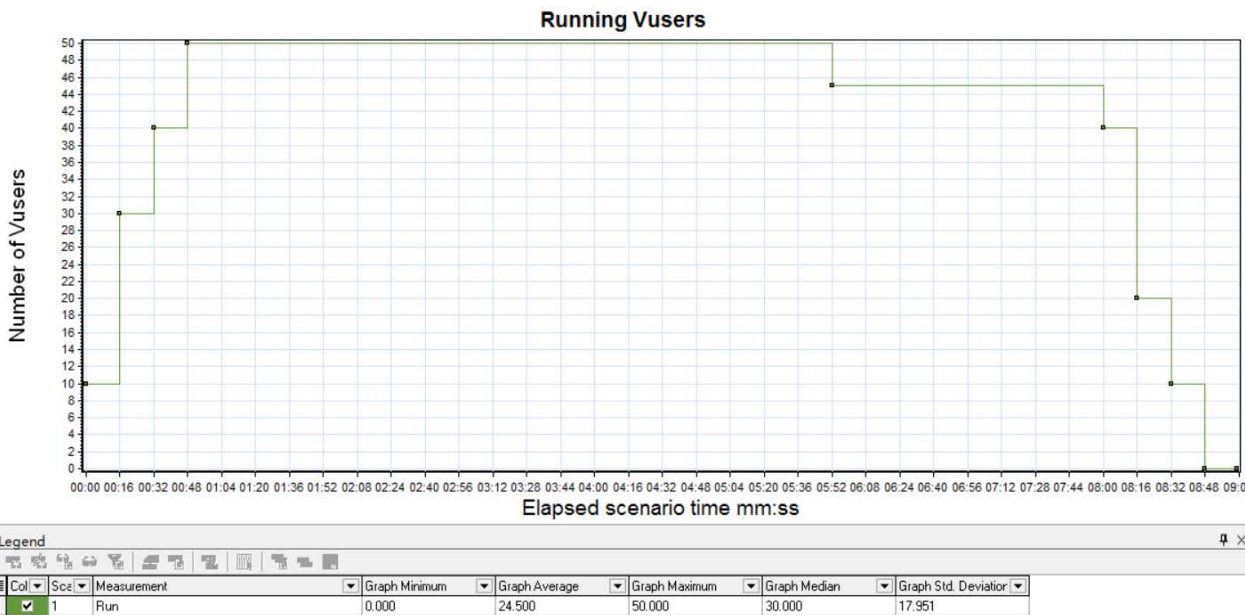
附图 5 录制脚本的播放演示

< “智能花园” 网站>	版本: 1.0
测试报告	Date: 27/07/2018
TestRep_1.4	



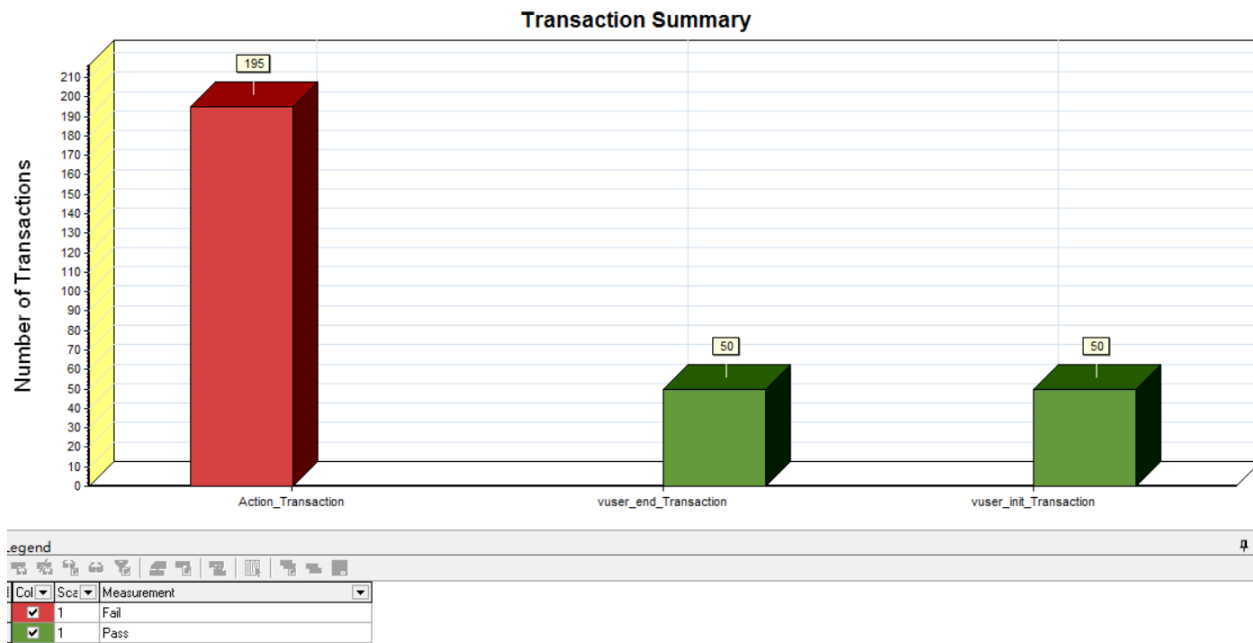
附图 6 录制脚本播放成功

5.3 补充的 Loadrunner Analysis 分析图表

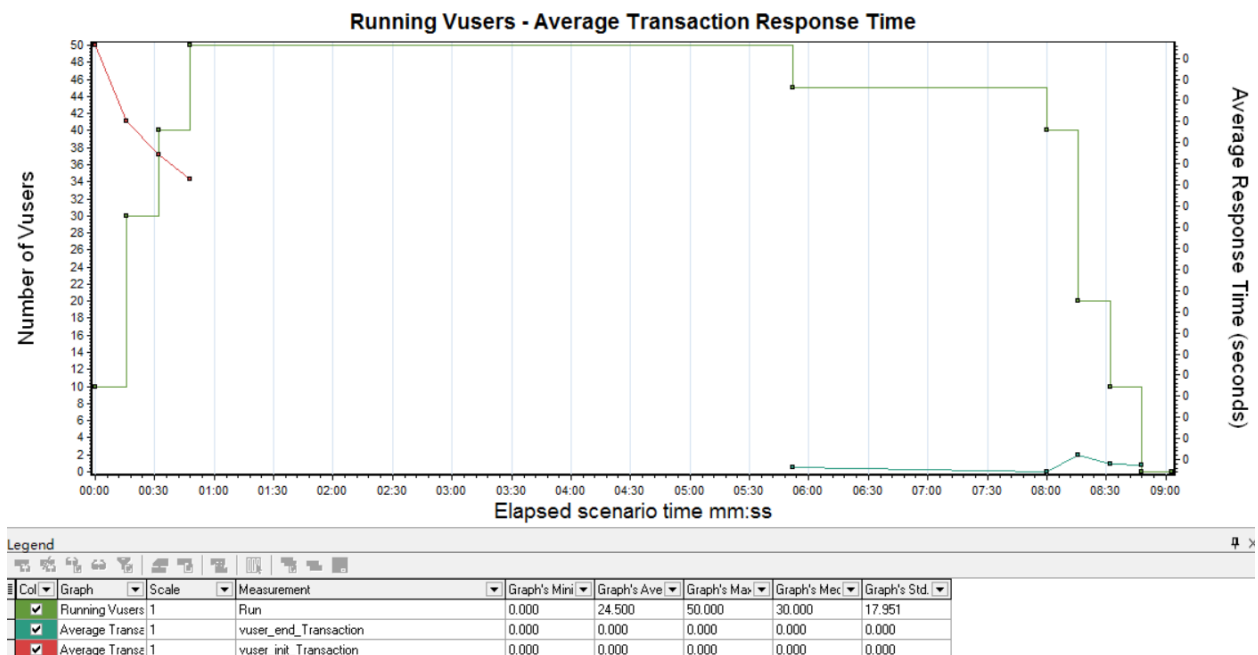


附图 7 用户数量变化

< “智能花园” 网站>	版本: 1.0
测试报告	Date: 27/07/2018
TestRep_1.4	



附图 8



附图 9 虚拟用户—用户响应时间图



< “智能花园” 网站>	版本: 1.0
测试报告	Date: 27/07/2018
TestRep_1.4	

此外，对于 loadrunner 的完整测试结果，我们提供了一份附加文档进行说明，详细内容可见提交的文件 “report.pdf”