

# 实验五 LoadRunner中进行Web性能测试

## 实验目的：

1. 掌握负载测试的原理
2. 掌握使用LoadRunner进行负载测试的方法；
3. 学习针对Web的性能测试的指标类型，及其在LoadRunner中的实现

## 实验内容：

模拟 10、20、30 个用户登陆航空订票系统时的平均响应时间，并进行分析。

## 实验提交：

测试报告（测试目的，测试过程，测试结果分析，实验感想或建议）

## LoadRunner 使用参考:

### 一、LoadRunner组成：

- 1、Virtual User Generator 虚拟用户生成器
- 2、Controller 控制器
- 3、Analysis 分析器

**Virtual User Generator** 通过录制典型最终用户在应用程序上执行的操作来生成虚拟用户（或称Vuser）。然后 VuGen 将这些操作录制到自动化 Vuser 脚本中，将其作为负载测试的基础。

**HP LoadRunner Controller** Controller 是用来设计、管理和监控负载测试的中央控制台。使用 Controller 可运行模拟真实用户操作的脚本，并通过让多个 Vuser 同时执行这些操作，从而在系统上施加负载。

**HP Analysis** 提供包含深入性能分析信息的图和报告。使用这些图和报告可以找出并确定应用程序的瓶颈，同时确定需要对系统进行哪些改进以提高其性能。

### 二、LoadRunner的工作原理：

LoadRunner的工作原理是通过用户执行被测程序的客户端，在VuGen中录制被测系统的客户端和服务器的协议交互，生成脚本，然后在Controller中控制Load Generator，按照一定

的配置（又称为场景），模拟一定数量的用户，对服务器产生压力，同时对被测系统涉及的操作系统，数据库，中间件等资源进行监控，收集压力情况下的资源信息，测试结束后形成测试结果和监控数据，在结果分析器中进行分析，最后生成测试结果报告。

### 三、LoadRunner测试流程

LoadRunner 测试流程由以下四个基本步骤组成：

步骤 1. 创建脚本：捕获在您的应用程序上执行的典型最终用户业务流程。

步骤 2. 设计模拟场景：通过定义测试期间发生的事件，设置负载测试环境。

步骤 3. 运行场景：运行、管理并监控负载测试。

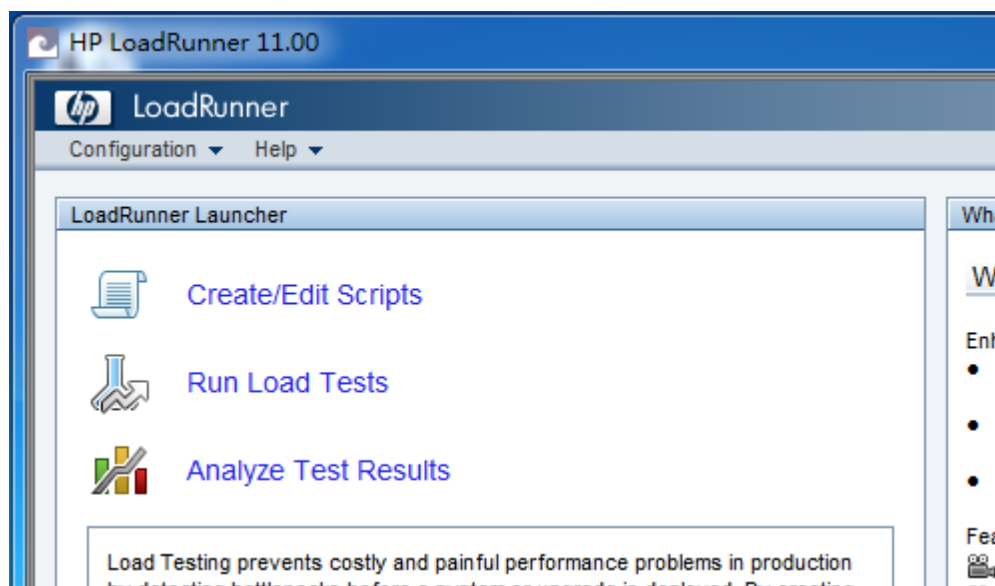
步骤 4. 分析结果：分析 LoadRunner 在负载测试期间生成的性能数据。

## 使用 VuGen 创建脚本

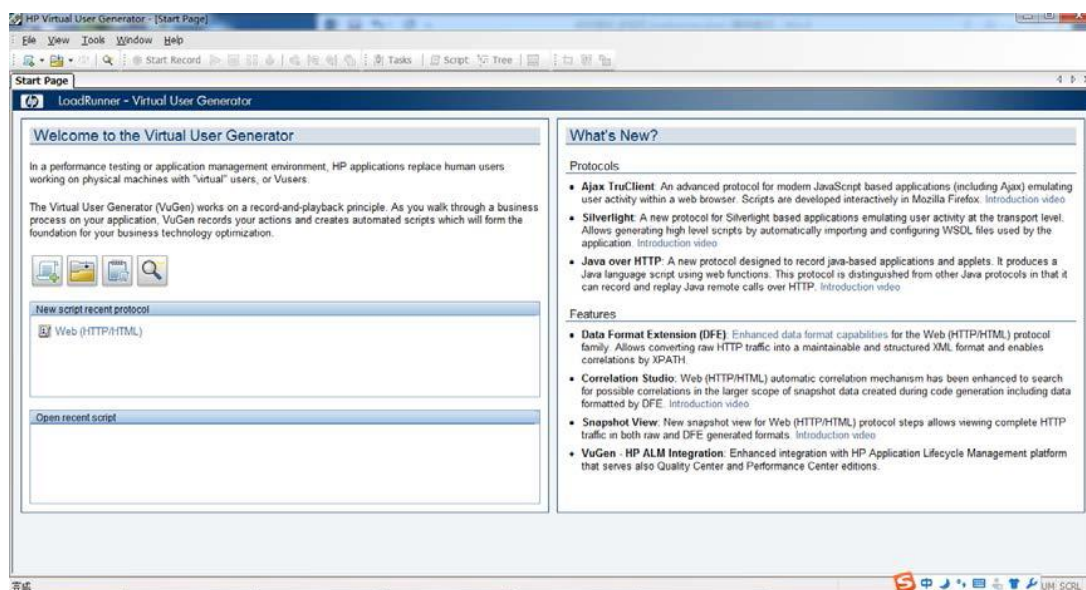
创建负载测试的第一步是使用 VuGen 录制典型最终用户业务流程。VuGen 以“录制-回放”的方式工作。当在应用程序中执行业务流程步骤时，VuGen 会将操作录制到自动化脚本中，并将其作为负载测试的基础。

### 一、启动LoadRunner，打开VuGen

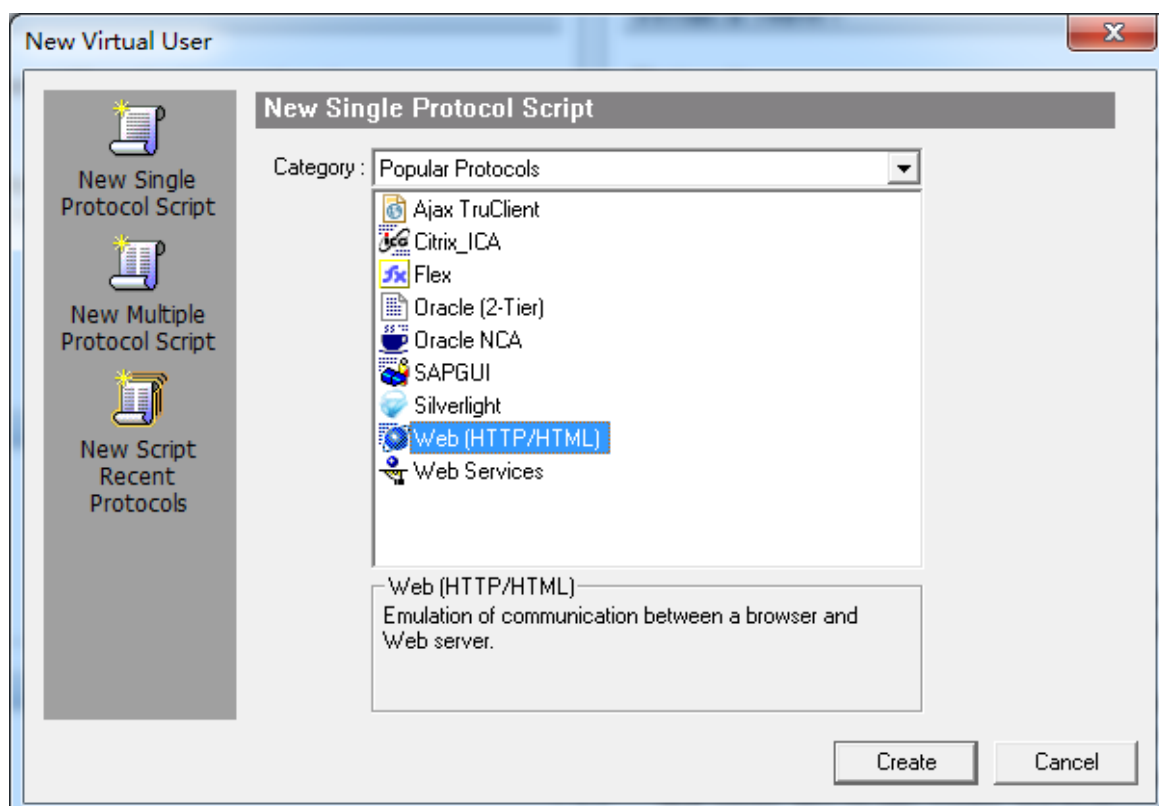
打开LoadRunner11，点击Create/Edit Scripts



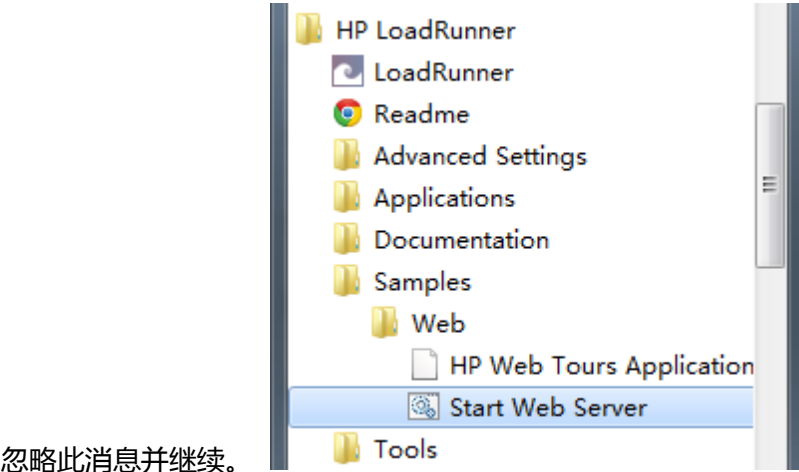
将打开HP Virtual User Generator界面



Virtual User Generator (VuGen) 通过执行业务流程来模拟用户的实际操作过程，简便易用。LoadRunner支持众多协议，我们以LoadRunner自带的航空订票系统Web版（HP Web Tours Application）为例，主要测试目标为Web应用，所以在脚本协议中选择“Web(HTTP/HTML)”项。



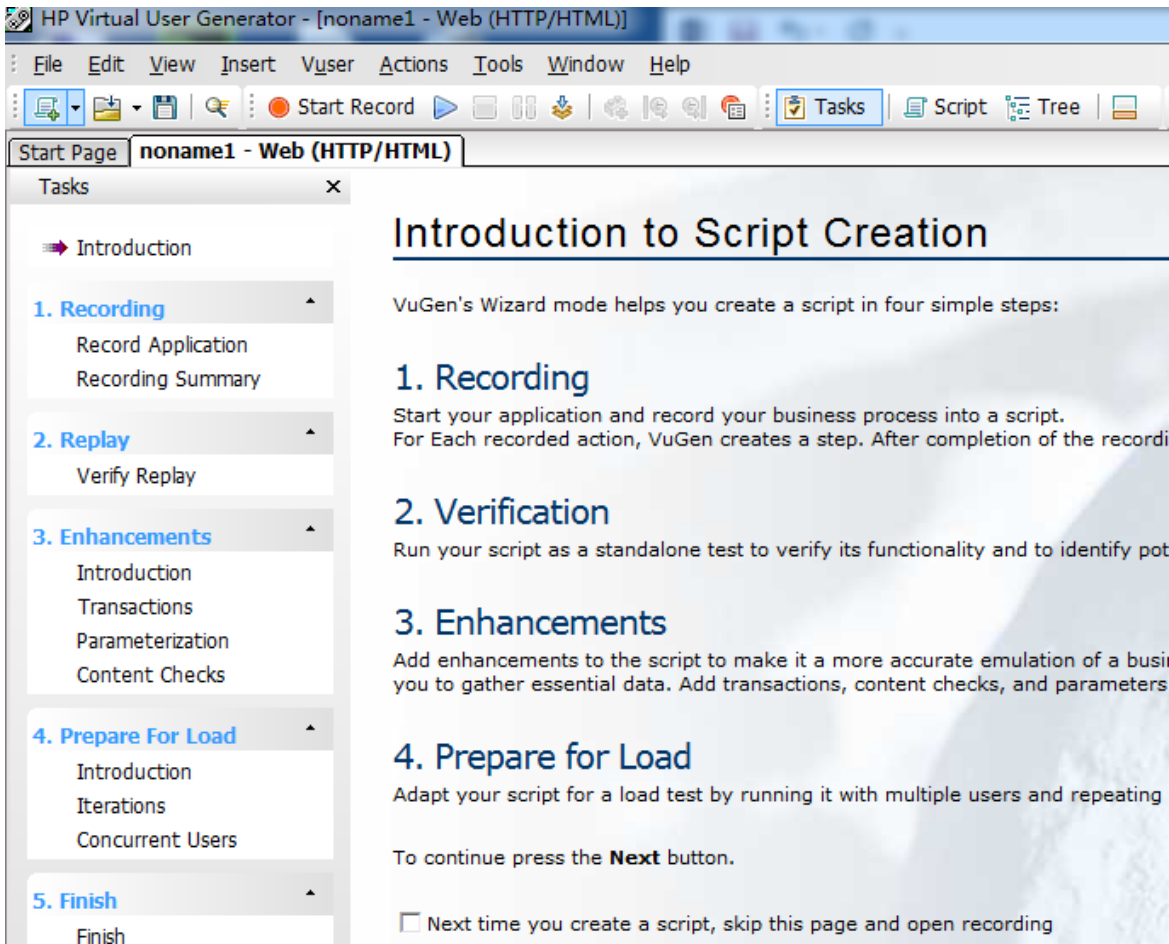
为运行HP Web Tours Application，需先确保它的网页服务器已启动。在Start菜单中找的HP LoadRunner，点击Start Web Server。如果Web 服务器已运行，将出现错误消息，可以



忽略此消息并继续。

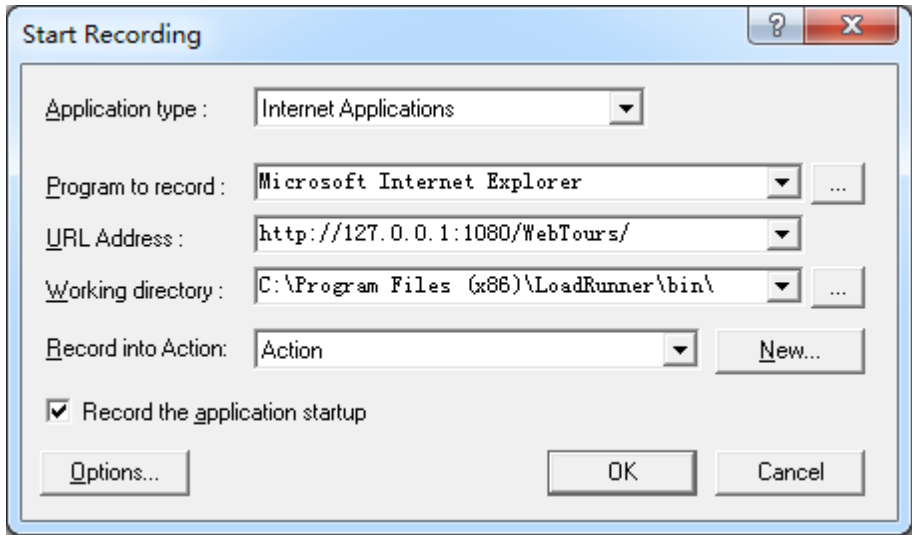
空白脚本以 VuGen 的向导模式打开，同时在左侧显示任务窗格。如果未显示任务窗格，可单击工具栏上的任务按钮调出。任务窗格列出脚本创建过程中的各个步骤或任务。在执行各个步骤的过程中，VuGen 将在窗口的主要区域显示详细说明和指示信息。

点击Recording，进入脚本录制

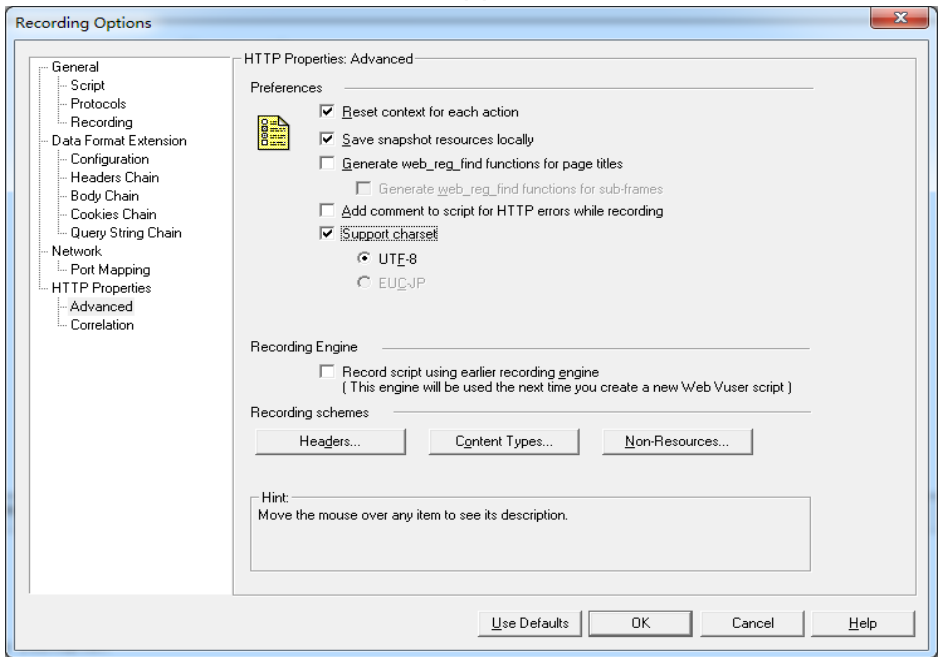


将弹出如下窗口，可以选择应用类型、使用的浏览器、设置被测系统的链接地址、工作的目录等。输入要测试网页的URL Address为：http://localhost:1080/WebTours/

{在录制脚本中，LoadRunner对服务器的返回中文内容是自动使用GB2312方式阅读的，但是所测系统是使用UTF-8的方式来编码，由于解码编码的方式不同最终会导致中文处理错误，在最终脚本中出现乱码现象。为了解决这一问题，可以选中上图左下角的【Options】，勾选录制选项的Support charset后选中UTF-8后进行录制：



注：在录制脚本中，LoadRunner对服务器的返回中文内容是自动使用GB2312方式阅读的，但是所测系统是使用UTF-8的方式来编码，由于解码编码的方式不同最终会导致中文处理错误，在最终脚本中出现乱码现象。为了解决这一问题，可以选中上图左下角的【Options】，勾选录制选项的Support charset后选中UTF-8



UTF-8后进行录制：

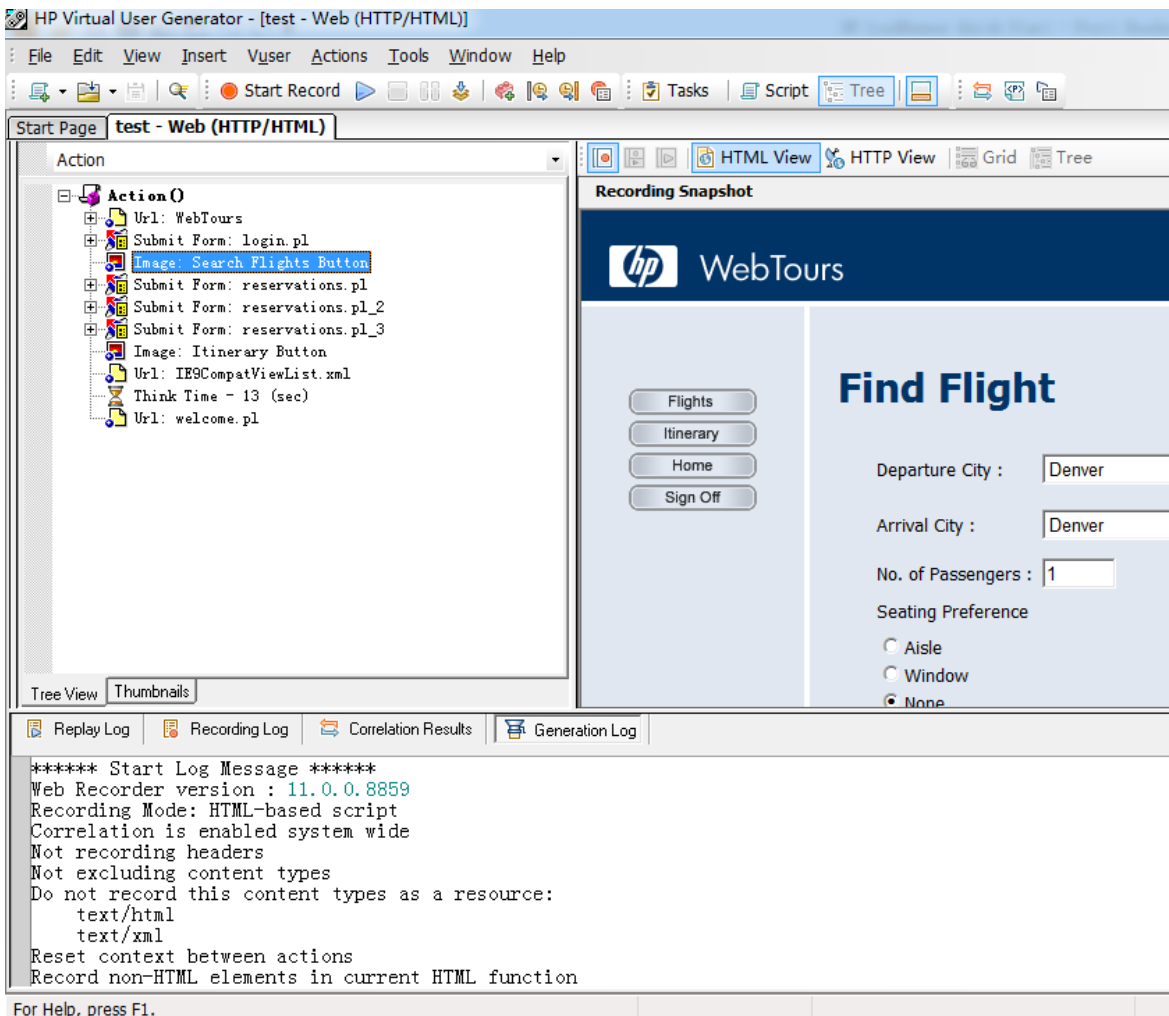
## 二、 录制Web Tours网站，录制订票过程

设置好参数后，即可点击Start Record按钮开始录制脚本，LoadRunner将记录录制过程中执行的所有事件，并自动生成脚本：

1. 首先注册一个用户名，然后以该用户名登录。
2. 输入航班详细信息；选择航班。
3. 输入支付信息并预订机票。
4. 查看航班路线。单击左窗格中的 itinerary以显示。
5. 退出当前用户。单击左窗格中的 Sign Off以退出。
6. 在浮动工具栏上单击停止以停止录制。

生成 Vuser 脚本后，VuGen 向导将自动继续执行任务窗格中的下一个步骤，并显示录制摘要信息（包括协议信息和会话期间创建的一系列操作）。VuGen 为录制期间执行的每个步骤生成一个快照，即录制期间各窗口的图片。这些录制的快照以缩略图的形式显示在右窗格中。

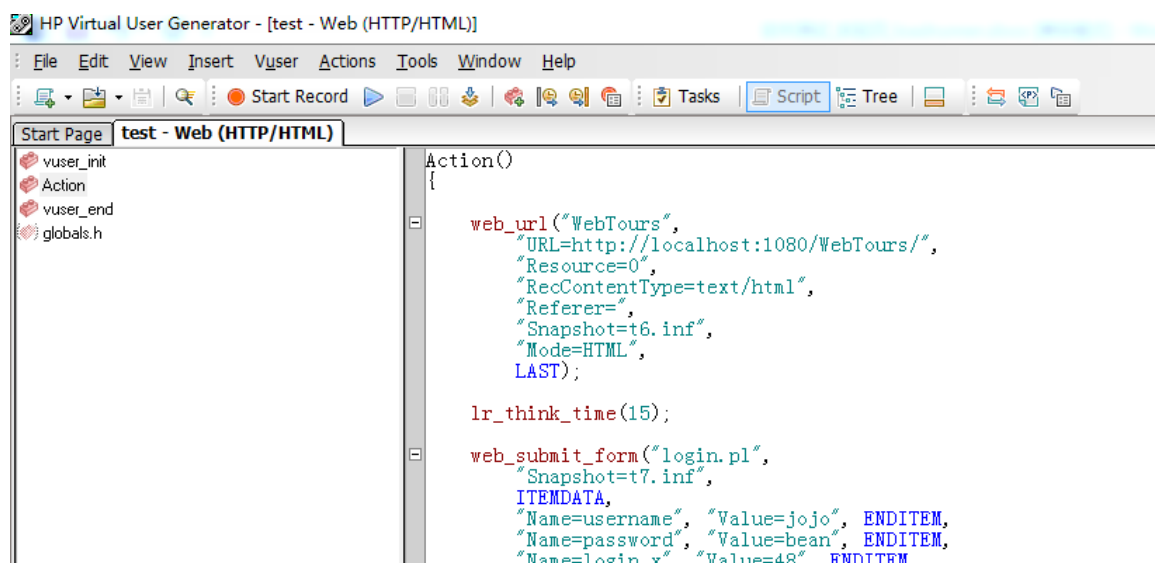
7. 选择文件→保存，或单击保存。VuGen 将该文件保存到 LoadRunner 脚本文件夹中，



### 三、查看录制脚本

可以在VuGen的树视图或脚本视图中查看已录制的脚本。树视图是一种基于图标视图，将Vuser的操作以步骤的形式列出。要在树视图中查看脚本，请选择视图 > 树视图，或者单击工具栏上的树按钮。对于录制期间执行的每个步骤，VuGen在脚本树中为其生成一个图标和一个标题。在树视图中，您将看到以脚本步骤的形式显示的用户操作。大多数步骤都附带相应的录制快照。

脚本视图是一种基于文本的视图，将Vuser的操作以函数的形式列出。要在脚本视图中查看脚本，请选择视图 > 脚本视图，或者单击工具栏上的脚本按钮。在脚本视图中，VuGen在编辑器中显示脚本，并用不同颜色表示函数及其参数值。可以在窗口中直接输入C或LoadRunner API函数以及控制流语句。





## 四、 脚本回放

可以回放脚本以验证其是否准确模拟了录制的操作。

1. 确保已显示任务窗格，如果未显示，请单击工具栏中的任务按钮。在任务窗格中单击Verify Replay( 验证回放 ) ,然后单击说明窗格底部的Start Replay( 开始回放 )按钮。稍后 VuGen 将开始运行脚本。
2. 当脚本停止运行后，可以在向导中查看关于这次回放的概要信息。

## 五、 编辑测试脚本

VuGen除了可以录制会话过程中的操作，将生成的各种函数插入到脚本中之外，用户还可在录制脚本的基础上进行修改，从而开发出一个逻辑功能和实际环境中完全一样的用户脚。

LoadRunner中有很多技术，例如集合点、参数、事务等，通过这些技术来解决脚本运行过程中产生的各种问题。下面将以登录模块为例，进行测试脚本的编辑。

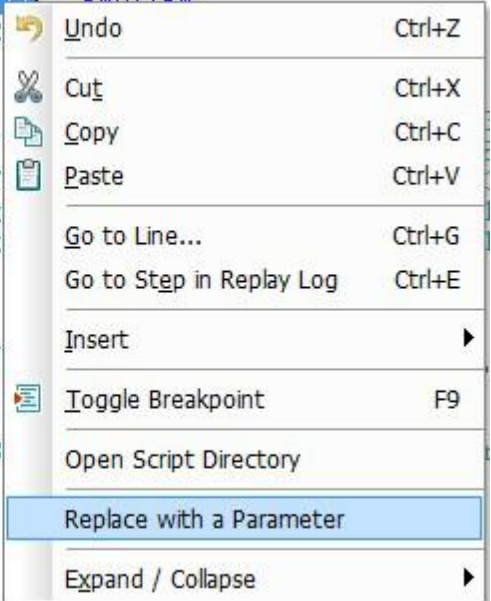
### 参数化

由于需要涉及到多用户并发登录查询，单个用户重复登录是无法成功的，所以在进行负载的时候为了模拟用户的真实操作，也需要输入不同的数据。要实现这一点，可以用参数替换固定的文本，也就是脚本的参数化。在数据库中预设的登录密码是相同的，因此只需对登录账号进行参数化：

```
ITEMDATA,
"Name=user.userNum", "Value=111111",
"Name=user.userPwd", "Value=2222",
"Name=user.userSort", "Value=0",
EXTRARES,
"Url=../images/left_nav_bg.jpg",
"Url=../images/manage_r2_c1.jpg",
"Url=../images/nav_bg.jpg", "Ref",
"Url=../images/manage_r2_c14.jpg",
"Url=../images/manage_r3_c11.jpg",
LAST);

url("manFrame",
"URL=http://172.29.136.183:8080/",
"Resource=0",
"RecContentType=text/html",
"Referer=http://172.29.136.183:8080/",
"Snapshot=t7.inf",
"Mode=HTML",
LAST);

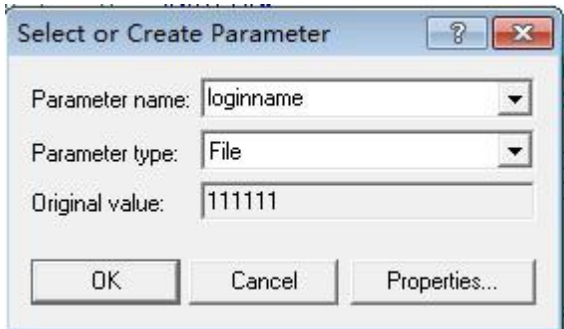
rrn 0;
```



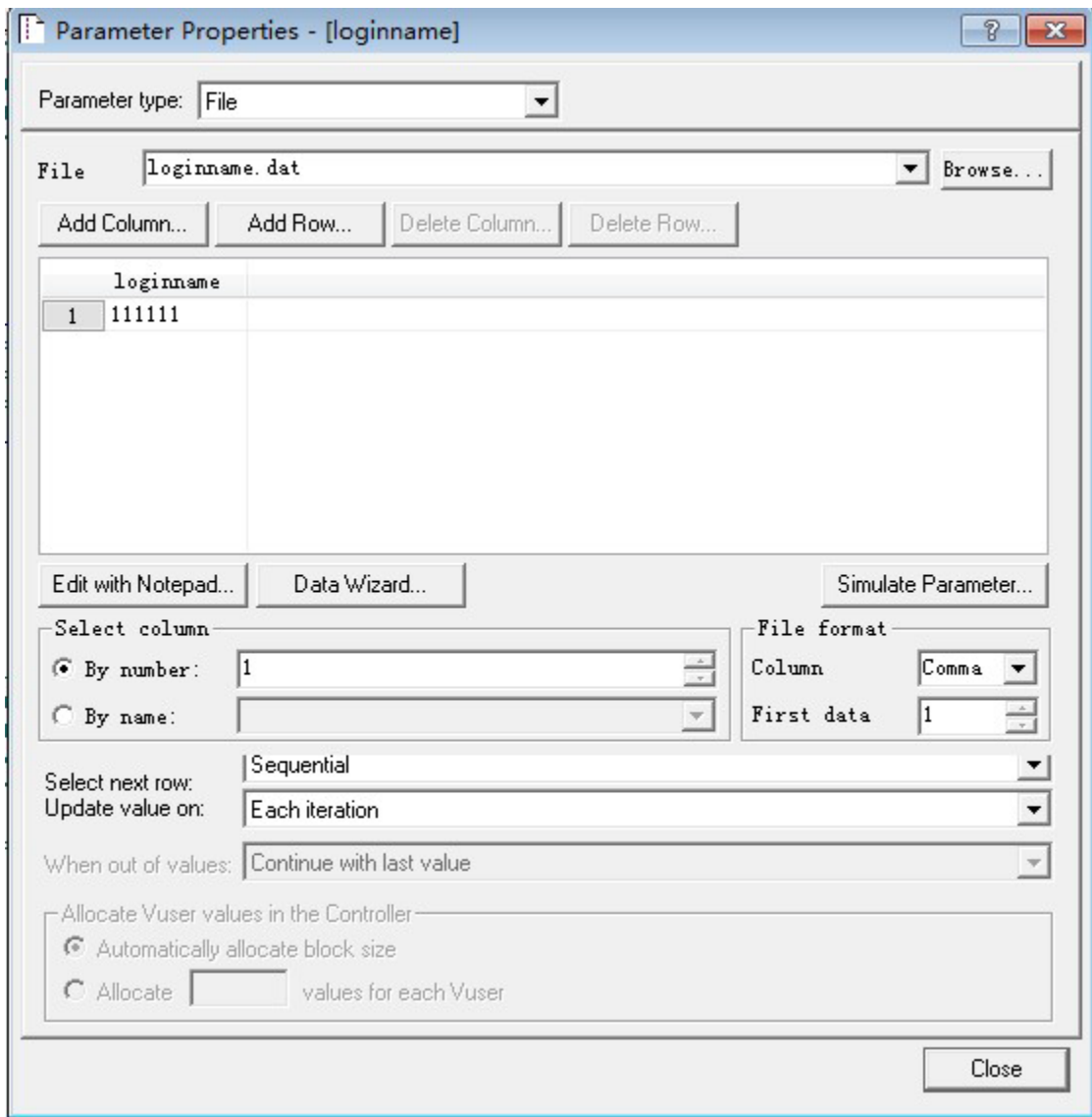
The screenshot shows a context menu open over a script line. The menu includes options like Undo, Cut, Copy, Paste, Go to Line..., Go to Step in Replay Log, Insert, Toggle Breakpoint, Open Script Directory, and Replace with a Parameter (which is highlighted). The script text on the left includes a comment 'ITEMDATA,' followed by several 'Name=Value' pairs, an 'EXTRARES' section with image URLs, a 'url' function call, and an 'rrn 0;' statement.



**右键菜单添加参数项** 选择了Replace with a Parameter选项后，会弹出创建参数的对话框



**创建参数的对话框** 点击Properties按钮，进入参数属性对话框界面：



参数属性

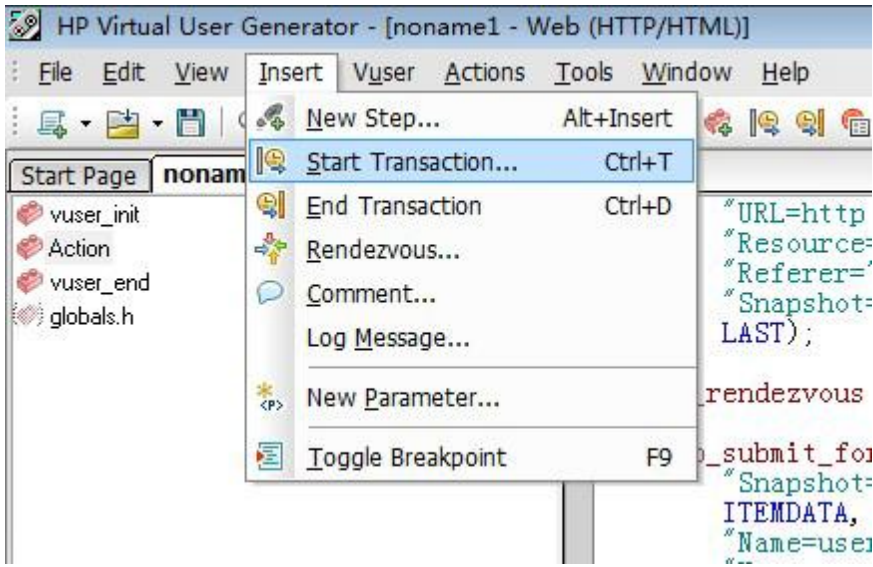
单击Edit with Notepad按钮，会出现记事本文档编辑框，可以在里面增加账号信息：

记事本参数编辑框

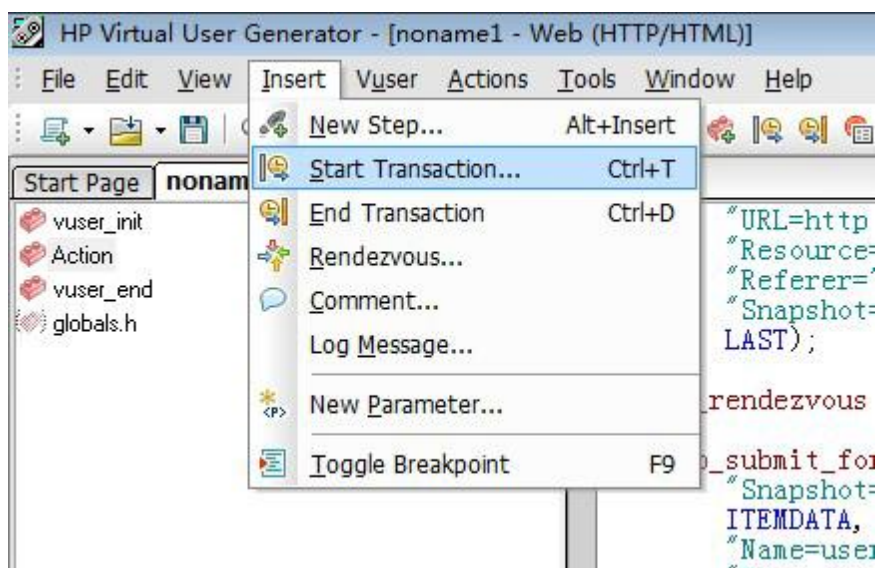


插入事务

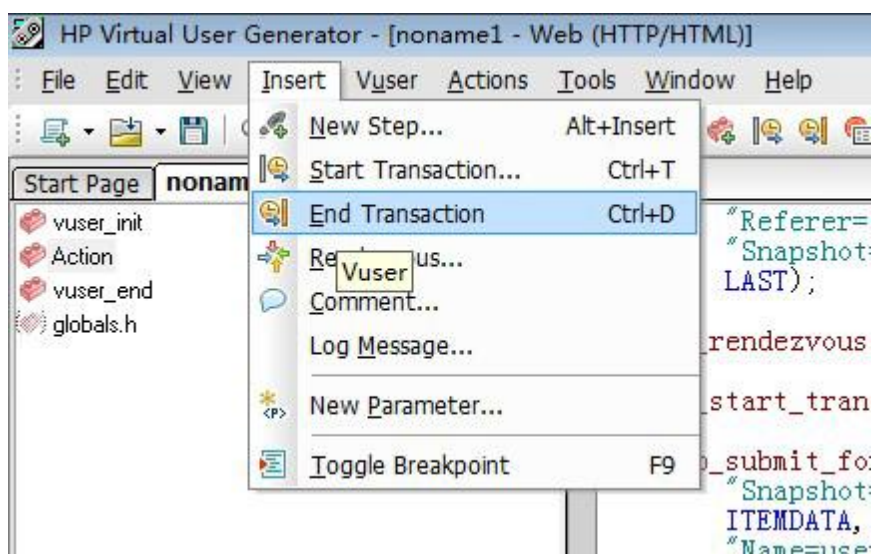
为了更好地衡量某个Action的性能，可以在Action的开始和结束位置插入一对事务，运行到该事务的开始点时开始计时，到结束点时计时结束，这段时间称为事务时间，事务时间反映了一个操作过程的响应时间。事务必须成对出现，即有始有终，而且通常不把思考时间添加到事务中间，添加方法如下图：



**事务开始点添加** 以下是管理员登录模块事务开始代码：lr\_start\_transaction("登录事务");



**事务结束点添加** 以下是管理员登录模块事务结束代码：lr\_end\_transaction("登录事务", LR\_AUTO);



## 插入集合点

为了能让多个虚拟用户能在同一时刻执行相同任务，还必须在脚本中插入集合点，可以通过如下图所示的选项来插入。

## 集合点添加

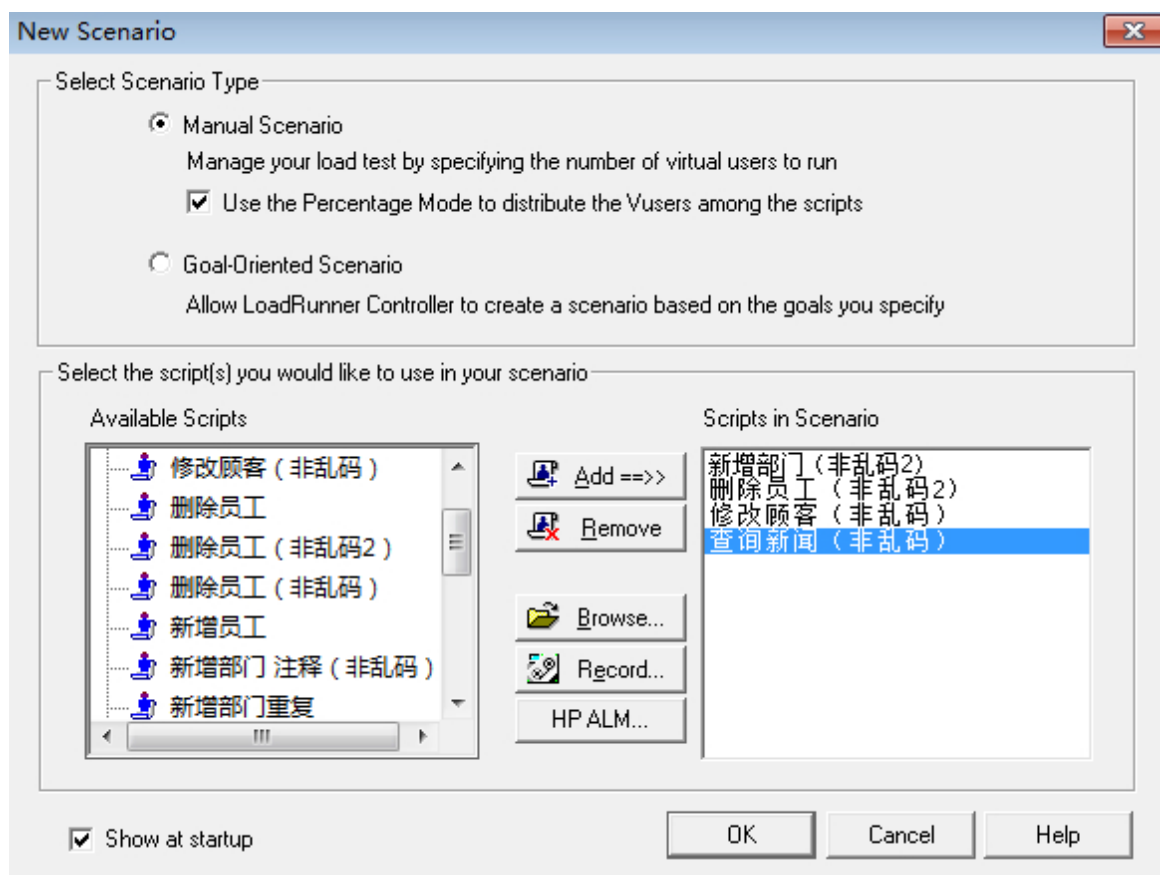
以下是管理员登录模块集合点代码：lr\_rendezvous("同时登录");

## 测试场景设计和执行过程

脚本录制工作完成以后，接下来就需要通过LoadRunner Controller来进行场景设置。

## 场景设置

场景的设置方式分为手动场景和基于目标场景两种，从场景设置对话框中可以选择。然后从对话框左侧Available Scripps中选择预先录制的脚本通过点击【Add】按钮将其添加到右侧的Scripps in Scenario中，如下图。为了更好地对系统进行性能评估，所以设置多组不同的用户数量进行比较，因此选择了手动方式。



场景设置对话框

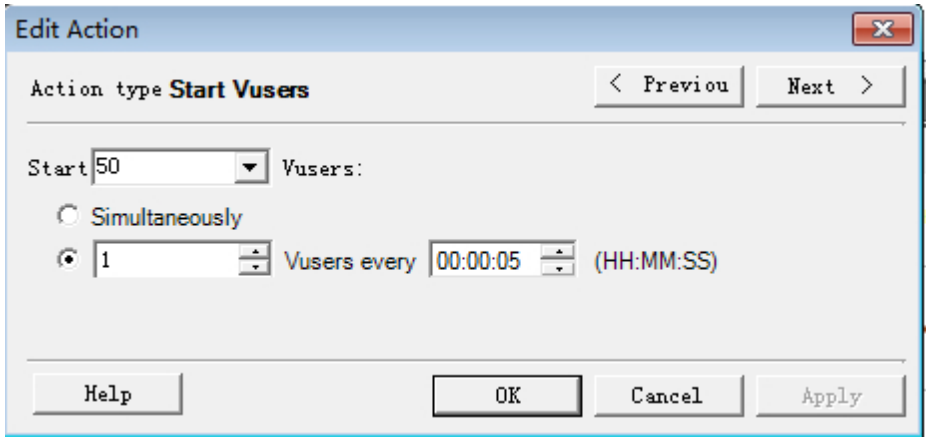
负载设置

负载设置主要有三个页面：加压页、持续时间页、减压页。以登录模块性能测试为例子，分别设定了数量为10、15、30、50个虚拟用户。

(1) 加压页

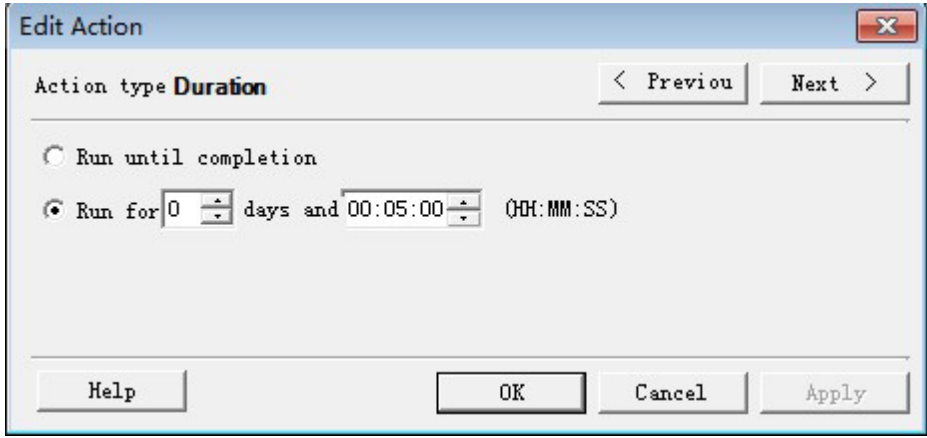
可用来指定一次性同时加载所有的虚拟用户(“Flat”测试)或者选择每隔一定的时间加载一定量的虚拟用户(“Ramp Up”测试)。如图所示是一共50个用户，每5秒钟增加1个用户：

加压页



(2) 持续时间页

可以用来指定场景是否要运行直到完成。对所有虚拟用户加压后，将场景运行一定的时间或者永久运行，如图所示是在所有50个用户增加完毕后，持续运行5分钟：



(3) 减压页

用来指定运行场景以后，选择立刻停止所有虚拟用户的运行或者每隔一定时间停止一定数量的虚拟用户，如图所示是5分钟持续运行完成后，每5秒停止1个用户，直至全部停止：

**Edit Action** ✕

---

Action type **Stop Vusers** < Previous    Next >

---

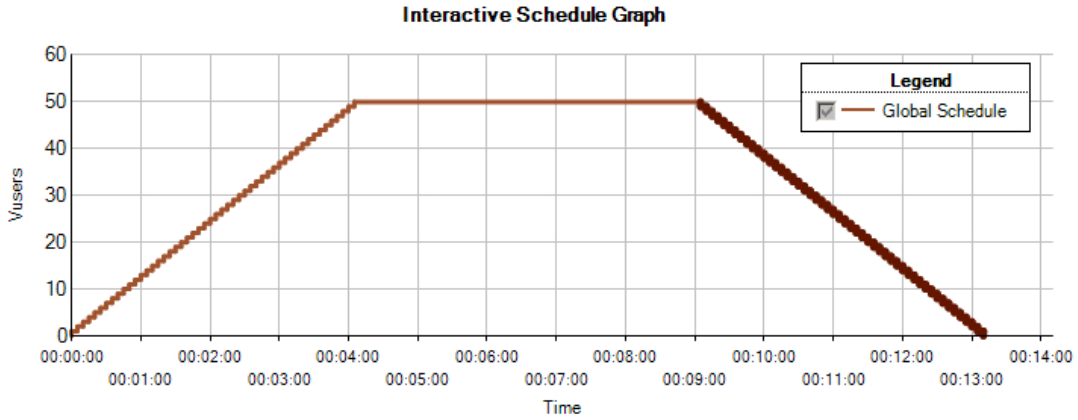
Stop All ▾ Vusers:

☐ Simultaneously

☒ 1 ▾ Vusers every 00:00:05 ▾ (HH:MM:SS)

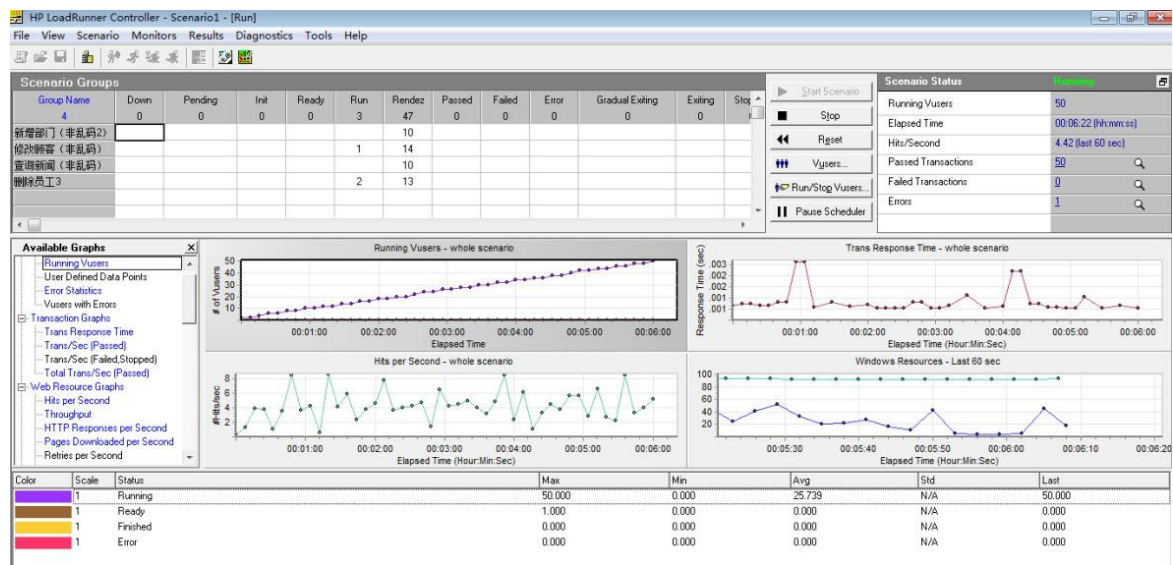
---

Help OK Cancel Apply



**场景设计图**

以上3个页面的设置完成以后，就可以点击执行按钮开始执行场景。



**混合业务测试场景运行图**

**测试结果分析**

(1) 事务响应时间

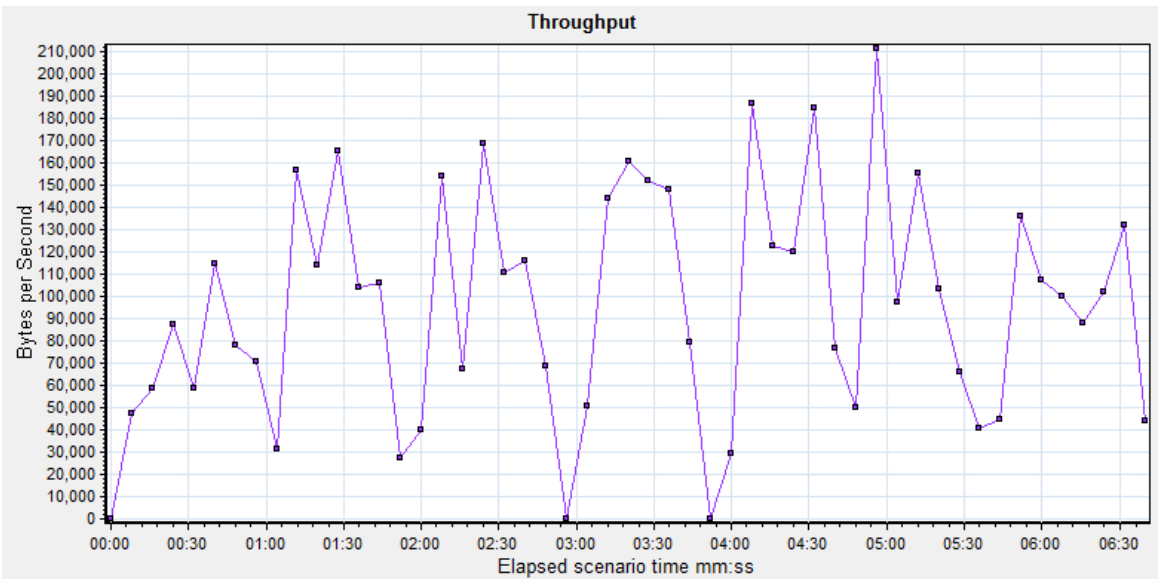


表 1 用户并发登录事务响应时间

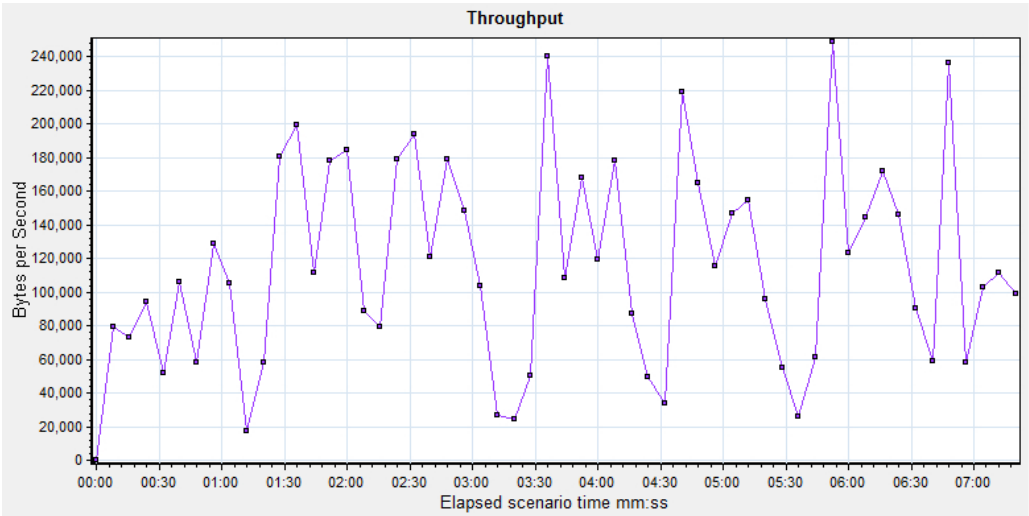
并发登录用户数	成功数	最小值(s)	平均值(s)	最大值(s)
10	10	2.279	5.960	11.804
15	15	1.953	8.295	30.957
30	30	2.439	15.012	83.569
50	50	1.816	23.057	108.952

( 2 ) 吞吐量

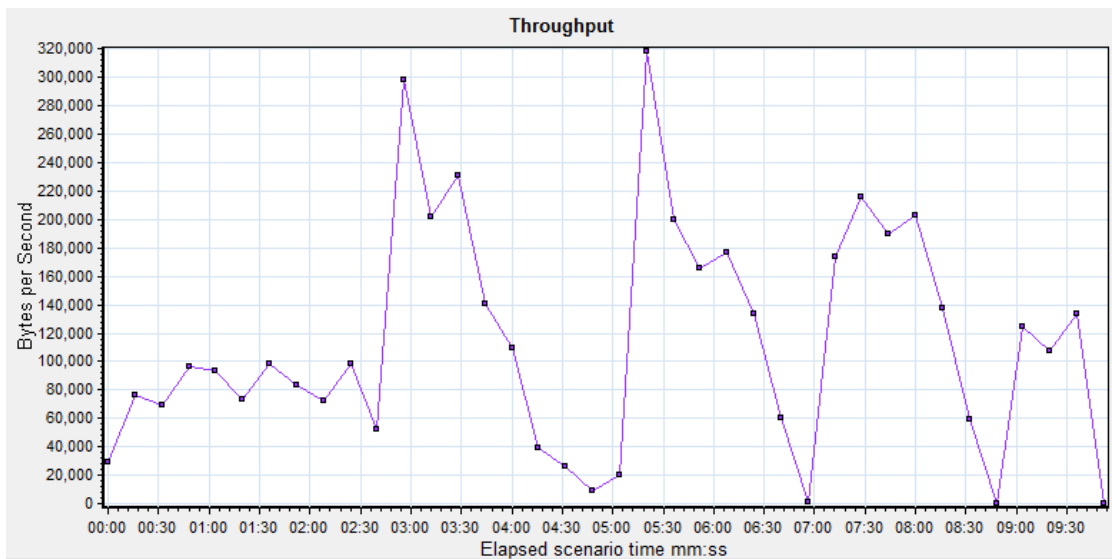
10 个用户并发登录的吞吐量：



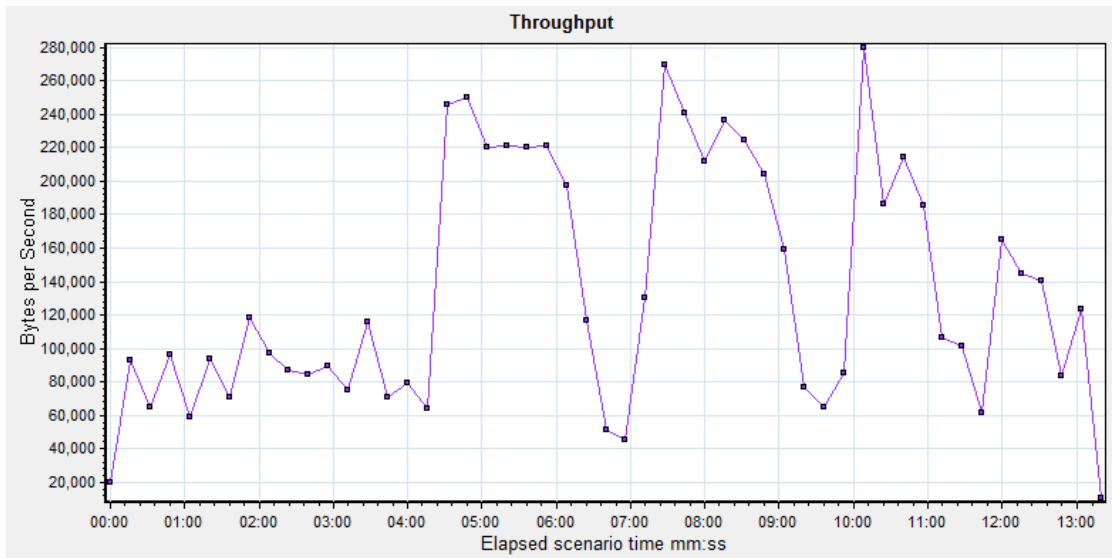
15 个用户并发登录的吞吐量：



30 个用户并发登录的吞吐量：



50 个用户并发登录的吞吐量：

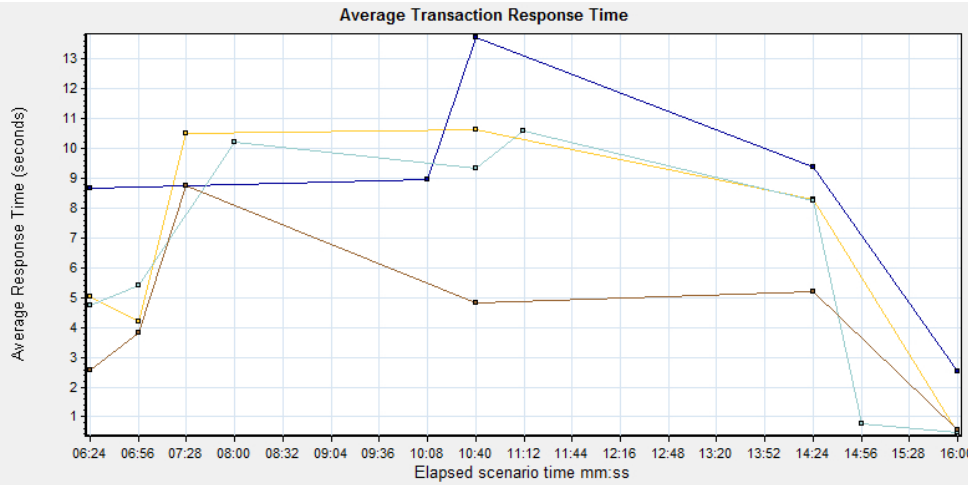


从表 1 和表 2 中，我们可以看出平均事务响应时间和平均吞吐量随着并发用户数的增加而增大。进一步分析比较，10 个虚拟用户的登录平均事务响应时间为 6.712s，15 个虚拟用户的时间为 8.749s，30 个虚拟用户的时间为 15.012s，50 个虚拟用户的时间为 23.057s。

对照之前性能测试用例设计中的期望输出结果 10 秒以内，只有 10 个和 15 个虚拟用户的平均事务响应时间是符合要求的。30 和 50 个的虚拟用户数的平均事务响应时间都是超出我们预期的，未达到了性能需求指标，还是有进一步提升和性能优化的空间。

## 业务组合测试结果分析

### （1） 平均事务响应时间

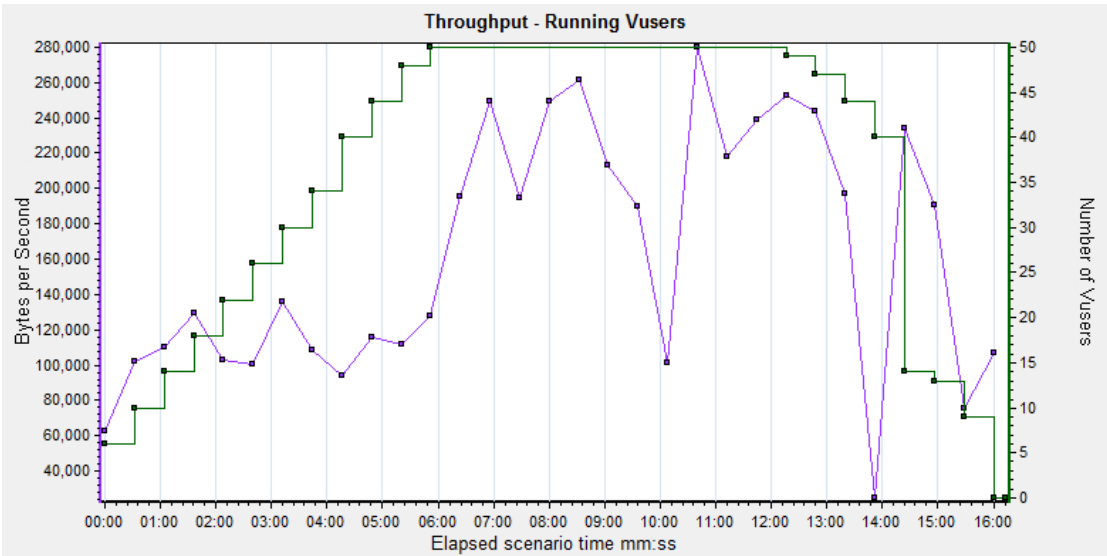


平均事物响应时间图（50个用户）

Color	Scale	Measurement	Graph's Minimum	Graph's Average	Graph's Maximum
	1	修改顾客邮编	0.485	6.524	10.621
	1	删除员工	2.529	8.647	13.714
	1	新增部门	0.478	6.208	10.562
	1	查询新闻	0.58	4.292	8.73

平均事物响应时间表

(2) 吞吐量

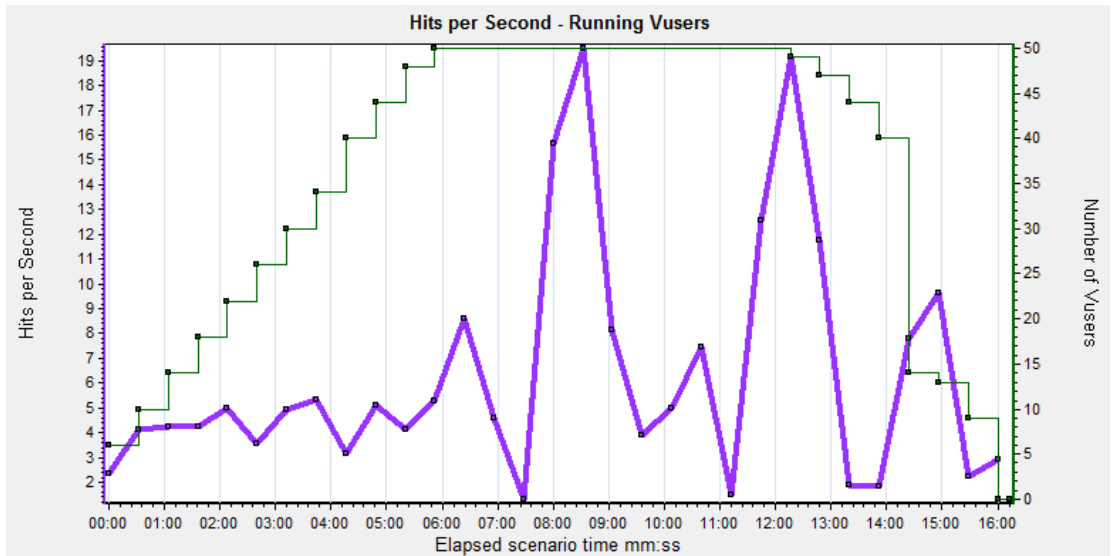


虚拟用户与吞吐量合并图

虚拟用户与吞吐量合并表

Color	Graph	Scale	Measurement	Graph's Minimum	Graph's Average
	Throughput	1	Throughput	24771.938	161832.277
	Running Vusers	1	Run	0	26.571

(3) 点击数



虚拟用户与每秒点击数合并图

虚拟用户与每秒点击数合并表

Color	Graph	Scale	Measurement	Graph's Minimum	Graph's Average	Graph's Maximum
Blue	Hits per Second	1	Hits	1.313	6.346	19.5
Red	Running Vusers	1	Run	0	26.571	50

首先对上图中的平均事务响应时间进行分析，增加部门、删除员工、修改客户信息、查询新闻4个业务操作的平均响应时间分别为6.208秒，8.647秒，6.524秒，4.292秒，都在10秒以内，符合预期的性能需求。

其次对系统吞吐量进行分析，从上图可以看出，接近10分和14分的时候，系统的吞吐量急剧向下，表现为一个明显的折线。这可能说明系统承载不了如此多的用户并发，也就是存在性能瓶颈。

最后分析每秒点击数，在上图中，可以看出起初点击数随着虚拟用户数的增加而增加，在8分多钟和12多分钟的时候达到了最高点击率，但在7分30秒、11分、13分至14分时点击率几乎为零，系统出现了性能瓶颈，需加以改善。