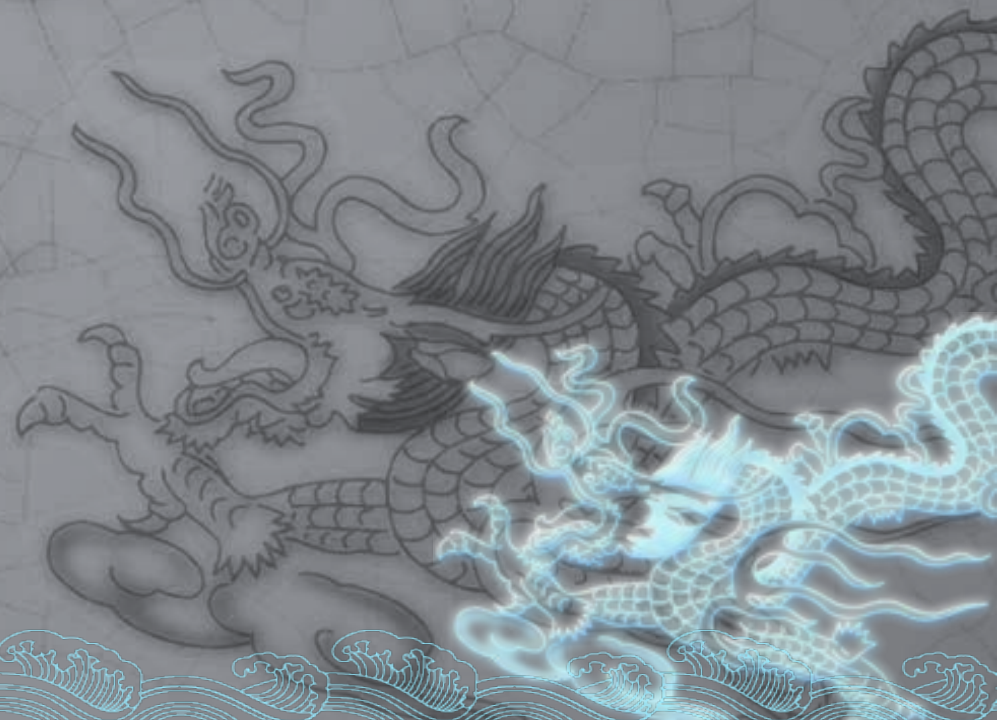
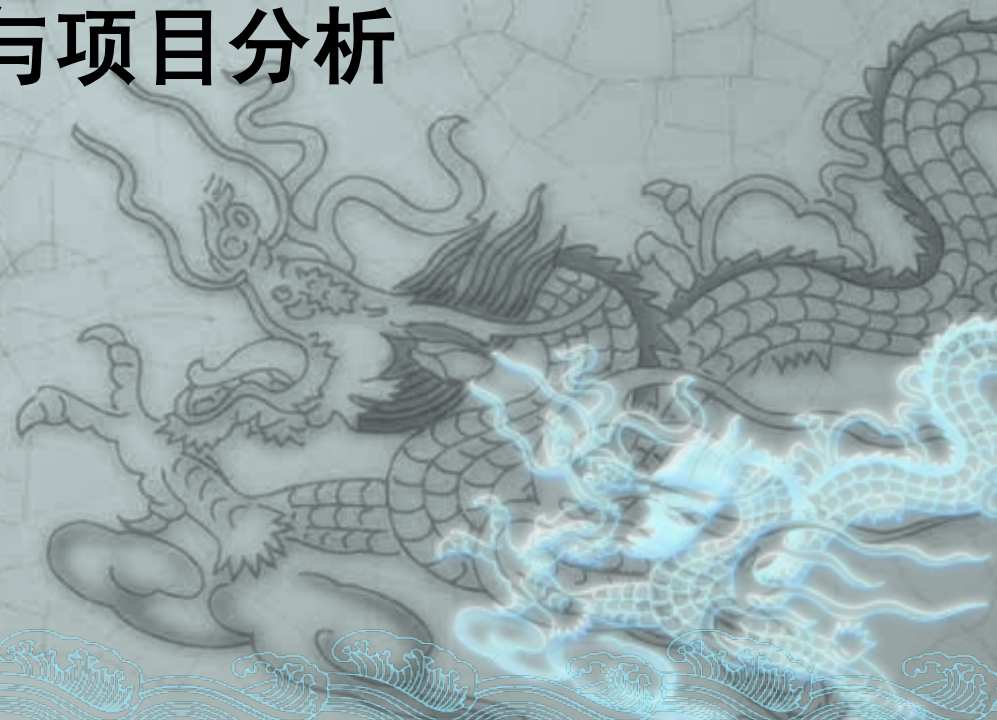


如何利用LoadRunner进行性能测试



分享提纲

- ◆ LoadRunner简介
- ◆ 脚本的录制与开发
- ◆ 场景的创建与执行
- ◆ 性能测试结果分析
- ◆ 逍遥问问性能测试流程与项目分析
- ◆ 总结逍遥问问性能测试



一、LoadRunner简介

■ 1 LoadRunner是什么

LoadRunner，是一种预测系统行为和性能的负载测试工具。通过以模拟上千万用户实施并发负载及实时性能监测的方式来确认和查找问题，LoadRunner能够对整个企业架构进行测试。通过使用 LoadRunner，企业能最大限度地缩短测试时间，优化性能和加速应用系统的发布周期。LoadRunner是一种适用于各种体系架构的自动负载测试工具，它能预测系统行为并优化系统性能。

■ 2 LoadRunner特点

- (1) 广泛支持业界标准协议
- (2) 支持多平台开发的脚本
- (3) 创建真实的系统负载
- (4) 强大的实时监控与数据采集功能
- (5) 精确分析结果，定位问题所在



由“性能”想到的。。。。

(1) 性能测试是什么？

在一定的负载情况下，系统的响应时间、资源利用、效率等特性是否满足特定的性能需求。

(2) 性能测试包含哪些方面？

压力测试、负载测试、并发测试、容量测试、配置测试、基准测试等。

(3) 应用系统性能指标主要有哪些？

响应时间、吞吐量、服务器资源利用

(4) 性能分析方法主要有哪些？

指标达成法、最优化分析（应用程序诊断、系统调优）。

(5) 性能测试的重要性，为什么要进行性能测试？

评估系统的能力

识别系统中的弱点

系统调优

验证可伸缩性和可靠性

.....



由“协议”想到的。。。

■ 协议分类表

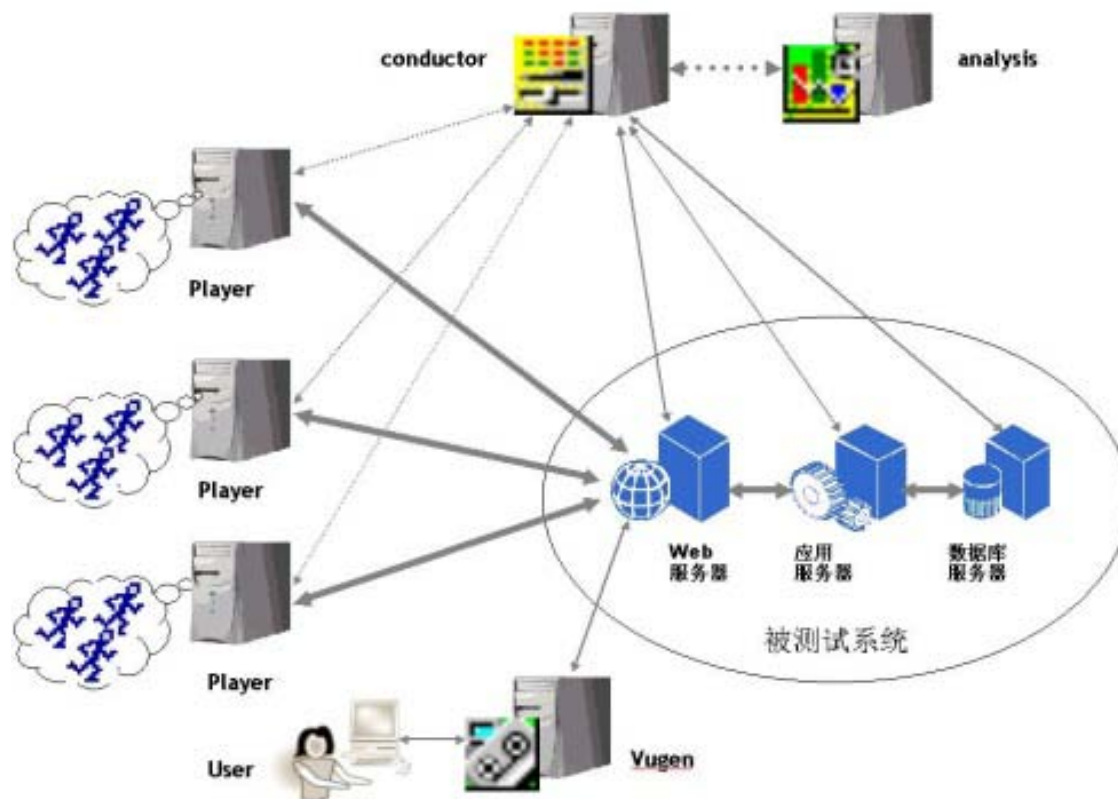
应用类型		建议选用协议
Web 网站		Web(HTTP/HTML)
FTP 服务器		File Transfer Protocol(FTP)
邮件服务器		Internet Messaging Application Protocol(IMAP)
		Post Office Protocol(POP3)
		Simple Mail Transport Protocol(SMTP)
C/S	客户端以 ADO、OLEDB 方式 后台数据库	Microsoft SQL Server Oracle、Sybase、DB2、Informix
	以 ODBC 方式连接后台数据库	ODBC
	没有后台数据库	Socket
分布式组件		COM/DCOM、EJB
无线应用		WAP、Palm



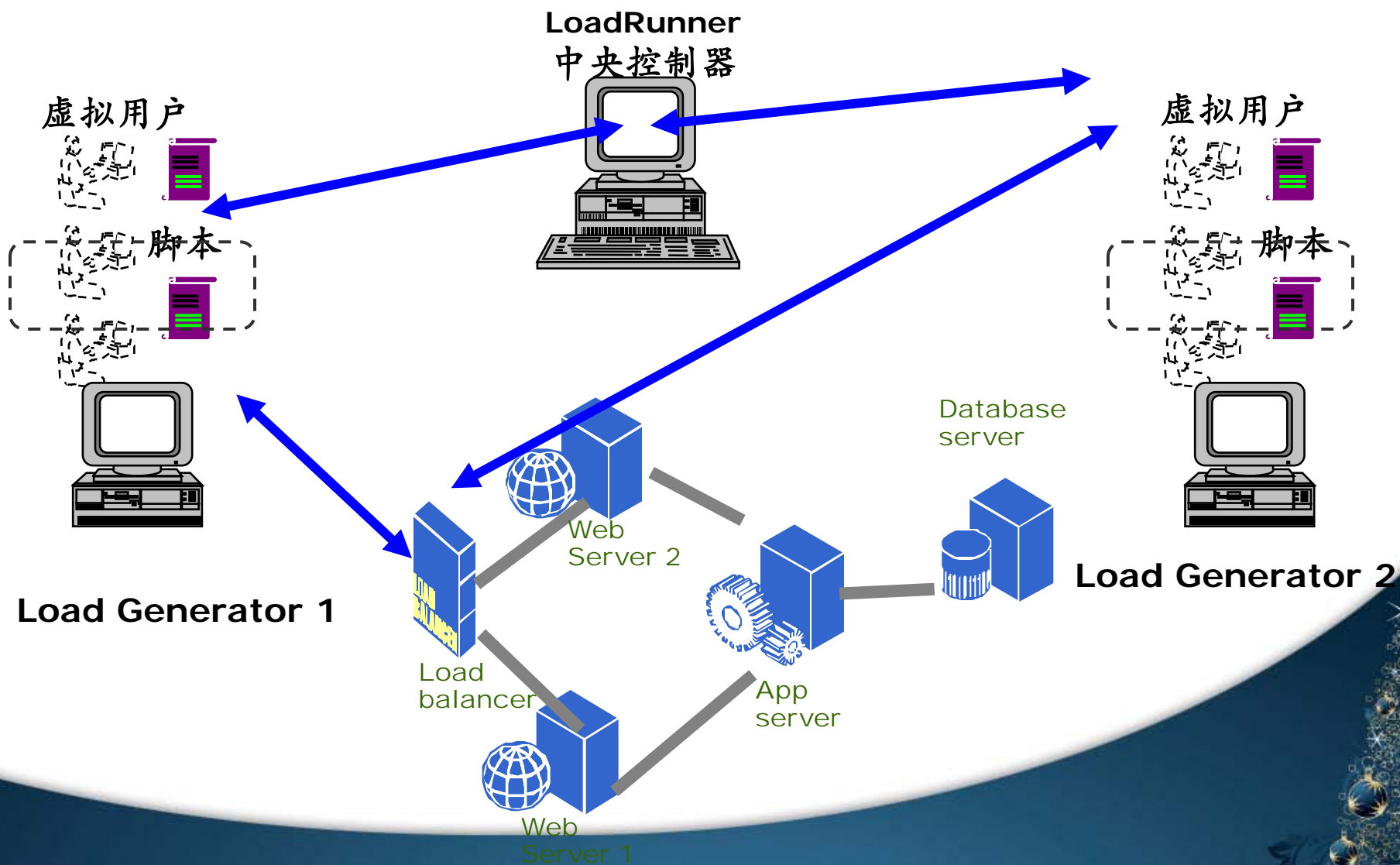
■ 3 LoadRunner工具组成

LoadRunner主要由以下几个部分组成：

- ❧ 脚本生成器VuGen
- ❧ 压力调度和监控系统Controller
- ❧ 结果分析工具Analysis



■ Load Generator-负载生成器（运行脚本的负载引擎）



4 LoadRunner实现原理

■ 用户行为模拟

- (1) 不同用户使用不同的数据（通过“参数化”实现）
- (2) 多用户并发操作（通过“集合点”实现）
- (3) 用户请求间的依赖关系（通过“关联”实现）
- (4) 请求间的延时时间（通过“思考时间”实现）

■ 性能指标监控

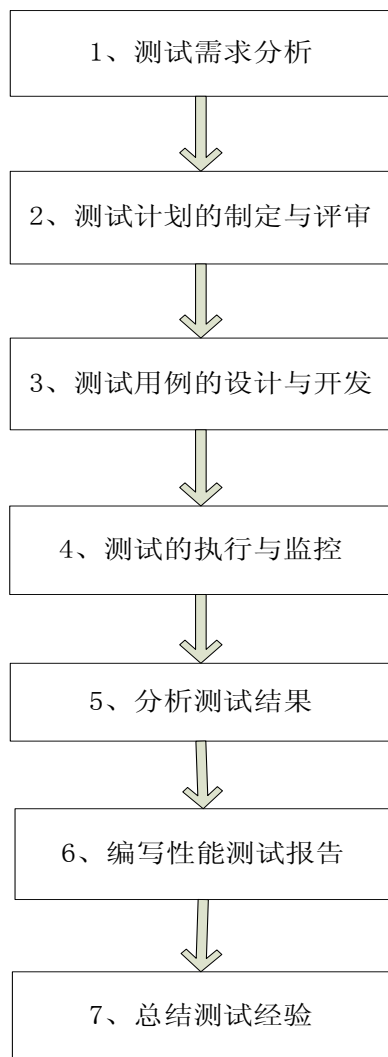
- (1) 请求响应时间监控（通过“事务”实现）
- (2) 服务器处理能力监控（通过“事务”计算吞吐量获得）
- (3) 服务器资源利用率监控（计数器接口）

■ 性能调优

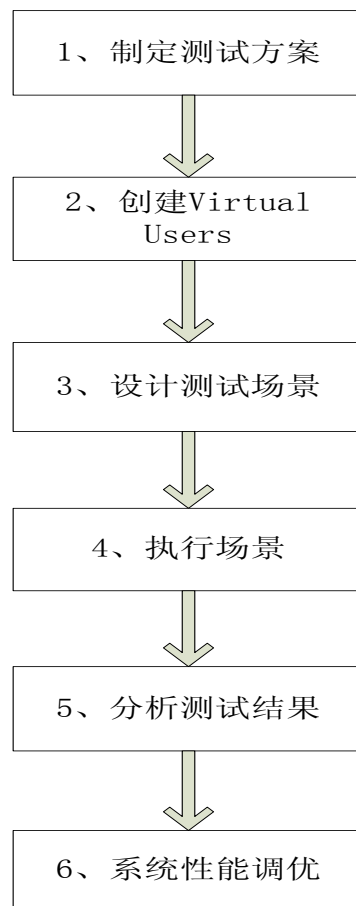
通过指标的监控发现系统存在的性能缺陷，利用分析工具定位并修正性能问题.



5 LoadRunner测试步骤



性能测试流程



LoadRunner工作流
程

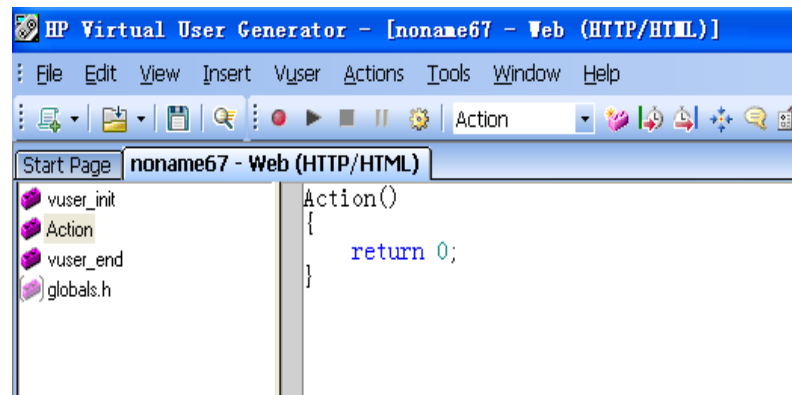
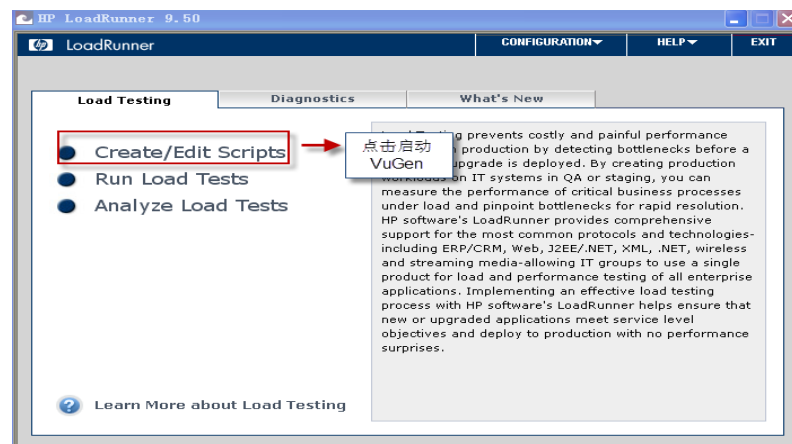


二、脚本的录制与开发

■ 1 启动Visual User Generator

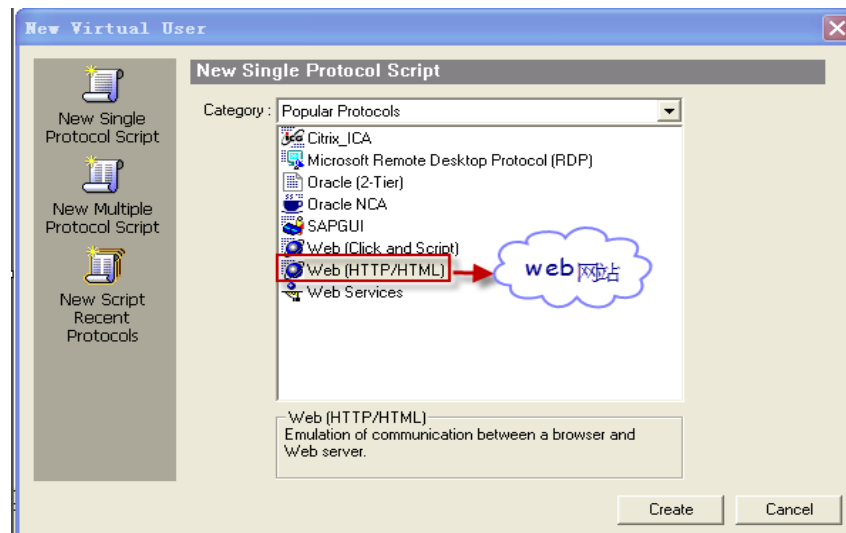
选择 Start > Programs > LoadRunner > Virtual User Generator 来开始vugen，打开vugen主窗口。

每一个脚本都包括三部分：
vuser_init，Actions 和vuser_end。
通常把登录部分放到 vuser_init
中，登录后的操作部分放到
Action 中，注销关闭部分放到
vuser_end 中。当需要重复运行脚
本的时候，仅仅脚本的action部分
被重复，而 vuser_init 和
vuser_end部分是不重复执行的。



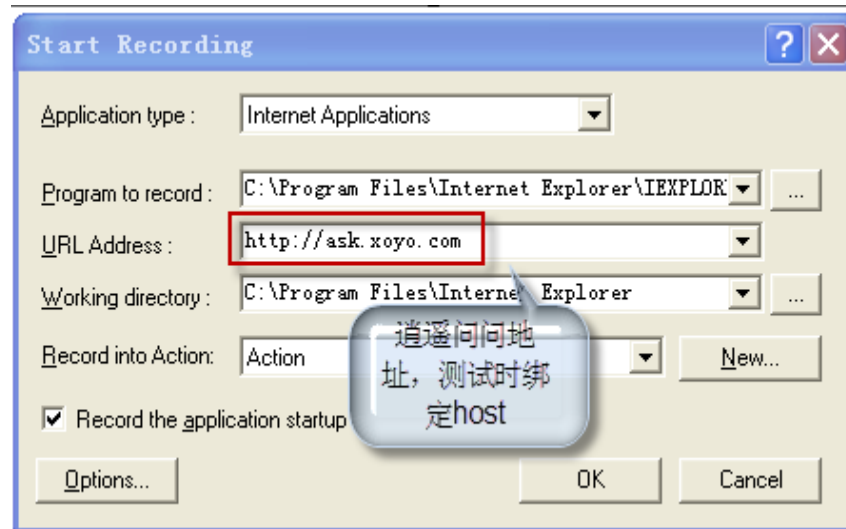
■ 2 新建测试脚本，选择系统通讯的协议。

Loadrunner提供单协议（single protocol）和多协议（multiple protocol）两种模式。在录制脚本时，不同的协议打开的参数对话框也不同。



■ 3 对于Web 应用，选择 Web(HTTP/HTML)协议。

在URL Address中输入要需要测试的网址。在Record into Action中选择要把录制的脚本放到哪一个部分。



■ 4 录制工具条介绍

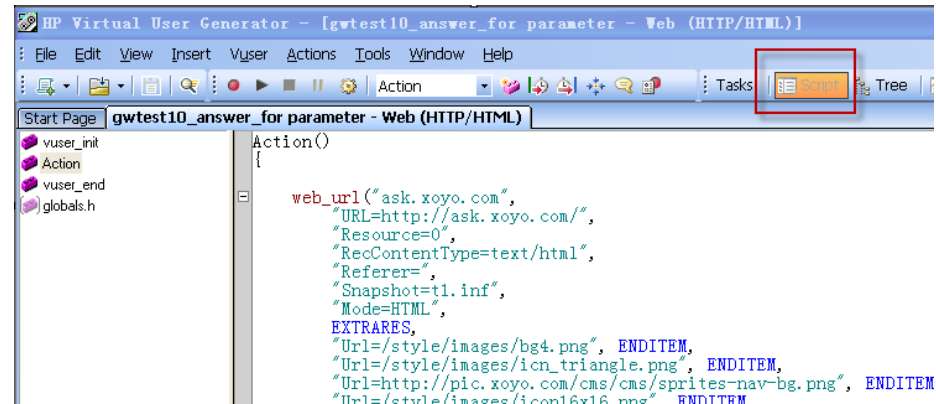
录制过程中，在屏幕上会出现有一个工具条。录制完成后，按下“结束录制”按钮，VuGen 自动生成用户脚本，完成录制过程。



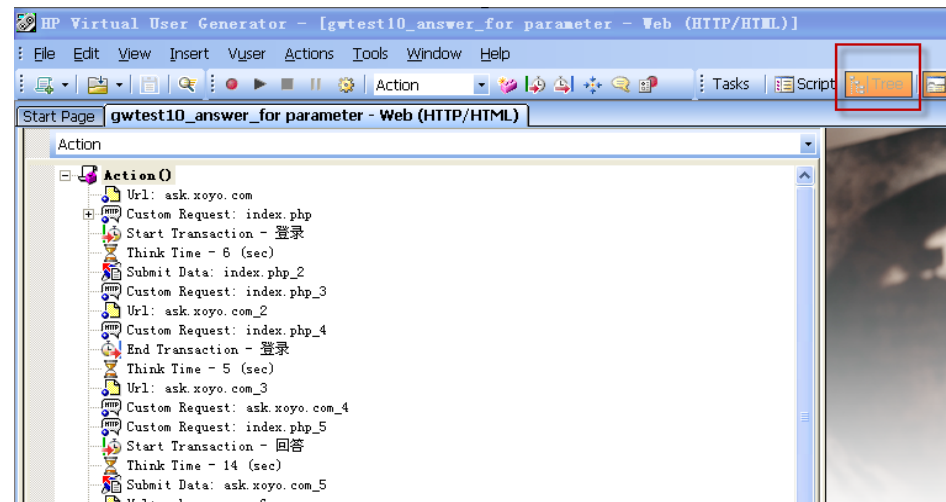
■ 5 查看脚本——两种方式：脚本视图和树视图

✓ 脚本视图：是一种基于文本的视图。

VuGen 将在编辑器中显示带有彩色编码的函数及其变量值的脚本。



✓ 树视图：是基于图标视图。对于录制期间所执行的每一步骤，VuGen 都在测试树中生成一个图标和一个标题，并附带相应的录制快照。



■ 6、完善优化测试脚本

■ a 插入事务 (Transaction)

事务是衡量服务器特定业务流程的性能。在运行测试脚本时，LoadRunner 运行到该事务的开始点时，就会开始计时，直到运行到该事务的结束点，计时结束。这个事务的运行时间会在结果中显示。

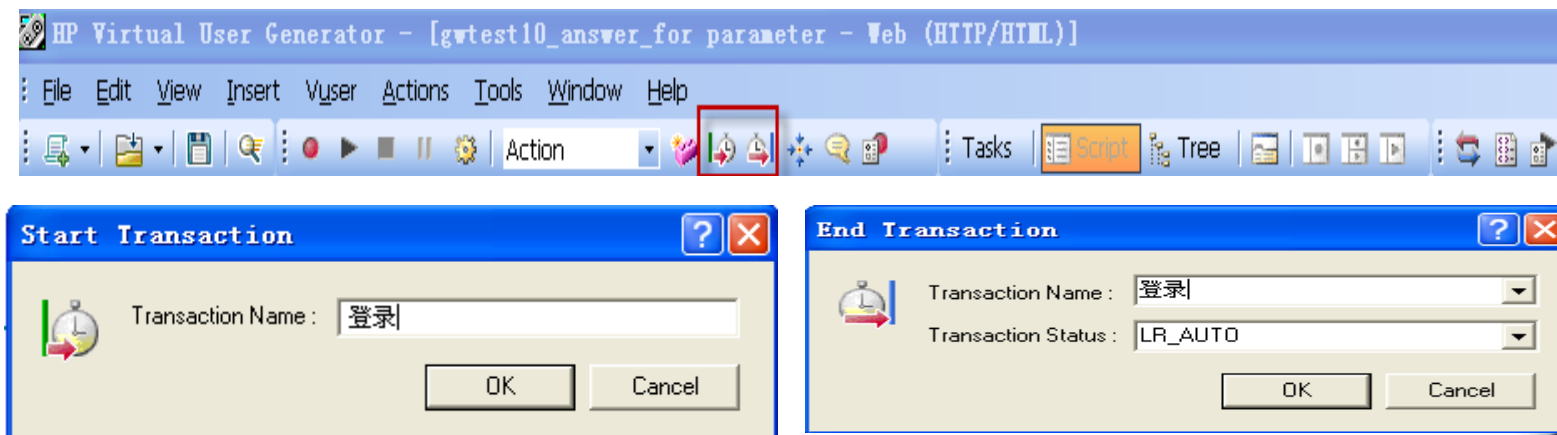
注意：（1）事务是成对出现的；

（2）两种添加方式：录制过程中、录制结束后的脚本中。

例：lr_start_transaction("登录");

.....

lr_end_transaction("登录",LR_AUTO);



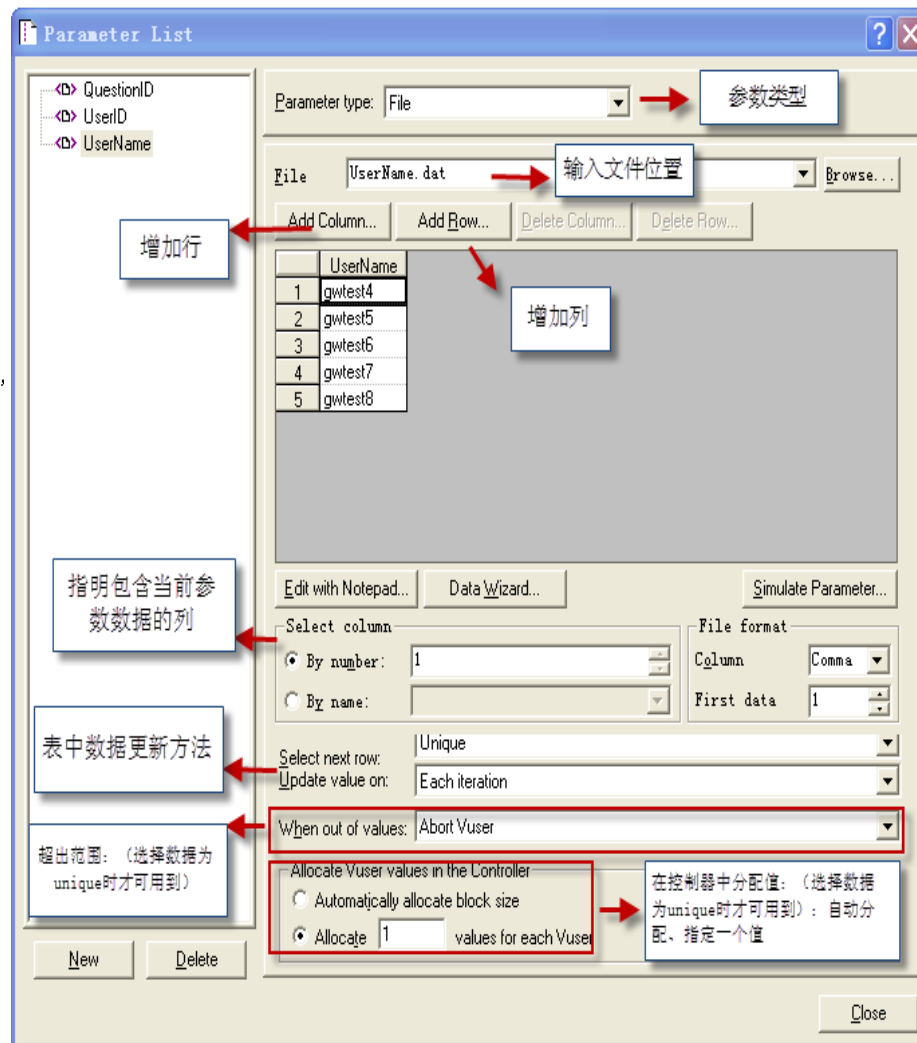
■ b 参数化

作用： 1) 在脚本中用参数取代常量值；
2) 设置参数的属性以及数据源。

```
web_url("ask.xoyo.com_3",
"URL=http://ask.xoyo.com/?act=q&id={QuestionID}",
"Resource=0",
"RecContentType=text/html",
"Referer=http://ask.xoyo.com/",
"Snapshot=t10.inf",
"Mode=HTML",
EXTRARES,
"Url=/style/images/icon_triangle_down.png", "Referer=http://ask.xoyo.com/?act=q&id={QuestionID}", ENDITEM,
"Url=/style/images/bg4.png", "Referer=http://ask.xoyo.com/?act=q&id={QuestionID}", ENDITEM,
"Url=/style/images/icon16x16.png", "Referer=http://ask.xoyo.com/?act=q&id={QuestionID}", ENDITEM,
LAST);

web_custom_request("ask.xoyo.com_4",
"URL=http://ask.xoyo.com/?act=topjs&id={QuestionID}&gameid=18&n=12",
"Method=GET",
"Resource=0",
"RecContentType=text/html",
"Referer=http://ask.xoyo.com/?act=q&id={QuestionID}",
"Snapshot=t11.inf",
"Mode=HTML",
"EncType=application/x-www-form-urlencoded",
"Body=act=topjs&id={QuestionID}&gameid=18&n=12",
EXTRARES,
```

参数设置	选择项	具体含义
Update value on	Each iteration	在每次迭代时更新参数的值
	Each occurrence	在参数每次出现的时候更新参数的值
	once	只在第一次迭代的时候取一次值，不在更改
Select next row	Sequential	按照顺序从参数表中取值
	Random	随机从参数表中取值
	Unique	每次取唯一值



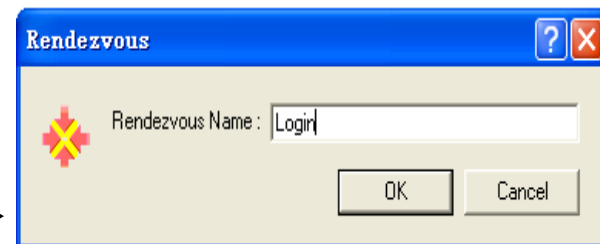
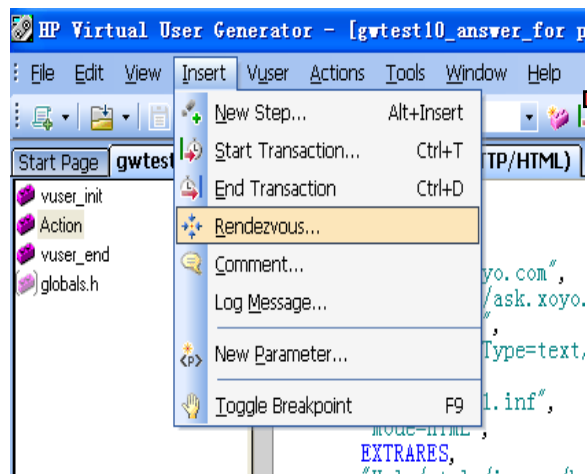
■ C 集合点:

集合点是为了衡量在加重负载的情况下服务器的性能情况。在测试计划中，可能会要求系统能够承受1000人同时提交数据，在LoadRunner 中可以通过在提交数据操作前面加入集合点，这样当虚拟用户运行到提交数据的集合点时，LoadRunner 就会检查同时有多少用户运行到集合点，如果不到1000 人，LoadRunner 就会命令已经到集合点的用户在此等待，当在集合点等待的用户达到1000 人时，LoadRunner 命令1000 人同时去提交数据，从而达到测试计划中的需求。

注意：集合点经常和事务结合起来使用。集合点只能插入到Action 部分，vuser_init 和vuser_end 中不能插入集合点。

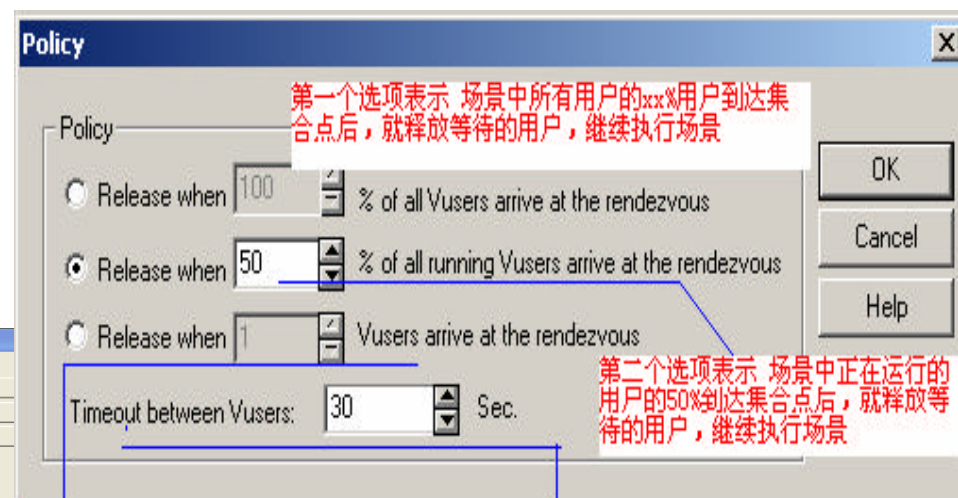
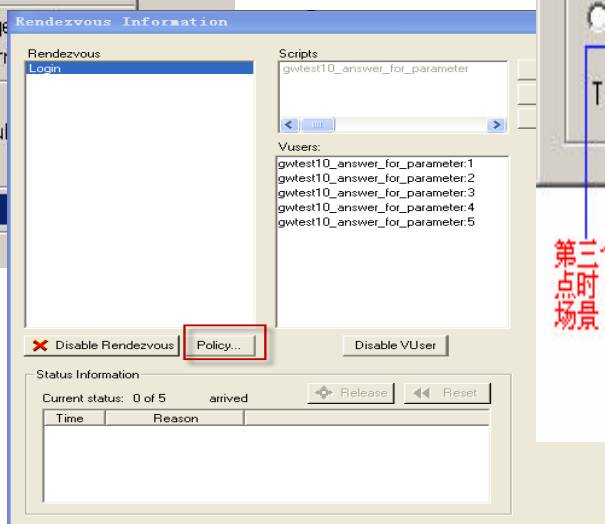
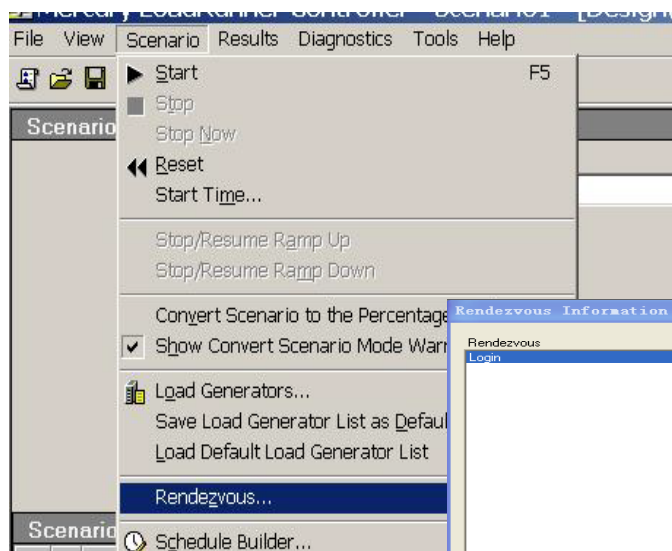
插入集合点方法：1) 在需要插入集合点的前面，选择菜单 Insert->Rendezvous。

2) 在弹出的Rendzvous对话框中，输入要插入的集合点的名称，点击OK后，集合点插入完毕。集合点在脚本中的代码：
`lr_rendezvous(“名称”)`



`lr_rendezvous("Login");`

如果在脚本中设置了集合点，还需要在Controller 中设置集合点策略，从菜单中调出集合点策略设置窗口，根据需求设置集合点策略。



第三个选项表示 当xx个用户到达集合点时，就释放等待的用户，继续执行场景

该选项表示 当第一个用户到达集合点时，等待30秒，如果在30秒内等到指定数量的用户到达集合点，就开始继续执行场景。如果在30秒时还没有等到指定数量的用户到达集合点，就不再等待，开始释放等待的用户，继续执行场景

■ d 检查点

在进行压力测试时，为了检查Web 服务器返回的网页是否正确，VuGen 允许我们插入Text/Imag 检查点，这些检查点验证网页上是否存在指定的Text 或者Imag，还可以测试在比较大的压力测试环境中，被测的网站功能是否保持正确。以下是插入检查点的步骤及检查点在脚本中的函数表示。

The screenshot shows the QTP interface with the 'Add Step' dialog box open. The 'Step Type' list on the left has 'Text Check' selected. The 'Text Check Properties' dialog box is also open, showing the 'General' tab with search parameters. The 'Insert' menu is open, showing options like 'New Step...', 'Start Transaction...', 'End Transaction...', 'Rendezvous...', 'Comment...', and 'Log Message...'. Red arrows point from the 'Add Step' dialog box to the 'Text Check Properties' dialog box and from the 'Insert' menu to the 'New Step...' option.

Add Step Dialog Box:

- Step Type: Text Check
- Find Function: w

Text Check Properties Dialog Box:

- General tab
- Search Parameters:
 - Search for: welcome
 - Right of: ask
 - Left of: name

Insert Menu:

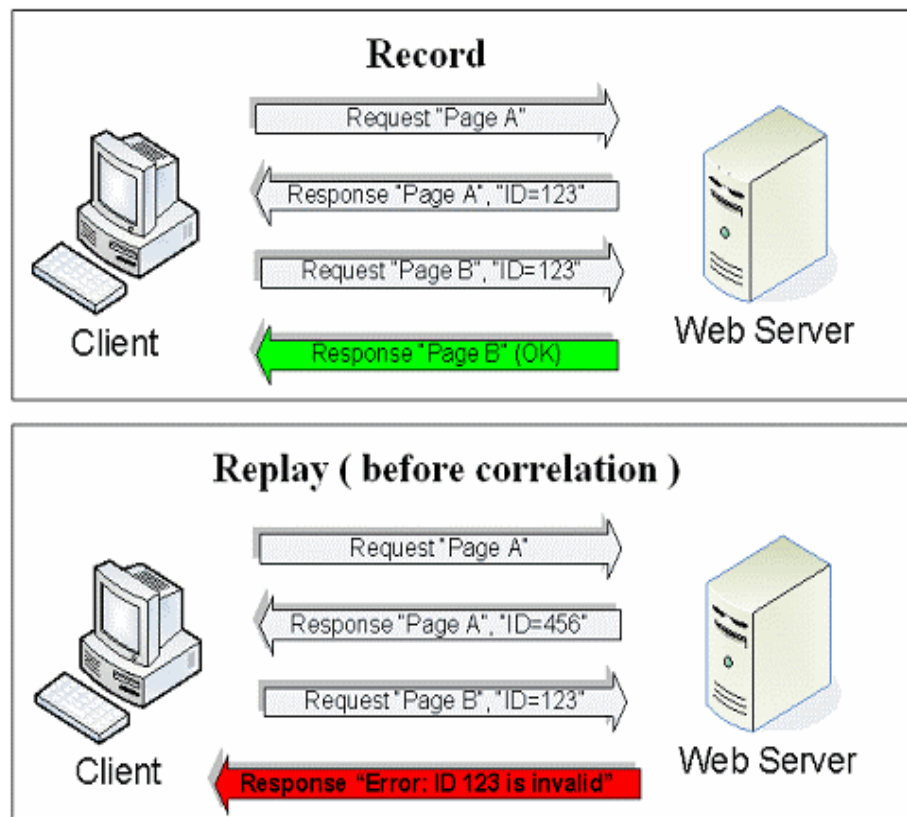
- New Step... (Alt+Insert)
- Start Transaction... (Ctrl+T)
- End Transaction... (Ctrl+D)
- Rendezvous...
- Comment...
- Log Message...

■ 关联

概念：所谓关联（correlation）就是把脚本中某些写死的（hard-coded）资料，转变成是来自服务器的、动态的、每次都不一样的资料。

原理：服务器在每个浏览器第一次跟它要资料时，都会在资料中夹带一个唯一的辨识码，接下来就会利用这个辨识码来辨识跟它要资料的是不是同一个浏览器。一般称这个辨识码为Session ID。对于每个新的交易，服务器都会产生新的Session ID给浏览器。这也就是为什么执行脚本会失败的原因，因为VuGen还是用旧的Session ID向服务器要资料，服务器会发现这个Session ID是失效的或是它根本不认识这个Session ID，当然就不会传送正确的网页资料给VuGen了。

要对付这种服务器，我们必须想办法找出这个Session ID到底是什么、位于何处，然后把它记录下来，放到某个参数中，并且取代掉脚本中有用到Session ID的部份，这样就可以成功骗过服务器，正确地整个交易了。



哪些错误代表着我应该做关联？

- (1)错误讯息有可能会提醒要重新登入；
- (2)有可能直接就显示HTTP 404的错误讯息。

总而言之：

当脚本中的数据每次回放都发生变化时，并且这个动态数据在后面的请求中需要发送给服务器，那么这个内容就需要通过关联来询问服务器，获得该数据的变化结果。

关联函数

(1)web_reg_save_param

```
web_reg_save_param("ParamName","LB=", "RB=", "Ord=1", "Search=ALL",  
LAST);
```

通过左右边界的限定，将关联出来的内容放在一个参数中，当关联对象寻找范围与寻找结果通过函数值体现出来。

(2)web_save_param_length

```
web_save_param_length("suname",LAST)
```

将创建一个名为 suname_Length的参数，函数执行完成后将参数suname的长度值保存到suname_Length中。



手动关联

- (1) 使用相近的业务流程与资料，录制二份脚本
- (2) 使用WinDiff工具帮助找出需要关联的资料
- (3) 使用web_reg_save_param函数手动建立关联
- (4) 将脚本中有用到关联的资料，以新数替代

自动关联

方案一.Rules Correlation

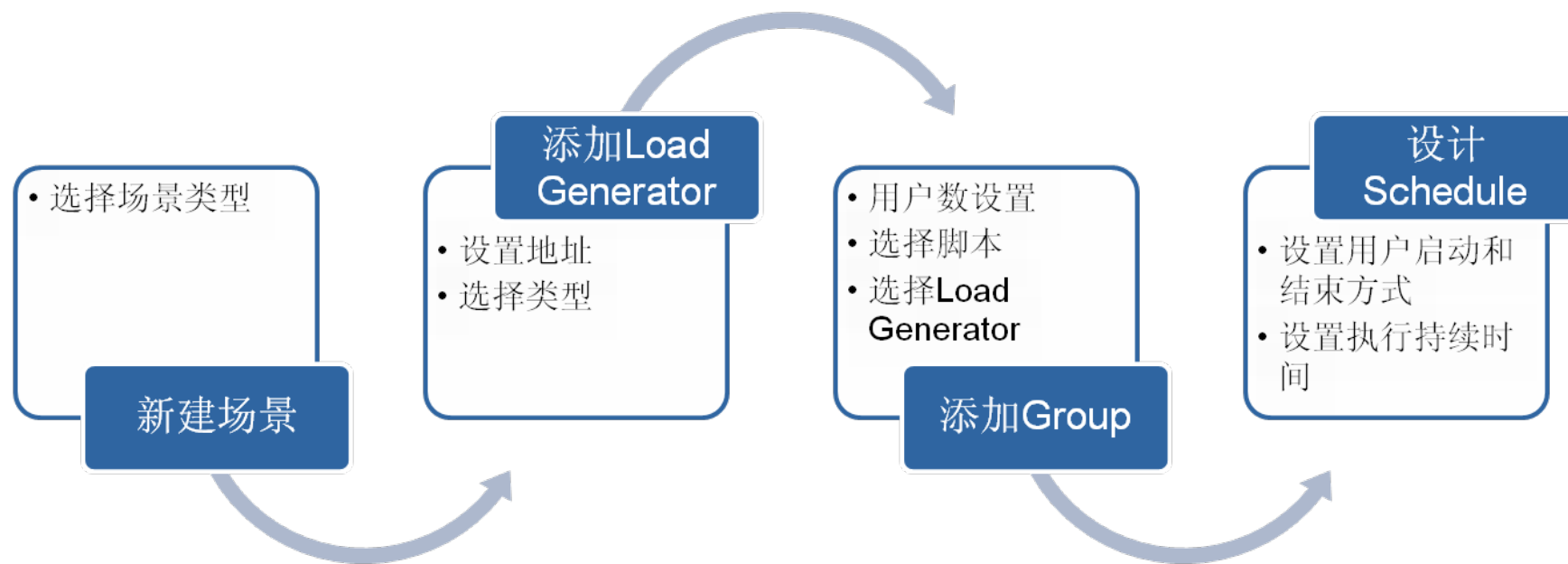
- (1) 内建 (Built-in Correlation) : 在【Recording Options】>【Internet Protocol】>【Correlation】中启用关联规则，则当录制这些应用系统的脚本时，VuGen会在脚本中自动建立关联。
- (2) 使用者自订 (User-defined Rules Correlation) : 在【Recording Options】>【Internet Protocol】>【Correlation】建立新的关联规则。

方案二.Correlation Studio

在执行脚本后才会建立关联，也就是说当录制完脚本后，脚本至少须被执行过一次，Correlation Studio才会作用。



三、场景的创建与执行



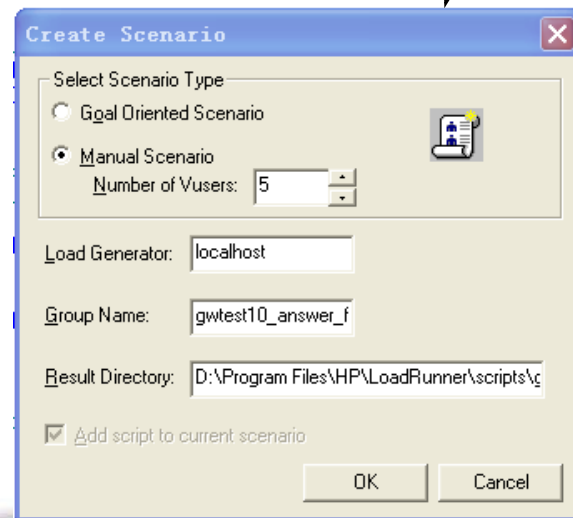
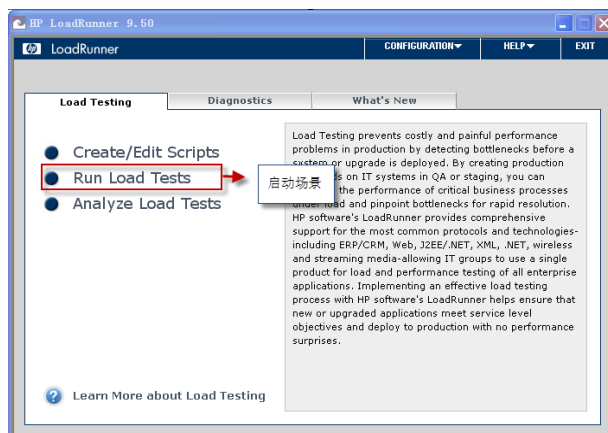
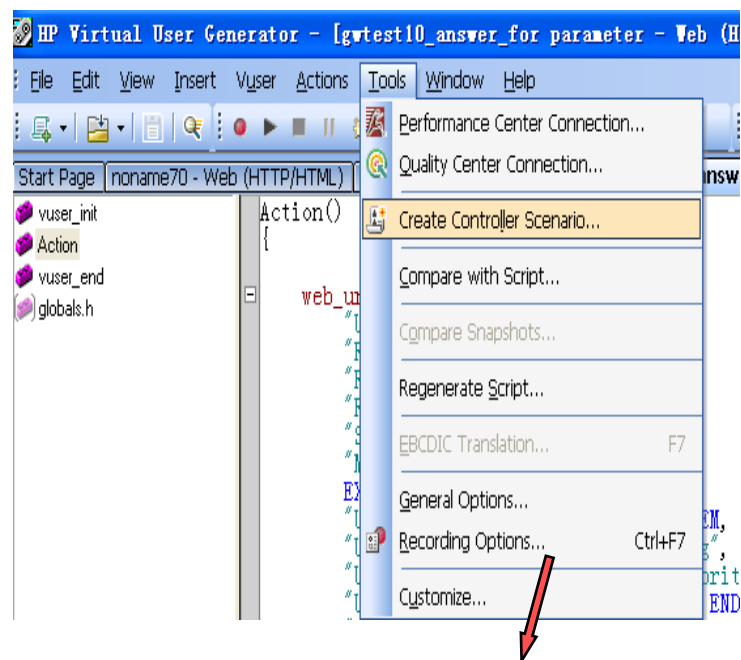
设计场景流程



■ 1 启动场景

Loadrunner的场景能够描述在测试活动中发生的各种事件。一个场景包括一个运行虚拟用户活动的Load Generator机器列表，一个测试脚本的列表以及大量的虚拟用户和虚拟用户组。

点击开始-> LoadRunner-> Controller来启动Controller程序。在VuGen下，也可点击Tools下的菜单，即可启动Controller，并将当前脚本加入到场景中。



■ 2、引入脚本选择脚本文件

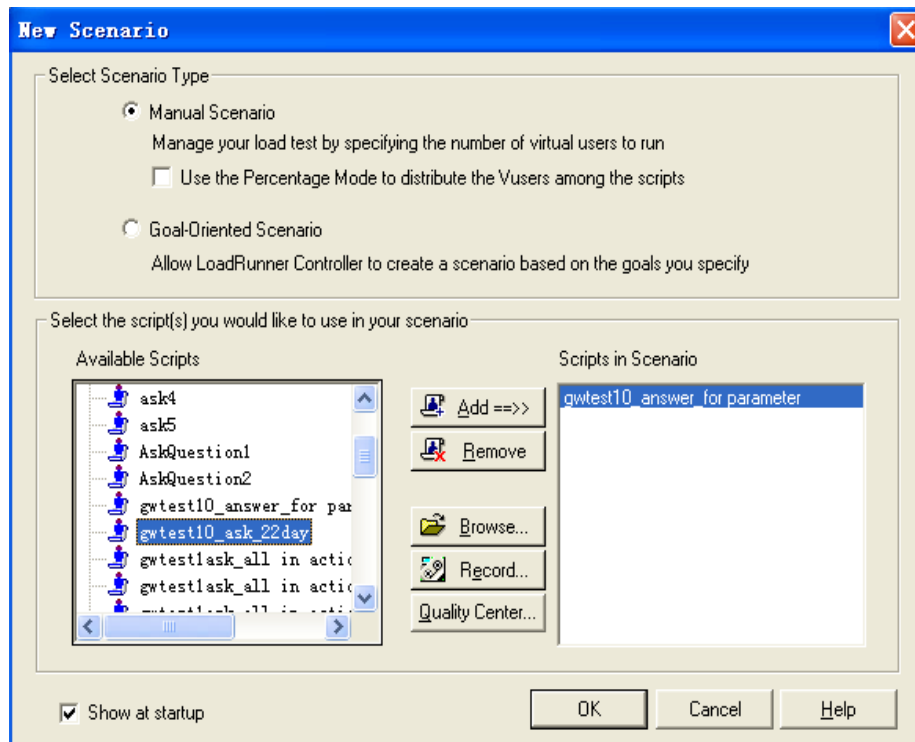
第一次打开Controller工具时会弹出下面窗体在Select Scenario Type中，有两个选项：

(1) Manual Scenario：指手动的设置测试的环境参数；

(2) Goal – Oriented Scenario：指选择预先设置的环境参数

通常选择Manual Scenario，手动进行测试场景设置。

在Available Scripts中选择脚本名称，然后点击Add，即可，这里我们可以同时引入若干个脚本，让不同的用户执行不同的脚本。

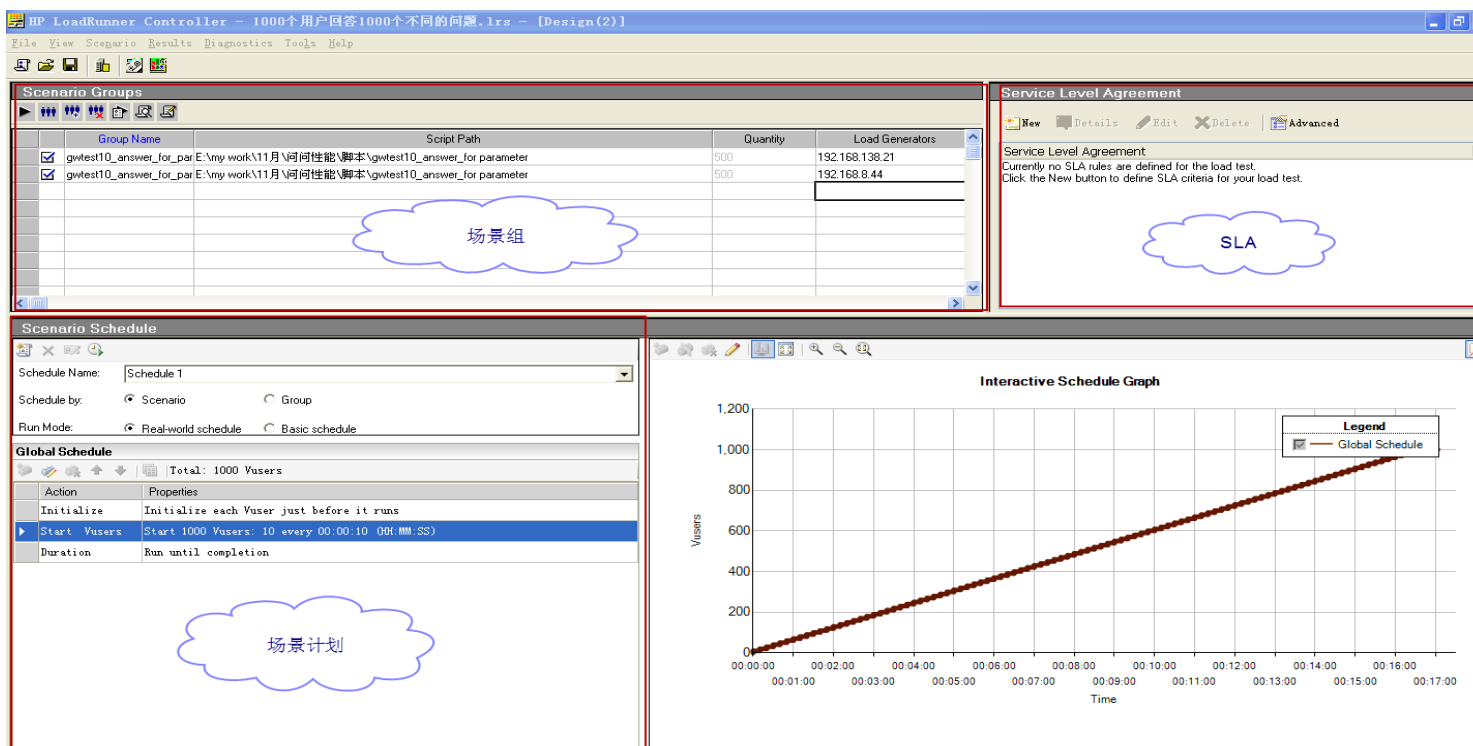


■ 3、设计测试场景

场景计划： 设置负载行为，确定将负载应用于应用程序的速率、负载测试持续时间以及如何停止负载。

场景组： 配置 Vuser 组。定义典型用户运行的操作、运行的 Vuser 数及 Vuser 运行时所用的计算机。

SLA: Service-Level Agreement，服务等级协议，运行场景时，LoadRunner 收集并存储与性能相关的数据。分析运行情况时，Analysis 将这些数据与 SLA 进行比较，并为预先定义的测量指标确定 SLA 状态。



■ 4 目标场景的场景设置

Create Scenario

Select Scenario Type

☒ Goal Oriented Scenario **目标场景**

☐ Manual Scenario

Number of Users: 5

Load Generator: localhost

Script Name: gwtest1ask_all in ac

Result Directory: E:\my work\11月\问问性能\脚本\gwtest1

☒ Add script to current scenario

OK Cancel

LoadRunner Controller - Scenario1 - (Design1)

File View Options Scenario Properties Help

Scenarios Single

Scenario Name	Script Path	Goal Target	Load Settings
gwtest1ask_all in ac	E:\my work\11月\问问性能\脚本\gwtest1	100%	100% Load Settings

Service Level Agreement

Service Level Agreement

Currently no SLA rules are defined for this load test.
Click the New button to define SLA criteria for your load test.

Scenarios Goal

Goal Profile Name: Hits per Second 200

Goal: 100.00 Hits per Second

Min Number of Users: 50

Max Number of Users: 150

Scenario Duration: 00:30:00 (Minutes:SS) after the target has been achieved

Load Behavior: Reach target hits per second using automatic ramp up

OK Cancel

Edit Scenario Goal

Goal Profile Name: Hits per Second 200

Rename Delete New

Scenario Start Time...

Define Scenario Goal

Goal Type: Virtual Users

Reach goal of: Virtual Users
Hits per Second
Transactions per Second
Transaction Response Time
Pages per Minute

Scenario Settings Load Behavior

Run Time

Run for 00:30:00 (HHH:MM:SS) after the target has been achieved

If target cannot be reached

☐ Stop scenario and save results

☒ Continue scenario without reaching goal

☒ Receive notification

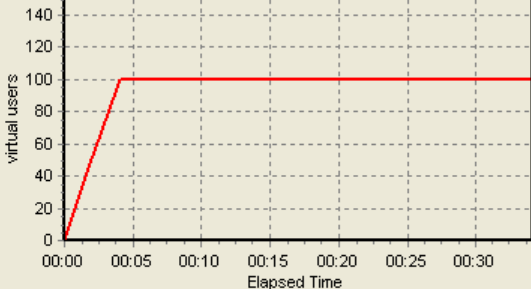
Do not change recorded think time

OK Cancel Help

Load Preview

virtual users

Elapsed Time



Elapsed Time	Virtual Users
00:00	0
00:05	100
00:10	100
00:15	100
00:20	100
00:25	100
00:30	100

■ 5 手动场景的场景设置

Create Scenario

Select Scenario Type

☐ Goal Oriented Scenario

☒ Manual Scenario

Number of Users: 5

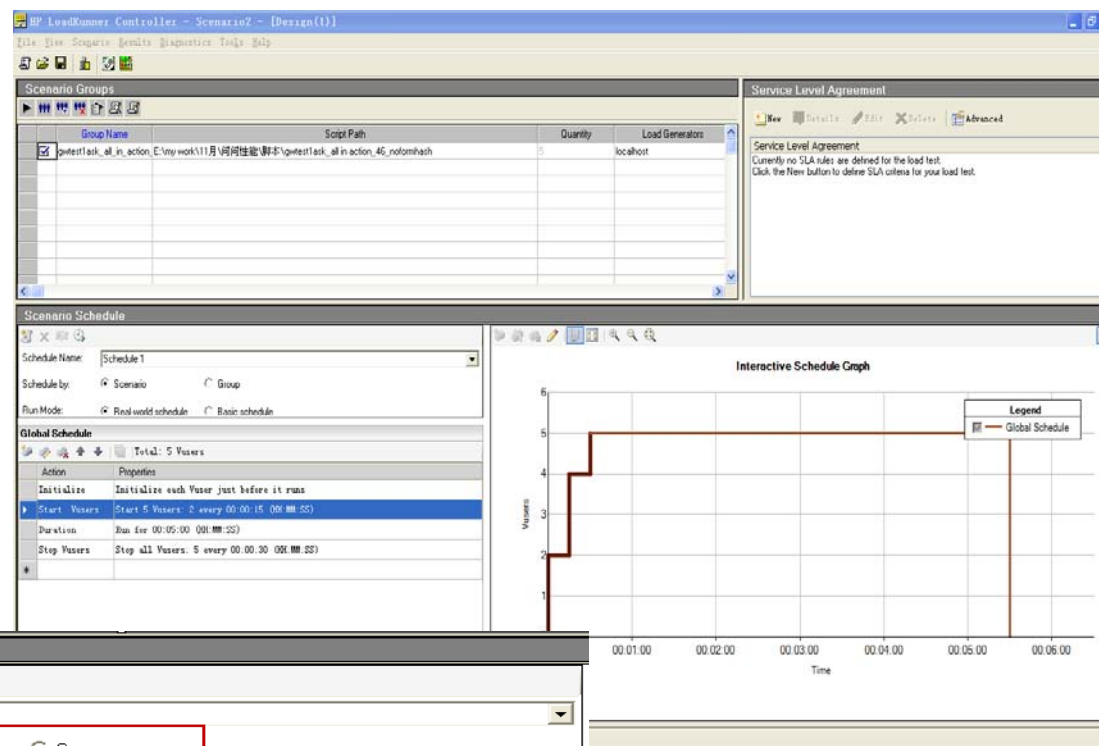
Load Generator: localhost

Group Name: gwtest1ask_all in action

Result Directory: E:\my work\11月\问问性能\脚本\gwtest1

☒ Add script to current scenario

OK



Scenario Schedule

Schedule Name: Schedule1

Schedule by: ☒ Scenario ☐ Group

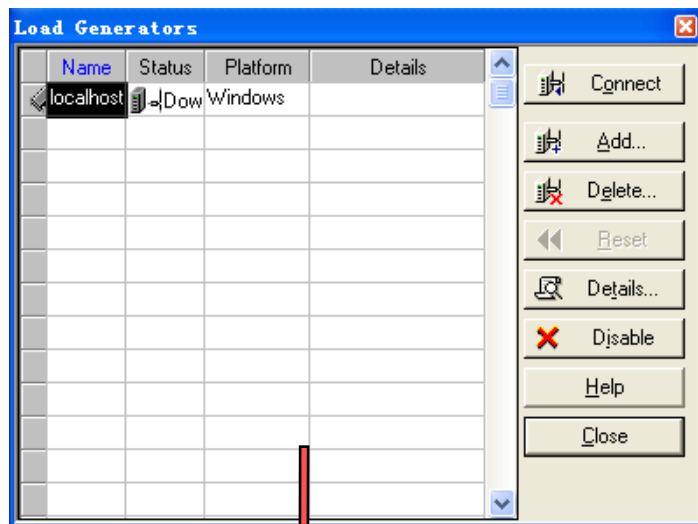
Run Mode: ☒ Real-world schedule ☐ Basic schedule

Global Schedule

Total: 5 Users

Action	Properties
Initialize	Initialize each User just before it runs
Start Users	Start 5 Users: 2 every 00:00:15 00:MM:SS
Duration	Run for 00:05:00 00:MM:SS
Stop Users	Stop all Users: 5 every 00:00:30 00:MM:SS

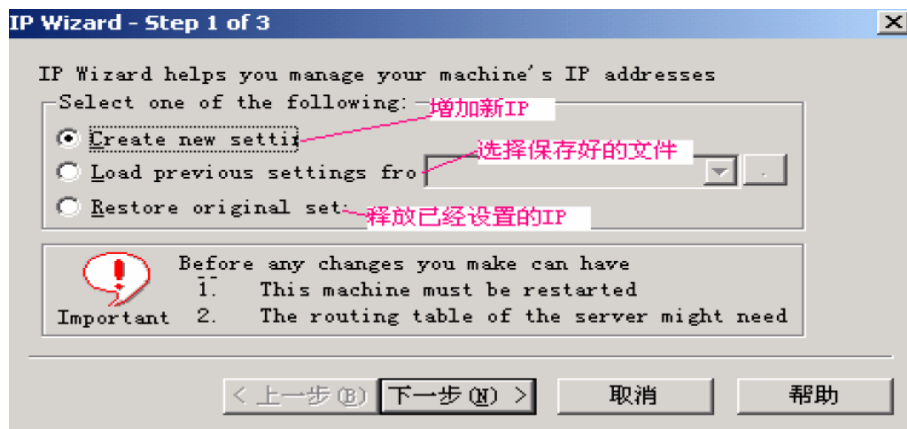
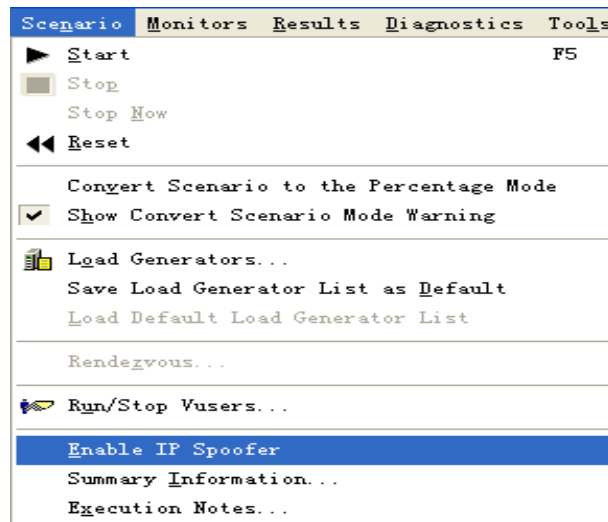
■ 6 负载生成器管理



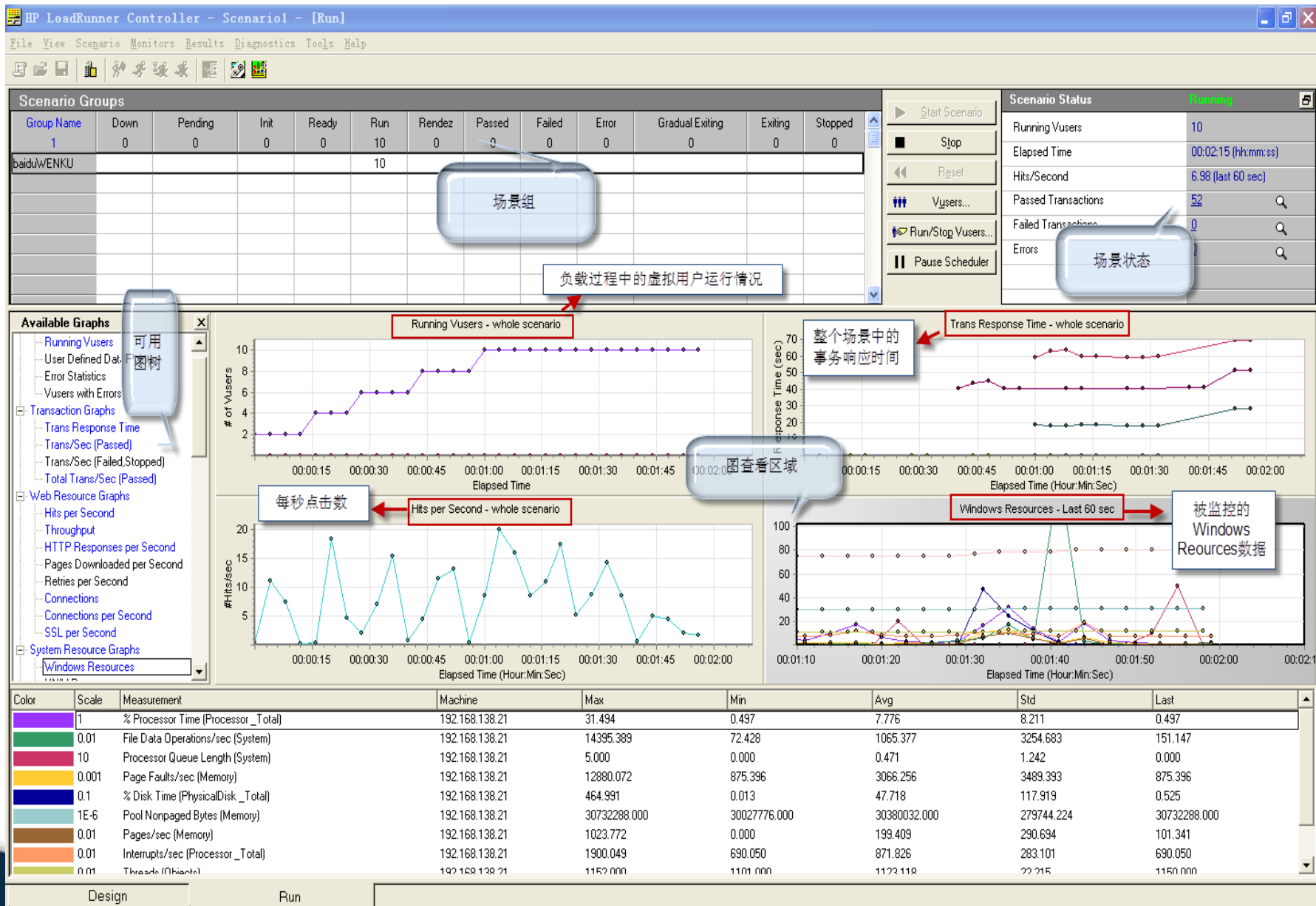
■ 7 IP虚拟（IP欺骗）

当运行场景时，虚拟用户使用它们所在的 Load Generator（负载生成器）的固定的 IP 地址。同时每个 Load Generator 上运行大量的虚拟用户，这样就造成了大量的用户使用同一 IP 同时访问一个网站的情况，这种情况和实际运行的情况不符，并且有一些网站会根据用户 IP 来分配资源，这些网站会限制同一个 IP 的登陆，使用等等。为了更加真实的模拟实际情况，LoadRunner允许运行的虚拟用户使用不同的 IP 访问统一网站，这种技术称为“IP 欺骗”。

注意：IP Spoofer 在连接 Load Generators 之前启用。要使用 IP 欺骗，各个 Load Generator 机器必须使用固定的 IP，不能使用动态 IP。即：运行 IP Wizard 程序的机器必须使用固定的 IP，不能使用动态 IP。



8 系统监控



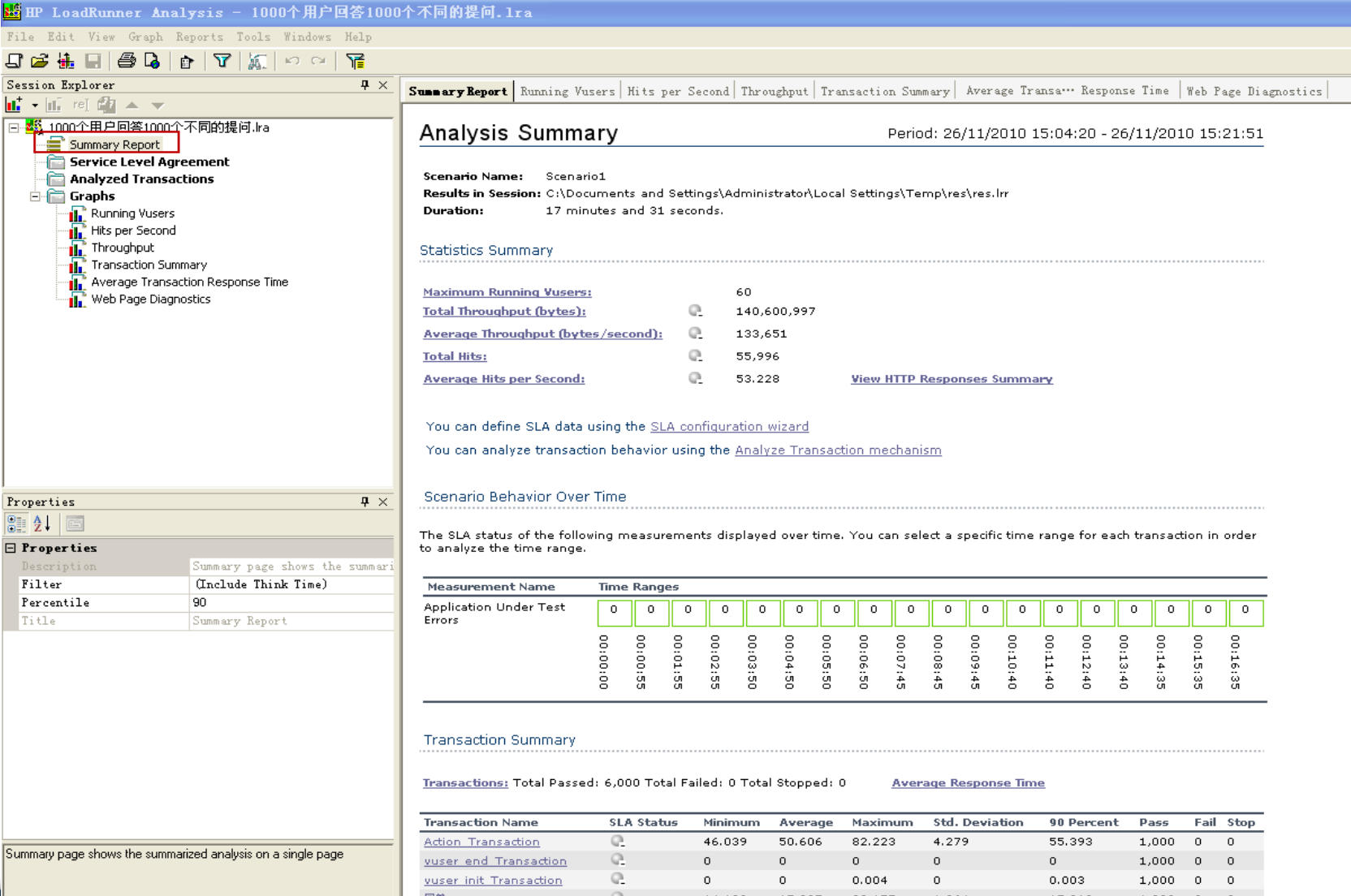
四、性能测试结果分析

从开始->LoadRunner-> Analysis来启动Analysis程序，或在Controller中Results菜单中选择Analysis Results，自动调出Analysis分析结果。Loadrunner提供多种性能参数的分析结果，如：

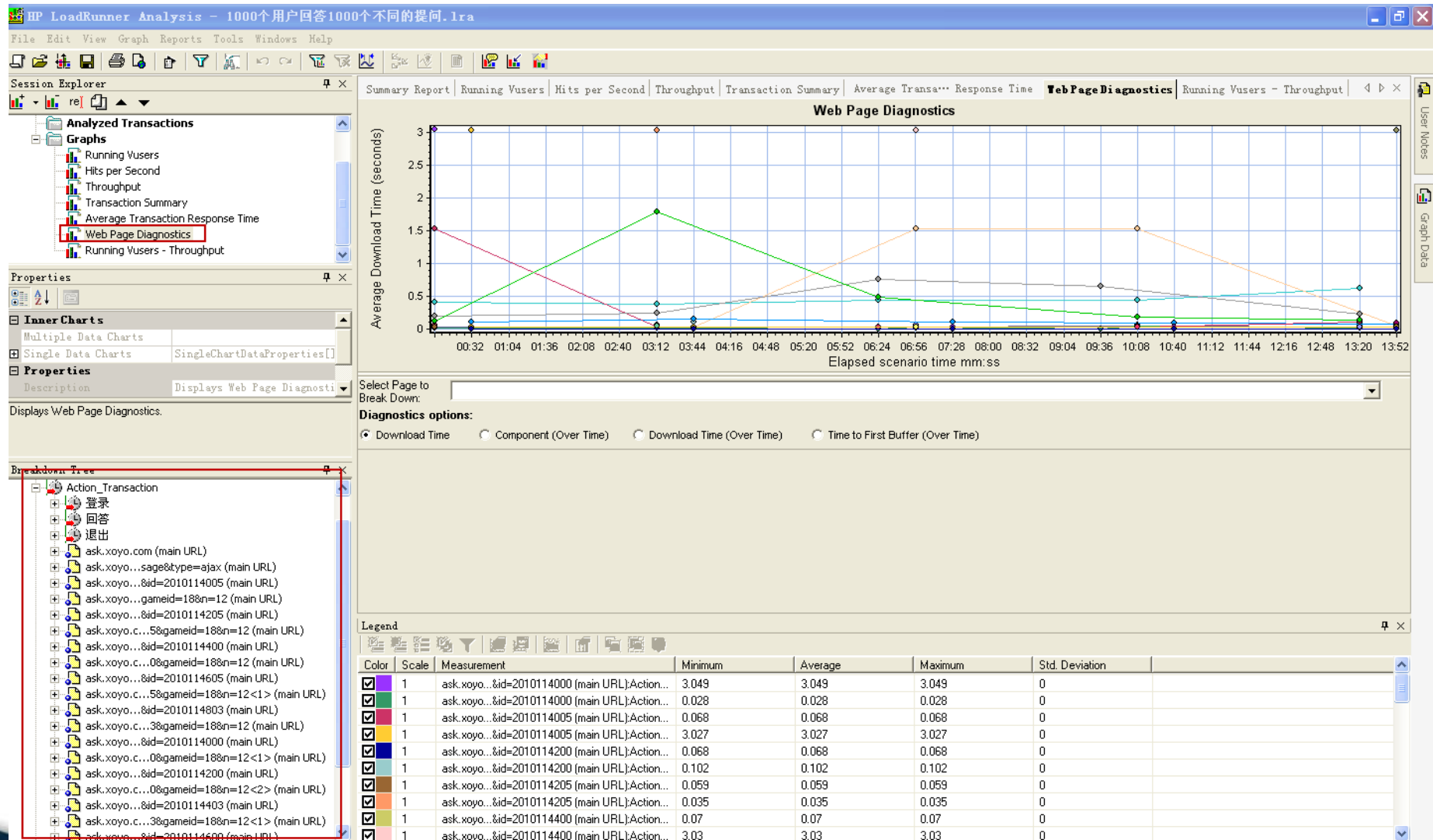
- Summary Report——分析概要报告
- Running Vusers——运行户数
- Hits per Second——每秒钟的点击次数
- Throughput——吞吐量
- Transaction Summary——事务分析
- Average Transaction Response Time——平均事务响应时间
- Web Page Breakdown——Web页面详细分析（需要添加）



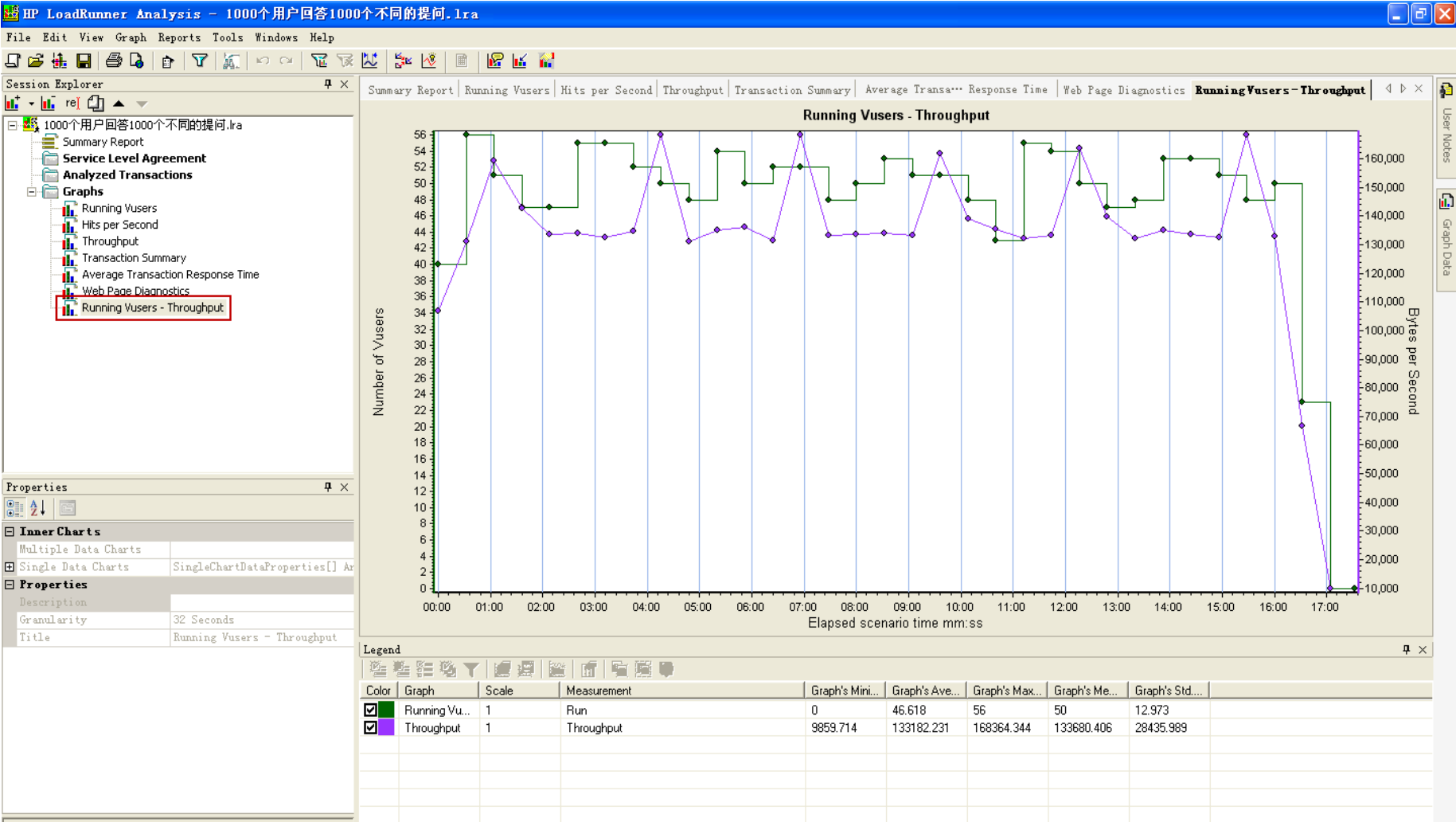
简要报告



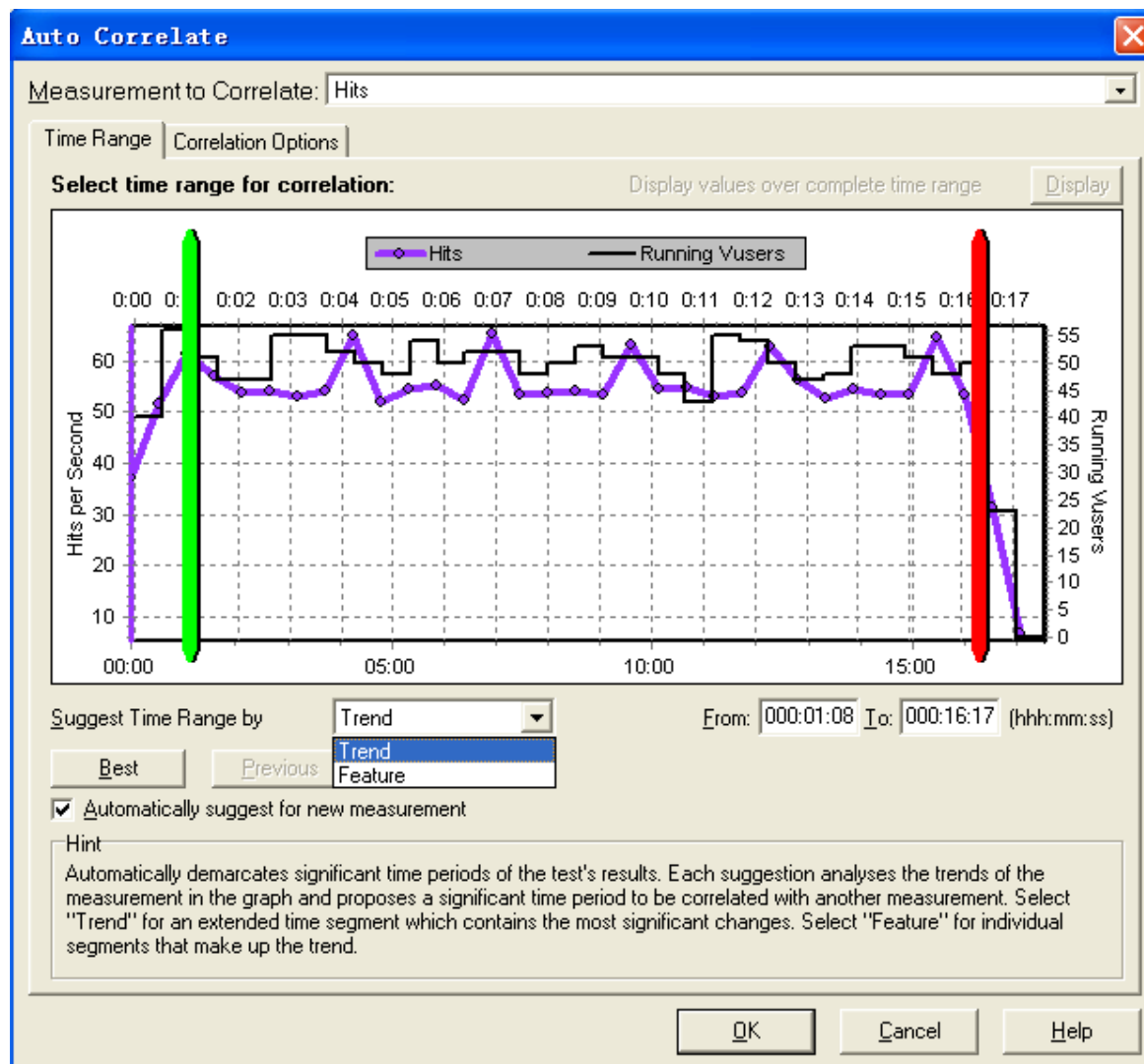
网页细分图



合并图

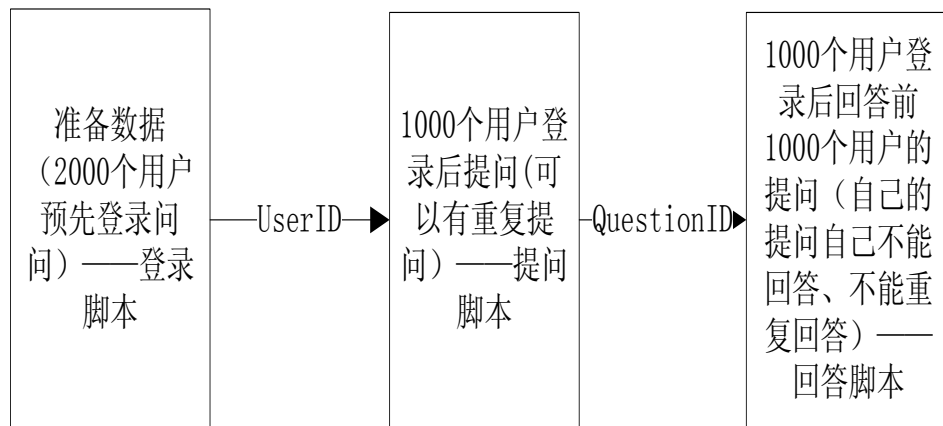


自动关联图



■ 3 实现问问项目的性能测试步骤

- (1) 研究实际项目、了解需求、准备数据
- (2) 设计场景
- (3) 录制脚本、完善脚本
- (4) 场景设置、生成负载、运行场景
- (5) 产生分析报告、分析系统瓶颈



■ 4 难点

- (1) 分析整个逍遥问问，获得数据流通思路与性能测试思路；
- (2) 在脚本中迭代和参数化 设置与在场景中的迭代和参数化设置是不同；
- (3) 如何设置合理的运行场景与过程，在场景运行过程中可以观察到每个用户进来后的具体动作。



■ 5 如何解决问题

(1) LoadRunner中如何实现参数化后每次取到唯一值。

只有当select next row = unique时，才能保证每个VU取不同的值；如果每个VU每次迭代取不同的值，采用unique+each iterationl设置方式；如果每个VU取不同值，每次迭代取相同的值，采用unique+once设置方式。

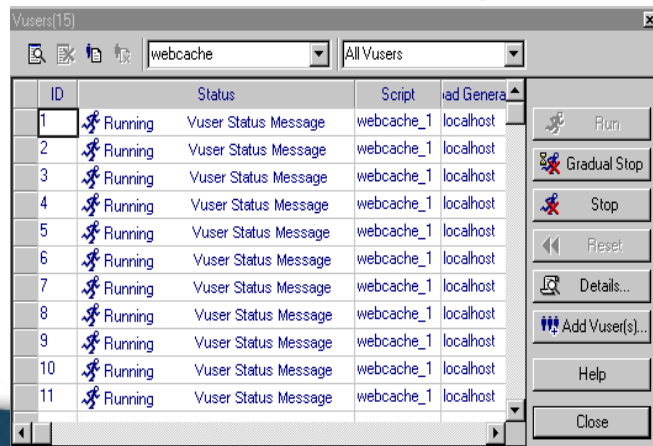
(2) LoadRunner中的多个参数之间的关系。

当UserName参数设置为Unique+each iteration时，UserID选择Same Line As UserName，多参数情况下都选择Same Line As UserName。

(3) 状态输出函数

函数格式：lr_vuser_status_message("Para is:%s,%dTimes Iteration",lr_eval_string("{username}"),++iteration);

查看地方：在Controller运行窗口,运行场景时在run界面中单击Vusers按钮（开始方案按钮的下面），在弹出窗口中可看到动态的username信息。



六、总结逍遥问问性能测试

■ 1、关于分析问题与解决问题

LoadRunner的学习与使用重要，更重要的是对实际项目的需求分析与实际情况中一些比较复杂关系的整理，需要明白数据的来龙去脉，需要清晰整理出解决问题的流程与方案。

在遇到问题的时候不能回避，通过查阅与咨询，不断地实践对比结果，每一个问题到最后都不是问题。

性能测试难点不在Loadrunner工具本身，难在技术攻关以及对系统的全局把握。



■ 2、关于脚本录制与完善

- (1) 根据实际情况，需要在录制脚本过程中将登录、操作、退出录制在同一个Action中；
- (2) 在录制过程中加入各种事务，在完善脚本的过程中，考虑参数化、关联等，删除不必要的脚本；
- (3) 登录、提问和回答三个脚本，明确每一个脚本的作用；
- (4) 脚本的成功与否在于不出错的前提下，是否能够按照预期的结果在数据库中插入合理的数据；
- (5) 每一次脚本的修改在对应场景中需要进行刷新；
- (6) 尽量在每一个脚本中加入状态输出代码，以便在场景运行中可以查看用户的状态变化。



■ 3、关于预期结果与实际场景设置

- (1) 预期结果与正确的场景设计离不开，所谓场景设计指的就是LoadRunner进行性能测试的测试用例，由具体的完善用例得到脚本，能够更准确的得到预期结果；
- (2) 在脚本中设置参数与迭代后，并不代表在场景中能够运行成功，需要了解场景中各个设置的具体意思与脚本中参数设置、Run Time Setting设置三者之间的关系。

■ 4、关于性能测试结果分析

- (1) 生成分析结果之后，对有关系的项要进行合并图，能更清晰地看到各个指标变化之间的关系；
- (2) 网页细分图需要在生成结果后另行设置后生成；
- (3) 合理使用Auto Correlate。

