

7

场景设计实践

前面已经讲述了场景设计的一些基础知识，鉴于场景设计在整个性能测试过程中的重要性，在实际使用过程中也就存在很多使用技巧或者说是一些高级使用方法。通过这些技巧或高级使用方法可以帮助更好地进行场景设计或监控场景执行的情况。本章主要对场景设计过程中常用的一些技巧进行讲解，主要涉及的内容有：

- 集合点设置
- IP 欺骗技术
- 负载均衡技术
- 执行路径转换
- 在 LoadRunner 中使用功能测试脚本

7.1 集合点设置

首先需要明确在性能测试过程中为什么要进行集合点设置，在 LoadRunner 测试过程中，其实并不能保证所有的 Vuser 真的在同一时刻进行操作，这样就达不到并发测试的目的，故需要使用到集合点技术，集合点的意思是如果在一个操作之前设置了一个集合点，LoadRunner 会等待所有的 Vuser 都准备好要执行该功能时才开始执行，其强调的是所有的 Vuser 都已准备好了，如果只是部分 Vuser 准备好了，该功能还是不会被执行。

选择 Scenario→Rendezvous，如图 7-1 所示。



注意

在场景设置集合点之前，脚本一定要插入集合点，否则会发现 Scenario 菜单中 Rendezvous 项是不可选的。

接下来会弹出集合点信息对话框，如图 7-2 所示。

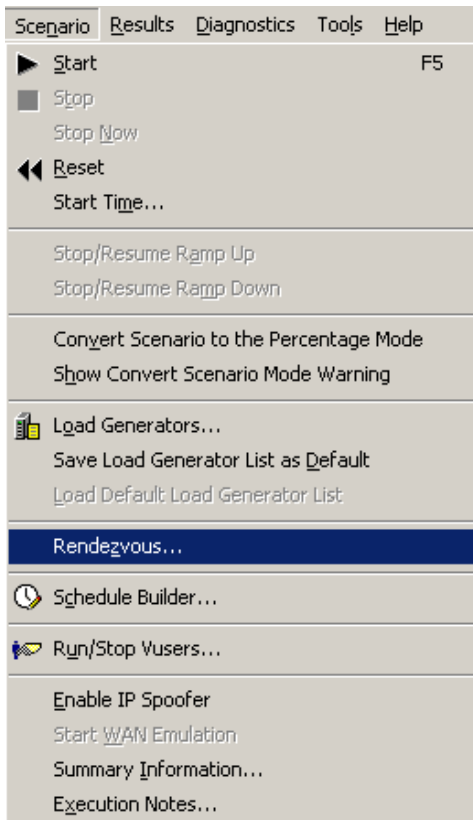





图 7-1 设置集合点操作

- **Rendezvous:** 这里显示出脚本中包含的所有集合点。默认情况下这些集合点处于启用状态。可以点击  **Disable Rendezvous** 按钮，将集合点设置为禁用状态。
- **Scripts:** 显示了场景运行的所有脚本。
- **Vusers:** 场景运行设置虚拟用户情况。默认情况下所有的 **Vuser** 都会参与到集合点的策略中来，可以手动点击  **Disable VUser** 按钮，将一些 **Vuser** 设置为不参与到该集合点策略中来。

点击  **Policy...** 按钮，系统弹出 **Policy** 对话框，如图 7-3 所示。在该对话框中可以设定集合点执行的策略。

第一项：表示当所有用户数的 **X%** 到达集合点时，开始释放等待的用户并继续执行场景。

第二项：表示当前正在运行用户数的 **X%** 到达集合点时，开始释放等待的用户并继续执行场景。

第三项：表示当 **X** 个用户到达集合点时，开始释放等待的用户并继续执行场景。

其中还有一项 **Timeout between Vusers**，就 30 秒来说，当第一个用户到达集合点后，再等待 30 秒，如果在 30 秒内到达的用户数达到指定的数量，就开始继续执行场景。如果在 30 秒内还没有达到指定的用户数量，就不再等待，开始释放等待的用户并继续执行场景。

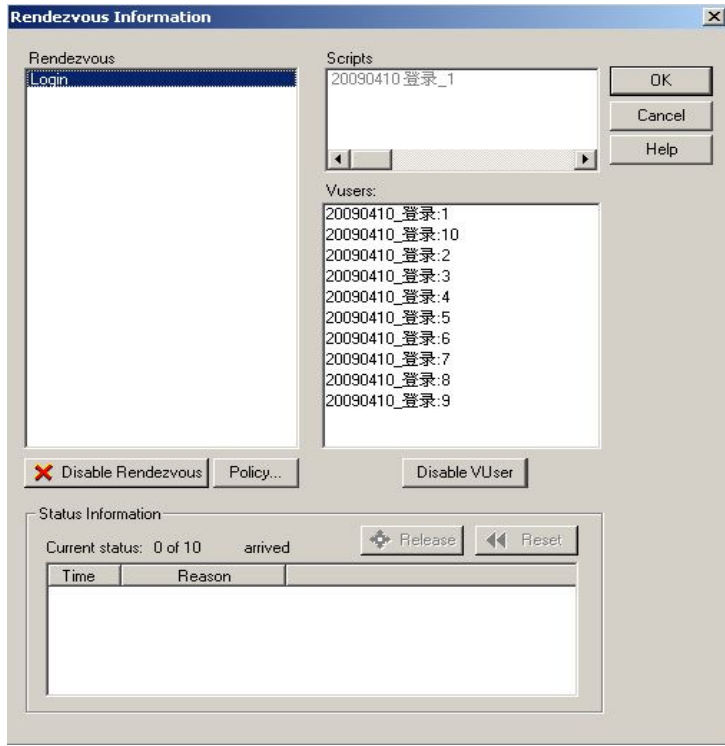


图 7-2 集合点设置

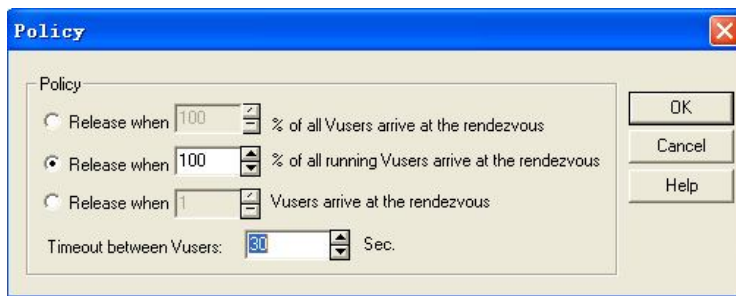





图 7-3 集合点策略设置

以上都是自动控制 Vuser 释放的情况，但在实际使用的过程中，也可以手动对 Vuser 进行释放，这涉及到手动释放 Vuser 技术。在正常情况下会发现在集合点信息对话框中，下面的 Release 按钮不可用，如图 7-2 所示。以下是手动释放 Vuser 的步骤：

- 1) 开始执行场景。
- 2) 在场景执行过程中，选择 Scenario→Rendezvous，打开集合点信息对话框。
- 3) 在场景运行过程中，下面的  Release 按钮和  Reset 按钮会变成可用状态，这时可以点击  Release 按钮，进行手动释放 Vuser，如图 7-4 所示。

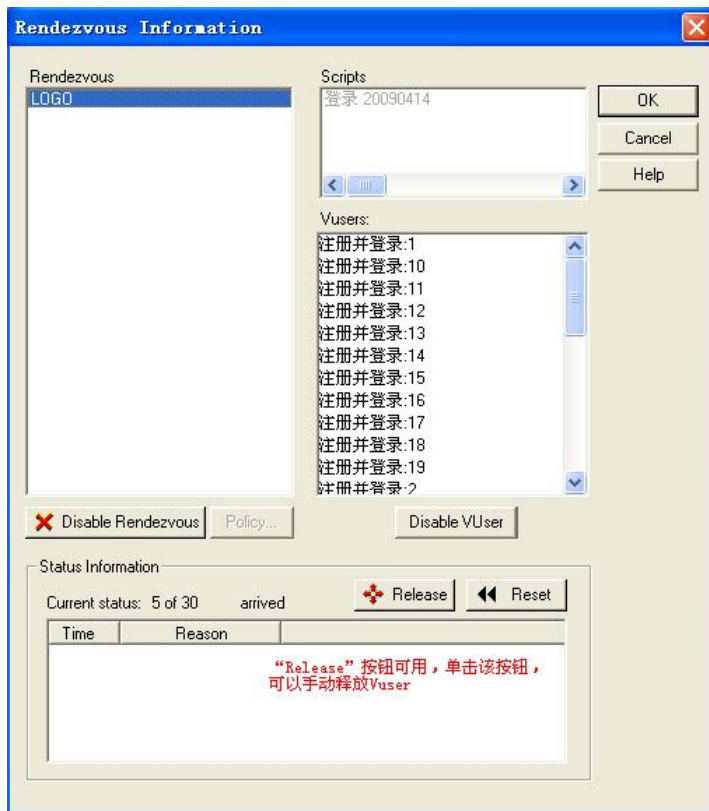


图 7-4 手动释放 Vuser

7.2 IP 欺骗技术

在场景运行时，每台负载发生器计算机上的 Vuser 都使用其计算机的固定 IP 地址。这样就不能模拟用户使用不同计算机的真实情况。

应用程序服务器和网络设备用 IP 地址标识客户端。应用程序服务器经常缓存来自同一台计算机的客户端信息。网络路由器则尝试缓存源信息和目标信息来优化吞吐量。如果许多用户使用同一个 IP 地址，服务器和路由器都将尝试进行优化处理。由于同一台负载发生器计算机上的 Vuser 具有相同的 IP 地址，因此服务器和路由器优化不反映真实情况。

使用 LoadRunner 的多 IP 地址功能，可以用许多 IP 地址来标识在一台计算机上运行的多个 Vuser。这样，服务器和路由器将认为 Vuser 来自不同的计算机，因此测试环境更加真实。

当然 LoadRunner 在设计的时候就想到这个问题了，这就是 IP 欺骗技术。但是在使用 IP 欺骗时需要注意 IP Spoofer 要在连接 Load Generator 之前启用，并且各负载发生器计算机必须使用固定 IP，不能使用动态 IP（即 DHCP）。

7.2.1 IP Spoofer 设置

1) 点击“开始”菜单→“所有程序”→Mercury LoadRunner→Tools→IP Wizard, 弹出 IP Wizard 配置对话框, 如图 7-5 所示。

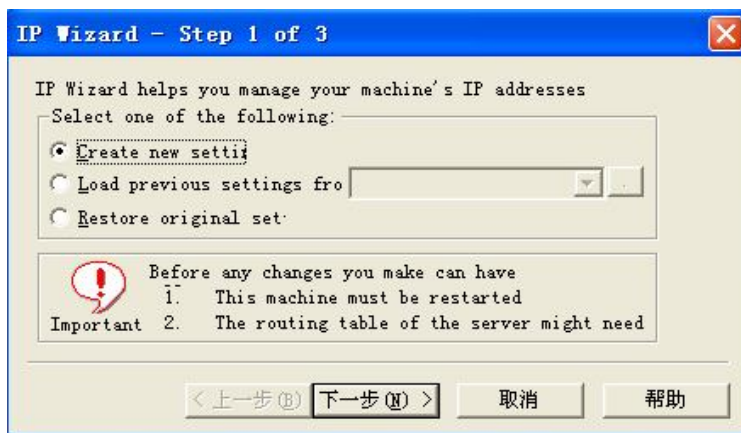


图 7-5 IP Wizard 设置首页

2) 在这里选择一种方式进行设置, 如果以前保存过这类文件, 可以选择“Load Previous settings from (从文件中加载以前的设置)”, 然后选择该文件即可。

3) 也可以选择“Create new settings (创建一个新的设置)”。

4) 点击“下一步”按钮。如果计算机中安装了多个网卡, 那么要选择用于这些 IP 地址的网卡, 然后点击“下一步”按钮, 在弹出的对话框中设置服务器的 IP 地址, 如图 7-6 所示。



图 7-6 设置服务器 IP 地址

5) 点击“下一步”按钮将看到该计算机的 IP 地址列表。点击“添加”按钮可以定义地址范围, 如图 7-7 所示。

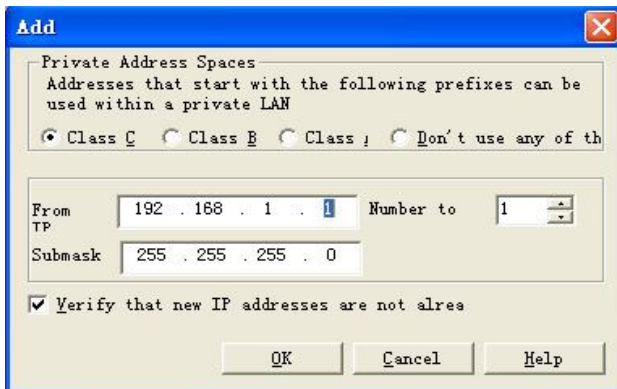


图 7-7 设置虚拟 IP 和子网掩码

6) 在该对话框中选择计算机的 IP 地址类型。指定要创建的 IP 地址数。选中“Verify that new IP addresses are not already (验证新的 IP 地址未被使用)”复选框，以指示 IP 向导对新地址进行检查，LoadRunner 会自动检查新添加的 IP 在同一网段中是否已被使用，如果 IP 已经被使用，那么这些 IP 将不会被添加进来，只有未被使用的 IP 地址才会被添加进来，点击“确定”按钮继续。

7) 完成之后，IP 欺骗向导会显示出 IP 变更统计的对话框，如图 7-8 所示。

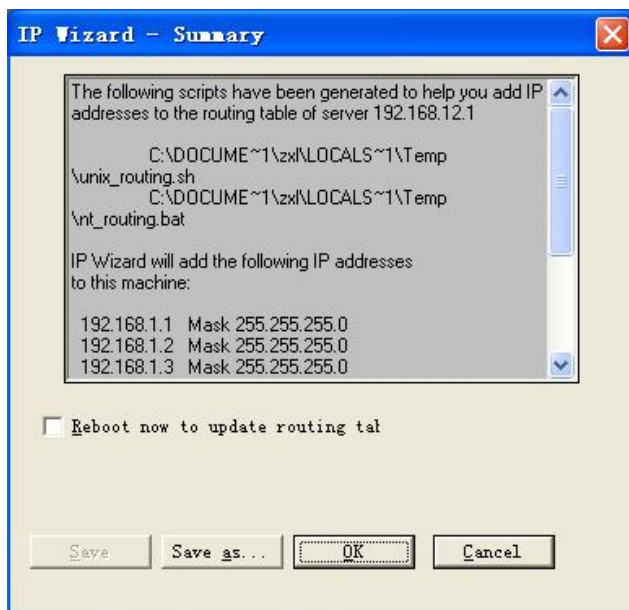


图 7-8 IP 变更统计

在该对话框中，可以点击 **Save as...** 按钮将本次添加的 IP 保存成“.ips”文件，下次再使用时就可以直接加载此文件。

完成之后，系统会提示重启计算机，重启计算机后，设置的 IP 欺骗有效，重启后为了确认虚

拟 IP 是否都生效，可以在“开始”→“运行”，输入“cmd”，在弹出的命令窗口输入“ipconfig/all”，就能看到生效的所有 IP。

还可以使用 `lr_get_vuser_ip` 函数来得到当前虚拟用户的 IP 地址，在脚本中加入下面的代码即可。

```
char *ip;
ip = lr_get_vuser_ip();
if(ip)
    lr_output_message("The IP address is %s",ip);
else
    lr_output_message("IP spoofing disabled");
```

场景运行时，如果虚拟 IP 生效的话，打开每个 Vuser 的日志，就可以看见他们各自的 IP。

而在实际使用过程中，经常发现，客户端和服务端之间存在一个路由器，出现这种情况怎么办呢？其实在使用 IP 欺骗技术时，服务器需要将这些地址添加到路由表中，以识别返回客户端的路由。如果服务器和客户端具有相同的子网掩码、IP 类和网络，那么不需要修改服务器的路由表。如果客户端和服务端之间有路由器，那么必须手动让服务器将这些地址添加到路由表中，这样服务器才能识别经过该路由器的路径。

在设置完成 IP 欺骗的最后一步，不要重启计算机，此时在临时文件夹 Temp 里生成两个文件，一个是用于 UNIX 操作系统的 `unix_routing.sh` 文件，一个是用于 Windows 操作系统的 `nt_routing.bat` 文件。两个文件的内容如下：

`nt_routing.bat` 文件：

```
REM This is a bat file to add IP addresses to the routing table of a server
REM Replace [CLIENT_IP] with the IP of this machine (Load Generator) that the server already recognizes
REM This script should be executed on the server machine
route ADD 192.168.14.22 MASK 255.255.255.255 [CLIENT_IP] METRIC 1
route ADD 192.168.14.23 MASK 255.255.255.255 [CLIENT_IP] METRIC 1
route ADD 192.168.14.25 MASK 255.255.255.255 [CLIENT_IP] METRIC 1
```

`unix_routing.sh` 文件：

```
# Bourne shell script to add IP addresses to the routing table of a server
# To run replace [CLIENT_IP] with the IP of this machine (Load Generator) that the server already recognizes
# Then chmod +x unix_routing.sh
# and finally execute this script on the server
#!/bin/sh
route add 192.168.14.22 [CLIENT_IP] 255.255.255.0
route add 192.168.14.23 [CLIENT_IP] 255.255.255.0
route add 192.168.14.25 [CLIENT_IP] 255.255.255.0
```

要更新服务器路由表，必须对这两个文件进行一定的修改。

1) 将 `[CLIENT_IP]` 用 LoadRunner 机器的 IP 地址替换。假设 LoadRunner 机器的 IP 地址为 192.168.14.26。

2) 修改的文件必须在服务器上运行该文件，Windows 操作系统使用 `nt_routing.bat` 文件，UNIX 操作系统使用 `unix_routing.sh` 文件。

修改后的两个文件内容如下：

nt_routing.bat 文件：

```
REM This is a bat file to add IP addresses to the routing table of a server
REM Replace [CLIENT_IP] with the IP of this machine (Load Generator) that the server already recognizes
REM This script should be executed on the server machine
route ADD 192.168.14.22 MASK 255.255.255.255 192.168.14.25 METRIC 1
route ADD 192.168.14.23 MASK 255.255.255.255 192.168.14.25 METRIC 1
route ADD 192.168.14.25 MASK 255.255.255.255 192.168.14.25 METRIC 1
```

unix_routing.sh 文件：

```
# Bourne shell script to add IP addresses to the routing table of a server
# To run replace [CLIENT_IP] with the IP of this machine (Load Generator) that the server already recognizes
# Then chmod +x unix_routing.sh
# and finally execute this script on the server
#!/bin/sh
route add 192.168.14.22 192.168.14.25 255.255.255.0
route add 192.168.14.23 192.168.14.25 255.255.255.0
route add 192.168.14.25 192.168.14.25 255.255.255.0
```

到这里跨网段 IP 欺骗技术已经完成，重启计算机，IP 欺骗将生效。

7.2.2 Controller 中启动 IP Spoofer

设置好 IP Spoofer 后，必须在 Controller 中启动 IP Spoofer，选择 Scenario→Enable IP Spoofer，启动 IP Spoofer 策略，如图 7-9 所示。

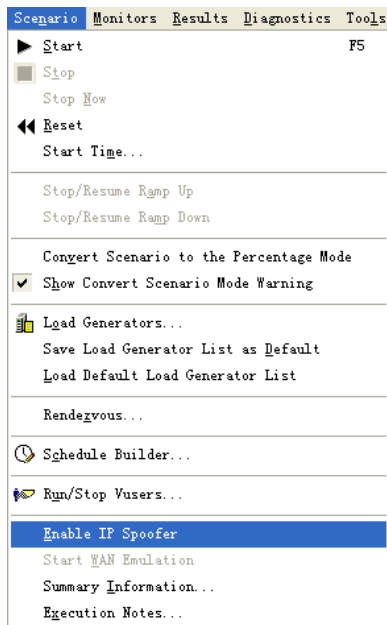


图 7-9 启动 IP 欺骗策略

当启动 IP Spoofer 后, 在 Run 选项卡的右下角会看到 IP Spoofer 的标记, 如图 7-10 所示。

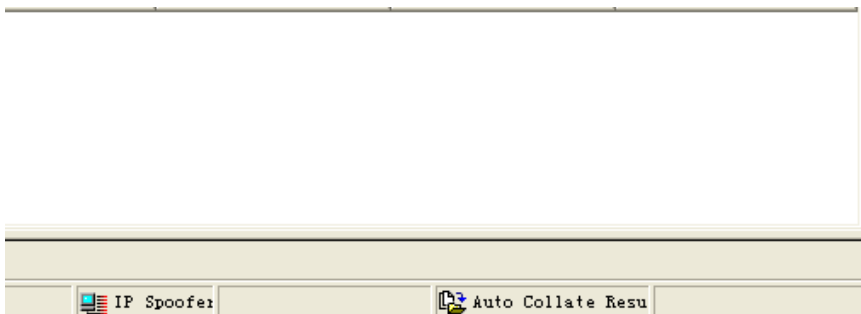


图 7-10 IP Spoofer 启动标记

启动多 IP 地址模式后, 可以在专家模式中对多 IP 地址进行全局设置。首先, 选择 Tools→Expert Mode 来启动专家模式。接下来选择 Tools→Options→General 选项卡, 来设置多 IP 地址模式分配 IP 地址的情况, 如图 7-11 所示。

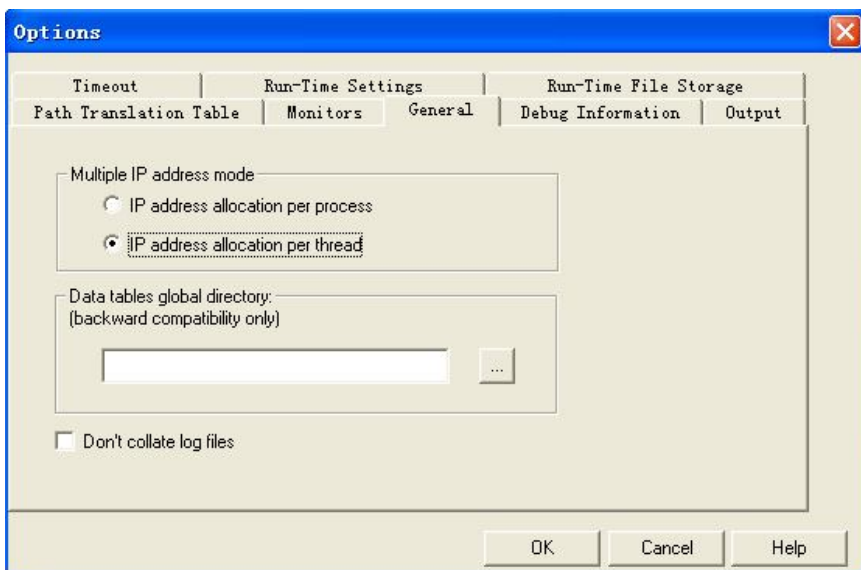


图 7-11 多 IP 地址模式下 IP 地址分配设置

可以选择按进程或线程分配 IP 地址, 按线程分配会导致场景中存在更多的 IP 地址范围。

选择 Scenario→Load Generators, 设置 Vuser 生成器, 将虚拟 IP 地址添加进去, 并连通 (点击 Connect 按钮, 每个 IP 状态由 Down 变为 Ready), 如图 7-12 所示。

当连接成功后, Vuser 生成器会在工具栏中显示, 如图 7-13 所示。

在使用完成之后, 一定要记得释放 IP 地址, 在 IP Wizard 设置首页, 如图 7-5 所示。选择 Restore original set 项, 进行 IP 地址释放, 重启计算机即可。

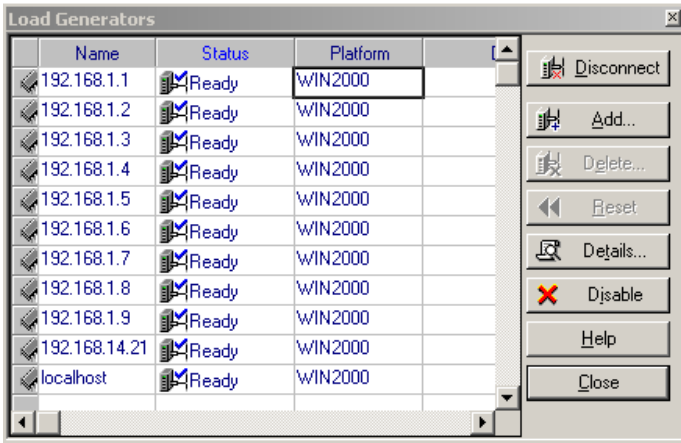


图 7-12 Vuser 生成器设置



图 7-13 连接成功标识

7.3 负载均衡技术

在测试过程中，测试机也就是负载发生器也可能成为瓶颈。那么为什么测试机也可能成为瓶颈呢？原因是由于负载不均匀造成的。例如，在测试过程中使用 4 台测试机作为负载发生器，虚拟用户为 1000 个，这时如果负载分配不均匀，可能出现这种现象，就是 1000 个虚拟用户从 4 台中的 2 台测试机中产生，这样就导致有 2 台机器特别忙，而另外 2 台机器就特别闲，这样这 2 台特别忙的机器，其本身就可能成为瓶颈了。

为了尽可能减少或者避免测试机成为测试过程中的瓶颈，在测试过程中，需要使用所有的测试机产生 Vuser，对被测试系统进行施压。

在默认模式下，会发现在 Controller 中添加多台 Load Generators 机器时，不管如何添加，最终只能选中一台机器，如图 7-14 所示。

Script Path	Quantity	Load Generators
090410 注册	10	192.168.14.14
090410 登录	10	<Add...>
		192.168.14.20
		192.168.14.14
		192.168.14.18
		192.168.14.19
		localhost

图 7-14 Load Generators 机器设置

这样的负载分配是不均匀的,为了解决这个问题,首先要更换场景模式,选择 Scenario→Convert Scenario to the Percentage Mode, 将场景模式由组模式更换为百分比模式,如图 7-15 所示。在弹出的对话框中点击“是”按钮即可。

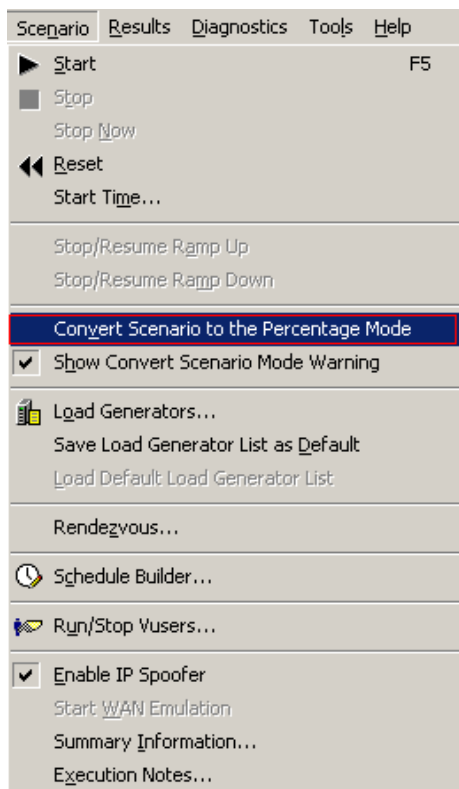


图 7-15 将组模式设置更改为百分比模式

这时,可以在已经添加好的 Load Generators 机器列表中选择需要的机器,如图 7-16 所示。

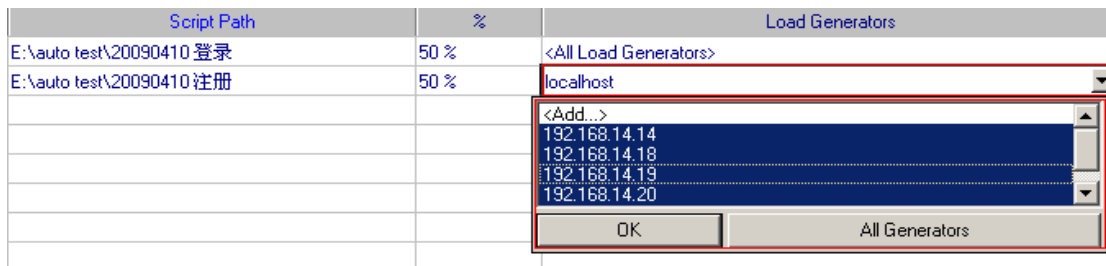


图 7-16 选择需要的负载机

选择需要的负载机,也可以选择所有的负载机,点击 OK 按钮,所选择的机器都将被添加进来,如图 7-17 所示。

Script Path	%	Load Generators
E:\auto test\20090410 登录	50 %	<All Load Generators>
E:\auto test\20090410 注册	50 %	192.168.14.20, 192.168.14.14, 192.168.14.18, 192.168.14.19, localhost

图 7-17 为 Load Generators 设置多台机器

如果需要将百分比模式切换回组模式，那么可以选择 Scenario→Convert Scenario to the Vuser Group Mode，将场景模式切换为组模式，如图 7-18 所示。

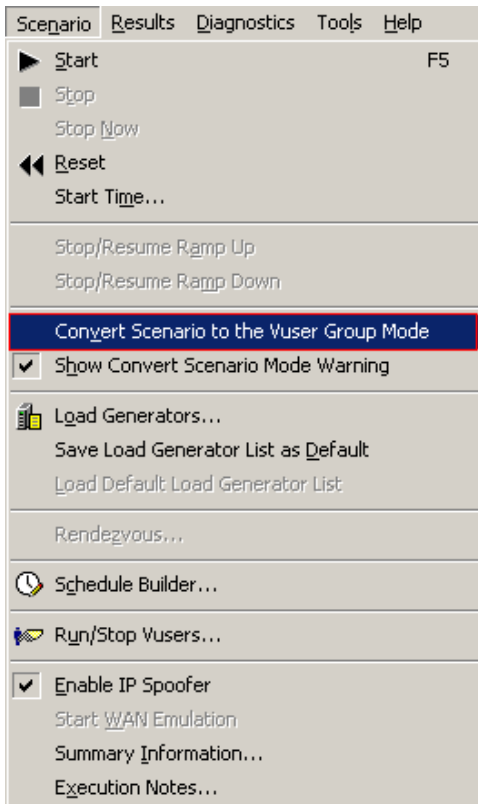


图 7-18 将百分比模式设置为组模式

这样就可以保证负载机均匀地对服务器进行施压。

7.4 执行路径转换

7.4.1 路径转换介绍

在运行场景时，LoadRunner 会从正在参与的 Vuser 中收集运行时的数据。默认情况下，

LoadRunner 将数据存储在每个 Vuser 计算机上的临时文件中。当场景运行完成后，会在常规结果目录中整理数据。但也可以设置为直接将运行时数据写入共享网络驱动器中。建议不要使用此方法，因为这样会增加网络流量且需要进行路径转换设置。

路径转换是 LoadRunner 在转换 Controller 的远程路径名时所使用的一种机制。一个远程负载发生器将网络驱动器映射为 F，而另一个负载发生器将同一个驱动器映射为 H。在如此复杂的场景中，需要确保所有参与的计算机能够识别同一个网络驱动器。

在 Tools→Options→Run-Time File Storage 选项卡中，可以设置将脚本和运行时数据结果存储在共享网络驱动器上，如图 7-19 所示。

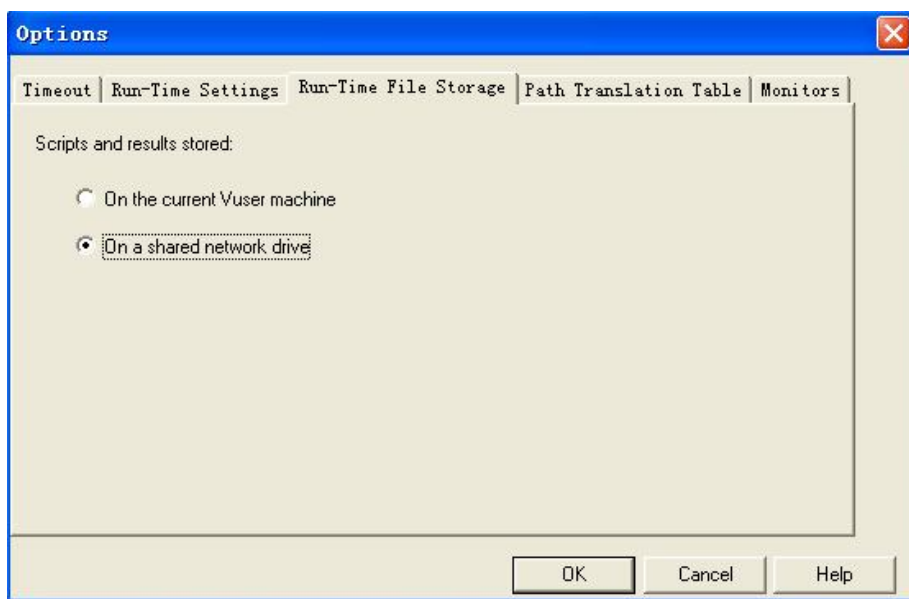


图 7-19 设置运行时文件存储位置

如果 Controller 和远程负载发生器对存储运行时结果文件的共享驱动器采用不同的映射，必须执行路径转换。在 Windows 和 UNIX 之间进行跨平台路径转换也是有效的。可以通过路径转换将基于 Windows 的路径（可以被 Controller 识别）转换成能被 UNIX Vuser 负载发生器识别的路径。

7.4.2 编辑路径转换表

LoadRunner 将路径转换表保存在一个 ASCII 文件（ppath.mnt）中。该文件存储在 LoadRunner 根目录/dat 中，对于每个要转换的网络路径都包含一行条目，如图 7-20 所示。

路径转换表中每一行条目的格式如下：

```
<controller_host> <controller_path> <remote_path> [<remote_host>]
```

controller_host：表示正在运行 Controller 的计算机名称或类型。例如，如果 Controller 运行在

一台 UNIX 计算机上，可以在“controller_host”字段中键入 unix。也可以输入 Controller 计算机的名称（如 PC1）。

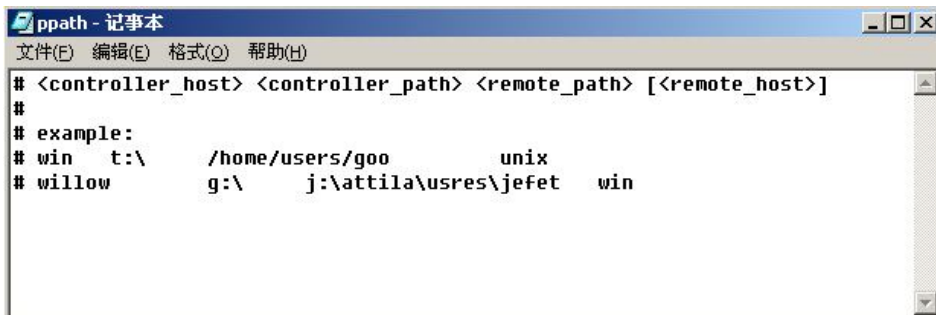


图 7-20 ppath 文件内容

controller_host 的值可以是：

- Hostname: Controller 运行的计算机名。
- Win: Controller 运行在 Windows 计算机上。
- Unix: Controller 运行在 UNIX 计算机上。
- All: Controller 运行在 Windows 或 UNIX 计算机上。

controller_path: 表示 Controller 能够识别的特定目录所在的路径。例如，如果脚本所在的目录位于 Controller 映射的网络驱动器 R 中，则在“controller_path”字段键入路径 R:\scripts。

remote_path: 远程计算机能够识别的特定目录所在的路径。例如，如果脚本所在的目录位于远程负载发生器映射的网络驱动器 N 中，则在“remote-path”字段键入路径 N:\scripts。如果远程 UNIX 负载发生器上的 Vuser 将上述路径识别为/m/tests，则应在“remote-path”字段中键入该路径。

remote_host: 表示负载发生器的名称或类型。其他可设置的值和 controller_host 一致，该参数是可选参数。

下面看两个例子：

```
arivn F:\ G:\loadtest\ huang
```

表示 Vuser 运行在 Windows 计算机 huang 上。Arivn 将网络驱动器映射为 F:，而 huang 将其映射为 G:\loadtest。

```
arivn F:\ /u/tests/load Unitr
```

表示 Vuser 运行在 UNIX 计算机 Unitr 上。Arivn 将网络驱动器映射为 F:，而 Unitr 将其映射为 /u/tests/load。

接下来就是编辑路径转换表了，具体步骤如下：

1) 在 Controller 控制器中，选择 Tools→Options，选择 Path Translation Table 选项卡，打开路径转换表视图，如图 7-21 所示。

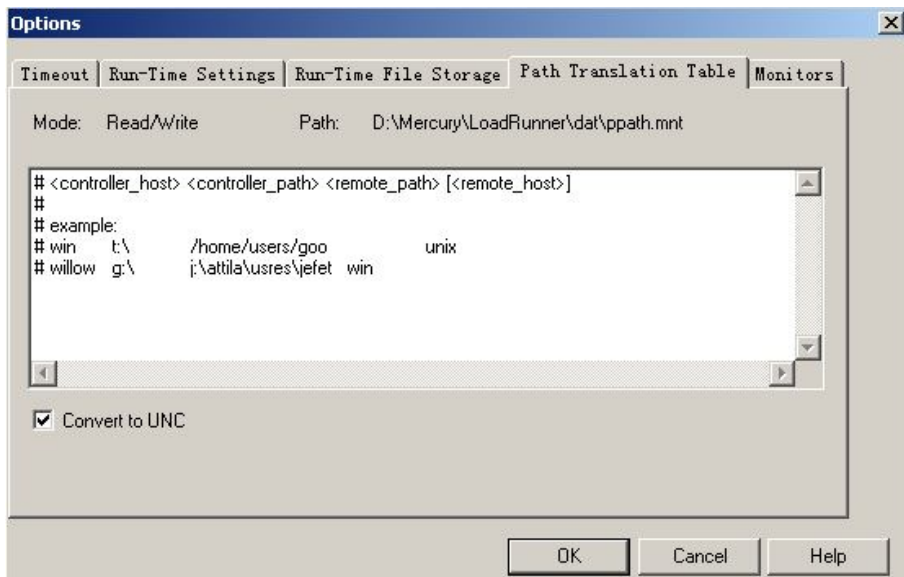


图 7-21 路径转换表

2) 输入路径转换信息之前, 请首先考虑使用通用命名约定方法。如果您的计算机是 Windows 计算机, 可以指示 Controller 将所有路径转换为 UNC。这样, 所有计算机都可以识别路径而无需进行路径转换。以下为 UNC 格式路径的一个示例: \\machine_a\results。选中 Convert to UNC 复选框, 表示 LoadRunner 将忽略路径转换表并将所有路径都转换为通用命名约定格式。

3) 如果您的计算机不是 Windows 计算机并需要路径转换, 则在该表中键入路径信息。可以通过在该表中行首处键入#符号来插入注释。点击 OK 按钮保存信息即可。

7.5 在 LoadRunner 中使用功能测试脚本

LoadRunner 能够将 GUI Vuser 脚本形式的功能测试脚本集成到负载测试场景中。这样就可以使用 LoadRunner 来测试和监控负载对应用程序功能的影响。LoadRunner 可以将 GUI Vuser 脚本形式的这些功能测试脚本集成到负载测试场景中, 在 QuickTest 或 WinRunner 中设计并调试过的这些脚本可用作负载测试的基础。

在 LoadRunner 中运行功能测试脚本主要有以下几个优点:

- 可检查高负载对应用程序的功能带来的影响。
- 可以度量应用程序在负载下运行时, 典型用户在客户端等待的响应时间(端到端的响应时间)。

通过 GUI Vuser 可以度量并监控客户端/服务器系统在负载下端到端的用户响应时间。端到端的响应时间表示用户在提交请求后等待响应的总时间。端到端的响应时间包括 GUI 响应时间、网络和服务器响应时间, 如图 7-22 所示。

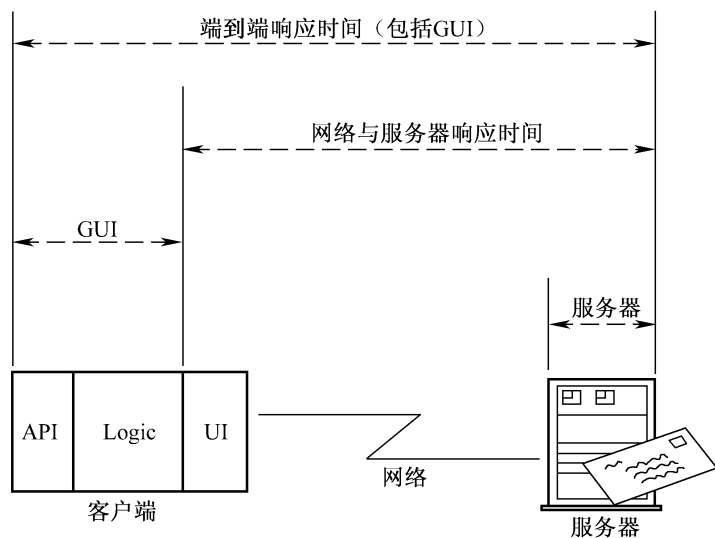


图 7-22 GUI 响应时间

7.5.1 QuickTest 创建 GUI Vuser 脚本

在 QuickTest 中创建用于 LoadRunner 测试场景中的 GUI Vuser 脚本时，需要遵循特定的准则以确保能够平衡集成脚本。QuickTest 具有多项与 LoadRunner 集成设计的功能。但部分 QuickTest 功能可能在与 LoadRunner 集成时不可用。QuickTest 与 LoadRunner 集成要注意以下几个方面。

1. 事务

在一些时候为了知道服务器对某一个业务的处理能力，需要定义事务。在 Vuser 脚本中可以通过嵌入适当的开始和结束事务语句脚本段来定义事务。但要注意的是 LoadRunner 仅提供事务中数据性能信息。因此，QuickTest 测试必须包括 LoadRunner 要使用的事务。通常在 QuickTest 中也使用插入开始事务和插入结束事务来插入事务。这样可以确保事务被正确地集成到 LoadRunner 中。

2. 将集成数据与测试一起保存

QuickTest 为了能与 LoadRunner 虚拟用户技术集成，必须生成特殊的集成文件。在默认情况下，生成这些数据的选项都是处于开启状态，但有时为了保留磁盘空间而禁用该选项。这样在录制时就要确保该选项处于开启状态。

在 QuickTest 中，选择 Tools→Options 对话框，打开 Run 选项卡，如图 7-23 所示。在这里一定要保证 Allow other Mercury products to run tests and components 选项处于选中的状态，如果没有选中，则将其选中并保存设置。

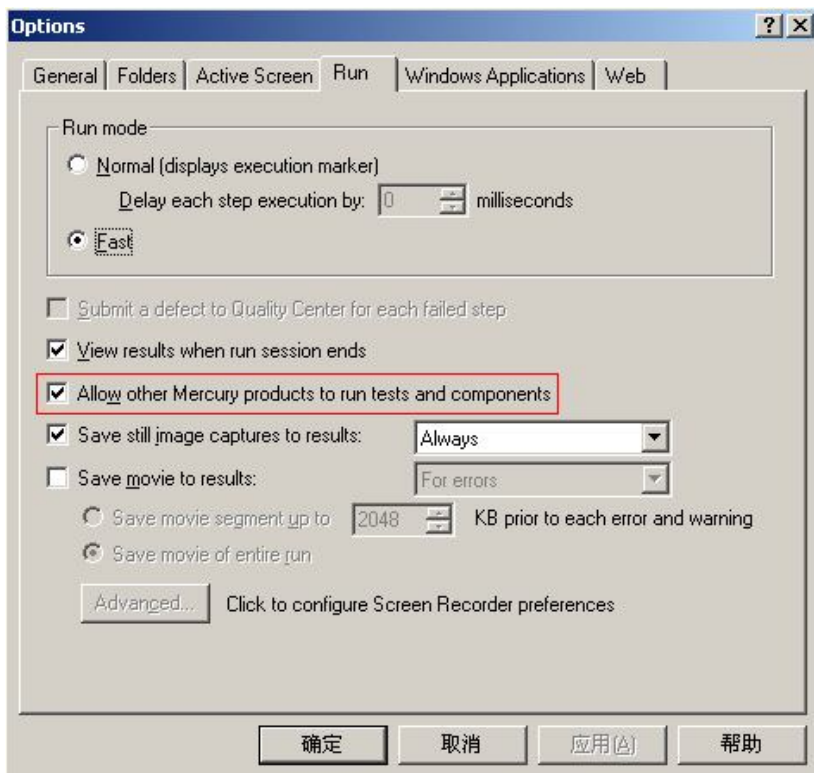


图 7-23 设置集成数据与测试一起保存

3. 添加语句

可以使用 Services 对象及其相关方法来插入与性能测试相关的语句，主要包括：Abort、GetEnvironmentAttribute、LogMessage、SetTransactionStatus、ThinkTime、UserDataPoint、StartTransaction 及 EndTransaction。

4. 为 LoadRunner 设计测试

在设计要用于 LoadRunner 的测试时，需要考虑以下设计准则：

- LoadRunner 中使用的 QuickTest 测试应该简单，并专门针对特定的操作。
- LoadRunner 无法运行嵌入式操作迭代。
- 请勿引用外部操作或其他外部资源，如外部数据表文件、环境变量文件和共享对象库等。

7.5.2 WinRunner 创建 GUI Vuser 脚本

WinRunner 是一个基于 Windows 平台，用于创建、编辑和调试 GUI Vuser 脚本的完整开发环境。通过 WinRunner 可记录真实用户在应用程序中的操作。

WinRunner 的 GUI Vuser 脚本是使用 TSL（Mercury Interactive 的测试脚本语言）编写的。TSL 是一种与 C 类似的高级编程语言。它结合了传统编程语言的功能和灵活性以及专为测试设计的功

能。但是 WinRunner 中创建的 Vuser 脚本是不能在 UNIX 计算机上运行的。

同样，使用 WinRunner 创建 GUI Vuser 脚本时，要注意以下几个方面：

1. 插入事务和集合点

插入事务和 QuickTest 一样，直接使用事务函数插入开始事务与结束事务，集合点方面将 Rendezvous 语句插入到 Vuser 脚本中即可。

2. 向 Controller 发送消息

运行场景时，Controller 的“输出”窗口显示有关脚本执行情况的重要信息。除了 WinRunner 自动发送的消息外，还可以在脚本中插入向 Controller 发送错误消息和通知消息的语句。

`error_message` 函数用于向 Controller 的“输出”窗口或 Console 发送错误消息。此函数的语法为：

```
error_message(message);
```

其中 `message` 为文本字符串。

`output_message` 函数用于发送不是错误消息的特殊通知。此函数的语法为：

```
output_message(message);
```

其中 `message` 为文本字符串。

在执行场景或会话步骤间，可以识别以下对象：

- 某特定时刻在场景中执行任务的 Vuser。
- 执行脚本的负载发生器。
- 运行 Controller 的计算机。

而有关 Vuser 和负载发生器的信息可以通过以下函数来获取：

`lr_whoami`：返回 Vuser 名称及其所属的 Vuser 组。


`get_host_name`：返回执行脚本的计算机名称。

`get_master_host_name`：返回运行 Controller 或 Console 的计算机名称。

7.5.3 场景中使用 GUI Vuser 脚本

在 QuickTest 或 WinRunner 中创建 GUI Vuser 脚本后，便可以将该脚本集成到 LoadRunner 场景中。在将 GUI Vuser 脚本添加到 LoadRunner 场景前，要注意以下几个问题：

- 每台计算机上每次只能运行一个 GUI Vuser。
- 请确保在运行场景前关闭 QuickTest 或 WinRunner。
- 在 Run-time Settings for script 设置对话框中，只有 General 类中的 Run Logic 和 Think Time 与 QuickTest 和 WinRunner 测试有关，如图 7-24 所示。

接下来就是要将 GUI Vuser 添加到 LoadRunner 场景中，在控制器界面，选择 File→Open，在弹出的对话框中选择要添加的 GUI Vuser 脚本。也可以选择 File→New，在弹出的 New Scenario 对话框中点击  按钮，弹出 Open Test 对话框，选择要添加的 GUI Vuser 脚本。这里有一个“文件类型”框，如图 7-25 所示。对于 WinRunner 来说，选择“GUI Scripts”文件类型。对于 QuickTest 来说，选择“Astra Tests”文件类型。确定文件类型后，选择合适的脚本并进行添加即可。

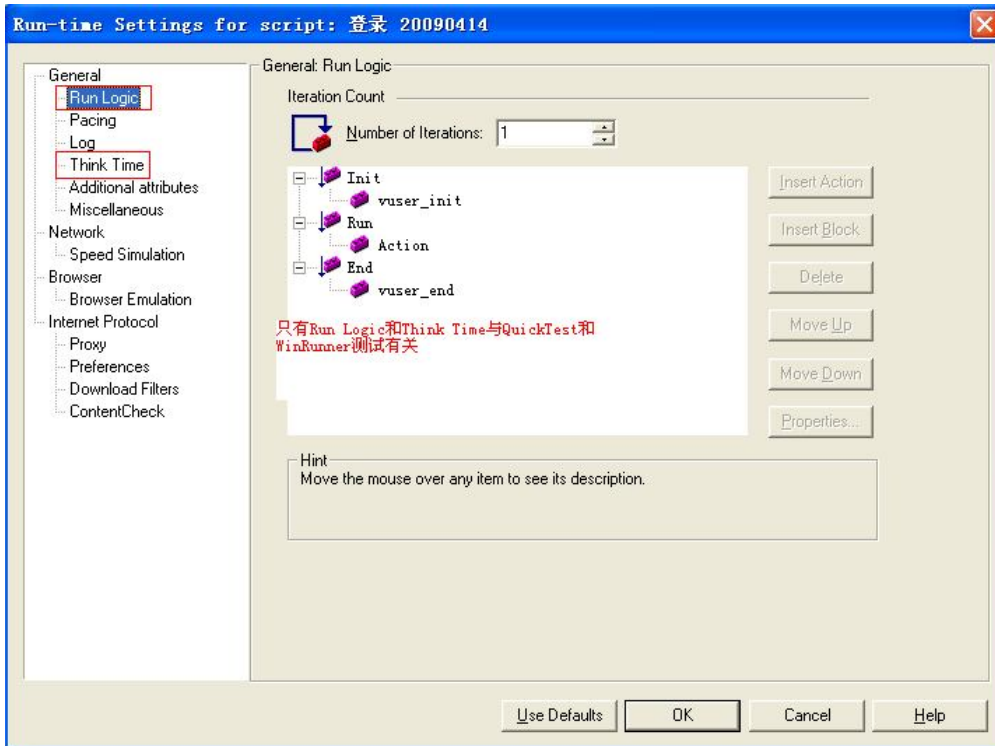


图 7-24 与 QuickTest 和 WinRunner 有关项

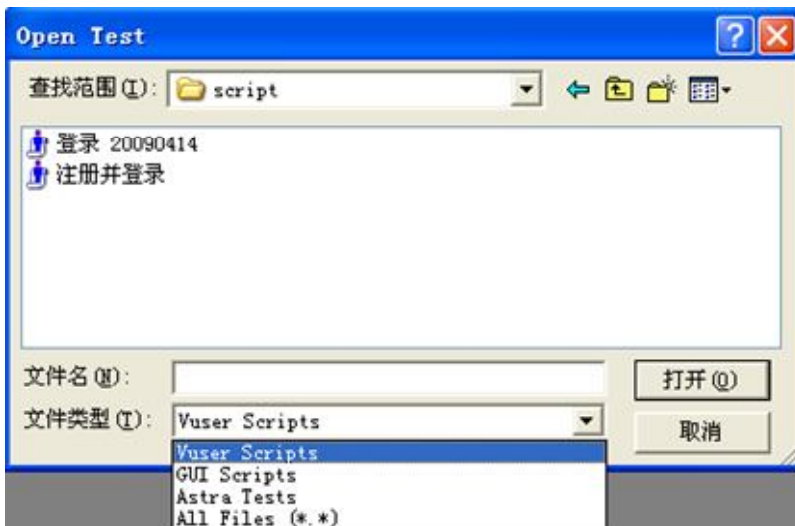


图 7-25 文件类型选择