LoadRunner自动化测试工具的应用
(讲稿)

目录

（由于工作繁忙，时间仓促，讲稿均在下班后写成，加上LR博大精深，本人水平有限，敬请各位提出批评和建议，文中参考了深圳软件测试论坛两位板主的部分观点，在此表示感谢）

第一部分：Loadrunner的简介

LoadRunner® 是一种预测系统行为和性能的工业级标准性能测试负载测试工具。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。

目前企业的网络应用环境都必须支持大量用户，网络体系架构中含各类应用环境且由不同供应商提供软件和硬件产品。难以预知的用户负载和愈来愈复杂的应用环境使公司时时担心会发生用户响应速度过慢，系统崩溃等问题。这些都不可避免地导致公司收益的损失。Mercury Interactive 的LoadRunner能让企业保护自己的收入来源，无需购置额外硬件而最大限度地利用现有的IT资源，并确保终端用户在应用系统的各个环节中对其测试应用的质量，可靠性和可扩展性都有良好的评价。

LoadRunner 是一种适用于各种体系架构的负载测试工具，它能预测系统行为并优化系统性能。LoadRunner 的测试对象是整个企业的系统，它通过模拟实际用户的操作行为和实行实时性能监测，来帮助您更快的查找和发现问题。此外，LoadRunner能支持广泛的协议和技术，为您的特殊环境提供特殊的解决方案。

1.1 安装注意事项

LR的版本：目前常见的:LR7.0,LR7.51,LR7.6,LR7.8,每个版本差别蛮大的，大家尽量使用高版本。本文以LR7.8为例。

操作系统选择：LR是个比较底层的软件，OS最好为Windows 2000，因为W2k的稳定性和兼容性都不错，需要的内存也低，有人把LR装在WinXp下面，是有问题的。出错现象：“应用程序正常初始化（0xc0000005）失败”。在win2003,winXP下安装LR后会出现如此情况。

License问题：LR的license是区分类型的，一般是按协议和时间，用户数量来区分的，比如：

License for LoadRunner 7.51 (Type：Global 500， Time Limited时效：1年 ） 就表示支持所有协议，最大500VU，时间：1年。

三种安装类型：Standalone Installation，Network Installation，Network Installation and shortcuts

四种安装方式：Typical Installation，Load Generator，MI Listener，Custom Installation 我们根据实际情况选择，我用Standalone Installation和Custom Installation安装，安装所有组件。

1.2 协议的选择或者 VUSER 类型的选取

我现在要用 LoadRunner 测一个 C/S or B/S 系统，请问该用什么协议？

经常有新手问：为什么我用 LR 录完之后 VuGen 里产生不了脚本？这就是协议选择的问题了，LR 支持的协议和应用非常广泛，很少有人能用完这么多协议，我们就常见的大多数人用的加以讨论：

B/S 系统：选择 Web(Http/Html),

C/S 系统：根据 C/S 结构所用到的后台数据库来选择不同的协议,如果后台数据库是 Sybase，则采用 sybaseCTlib 协议，如果是 Sql server,则使用 MS Sql server 的协议，至于 oracle 数据库系统，当然就使用 oracle 2-tier 协议。

对于没有数据库的 c/s（ftp, SMTP）这些可以选择 windows sockets 协议。

至于其他的 ERP，EJB（需要 ejbdetector.jar），选择相应的协议即可.

1.3 LR 的基本原理

LR 启动以后，在任务栏会有一个 Agent 进程，通过 Agent 进程，监视各种协议的 Client 与 Server 端的通讯，用 LR 的一套 C 语言函数来录制脚本，所以只要 LR 支持的协议，就不会存在录制不到的，这是它与 Load test，WR，Robot（Gui）录制脚本的很大一个区别。(WR 必须识别对象，才能录制到)。然后 LR 调用这些脚本向服务器端发出请求，接受服务器的响应。至于服务器内部如何处理，它不关心。

1.4 测试脚本录制/分配所遵循的几个原则

1. 脚本越小越好。就像写 code 一样的，不要太长，尽量做到一个功能（Transaction）一个脚本。如果那些功能是连续有序的，必须先做上一个，才能工作下一个，那就只好放在一起了。
2. 选择使用频率最高的。有些人喜欢在 LR 中测试几乎所有的功能，其实这样不合适，我们把最常用的、使用频率最高的、拿出来测试。但是也要结合用户实际使用情况，一般在一个系统中是多个用户使用多个功能，某些功能使用的频率更大一些，我们在录制脚本之前就要设计好，哪个脚本会跑几个用户，一共需要多少个脚本，能满足性能测试的需求。
3. 选择你所需要的进行录制。对于WEB的程序,对于你所关注的内容没什么影响

的操作,你可以不录制，可以使用暂停，这需要试的，对被测功能有一个清楚的认识和了解，要能把握住哪些地方是对整个过程没有影响的，比如一些查询,通常,选择条件的页面都可以不录制,但对于一些页面有可能要传递参数,就需要录制了，如何确定哪些点可以不录制，一是可以找开发人员了解清楚程序设计的结构，再就是靠自己的经验，作的多了，就心中有数了。

例子：

Test case name	Vuser number Total 50	Vuser number Total 20	Iteration
Test case 1: merchants create schedule and costing sheet	10	4	200
Test case 2: Merchant run report-5 concurrent users (3 merchants, 2 vendors).	5	1	200
Test case 3: merchant edit costing sheet FOB and create production schedule	9	4	200
Test case 4: merchant reply schedule tasks	6	3	300
Test case 5: vendor reply schedule tasks	7	3	200
Test case 6: vendor edit costing sheet	7	3	200
Test case 7: merchant create topic, vendor reply topic	4	1	100
Test case 8: imports reply schedule and do classification.	2	1	10

第二部分：录制脚本

2.1 录制脚本前需要理解的几个基本概念

2.1.1 事务（Transaction）

事务（Transaction）是这样一个点，我们为了衡量某个action的性能，需要在action的开始和结束位置插入这样一个范围，这就定义了一个transaction，LoadRunner 运行到该事务的开始点时，LoadRunner 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在结果中会有反映。

插入事务操作可以在录制过程中进行，也可以在录制结束后进行。LoadRunner可以在脚本中插入不限数量的事务。

举个例子：比如一个单据,把从登录到保存成功退出整个作为一个脚本,对于需要关注的保存时间,定义为单独的事务,以取得响应时间，事务脚本函数如下：

```
lr_start_transaction("SubmitBookData");

/*
 * 中间代码是具体事务的操作
 */

lr_end_transaction("SubmitBookData", LR_AUTO);
```

2.1.2 集合点（Rendezvous）

集合点：是一个并发访问的点，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，并发访问的目的。

注意：集合点经常和事务结合起来使用，常放在事务的前面，集合点只能插入到Action 部分，vuser_init和vuser_end 中不能插入集合点。集合点函数如下，参数不能加空格：

```
lr_rendezvous("SubmitQueryData");
```

加入集合点之后，在后面运行过程中可以看到VU的状态，会等待集合。

2.1.4 IP Spoofer（IP 欺骗）

当运行场景时，虚拟用户使用它们所在的Load Generator 的固定的IP 地址。每个Load Generator 上（同时）运行大量的虚拟用户（*不明白），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，并且有一些网站会限制同一个IP 的登陆。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 访问同一网站，这种技术称为“IP 欺骗”。
启用该选项后，场景中运行的虚拟用户将模拟从不同的IP 地址发送请求。该选项非常
的有用。**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DHCP）。**

IP Wizard工具就提供了生成多个ip的功能，IP Wizard是一个单独的程序，我们可以在开始菜单里面找到，你可以添加一个局域网内的IP段。添加后重启，在Win2k下使用Ipconfig/all查看到很多虚拟的IP，最后要在Controller里面选择enable ip spoofer.

2.1.5（Text/Image）检查和 contents check 点

对于查询类的脚本,一定要添加检查点,以保证在测试时结果的正确性.因为LR只要检测到网页的响应,就认为是pass而并不管当前网页内容的正确性.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Image，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。检查点的含义和WinRunner中的检查点功能基本上一致，这里就不再作过多的说明。

举个例子：当我用loadrunner做压力测试的时候，
它的确能反馈给我各种服务器性能的数据，
但是在做B/S结构系统的测试的时候，
却发现如下问题：
loadrunner不能正确判断操作是否成功，
比如登录，我要测试200人同时登录，
但是我的login.jsp里面没有正确的关闭数据库的连接，
导致登录100人后，建立了100个数据库连接，
第101人一个人登录的时候，由于超出数据库连接的最大数，
所以，jsp程序抛出了一个数据库异常。
但是页面的走向是正确的，所以loadrunner会认为程序是正确执行的，但是事实却并非如此。

对于有的页面是无法添加文本和图像检查点的，就加入contents check点。

2.1.6 LR 脚本复用问题

作为一款优秀的负载测试工具，LR的测试脚本有很好的复用性，参数化后的脚本，在应用没什么大的变化的情况下，一直是可以用的。甚至你在A服务器录制的脚本，如果做测试的时候，需要转移到B服务器上，你只需要用查找替换的功能将A服务器的IP地址换成B服务器的IP地址就可以使用。

2.1.7 理解 Correlation（关联）

关联是用来解决脚本中存在的动态数据问题的.在7.8中,当你回放一次后,LR会自动录找你录制的时候和回放时候的差别,找出动态数据,并作成参数。举个我作的动态数据的例子，当用户登录时，会产生一个Sessionid号，访问结束后，该Sessionid便会失效。我录制的时候，在脚本里面获取到了该Sessionid，但当我再回放的时候，这个Sessionid已经无效了，所以我需要把这个Sessionid作为一个动态数据，当我一登录的时候，便获取一个新的有效的Sessionid，然后通过函数把它保存下来

```
web_reg_save_param("WCSParam_Text2",
"LB=ProcessID=",
"RB=;",
"Ord=1",
"RelFrameId=1",
"Search=body",
LAST);
```

那么后面用到的页面中就可以调用**WCSParam_Text2**这个参数使用这个新的Sessionid号。这是个动态数据很典型的例子，

```
web_submit_data("w_onload_check.asp",
"Action=http://gsserver3/cwbase/sys/userlogin/w_onload_check.asp",
"Method=POST",
"TargetFrame=",
"RecContentType=text/html",
"Referer=http://gsserver3/cwbase/sys/userlogin/index.asp",
"Snapshot=t3.inf",
"Mode=HTML",
ITEMDATA,
"Name=hdclentip", "Value=chenjing", ENDITEM,
"Name=hdProcessID",
"Value=F8E5ACCD372845C38C7E1981A342F703", ENDITEM,
"Name=selInstanceid", "Value=T01", ENDITEM,
```

```
"Name=TxtUserID", "Value={RYBH}", ENDITEM,
"Name=TxtPassword", "Value=cwpass", ENDITEM,
"Name=Txtkjdate", "Value=2003.01.06", ENDITEM,
"Name=selInstancetxt", "Value=性能测试用(中型数据库)", ENDITEM,
"Name=selInstance", "Value=T01", ENDITEM,
LAST);
web_url("loginpage.aspx",
"URL=http://gsserver3/cwbase/sys/menushow/loginpage.aspx?Proces
sID={WCSParam_Text2}",
"TargetFrame=",
"Resource=0",
"RecContentType=text/html",
"Referer=",
"Snapshot=t4.inf",
"Mode=HTML",
LAST);
```

理解web_reg_save_param函数，
int web_reg_save_param (const char *ParamName, <List of Attributes>, LAST);

第一部分：参数名字，用双引号括起，逗号分开；
第二部分:List of Attributes,包括:LB、RB、RelFrameID、Ord、Search、SaveOffset、SaveLen等，
第三部分：LAST，结束标志。

左边界，右边界到底是个什么概念？
LB是左边界，要查找的字符串左面的边界值，即位于查找字符串的最左边的字符串，RB是右边界，要查找的字符串右面的边界值，即位于查找字符串的最右边的字符串，比如说吧，程序中有这么一个赋值，ProcessID =A53625E18440FCE81F26DCE712E65EBA;如果ProcessID的值是动态的，我想使用动态变量，那我设定查找左边界为LB=ProcessID=,右边界为RB=;的字符，如果找到了，就替换成变量.
Search是指查找范围，就是说在哪里查找这些值，可以取这样几个值，我们一般设为ALL，Body等即可。
Headers (Search only the headers),
Body (search only Body data, not headers),
Noresource (search only the html body, excluding all headers and resources),
ALL (search Body and headers). The default value is ALL.
RelFrameID: The hierarchy level of the HTML page relative to the requested URL.一般取1

ORD: This parameter, also known as Instance, indicates the ordinal or

instance of the match. 一般取 1

LR7.8已经为用户定义好了很多类型的关联，我们可以自己定义New rule，不过我在录制脚本的时候一般把系统的那些都关掉，定义自己的，只是有的时候，它不能自动关联，就干脆手工作了。需要关联的地方：一般是在有主键，不允许记录重复，和一些主程序中传递过来的参数，在后面要用到等情况才使用关联的，你可以先执行一遍，如果有些地方不允许重复，参数值无效，就会出错的，根据错误提示你可以判断出来，就知道需要关联了。一般需要关联的不多，我现在遇到的就是进程号，以及表的主键。我这里有个Correlation的官方文档，大家下去可以自己看看。

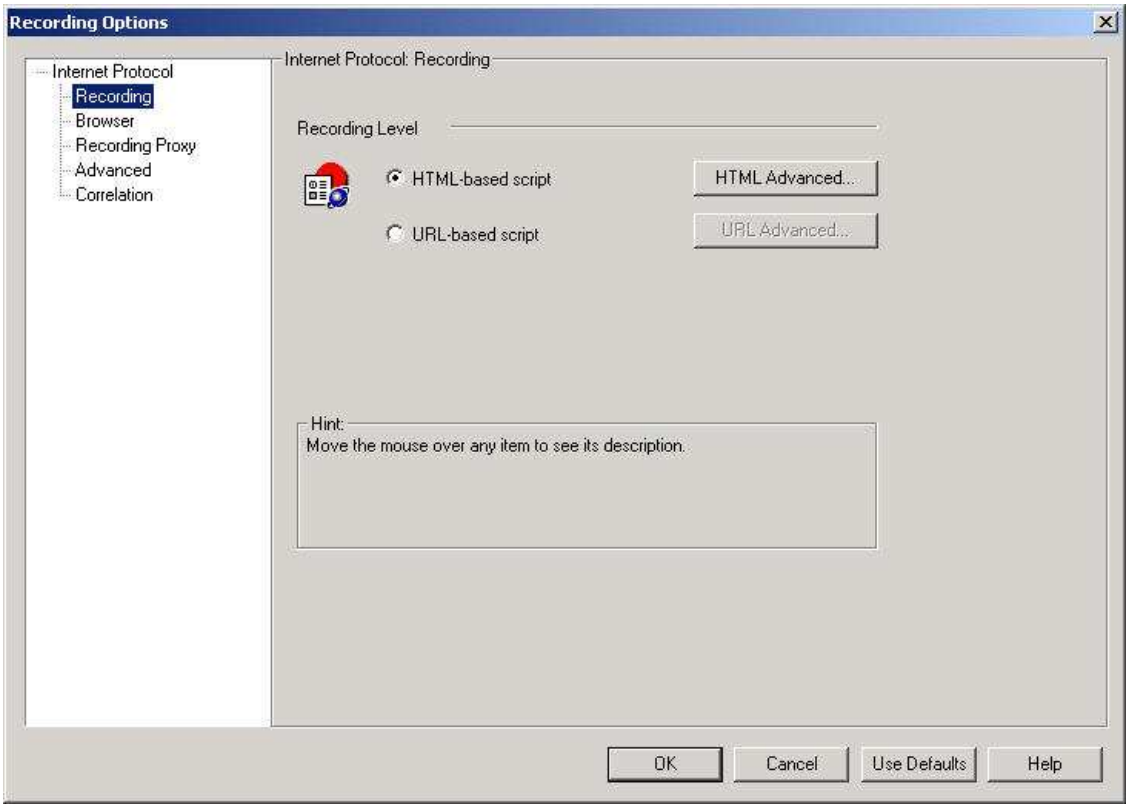
2.1.8 以录制 Web(Http/Html)协议为例讲述一下 LR 的脚本的录制

LR的脚本是C语言代码，LR有自己的一整套函数接口，可以供外部调用，在VUGen里面敲Lr_就可以看到了。Web(Http/Html)脚本本身分INIT，ACTION，END三部分，各部分的解释：INIT部分可以理解为初始部分，ACTION可以理解为事务部分，也是测试的主体，END是退出结束。重复的时候，仅重复action部分。我们一般把登录部分放在init，退出放到end，只会执行一次，或者有的时候，各部分反复的次数不一样，分成多个action,可以单独设定反复次数。（如果需要在登陆操作设集合点，那么登陆操作也要放到Action 中，因为vuser_init 中不能添加集合点）

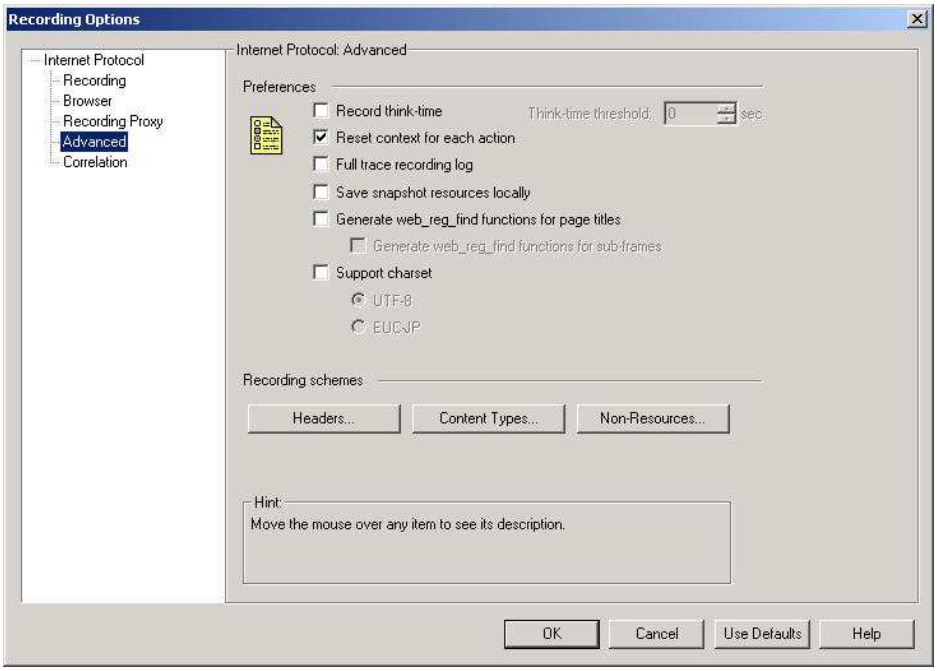

```
Init，login
#include "web_api.h"
#include "lrw_custom_body.h"


vuser_init()
{
//登录的脚本可以放在这里
    }
```

录制前的选项设置：



解释：1.基于浏览器的应用程序推荐使用HTML-based Script，脚本中采用HTML页面的形式来表示，这种方式的Script脚本容易维护，容易理解，使用该选项中的advance中的第一个选项，如果单纯的HTML方式，是不允许使用关联的。
2．不是基于浏览器的应用程序推荐使用URL-based Script，脚本中的表示采用基于URL 的方式，不是很好阅读。



解释：1.是否记录录制过程中的ThinkTime，如果记录，还可以设置最大值，一般我不记录这个值。

2．通知Vugen去重新设置每个action之间的Http context，缺省是需要的。

3．完整记录录制过程的log，

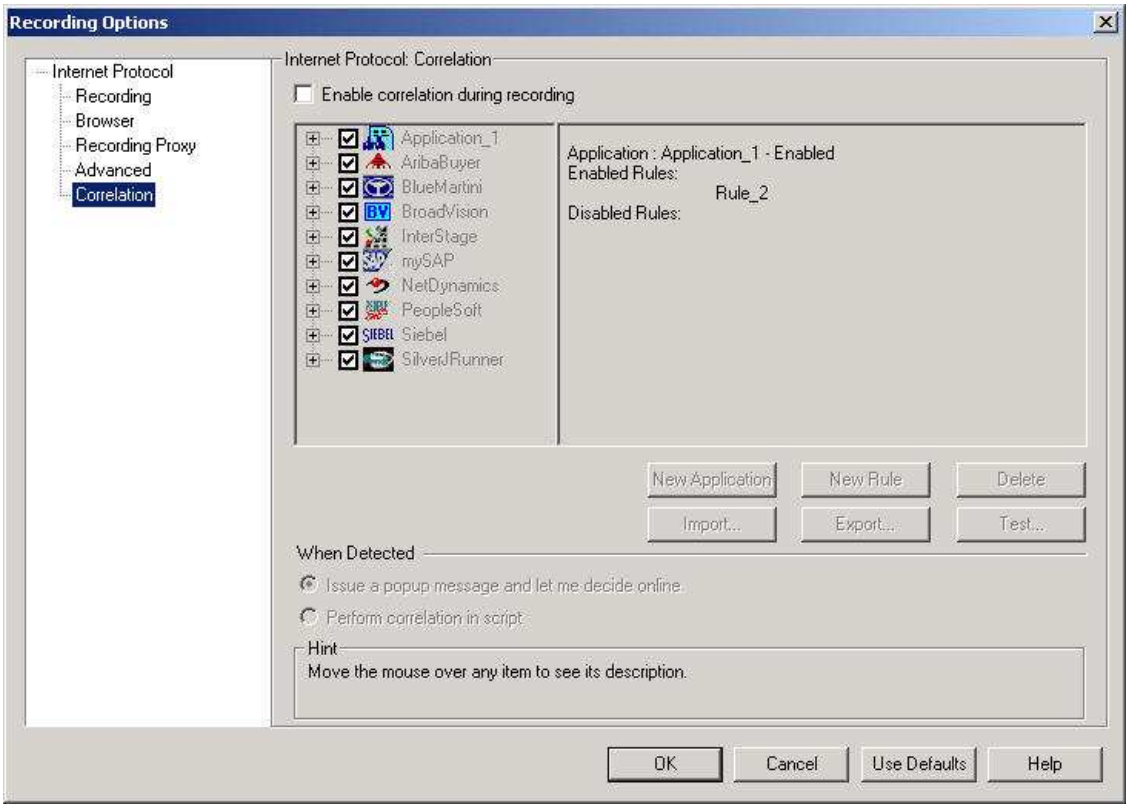
4．保存一个本地的snapshot，可以加速显示

5．把html的title放到web_reg_find函数里面

6.支持的字符集标准

7．Http header的录制，我们采用缺省即可，不需要用web_add_header去录制非标准的header信息。

对录制的content的内容进行filter，不作为resource处理的。



解释：这个就是我前面提到的关联，系统已经预先设置好了一些常见的关联rules，我们录制脚本之前，可以把系统的都关掉，定义自己的，只是有的时候，它不能自动关联，就干脆手工关联。这里比较重要，我还有一个专门的PPT文档是详细讲这个的，大家可以到我的网站上下载。

2.2 脚本录制

现在可以开始录制脚本了，我给出几段已经录制好的脚本。

```
lr_rendezvous("createpreproduction schedule ");
lr_start_transaction("create pre production schedule");

    web_url("Folder.jsp_4",
        "URL=http://172.17.16.5/xpc71/jsp/com/folder/Folder.jsp",
        "Resource=0",
        "RecContentType=text/html",
        "Referer=http://172.17.16.5/xpc71/LoginAction.do",
        "Snapshot=t10.inf",
        "Mode=HTML",
        LAST);
```

```
	web_url("TemplateAdminAction.do",
		"URL=http://172.17.16.5/xpc71/TemplateAdminAction.do?forwardID=1",
		"Resource=0",
		"RecContentType=text/html",
		"Referer=http://172.17.16.5/xpc71/LoginAction.do",
		"Snapshot=t11.inf",
		"Mode=HTML",
		LAST);
//可以自己手工，也可以让 correlate 定义一个动态参数
web_reg_save_param( "WCSParam_Diff2", "LB= name='", "RB='", "Ord=7",
"Search=Body", "RelFrameId=1", LAST );

	web_submit_form("ScheduleCreationAction.do",
		"Snapshot=t12.inf",
		ITEMDATA,
		"Name=PTName", "Value={schedule_name}", ENDITEM,
		"Name=headerTempID", "Value=preProductionHeaderTemplate", ENDITEM,
		"Name=selectHeader", "Value=1", ENDITEM,
		"Name=schTempID", "Value=preProductionDetailTemplate", ENDITEM,
		LAST);

	web_url("Folder.jsp_5",
		"URL=http://172.17.16.5/xpc71/jsp/com/folder/Folder.jsp",
		"Resource=0",
		"RecContentType=text/html",
		"Referer=http://172.17.16.5/xpc71/LoginAction.do",
		"Snapshot=t13.inf",
		"Mode=HTML",
		LAST);

	lr_end_transaction("create pre production schedule", LR_AUTO);

	 lr_rendezvous("createcs");
	lr_start_transaction("create cs");

	web_submit_data("CostingPreCreateAction.do",
		"Action=http://172.17.16.5/xpc71/CostingPreCreateAction.do",
		"Method=POST",
		"RecContentType=text/html",
		"Referer=http://172.17.16.5/xpc71/ScheduleCreationAction.do",
		"Snapshot=t14.inf",
		"Mode=HTML",
		ITEMDATA,
		"Name=scheduleID", "Value={schedule_id}", ENDITEM,
		"Name=scheduleName", "Value={schedule_name}", ENDITEM,
		"Name=VS022264", "Value=Spring 2004", ENDITEM,
```

```
        "Name=VS372264", "Value=ANF", ENDITEM,
        "Name={WCSParam_Diff2}", "Value=test001", ENDITEM,
        "Name=dParentID", "Value=", ENDITEM,
        "Name=itemID", "Value=", ENDITEM,
        "Name=ifRegen", "Value=Y", ENDITEM,
        "Name=editable", "Value=true", ENDITEM,
        "Name=calendar", "Value=", ENDITEM,
        "Name=respPartyID", "Value=", ENDITEM,
        "Name=respPartyName", "Value=", ENDITEM,
        "Name=listParentID", "Value=-1", ENDITEM,
        "Name=shareParty", "Value=", ENDITEM,
        LAST);

    web_url("menuAction.do_3",

    "URL=http://172.17.16.5/xpc71/menuAction.do?currentMenu=3&operation=change
MenuOnly",
        "Resource=0",
        "RecContentType=text/html",
        "Referer=http://172.17.16.5/xpc71/LoginAction.do",
        "Snapshot=t15.inf",
        "Mode=HTML",
        LAST);

    web_url("Folder.jsp_6",
        "URL=http://172.17.16.5/xpc71/jsp/com/folder/Folder.jsp",
        "Resource=0",
        "RecContentType=text/html",
        "Referer=http://172.17.16.5/xpc71/LoginAction.do",
        "Snapshot=t16.inf",
        "Mode=HTML",
        LAST);

    lr_end_transaction("create cs", LR_AUTO);
```

脚本里面有 2 个函数，解释一下：1.几个函数的解释：

1）int web_url (const char *Name, const char * url, <List of Attributes>,

[EXTRARES, <List of Resource Attributes>,] LAST);

这个函数 load 指定的 web 页面 .

*Name：页面的 name；

- url：页面的 url，Resource：指示 the URL 是否是一个资源。0，不是，1，是。

- RecContentType：录制脚本过程中，Header 响应的类型，e.g. *text/html*, *application/x- javascript*
- **Referer** - 参考 web 页的 the URL
- **Snapshot** - snapshot 文件名(扩展名 *inf*)，correlation 的时候要的。
- **Mode** - 录制的级别：HTML or HTTP
- **Last**- 属性列表的结束标志。

2） int web_submit_data (const char *StepName,//页面文件名， <List of attributes>,

ITEMDATA,//Item 数据

<List of data>,

[EXTRARES, <List of Resource Attributes>,]

LAST);

这个函数以 GET and POST requests 方式发送 form。

*StepName：

这里有个例子, the web_submit_data function submits a form using the POST method:

```
web_submit_data("customerinfo.asp",
    "Action=http://lazarus/webflight/customerinfo.asp",
    "Method=POST",
    "TargetFrame=",
    "EncType=multipart/www-urlencoded"
    "RecContentType=text/html"
    ITEMDATA,
    "name=flight", "value=6593", ENDITEM,
    "name=reserveFlight", "value=Next >",ENDITEM,
   LAST);
```

2.3 脚本的参数化

如果用户在录制脚本过程中，填写提交了一些数据，比如创建一个新的document。这些操作都被记录到了脚本中。当多个虚拟用户运行脚本时，都会提交相同的记录，这样做会被应用禁止，会出错，这样也不符合实际的运行情况，而且有可能引起冲突。为了更加真实的模拟实际环境，需要各种各样的输入。参数化输入是一种不错的方法

参数化包含以下两项任务：

① 在脚本中用参数取代常量值。

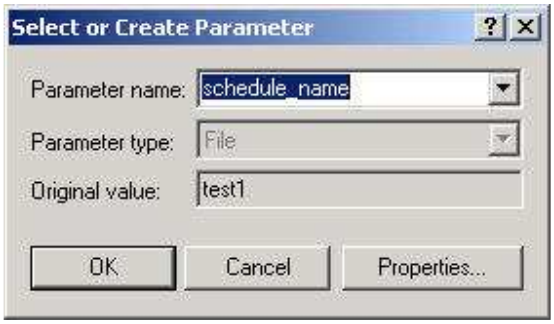
② 设置参数的属性以及数据源。

参数化仅可以用于一个函数中的参量。你不能用参数表示非函数参数的字符串。另外，不是所有的函数都可以参数化的。

我举一个例子来说明，还是我上面那个脚本，

```
web_submit_form("ScheduleCreationAction.do",
    "Snapshot=t12.inf",
    ITEMDATA,
    "Name=PTName", "Value=performance0001", ENDITEM,
    "Name=headerTempID", "Value=preProductionHeaderTemplate", ENDITEM,
    "Name=selectHeader", "Value=1", ENDITEM,
    "Name=schTempID", "Value=preProductionDetailTemplate", ENDITEM,
    LAST);
```

因为每次创建文档的时候，需要用不同的名字，系统禁止同名，如果同名就会出错，所以要把PTName值参数化，我们只要选中“performance0001”，然后点鼠标右键，选择“Replace with a parameter. ”，出现以下窗口：



参数类型解释：

DateTime：很简单，在需要输入日期/时间的地方，可以用DateTime 类型来替代。其属性设置也很简单，选择一种格式即可。当然也可以定制格式。

Group Name：暂时不知道何处能用到，但设置比较简单。在实际运行中，LoadRunner使用该虚拟用户所在的Vuser Group 来代替。但是在VuGen 中运行时，Group Name将会是None

Load Generator Name：在实际运行中，LoadRunner 使用该虚拟用户所在Load

Generator 的机器名来代替。

Iteration Number：在实际运行中，LoadRunner使用该测试脚本当前循环的次数来代替。

Random Number：随机数。很简单。在属性设置中可以设置产生随机数的范围

Unique Number：唯一的数。在属性设置中可以设置第一个数以及递增的数的大小。

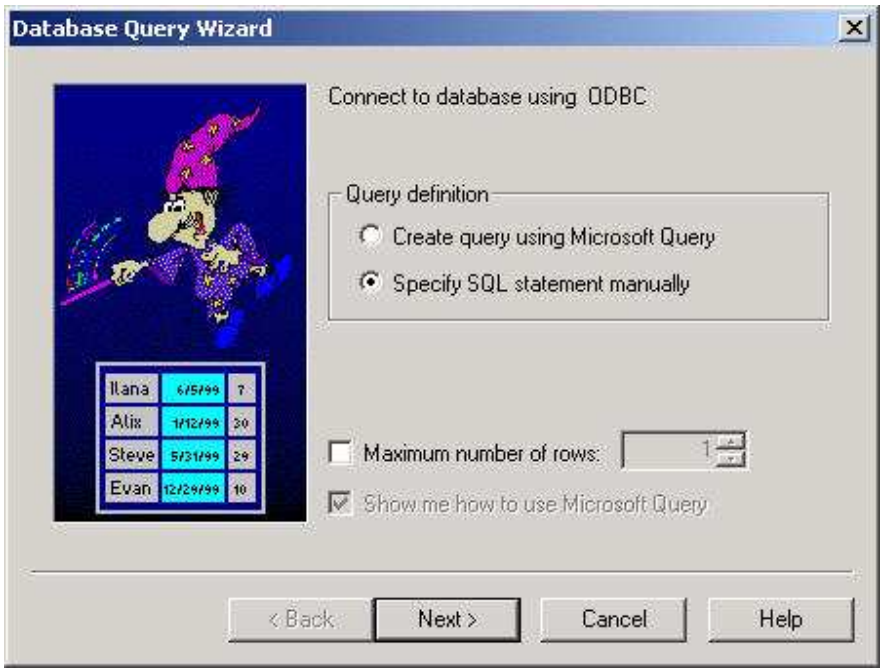
注意：使用该参数类型必须注意可以接受的最大数。例如：某个文本框能接受的最大数为99。当使用该参数类型时，设置第一个数为1，递增的数为1，但100个虚拟用户同时运行时，第100 个虚拟用户输入的将是100，这样脚本运行将会出错。注意：这里说的递增意思是各个用户取第一个值的递增数，每个用户相邻的两次循环之间的差值为1。举例说明：假如起始数为1，递增为5，那么第一个用户第一次循环取值1，第二次循环取值2；第二个用户第一次循环取值为6，第二次为7；依次类推。

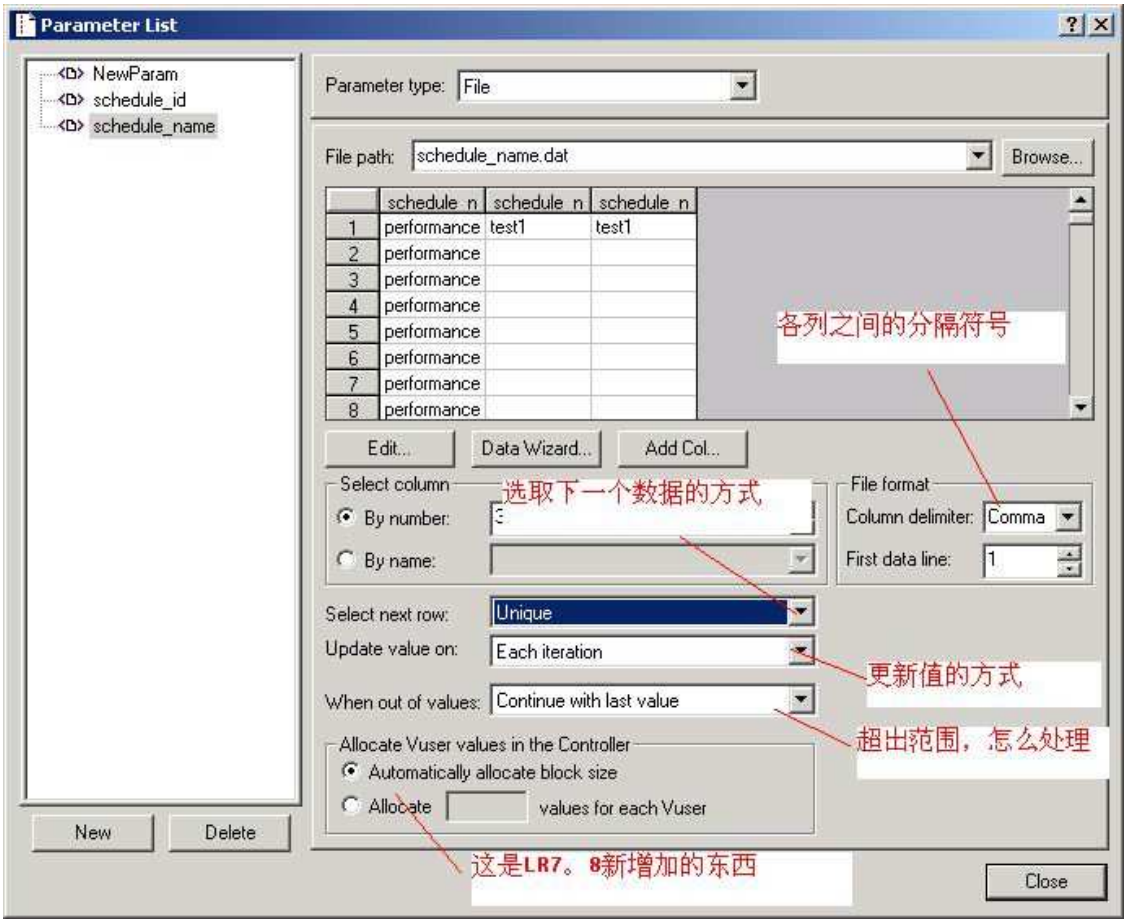
Vuser ID：设置比较简单。在实际运行中，LoadRunner 使用该虚拟用户的ID 来代
替，该ID 是由Controller 来控制的。但是在VuGen 中运行时，Vuser ID 将会是－1。

User Defined Function：从用户开发的dll 文件提取数据。

File：需要在属性设置中编辑文件，添加内容，也可以从现成的数据库中取数据

我们将会重点介绍这种参数类型，这也是 LR 的缺省参数类型，就是把准备好的数据放在文件或者用 sql 语句从数据库中取出来，让 VU 来读取。





“Select next row ”有以下几种选择：多个VU如何取值

Sequential：按照顺序一行行的读取。每一个虚拟用户都会按照相同的顺序读取

Random：在每次循环里随机的读取一个，但是在循环中一直保持不变

Unique ：每个VU取唯一的值。**注意：使用该类型必须注意数据表有足够多的数。比如Controller 中设定20 个虚拟用户进行5 次循环，那么编号为1 的虚拟用户取前5个数，编号为2 的虚拟用户取6-10 的数，依次类推，这样数据表中至少要有100个数据，否则Controller 运行过程中会返回一个错误。**

Same Line As 某个参数（比如Name）：和前面定义的参数Name 取同行的记录。通常用在有关联性的数据上面。这个也是很有用的，比如有时候我们要求指定VU取读取指定数据，就可以这样定义：

创建参数文件，共两列，假设userID，DOCID，userid设定取数方式是unique,DOCID则设成the same line as userid,如果第一行数据为0001,DOC1,则如果用户0001登录成功，在打开文档时，便会打开DOC1文档。

“Update value on”有如下几种选择：多次迭代如何取值

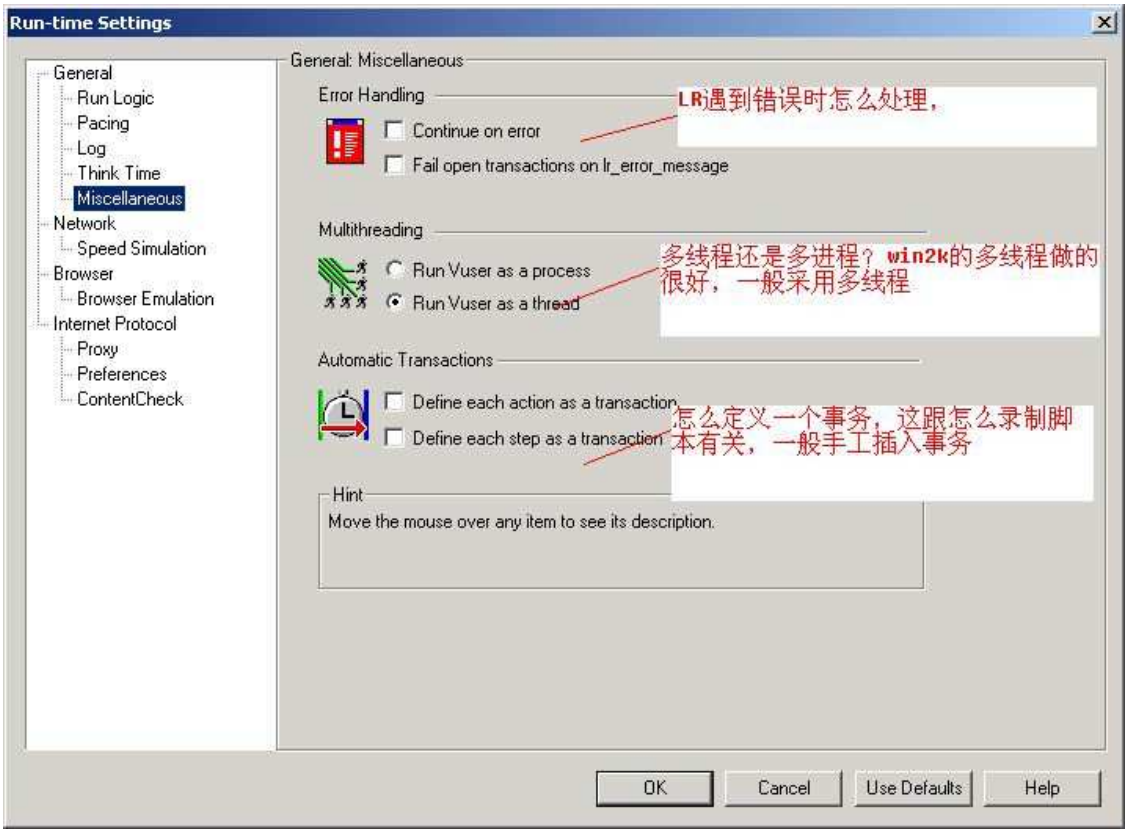
once 在所有的反复中都使用同一个值，

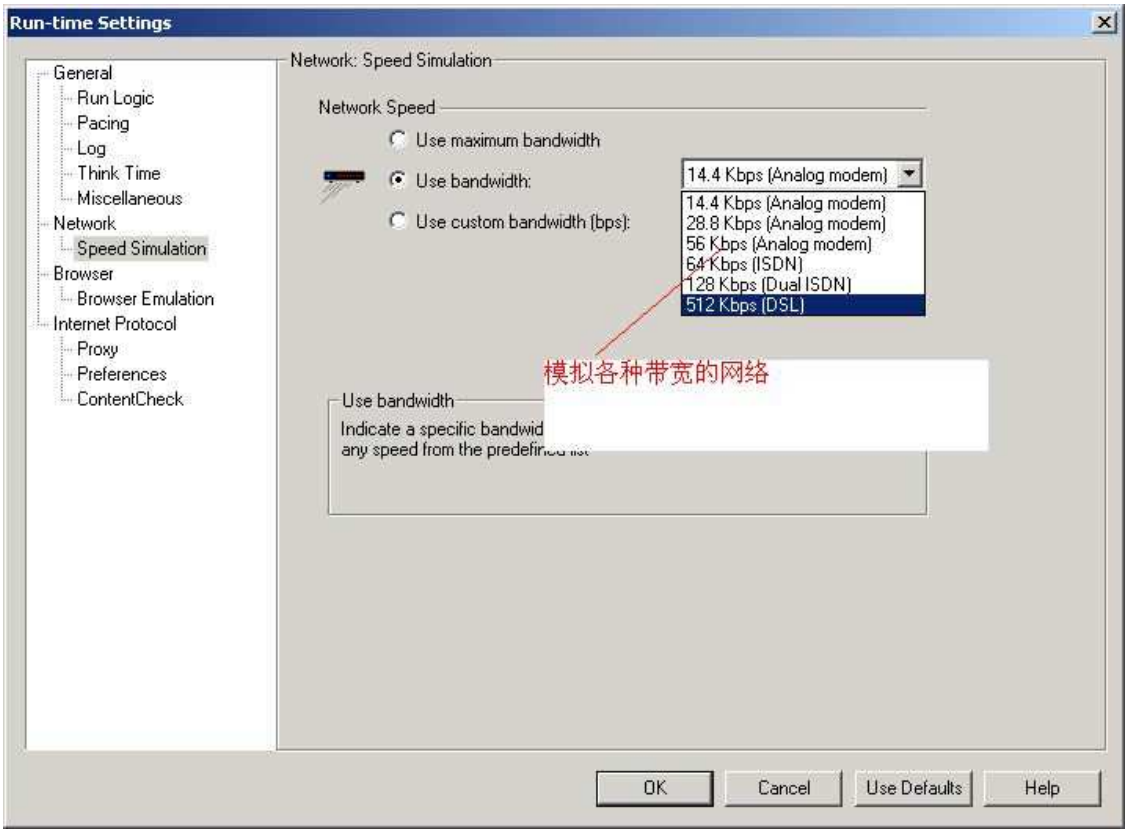
each iteration 则每次反复都要取新值，

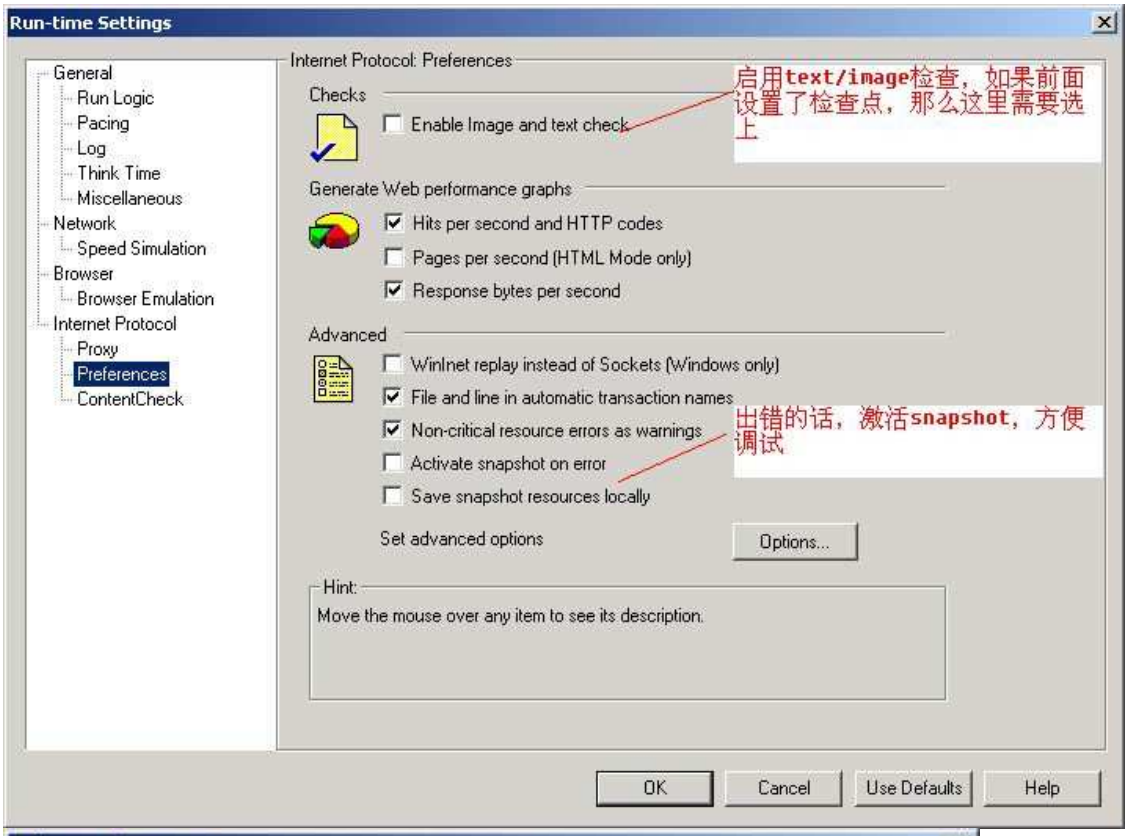
each occurrence 则只要发现该参数就要重新取值，也就是如果一个 action 中有多个该参数，每遇到一个就要重新取一个值。

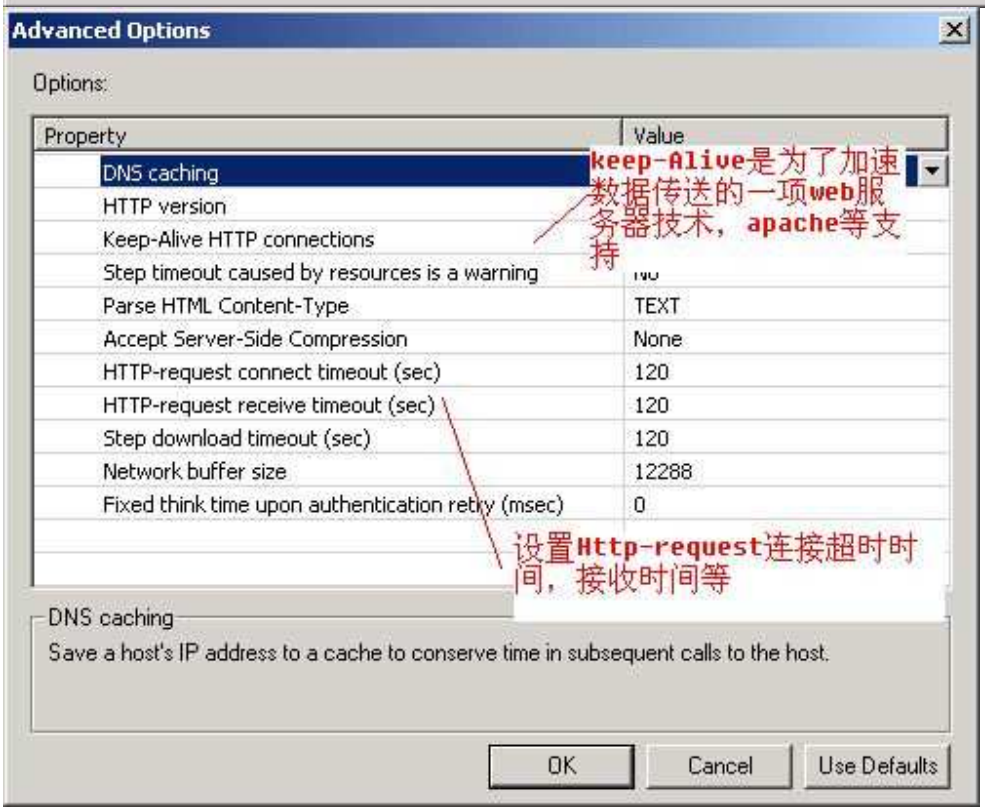
第三部分：创建运行场景

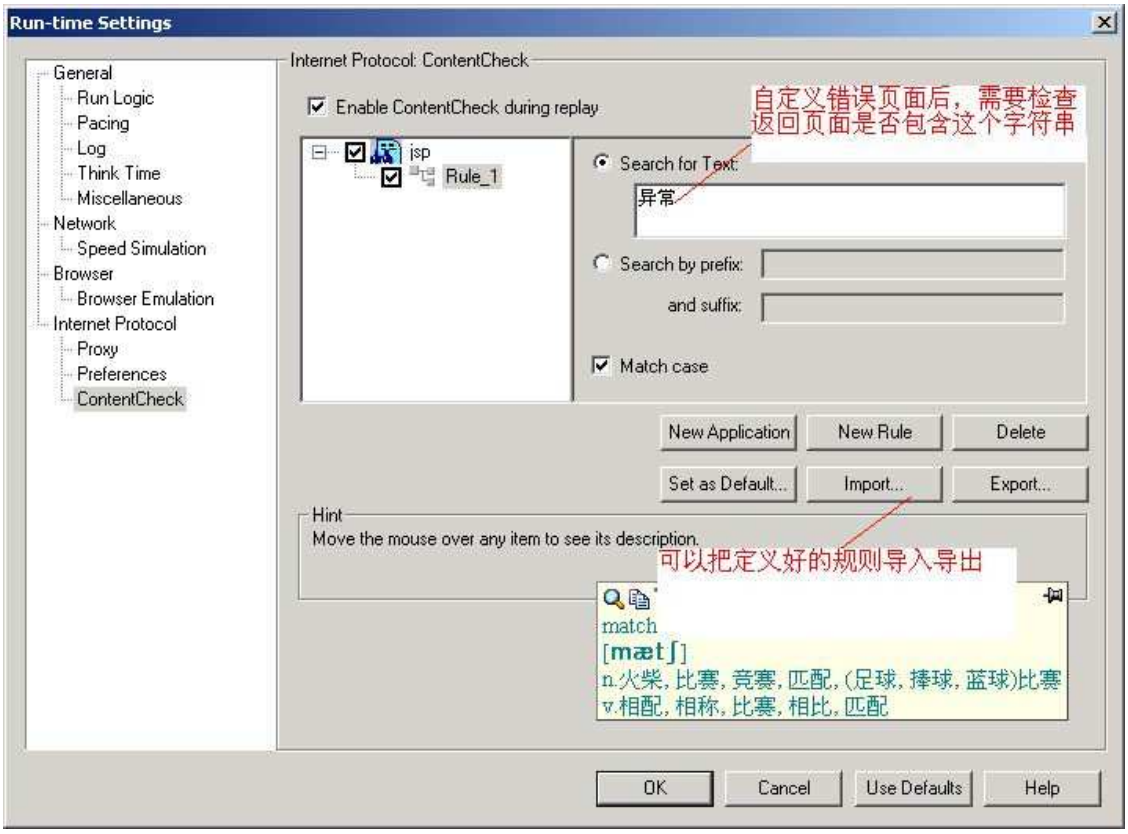
3.1 Run-Time Setting

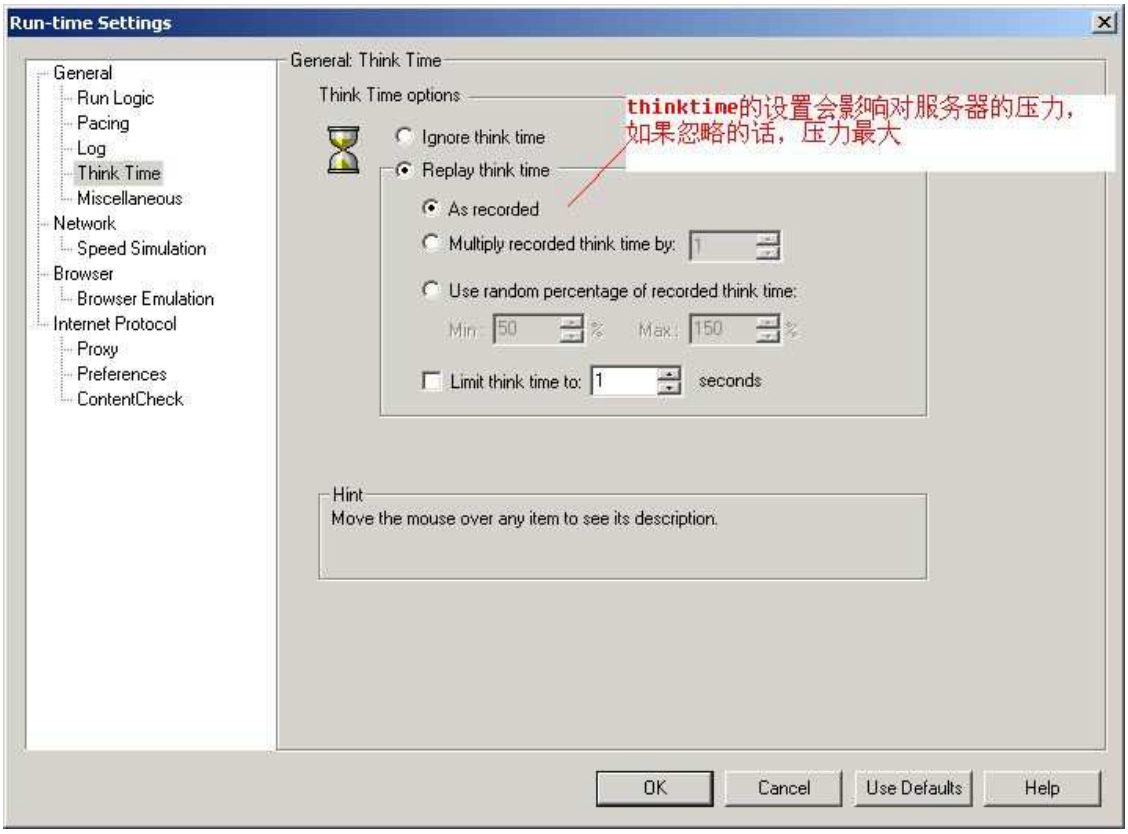



Run-time Settings
General
Run Logic
Pacing
Log
Think Time
Miscellaneous
Network
Speed Simulation
Browser
Browser Emulation
Internet Protocol
Proxy
Preferences
ContentCheck
Network: Speed Simulation
Network Speed
Use maximum bandwidth
Use bandwidth:
Use custom bandwidth (bps):
14.4 Kbps (Analog modem)
14.4 Kbps (Analog modem)
28.8 Kbps (Analog modem)
56 Kbps (Analog modem)
64 Kbps (ISDN)
128 Kbps (Dual ISDN)
512 Kbps (DSL)
模拟各种带宽的网络
Use bandwidth
Indicate a specific bandwid
any speed from the predefi
OK
Cancel
Use Defaults
Help


Run-time Settings
General
Run Logic
Pacing
Log
Think Time
Miscellaneous
Network
Speed Simulation
Browser
Browser Emulation
Internet Protocol
Proxy
Preferences
ContentCheck
Internet Protocol: Preferences
Checks
Enable Image and text check
启用text/image检查，如果前面设置了检查点，那么这里需要选上
Generate Web performance graphs
Hits per second and HTTP codes
Pages per second (HTML Mode only)
Response bytes per second
Advanced
WinInet replay instead of Sockets (Windows only)
File and line in automatic transaction names
Non-critical resource errors as warnings
Activate snapshot on error
Save snapshot resources locally
出错的话，激活snapshot，方便调试
Set advanced options
Options...
Hint:
Move the mouse over any item to see its description.
OK
Cancel
Use Defaults
Help


Advanced Options
Options:
Property
Value
DNS caching
HTTP version
Keep-Alive HTTP connections
keep-Alive是为了加速数据传送的一项web服务器技术，apache等支持
Step timeout caused by resources is a warning
Parse HTML Content-Type
TEXT
Accept Server-Side Compression
None
HTTP-request connect timeout (sec)
120
HTTP-request receive timeout (sec)
120
Step download timeout (sec)
120
Network buffer size
12288
Fixed think time upon authentication retry (msec)
0
设置Http-request连接超时时间，接收时间等
DNS caching
Save a host's IP address to a cache to conserve time in subsequent calls to the host.
OK
Cancel
Use Defaults

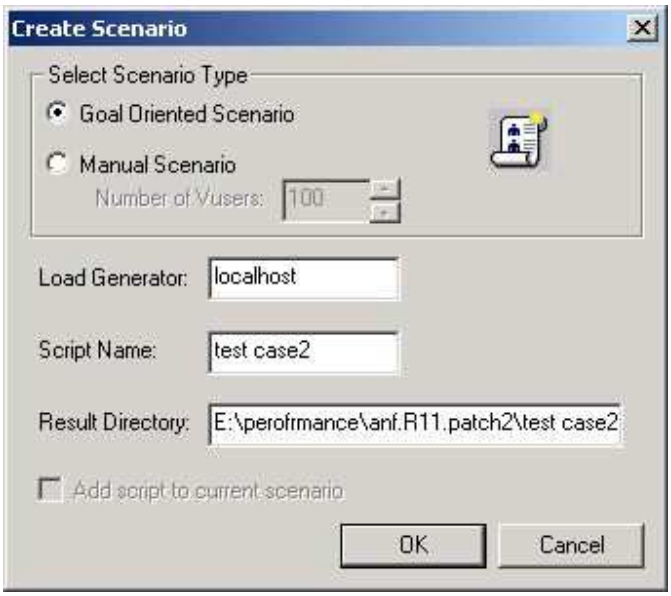

Run-time Settings
General
Run Logic
Pacing
Log
Think Time
Miscellaneous
Network
Speed Simulation
Browser
Browser Emulation
Internet Protocol
Proxy
Preferences
ContentCheck
Internet Protocol: ContentCheck
Enable ContentCheck during replay
自定义错误页面后，需要检查返回页面是否包含这个字符串
jsp
Rule_1
Search for Text:
异常
Search by prefix:
and suffix:
Match case
New Application
New Rule
Delete
Set as Default...
Import...
Export...
Hint
Move the mouse over any item to see its description.
可以把定义好的规则导入导出
match
[mætʃ]
n.火柴, 比赛, 竞赛, 匹配, (足球, 捧球, 蓝球)比赛
v.相配, 相称, 比赛, 相比, 匹配
OK
Cancel
Use Defaults
Help



和 TD 的连接，LR 可以很方便的和 TD 连接，把脚本放在 TD 中，

3.2 几种场景类型的选择

录制好脚本之后，就可以把脚本加入到场景里面去了，这里首先介绍一下 LR 的场景类型，LR 有 2 种大的场景类型：

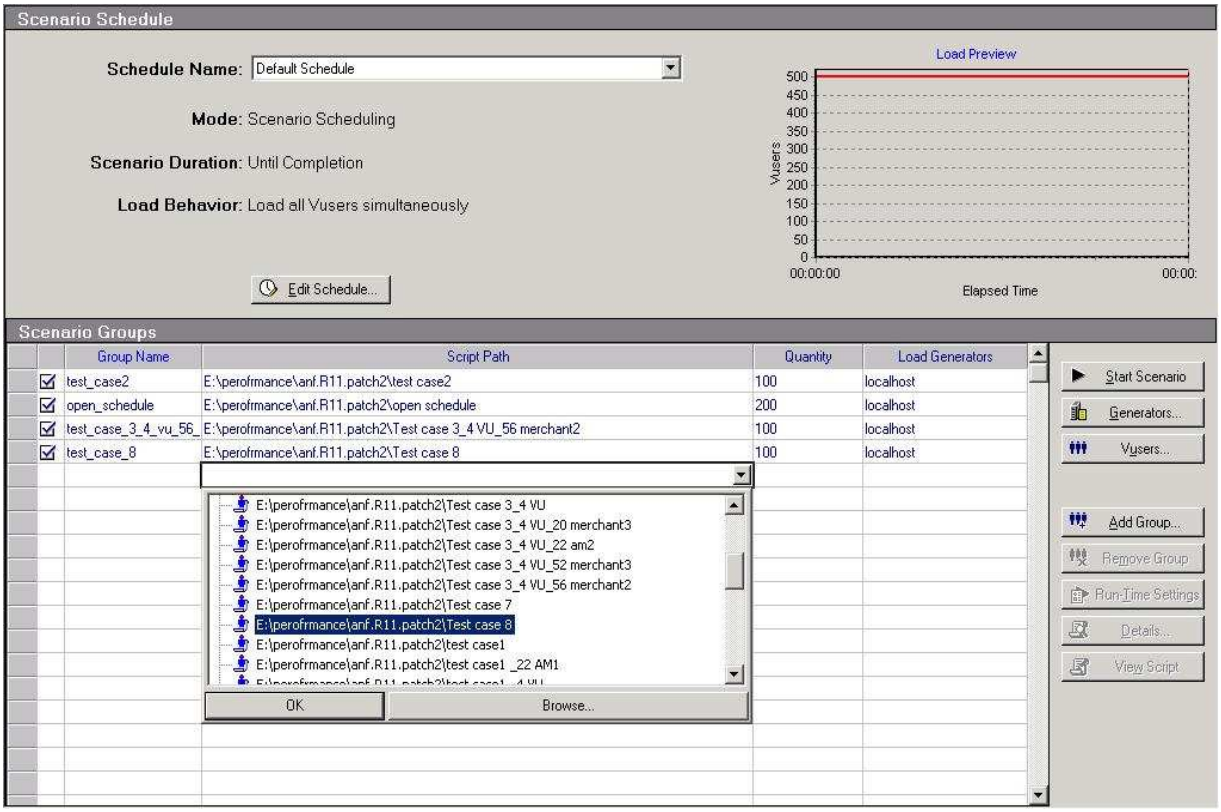


1. Manual Scenario：该项要完全手动的设置场景，这项下面还可以设置为每一个

脚本分配要运行的虚拟用户的百分比，可在Controller的Scenario菜单下设置。

2. Goal—Oriented Scenario：如果你的测试计划是要达到某个性能指标，比如：每秒多少点击，每秒多少transactions，能到达多少VU，某个Transaction在某个范围VU（500－1000）内的反应时间等等，那么就可以使用面向目标的场景。

选择手工场景，添加脚本到场景中，

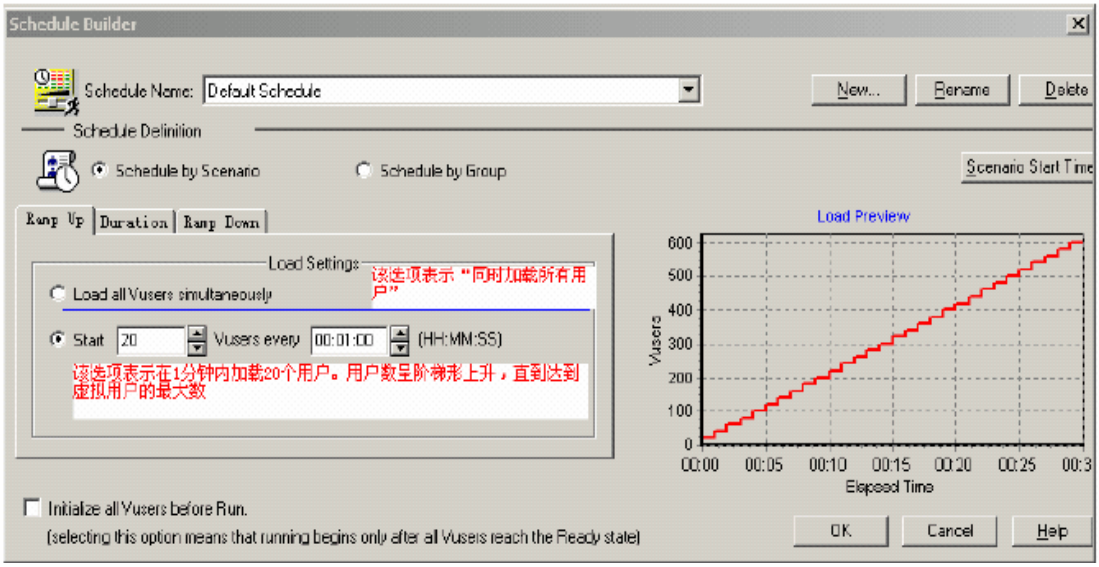


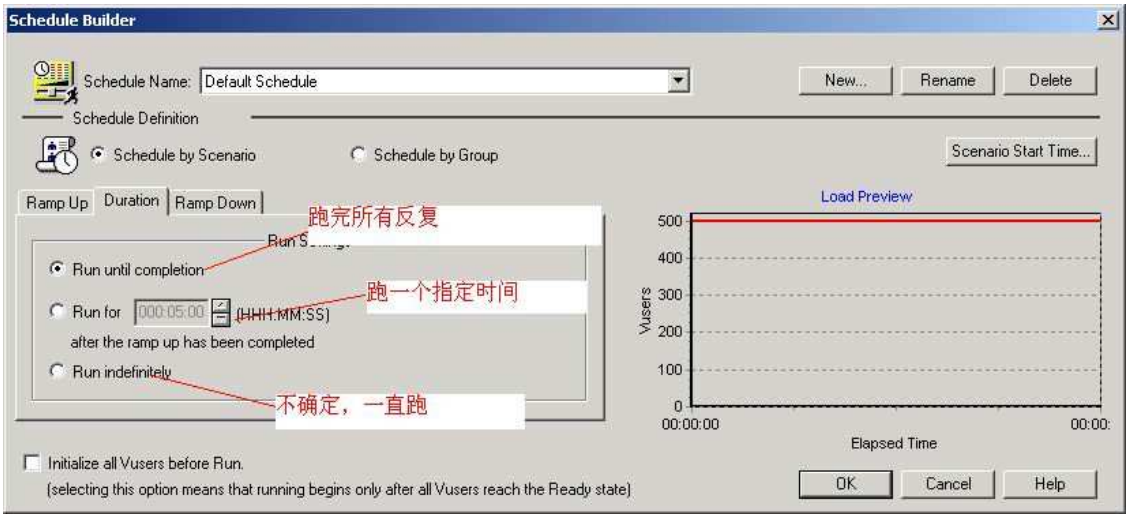
3.3 场景的设置

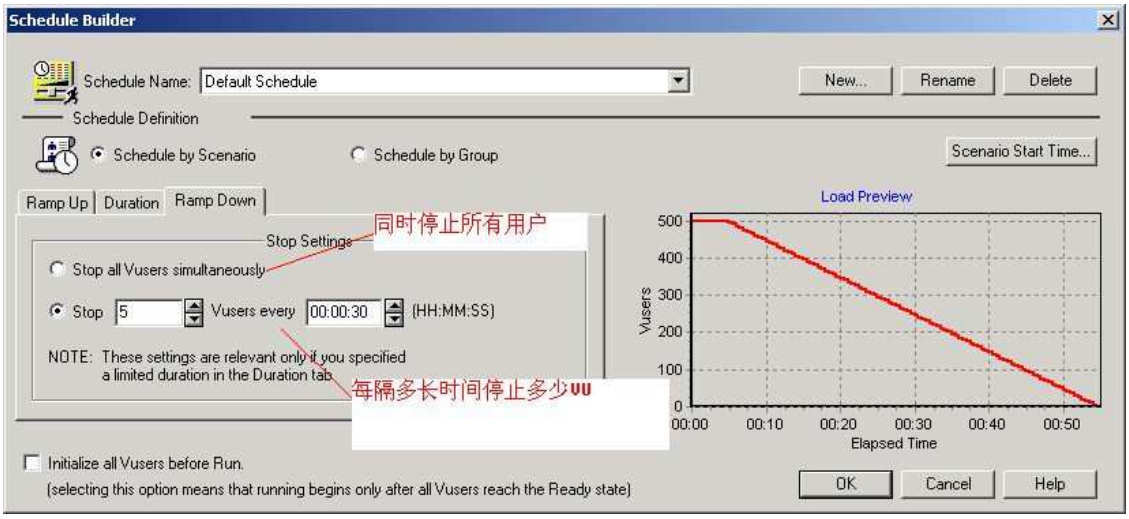
Design：设计测试场景的静态部分，设置模拟用户生成器、模拟用户数量、 模拟用户组等。

Run：设计测试的动态部分，主要指添加性能计数器，在脚本运行的过程中可以通过这些计数器反馈的数据。

建立了测试场景后，我们可以对 Edit Schedule 进行设置，设置测试开始执行的时间，对于手动设计的测试还可以设定它的持续时间，以及何时起用或禁止调用模拟用户。

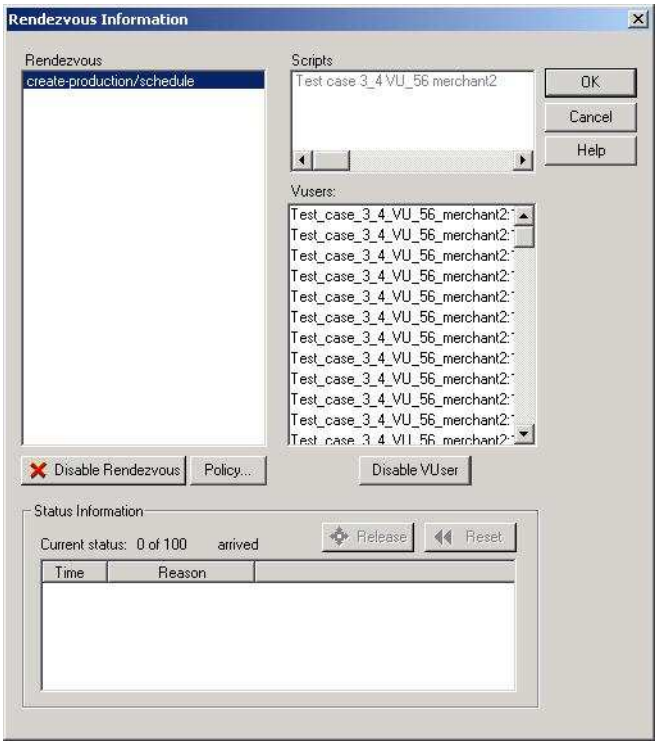

Schedule Builder
Schedule Name: Default Schedule
New...
Rename
Delete
Schedule Definition
Schedule by Scenario
Schedule by Group
Scenario Start Time...
Ramp Up
Duration
Ramp Down
Load Settings
Load all Vusers simultaneously
该选项表示“同时加载所有用户”
Start 20 Vusers every 00:01:00 (HH:MM:SS)
该选项表示在1分钟内加载20个用户。用户数呈阶梯形上升，直到达到虚拟用户的最大数
Load Preview
Vusers
Elapsed Time
Initialize all Vusers before Run.
(selecting this option means that running begins only after all Vusers reach the Ready state)
OK
Cancel
Help


Schedule Builder
Schedule Name: Default Schedule
New...
Rename
Delete
Schedule Definition
Schedule by Scenario
Schedule by Group
Scenario Start Time...
Ramp Up
Duration
Ramp Down
跑完所有反复
Run until completion
跑一个指定时间
Run for 000:05:00 (HHH:MM:SS)
after the ramp up has been completed
Run indefinitely
不确定，一直跑
Load Preview
Vusers
Elapsed Time
Initialize all Vusers before Run.
(selecting this option means that running begins only after all Vusers reach the Ready state)
OK
Cancel
Help

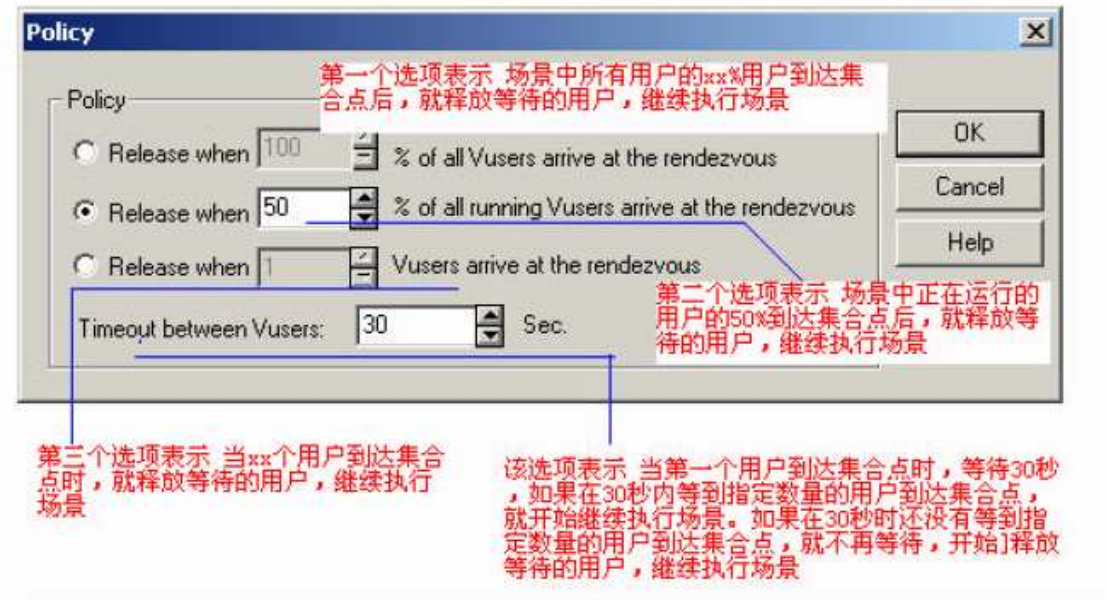


3.3.1 设置集合点

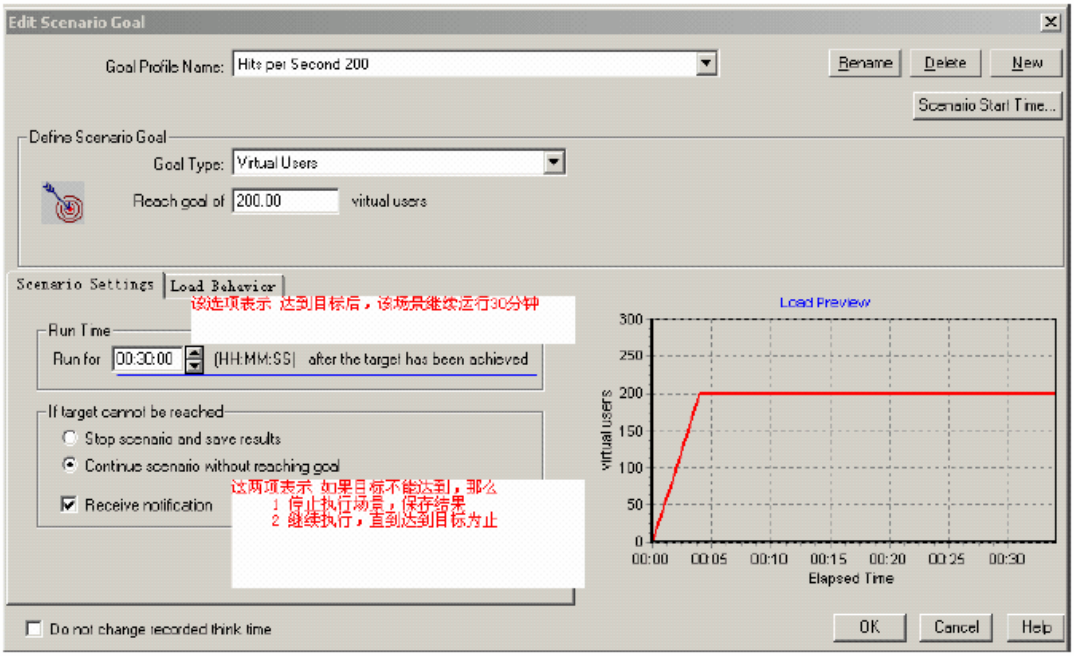
如果在脚本中设置了集合点，还需要在Controller中设置集合点策略。在菜单中调出设置集合点策略的窗口.，脚本运行的时候，在这里可以看到VU的状态。



3.3.2 设置集合点策略



如果选择场景类型为Goal—Oriented Scenario，添加脚本到场景中，它的场景设置是这样的



（以上的说明是以选择的目标为Virtual Users 时的）

各种目标类型的应用范围：

Virtual Users Goal

如果需要测试多少人可以同时运行Web 应用，那么推荐定义Virtual Users Goal。运行定义该目标类型的场景和运行Manual 类型的场景类似。

Hits per Second

如果想测试Web Server 的真正实力，推荐定义目标类型为：Hits per Second、Pages perMinute 或者Transactions per Second，这些类型都需要指定一个虚拟用户的最小值和最大值的范围。

Controller 试图使用最少的虚拟用户来达到定义的目标。如果使用最少的用户，不能达到目标，Controller 会增加用户数，直到达到定义的最大值。如果使用了最多的虚拟用户数，定义的目标还没有实现，那么需要增加最大用户数，重新执行场景。

Transactions per Second

可以选择一个在脚本里面已经定义好的事务

Transactions Response Time

如果想知道在多少用户并发访问网站时，事务的响应时间达到性能指标说明书中规定响
应时间的最大值，那么推荐使用Transactions Response Time 类型。指定需要测试的事务的名
称，虚拟用户数量的最小值和最大值，还有预先定义好的事务的响应时间。

在场景运行中，如果使用了最多的虚拟用户，还不能达到定义的最大响应时间，说明Web Server 还有能力接纳定义的虚拟用户的最多数量；如果在使用了部分虚拟用户，就达到了定义的最大的响应时间，或者LoadRunner 提示如果使用最多数量的虚拟用户时将要超过最大响应时间，那么需要重新设计或者修补应用程序，同时可能需要升级Web Server 的软硬件。

Pages per Minute

每分钟多少页面

如果你定义的类型是Pages per Minute、Hits/Transactions per Second，Controller 首先用最小用户数除以定义的目标，得到一个值，然后确定每个用户应该达到的hits/transactions或者pages per minute，然后controller 开始按照以下的策略加载用户：

- 如果选择的是自动的加载虚拟用户，LoadRunner 会首先加载50 个用户。如果定义的最
大用户数小于50，LoadRunner 就会一次加载所有的虚拟用户。
- 如果选择的是在场景运行一段时间后达到目标，LoadRunner 就会尝试在定义的这段时
间内达到目标，根据时间限制和计算出的每个用户的hits、transactions 或者pages，

LoadRunner 确定第一批加载多少用户。

- 如果选择的是按照一定的阶段达到目标（也就是先在x 长时间内达到y pages/hits，然后再达到下一个目标），LoadRunner 计算每个用户应该达到的数字后，再确定第一批加载多少用户。每加载一批用户后，LoadRunner 会判断是否达到这批用户的目标。如果这批用户的目标没有达到，LoadRunner 重新计算每一个用户应该达到的目标数字后，重新调整下一批加载用户的数量。默认情况下，LoadRunner 每两分钟加载一批用户。如果Controller 加载了最多数量的用户还没有达到预定的目标，LoadRunner 会重新计算每个用户的目标，然后同时运行最大数量的用户，尝试达到预定的目标。

如果出现以下情况，Pages per Minute、Hits/Transactions per Second 类型的场景会置于“Failed”状态：

- Controller 使用了指定的最大数量的用户，并且两次都没有达到目标
- 所有的用户运行都失败
- 没有足够的Load Generators 机器（现有的机器已经超载运行的情况下）
- Controller 增加了几批用户后，pages per minute 或者hits/transactions per second 没有增加
- Controlller 加载第一批用户后，定义的目标没有被捕捉到

3.3.3 这里介绍一下多机联合产生负载

通过 Generator 我们可以设定生成模拟用户的机器，这些机器需要安装 LR Generator，启动 Agent 进程，不需要脚本，只是调用它的资源。这些机器可以是实际存在的，也可是通过模拟 IP 模拟的机器。模拟 IP 通过 IP Wiazrd 工具生成，在上图中点击 Add，可以添加模拟的生成器，Connect 即可。

3.3.4 LR 对服务器资源的监视

LR只能监视它支持的服务器的资源，它支持大部分常见的服务器。

System Resource：包括windows平台，Unix平台等

Web Server：包括Apache、IIS、Sun的iplanet等

Application server：包括Weblogic、WebSphere等

Database server：包括DB2，Oracle，Sql server，Sybase等

Java： ejb，J2ee等，需要一个ejbdetector.jar文件

1.对Windows（Win2k server）的监视：

对windows的监视相对比较简单,监视前首先需要用有管理员权限的帐号连接被监server，例如：net use \\qa-test /user:donny ,输入密码。然后就可以添加计数器，比较常用的计数器有：

Memory：Available Mbytes 物理内存的可用数（单位 Mbytes）至少要有10% 的物理内存值

Processor：%Processor Time CPU 使用率。这是查看处理器饱和状况的最佳计数器。显示所有 CPU 的线程处理时间。如果一个或多个处理器的该数值持续超过 90%，则表示此测试的负

载对于目前的硬件过于沉重。为多处理器服务器添加该计数器的 0 到 x 个实例。

Processor Queue Length：是指处理列队中的线程数，小于2。处理器瓶颈会导致该值持续大于2。

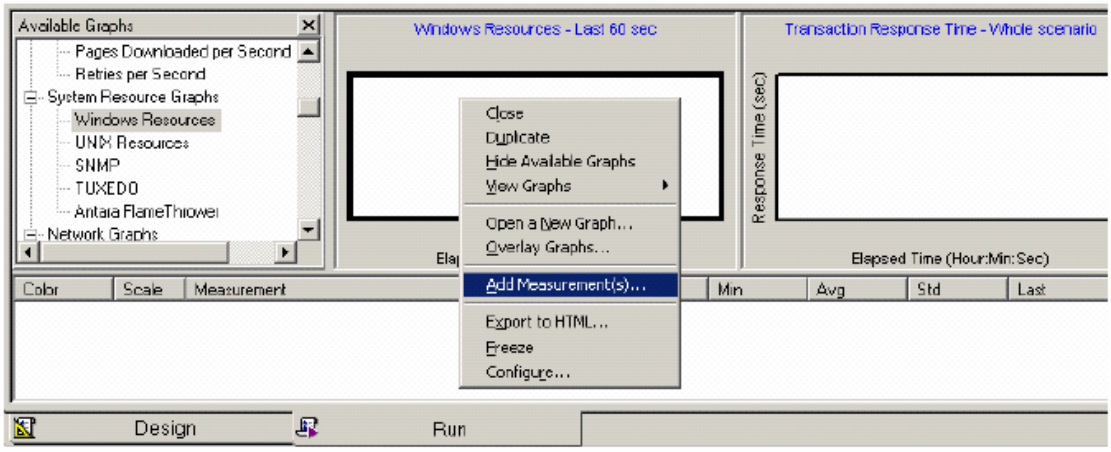
Context Switches/sec：如果切换次数到5000*CPU个数和10000*CPU个数中，说明它忙于切换线程

Network Interface：Bytes Total/sec 为发送和接收字节的速率，包括帧字符在内。判断网络连接速度是否是瓶颈，可以用该计数器的值和目前网络的带宽比较。

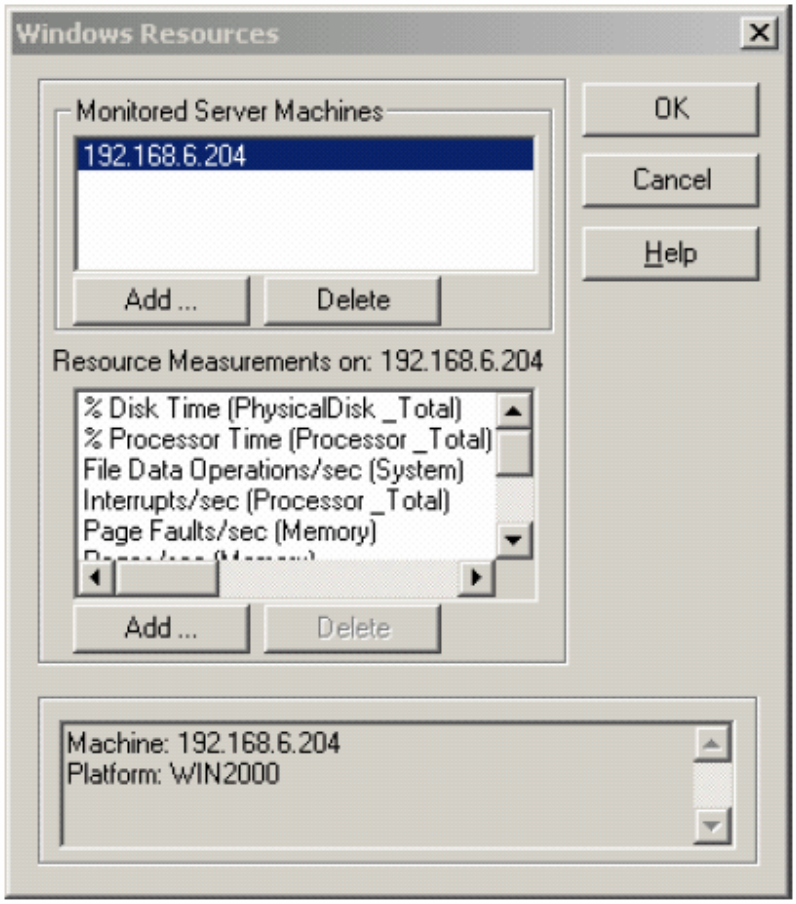
SQL Server2000：%Processor Time，CPU 使用率

General Statistics，Logins/sec，这是每秒登录到 SQL Server 的计数。

SQL Statistics： Batch Requests/sec，每秒收到的 Transact-SQL 命令批数。这一统计信息受所有约束（如I/O、用户数、高速缓存大小、请求每秒收到的 Transact-SQL 命令批数。这一统计信息受所有约束（如I/O、用户数、高速缓存大小、请求的复杂程度等）影响。批请求数值高意味着吞吐量很好。



然后，出现添加计数器的对话框



2.对Unix（Linux等）的监视，需要配置相应的服务器端，可以查看帮助文件，这里就只举一个例子了。

1） LoadRunner 如何监控Apache，需要修改apache的配置文件httpd.conf.

```
<Location /server-status>
SetHandler server-status
Order deny,allow
Allow from all
Allow from .your-domain.com
</Location>
```

把这节加在httpd.conf里面， restart apache即可。

第四部分：利用Analysis 分析结果

LR的报表分析功能也异常强大，有各种各样的报表，甚至可以将单个报表组合，也可以导出到Excel文件和Html文件。这里重点谈谈页面分解和报表组合。

4.1 页面分解

如果某个transaction的时间过长，为了分析问题出在哪里？就可以利用页面分解了，它可以把每个页面分解成：

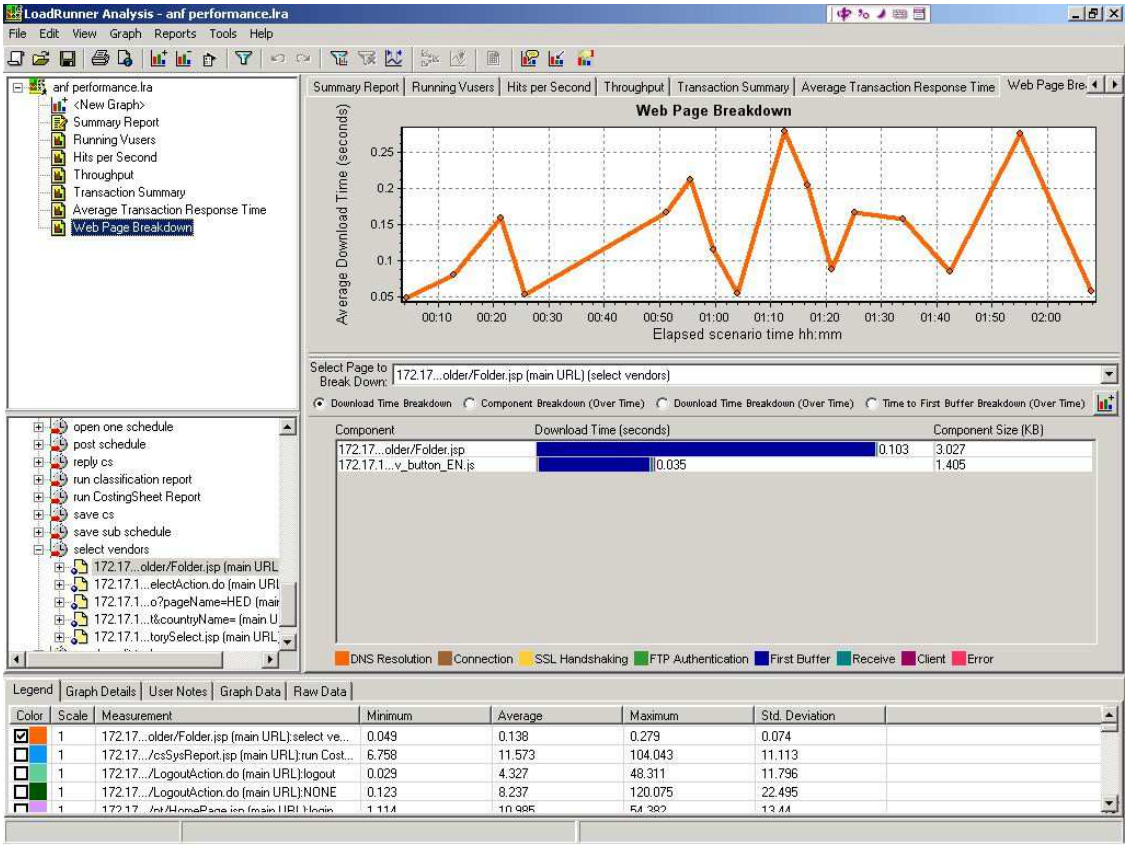
DNS解析时间：浏览器访问一个网站的时候，一般用的是域名，需要dns服务器把这个域名解析为IP，这个过程就是域名解析时间，如果我们在局域网内直接使用IP访问的话，就没有这个时间了。

Connection：解析出Web Server 的IP地址后，浏览器请求被送到了Web Server，然后浏览器和Web Server 之间需要建立一个初始化HTTP连接，服务器端需要做2件事：一是接收请求，二是分配进程，建立该连接的过程就是connection时间。

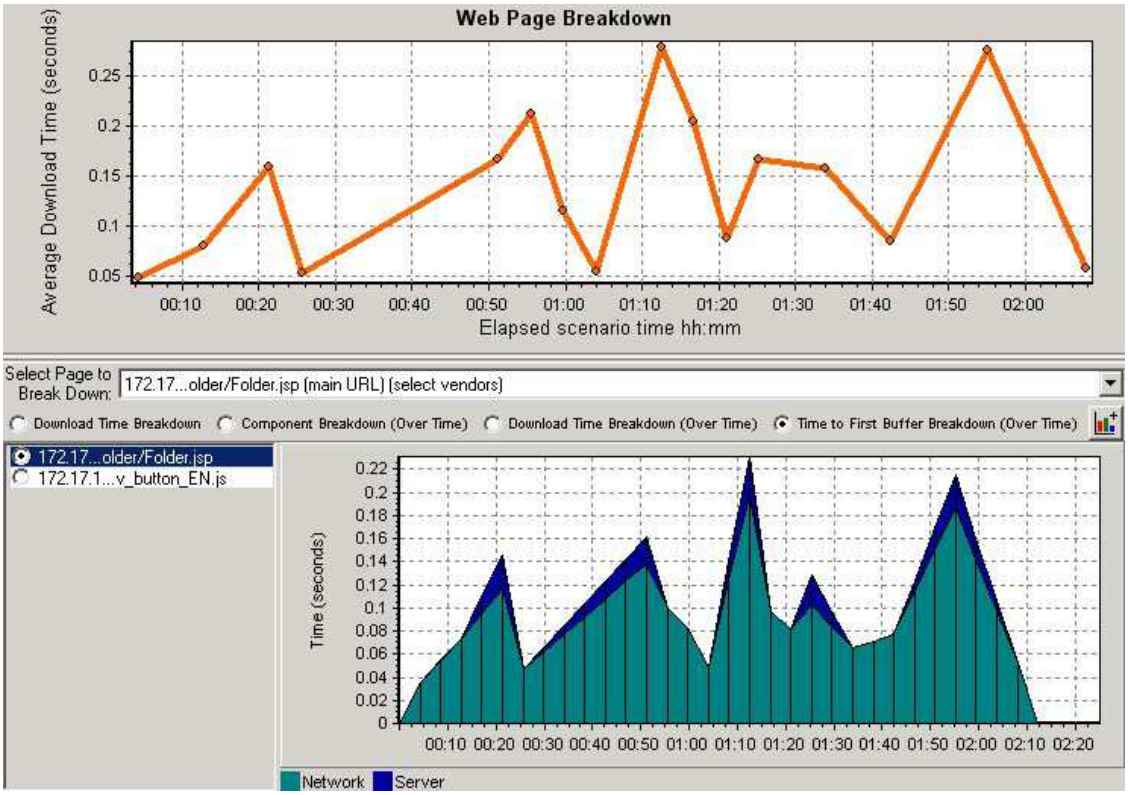
First Buffer：建立连接后，从Web Server 发出第一个数据包，经过网络传输到客户端，浏览器成功接受到第一字节的时间就是First Buffer。这个度量时间不仅可以表示Web Server 的延迟时间，还可以表示出网络的反应时间。

Receive：从浏览器接收到第一个字节起，直到成功收到最后一个字节，下载完成止，这段时间就是receive时间。

其他的时间还有SSL Handshaking（SSL 握手协议，用到该协议的页面比较少）、ClientTime（请求在客户端浏览器延迟的时间，可能是由于客户端浏览器的think time 或者客户端其他方面引起的延迟）、Error Time（从发送了一个HTTP 请求，到Web Server发送回一个HTTP 错误信息，需要的时间）



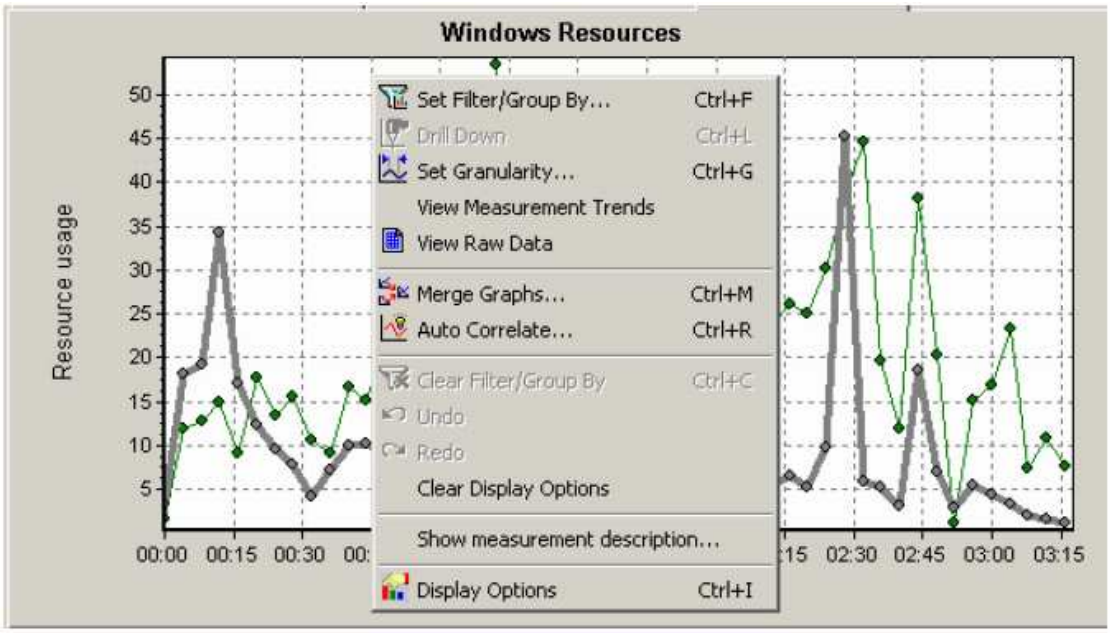
为了确认问题缘由到底是服务器还是网络，选择“Time to First Buffer Breakdown”，



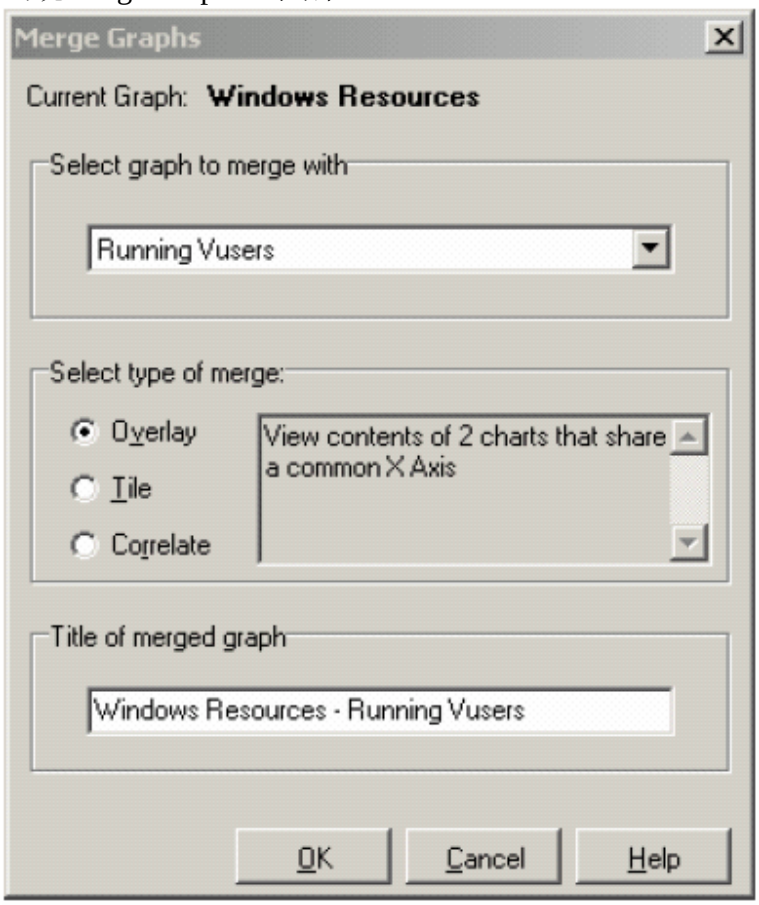
发现network时间比Server时间要高的多，从而确定问题是network引起的。

4.2 报表组合

Analysis 默认的图表都是以时间作为横坐标，然而在分析结果的过程中，我们可能需要以“运行的用户数”作为横坐标，来比较结果。假如我们要画出 Windows Resources ——VUsers 的图表，可以这样操作。首先打开Windows Resources 图表，然后在图表上点鼠标右键，选择Merge Graphs。



出现Merge Graphs 对话框



选择第一项“Overlay”，出现以下的图表，这样是把两个图表进行了合并，两条曲线的纵轴

共用一个原点，横轴还是时间轴。
选择第二项“Title”，出现以下的图表，这样是把两个图表进行了合并，两条曲线的纵轴不再共用一个原点，VUsers 的原点在Windows Resouces 的上面，横轴还是时间轴。
选择第三项“Correlate”，LoadRunner 提示信息