LoadRunner 压力测试实例

摘要：本文通过实例讲解介绍了 LoadRunner 工具的使用，介于公司的实际情况，文中主要是对工具的基本使用做了详细描述，高级运用方面除性能计数器与参数设置外其它均未涉及，待以后补充。目的是使公司人员根据该手册便可以独立运用 Loadrunner 进行压力测试

主题词：Loadrunner　工具　压力测试

1 LoadRunner 概要介绍

LoadRunner 是一种预测系统行为和性能的工业标准级负载测试工具。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自动负载测试工具， 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 能支持广范的协议和技术， 为您的特殊环境提供特殊的解决方案。

1.1 基本步骤

使用 LoadRunner 完成测试一般分为四个步骤：

1）Vvitrual User Generator 创建脚本

- ✧ 创建脚本，选择协议

- ✧ 录制脚本
- ✧ 编辑脚本
- ✧ 检查修改脚本是否有误

2）中央控制器（Controller）来调度虚拟用户

- ✧ 创建 Scenario，选择脚本
- ✧ 设置机器虚拟用户数
- ✧ 设置 Schedule
- ✧ 如果模拟多机测试，设置 Ip Spoofer

3）运行脚本

- ✧ 分析 scenario

4）分析测试结果

2 安装 LoadRunner 中文版

LoadRunner 分为 Windows 版本和 Unix 版本。如果我们的所有测试环境基于 Windows 平台， 那么我们只要安装 Windows 版本即可。本章讲解的安装过程就是 LoadRunner7.8 中文的 Windows 版本的安装。

2.1 系统要求

目前部门的测试机和工作机器足可以满足 LoadRunner7.8 的最低要求。不过要比较好的运行 LoadRunner， 内存最好在 512M 以上， 安装 LoadRunner 的磁盘空间至少剩余 500M。操作系统最好为 Windows 2000。

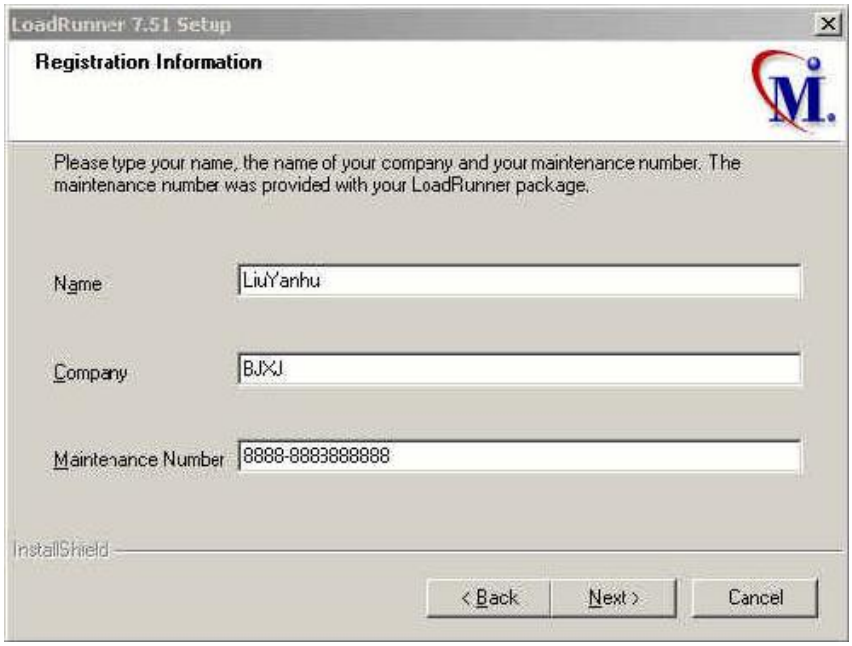
2.2 安装过程

LoadRunner7.8 中文版安装基本分两个步骤：首先安装 LoadRunner7.8 英文原版，然后安装中文语言插件包

LoadRunner7.8 英文原版存放位置：\\10.138.149.139\ test tools\LR7.8nt.rar 将压缩文件拷贝解压到本机的安装，过程比较简单要开始安装 LoadRunner，以 Administrator 的身份登陆 Windows2000 后，运行 LoadRunner 安装目录下 Setup.exe 即可进入安装程序。

1. 在“Registration Information” 界面中， 输入序列号（ 不用改动， 就是 n 个 8）

2. 在安装类型界面中， 选择一种安装类型



下面简单的对这三种安装类型进行介绍

●Standalone Installation 将要安装 LoadRunner 在一台计算机上

●Network Installation 把 LoadRunner 安装在一个网络驱动器上， 这样任何能连接到这个网络驱动器的计算机都可以使用 LoadRunner 的部分或者全部组件。

●Network Installation and shortcuts 和 Network Installation 类似，不同的只是这种类型将把自己的计算机配置成 Workstation 来运行 LoadRunner。如果选择了第二项， 我们还需要进行 2.3 的安装来配置 Workstation.。考虑到我们是自己学习研究学习， 选择第一种安装方法。

3. 在安装方式界面中， 需要选择一种安装方式。建议选择“ 自定义安装”， 这样所有的组件都会一次安装。

下面简单的对各个安装方式进行介绍

●Typical Installation 安装比较通用的组件， 包括 Controller、Vuser、在线帮助和脚该选项适合于控制 Vusers 的机器。

●Load Generator 只安装运行 Vusers 产生负载的组件。该选项适合于只产生负载，而不控制 Vusers 的机器。

●MI Listener 安装 MI Listener 组件， 用来透过防火墙来运行 Vusers 并且监视性能。

●Custom Installation 自定义安装， 我们将使用该选项， 安装全部的组件。

4. 在“License Information” 中输入 License Key 后，Next， 继续

100 个用户（无时间限制）：AEAMAUIK-YAFEKEKJJKEEA-BCJGI

10000 个用户（有时间限制）：AEABEXFR-YTIEKEKJJMFKEKEKWBRAUNQJU-KBYGB

5. 如果是网络安装，最好把网络驱动器映射成本机的一个盘符， 安装 LoadRunner 的各级目录不要包含中文字符。

6. Next 后进入拷贝文件的界面

7. 拷贝文件完成后， 进入“User Login Settings” 界面。

●Allow virtual users to run on this machine without user login 需要在下面输入域、用户名和密码， 这样运行 Load Generator 的机器会自动登陆到网络，

●Manual log in to the Load Generator machine 运行 Vusers 时， 自动登陆到网络，无需登陆用户名和密码， 这样 Vusers 就会不用任何干预自动的启动运行。推荐选择该项。这里选择第一项和第二项都可以。

8. 重新启动， 安装完成

LoadRunner7.8 英文原版存放位置：\\10.138.149.139\test tools\ LoadRunner7.8 中文版.rar 将压缩文件拷贝解压到本机的安装.。过程比较简单要开始安装以 Administrator 的身份登陆 Windows2000 后，（注意要退出已经运行的英文原版）运行安装目录下 Setup.exe 即可进入安装程序，安装过程中一切人机交流窗口多选择默认“下一步”即可

注意：解压文件存放的文件夹不可起中文名字，安装目录最好使用默认，如果更改则安装目录不要使用中文名！

3. 项目背景介绍

3.1 背景概述

“LMS 网校考试平台”是一个典型的三层 B/S 架构的 MIS 系统（客户端/应用服务器/数据库管），中间层是业务逻辑层，应用服务器处理所有的业务逻辑，但应用服务器本身不提供负载均衡的能力，而是利用开发工具提供的 ORB（对象请求代理）软件保证多个应用服务器间的负载均衡。本次测试的目的是：进行应用服务器的压力测试，找出应用服务器能够支持的最大客户端数。方法是：按照正常业务压力估算值的 1~10 倍进行测试，考察应用服

务器的运行情况。

3.2 压力测试用例

场景描述一：

1. 用户登录的 lmm 模块，总共登陆 24 个用户，所有用户都同时并发操作。
2. 用户点击“登记的教程”
3. 用户点击“启动”，进行课程学习，进入 DS 模块
4. 在 DS 模块中进行学习，过程包括：首先，点击一次课程结构树；然后，进行课程内容的学习。
5. 点击“返回 LMSò 按钮，返回到 lmm 模块，点击“退出”按钮，退出系统

场景描述二：

1. 用户登陆 lmm 模块，总共登录 48 个用户，每 1 秒登录 1 个用户
2. 用户点击“已登记教程”
3. 用户点击“启动”，进行课程学习，进入 DS 模块
4. 在 DS 模块中进行学习，过程包括：首先，点击一次课程结构树；然后，进行课程内容的学习；
5. 点击“返回 LMSò 按钮，返回到 lmm 模块，点击“退出”按钮，退出系统

场景描述三：

1. 用户登录的 lmm 模块，总共登陆 48 个用户，所有用户都同时并发操作。
2. 用户点击“登记的教程”
3. 用户点击“启动”，进行课程学习，进入 DS 模块
4. 在 DS 模块中进行学习，过程包括：首先，点击一次课程结构树；然后，进行课程内容的学习。
5. 点击“返回 LMSò 按钮，返回到 lmm 模块

点击“退出”按钮，退出系统

场景描述四：

1. 用户登录的 lmm 模块，总共登陆 48 个用户，每秒同时登录 10 个用户。
2. 用户点击“登记的教程”
3. 用户点击“启动”，进行课程学习，进入 DS 模块
4. 在 DS 模块中进行学习，过程包括：首先，点击一次课程结构树；然后，进行课程内容的学习。
5. 点击“返回 LMSò 按钮，返回到 lmm 模块，点击“退出”按钮，退出系统

场景描述五：

1. 用户登录的 lmm 模块，总共登陆 100 个用户，所有用户同时并发操作。
2. 用户点击“登记的教程”
3. 用户点击“启动”，进行课程学习，进入 DS 模块
4. 在 DS 模块中进行学习，过程包括：首先，点击一次课程结构树；然后，进行课程

内容的学习。

5. 点击“返回 LMSò 按钮，返回到 lmm 模块

场景描述六：

1. 用户登录的 lmm 模块，总共登陆 200 个用户，所有用户同时并发操作
2. 用户点击“登记的教程”
3. 用户点击“启动”，进行课程学习，进入 DS 模块
4. 在 DS 模块中进行学习，过程包括：首先，点击一次课程结构树；然后，进行课程内容的学习。
5. 点击“返回 LMSò 按钮，返回到 lmm 模块，点击“退出”按钮，退出系统

场景描述七：

1. 户登录的 lmm 模块，总共登陆 24 个用户。所有用户都同时并发操作
2. 所有用户都同时并发操作，户点击“登记的教程”中“test”课件

使用自发测试工具，目的测试 24 个用户同时打开课件时服务器性能

场景描述八：

1. 登录的 lmm 模块，总共登陆 60 个用户。所有用户都同时并发操作
2. 有用户都同时并发操作，户点击“登记的教程”中“test”课件

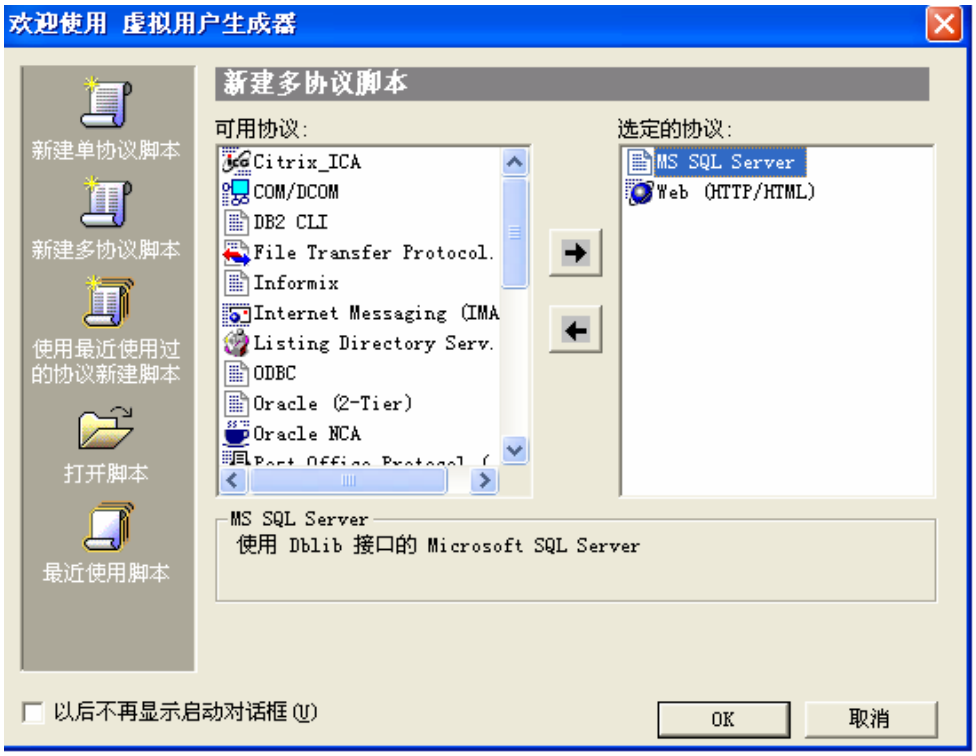
使用自发测试工具，目的测试 60 个用户同时打开课件时服务器性能

4. 使用 LoadRunner 进行负载/压力测试

4.1 录制基本的用户脚本

创建用户脚本需要用到 VuGen。提示： 运行 VuGen 最好在 1024*768 的分辨率下， 否则有些工具栏会看不到。

启动 Visual User Generator 后， 通过菜单新建一个用户脚本， 选择系统通讯的协议。这里我们需要测试的是 Web 应用，同时考虑到后台 SQL 数据库所以我们需要选择 Web(HTTP/HTML)协议＋SQL SERVER 协议，确定后， 进入主窗体。通过菜单来启动录制脚本的命令。

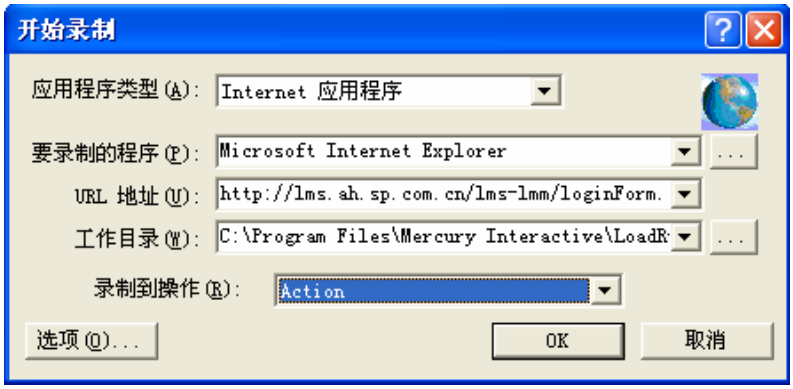


●在 URL 中添入要测试的 Web 站点地址..。

●测试 http://lms.ah.sp.com.cn/lms-lmm/loginForm.do 选择要把录制的脚本放到哪一个部分，默认情况下是“Action”。

这里简单说明一下：VuGen 中的脚本分为三部分：vuser_init、vuser_end 和 Action。其中 vuser_init 和 vuser_end 都只能存在一个， 不能再分割， 而 Action 还可以分成无数多个部分（ 通过点击 New 按钮， 新建 ActionXXX）。在录制需要登陆的系统时， 我们把登陆部分放到 vuser_init 中， 把登陆后的操作部分放到 Action 中， 把注销关闭登陆部分放到 vuser_end 中。（ 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） 在其他情况下， 我们只要把操作部分放到 Action 中即可。注意： 在重复执行测试脚本时，vuser_init 和 vuser_end 中的内容只会执行一次， 重复执行的只是 Action 中的部分。

作者：徐涛 msn： szxutao@hotmail.com



●点“ 选项 ”按钮， 进入录制的设置窗体， 这里一般情况下不需要改动。

●然后点“OK” 后，VuGen 开始录制脚本。在录制过程中， 不要使用浏览器的“ 后退”功能，LoadRunner 支持不太好！ 录制过程中， 在屏幕上会有一个工具条出现。录制的过程和 WinRunner 有些类似， 不再多介绍。录制完成后， 按下“ 结束录制” 按钮，VuGen 自动生成用户脚本， 退出录制过程。

4.2 完善测试脚本

当录制完一个基本的用户脚本后， 在正式使用前我们还需要完善测试脚本， 增强脚本的灵活性。一般情况下， 我们通过以下几种方法来完善测试脚本。插入事务、插入结合点、插入注解、参数化输入。这里只举例介绍参数化如何设置，其它只作简单介绍。

4.2.1 插入事务

事务（Transaction）： 为了衡量服务器的性能， 我们需要定义事务。比如： 我们在脚本中有一个数据查询操作， 为了衡量服务器执行查询操作的性能， 我们把这个操作定义为一个事务， 这样在运行测试脚本时，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 运行在脚本中插入不限数量的事务。

具体的操作方法如下： 在需要定义事务的操作前面， 通过菜单或者工具栏插入。输入该事务的名称。注意： 事务的名称最好要有意义， 能够清楚的说明该事务完成的动作。插入事务的开始点后， 下面需要在需要定义事务的操作后面插入事务的“ 结束点”。同样可以

通过菜单或者工具栏插入。默认情况下， 事务的名称列出最近的一个事务名称。一般情况下， 事务名称不用修改。事务的状态默认情况下是 LR_AUTO。一般情况下， 我们也不需要修改， 除非在手工编写代码时， 有可能需要手动设置事务的状态。

4.2.2 插入集合点

插入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。具体的操作方法如下： 在需要插入集合点的前面， 通过菜单或者工具栏操作输入该集合点的名称。注意： 集合点的名称最好要有意义， 能够清楚的说明该集合点完
成的动作。

4.2.3 插入注释

注释的作用就不多说了， 不过插入注释最好是在录制过程中。具体的操作方法如下： 在需要插入注释的前面， 通过菜单或者工具栏操作

4.2.4 参数化输入

如果用户在录制脚本过程中， 填写提交了一些数据， 比如要增加数据库记录。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。

用参数表示用户的脚本有两个优点：

① 可以使脚本的长度变短。

② 可以使用不同的数值来测试你的脚本。例如， 如果你企图搜索不同名称的图书， 你

仅仅需要写提交函数一次。在回放的过程中， 你可以使用不同的参数值， 而不只搜索一个特定名称的值。

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。

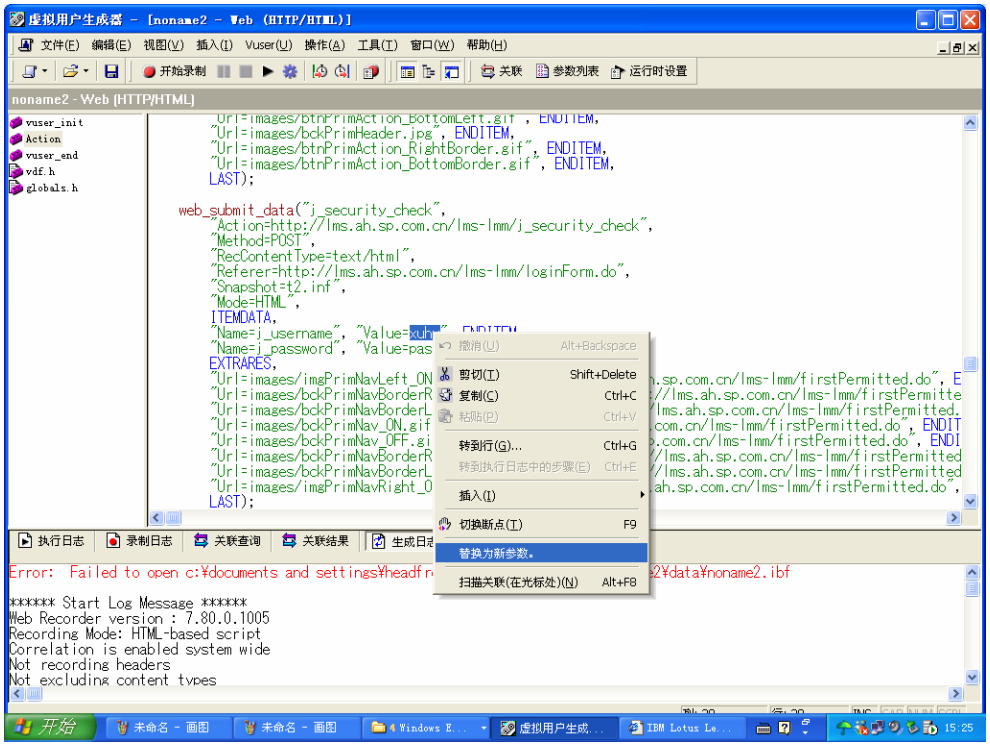
参数化仅可以用于一个函数中的参量。你不能用参数表示非函数参数的字符串。

另外， 不是所有的函数都可以参数化的。

参数化输入的讲解， 我们采用一个例子的方式来进行。

在本例中我们参数化用户的登陆名：

先看如下脚本，通过脚本录制找到用户登陆部分，如图



框选住登陆名，点鼠标右键，弹出对话框，选择“替换为新参数”弹出对话框

作者：徐涛 msn： szxutao@hotmail.com

参数名随意取，建议取通俗易懂的名字，下面我们重点介绍一下参数的类型。

●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。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（ 下面我们将会介绍）

●User Defined Function： 从用户开发的 dll 文件提取数据。就目前我认为， 这种方式没有必要。VuGen 支持 C 语言的语法，在 VuGen 中重新编写类似的函数应该不难。上面的例子中， 我们取随机数即可。点“Properties… ..” 按钮， 进行属性设置窗口添入随机数的取值范围为（1-50）， 选择一种数据格式。在“属性” 中有以下几个选项：

◆Each Occurrence：在运行时， 每遇到一次该参数， 便会取一个新的值

◆Each iteration：运行时， 在每一次循环中都取相同的值

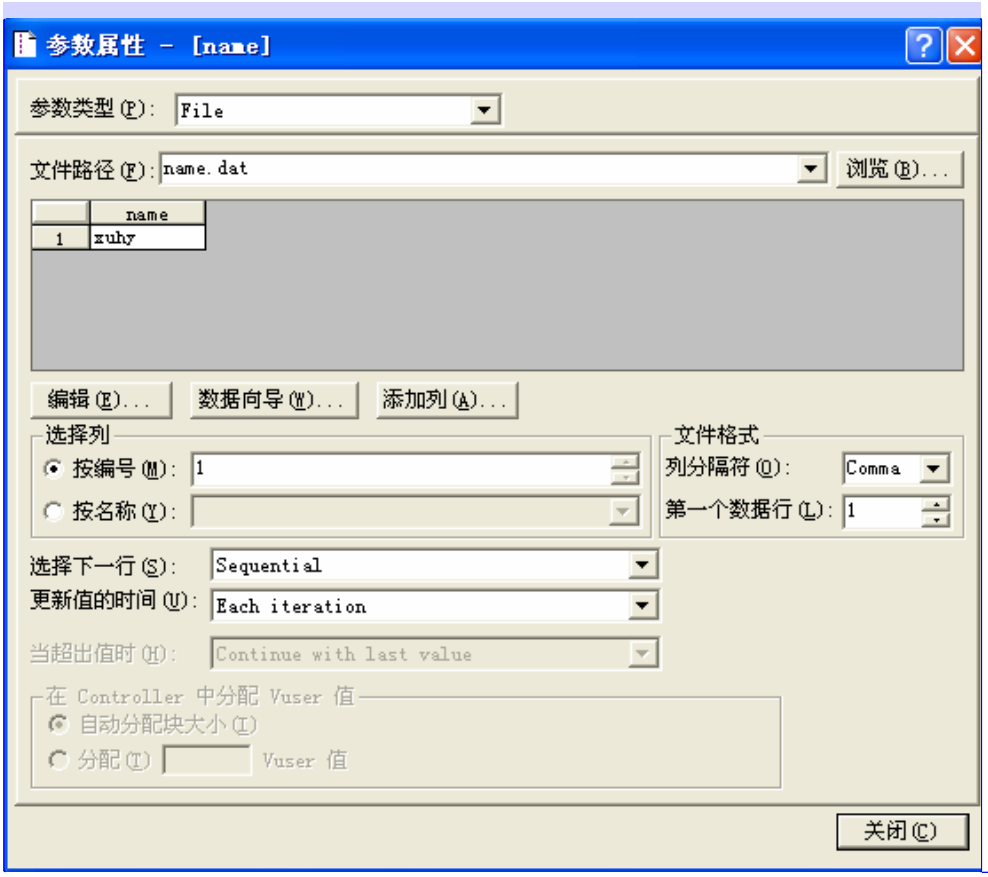
◆Once：运行时， 在每次循环中， 该参数只取一次值

这里我们用的是随机数， 选择 Each Occurrence 非常合适。

下面我们再介绍用数据库中的用户名来参数化登陆用户名。

框选住登陆名，点鼠标右键，弹出对话框，选择“替换为新参数”弹出对话框，此时参数名输入：name，参数类型选择 File，如图

点“属性”按钮， 出现以下窗口



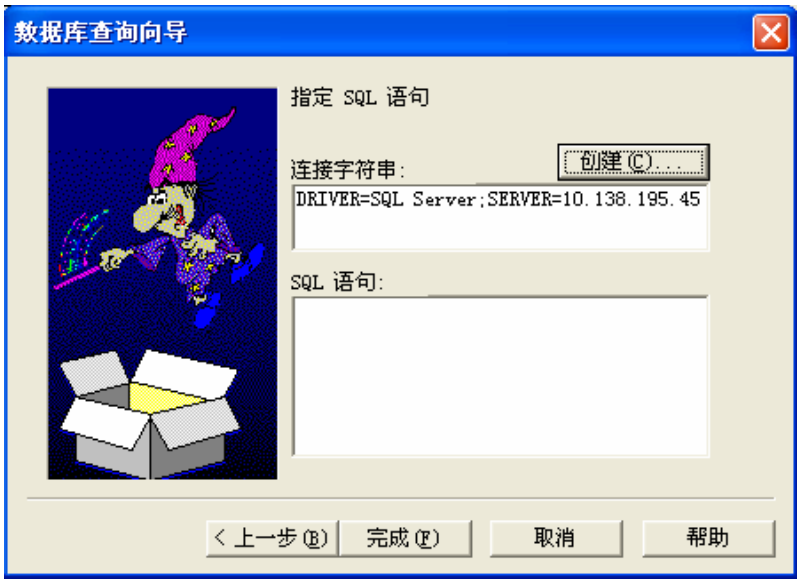
批注 [徐涛1]:

批注 [徐涛2R1]:

注意： 参数的文件名不要使用 con.dat、pm.dat 或者 lpt*.dat 等系统装置名下面我们将会连接数据库， 从数据表中选择用户名。点“数据向导” 按钮，显示如图

使用第 2 项， 选择“使用手动指定 SQL 语句”点下一步，出现如图窗口

批注 [徐涛3]:



添入连接字符串， 点“创建” 按钮，选择事先配置好的 ODBC 连接。在 SQL 语句里输入 select 查询语句，出现如图窗口

作者：徐涛 msn： szxutao@hotmail.com

提醒： 在参数数据显示区， 最多只能看到 100 行， 如果数据超过 100 行， 只能点“编辑” 按钮， 进入记事本看。

“选择下一行 ” 有以下几种选择：

●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 ： 唯一的数。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。

“按编号”指选择列表中的那一列数据，从左到右分别是 1、2、3 依次

通常用在有关联性的数据上面。我们这里取值 Sequential 即可。完成设置关闭即可

4.3 单机运行测试脚本

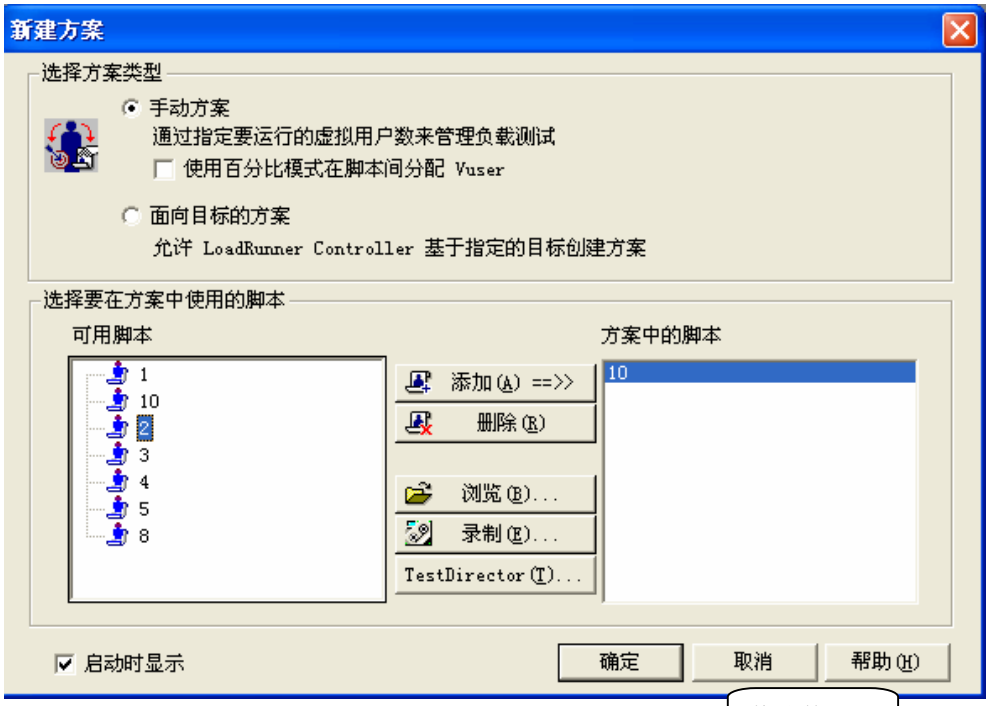
经过以上的各个步骤后，脚本就可以运行了。运行脚本可以通过菜单或者工具栏来操作。

执行“ 运行” 命令后，VuGen 先编译脚本， 检查是否有语法等错误。如果有错误，VuGen 将会提示错误。双击错误提示，VuGen 能够定位到出现错误的那一行。为了验证脚本的正确性， 我们还可以调试脚本， 比如在脚本中加断点等， 操作和在 VC 中完全一样， 相信大家谁都不会感到陌生。如果编译通过， 就会开始运行。然后会出现运行结果。

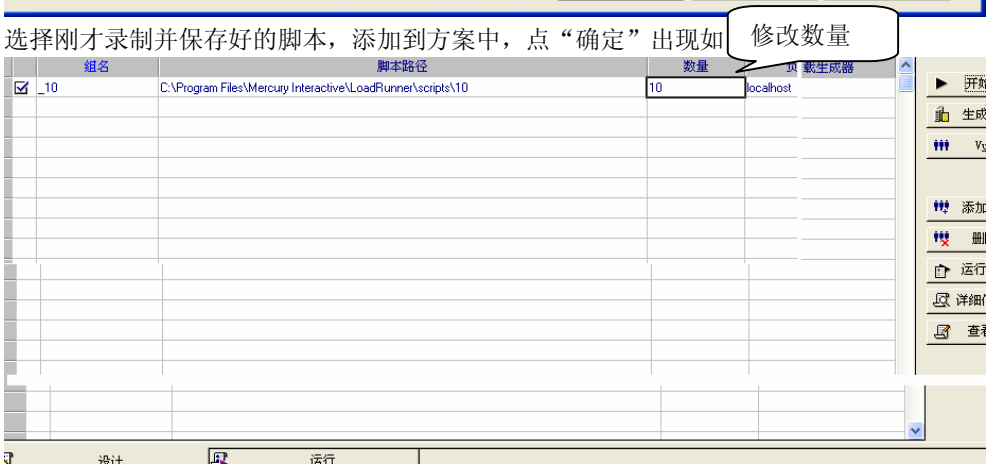
5 实施测试

5.1 选择脚本，创建虚拟用户

启用“controller”弹出如图窗口

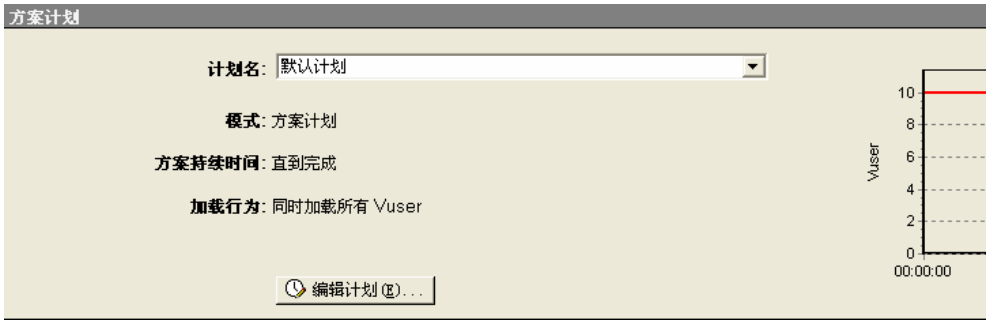


选择刚才录制并保存好的脚本，添加到方案中，点“确定”出现如



作者：徐涛 msn： szxutao@hotmail.com

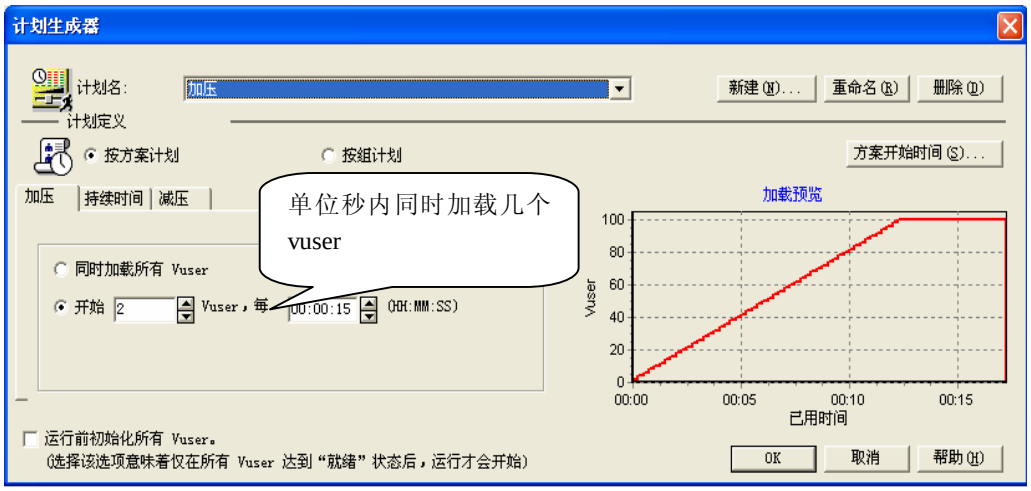
根据需要修改虚拟用户数量，这里我们取“100”根据实现场景设计，取不同数字



点“编辑计划”细化方案，计划名里选择计划种类：加压，缓慢加压、默认计划或新建立计划。

- ✧ 默认计划：同时加载所有 vuser，直到完成
- ✧ 加压：每 15 秒启动 2 个 vuser 持续时间 5 分种
- ✧ 缓慢加压：：每 2 分种启动 2 个 vuser 持续时间 10 分种

这里我们选择“加压” 出现如图



点“加压”标签设置加压方法，点“持续时间”标签选择完成时间，点“加压”标签选择退出方法，点“方案开始时间”可以定义时间后自动到点执行，并在一个限定的时间范围内结束，所有设置完毕后，点“ok”返回上一级窗口，点“开始方案”启动运行，出现如图窗口

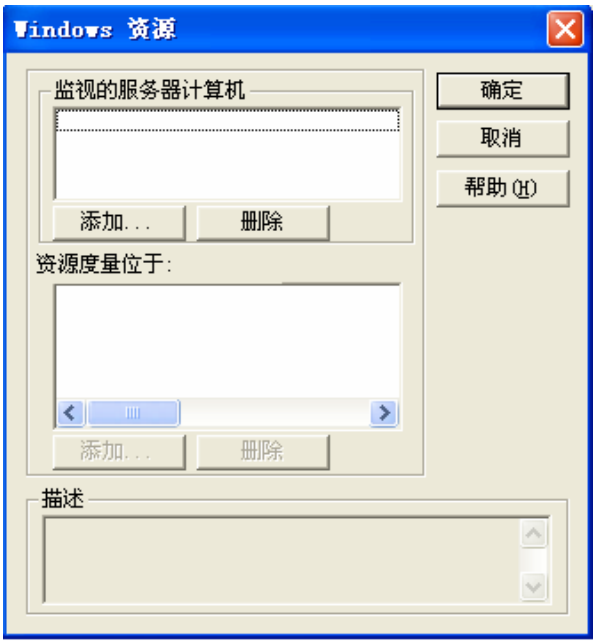
5.2 添加 windows 资源监视窗口

loadruner 默认性能监视窗口四个，分别是“运行 vuser“、”事务响应时间“、“每秒点击次数”最后一个可以根据用户自己选择现实什么窗口。打开可用图中目录树，选择系统资源，找到 windows 资源双击，则 windows 资源监视窗口便自动替换原窗口如上图。当然 loadrunner 也可以同时显示 1－16 个窗口，方法是点右键，在弹出菜单中选择“查看图”选择显示的图数，也可以自定义数字。

5.3 添加 windows 性能计数器

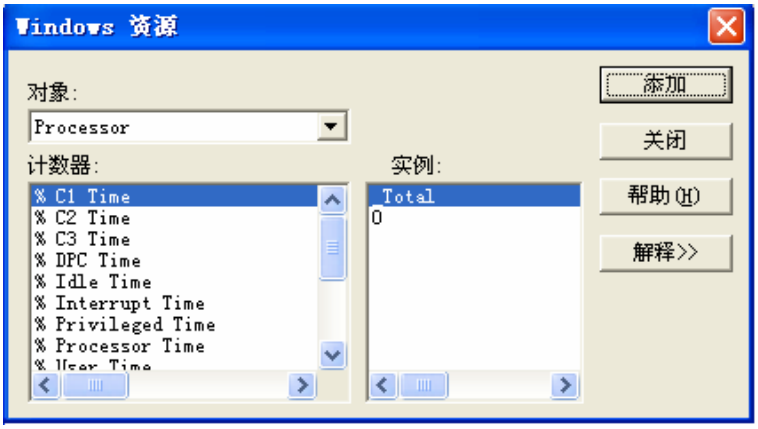
鼠标选择 windows 资源监视窗口，点击右键弹出菜单中选择“ADD Measurements..ò弹出如图窗口

作者：徐涛 msn： szxutao@hotmail.com



点“添加”把监视的服务器 ip 地址输入，点确定，如图

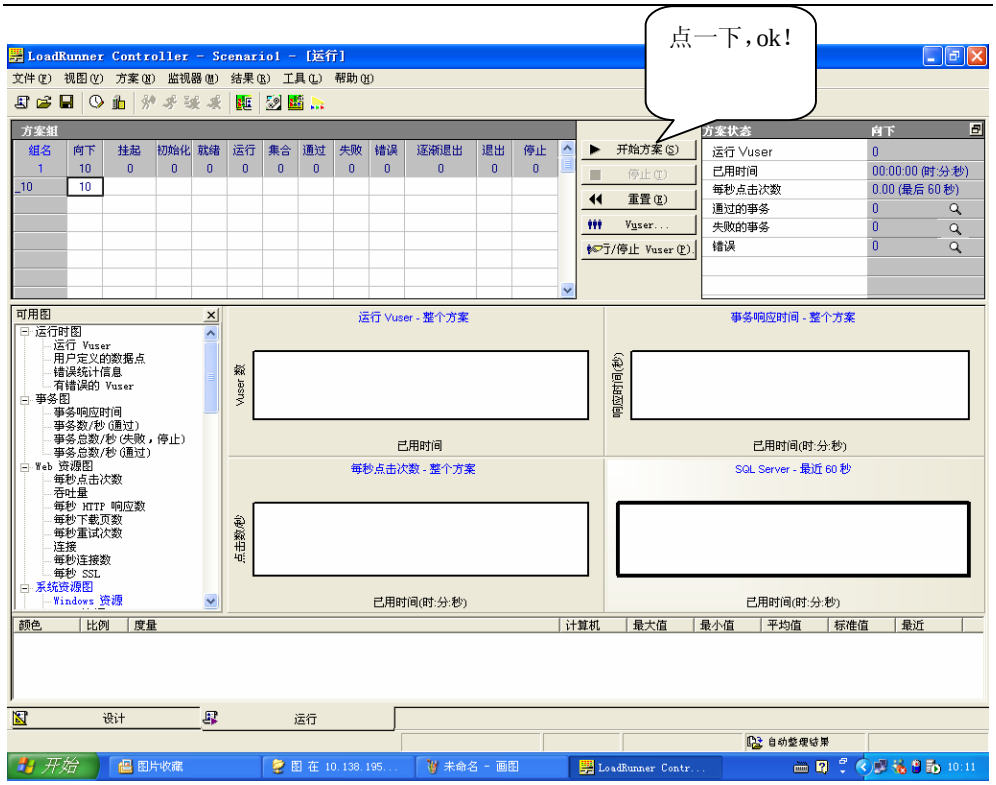
如果可以正常联机到服务器，则在资源度量中会显示全部计数器，此时如果点“确定”则系统默认全部选中，在监视窗口中会显示所有性能曲线，无法单独过滤显示某条曲线，如果选中某个计数器后点“添加”则弹出该项目下的其它性能指标，选择需要的计数器后点“添加”如图



此时要注意，你登陆客户端（也就是你装有 loadrunner 机器）的用户应该是管理员身份，同时还要保证该用户在被监视的服务器上也是管理员身份。这样选择虽然监视窗口中仍会显示所有性能曲线，但是可以通过鼠标右键弹出菜单，选中你指定的某条曲线单独显示。方法是双击监视窗口放大显示，然后右键选择“仅显示指定图”监视窗口还可以互相叠加等操作，功能强大，通过右键菜单选择可以进行复杂显示操作。常用的还有 web 程序服务器图、数据库服务器资源图等，添加方法雷同。计数器有那些，有什么含义，理想值是多少，可以参见第六章节。

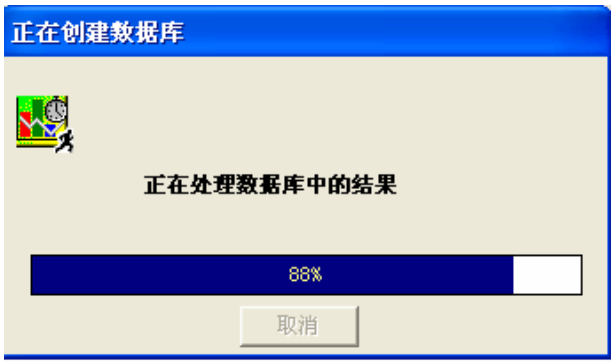
5.4 执行脚本

此时设置完毕后，那就简单了，点击“开始方案”注意观察吧。

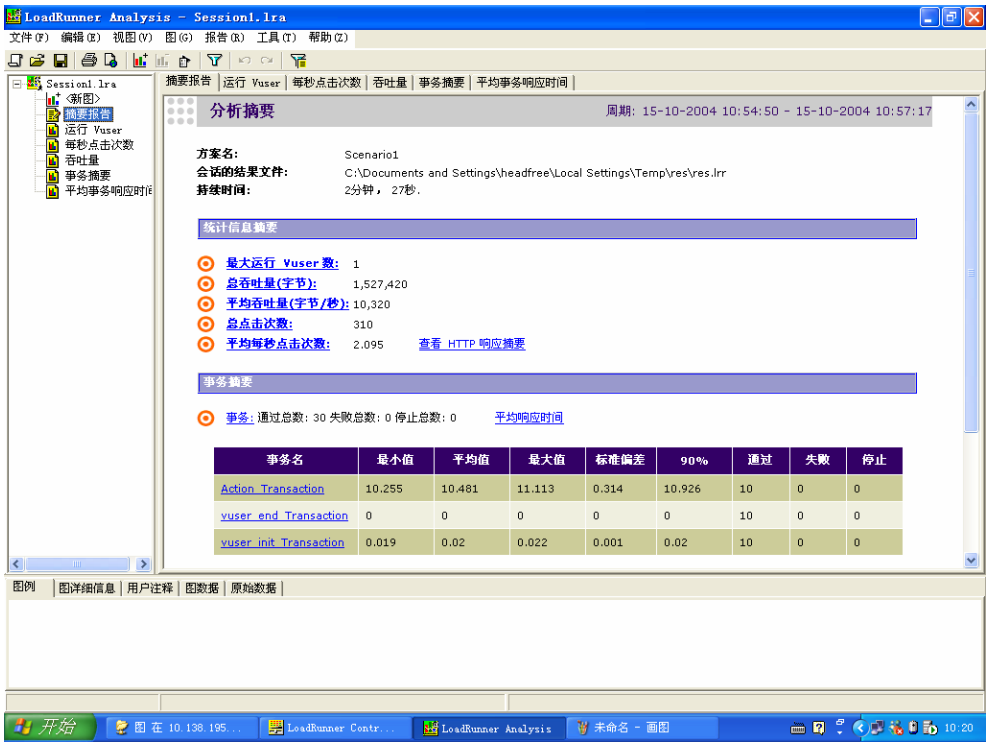


5.4.1 分析结果

脚本执行完毕后，loadrunner 会自动分析结果，生成分析结果图或表，方法是点导航栏“结果”选现，在弹出窗口中选择“分析结果”



作者：徐涛 msn： szxutao@hotmail.com



6 分析以及监视场景

在运行过程中， 可以监视各个服务器的运行情况（DataBase Server、Web Server 等）。监视场景通过添加性能计数器来实现。这一章非常的重要， 确定系统瓶颈全靠它了。下面重点讲讲需要添加那些计数器， 以及那些计数器代表什么意思。由于 Win2000 Professional、Server 以及 Advanced Server 提供的计数器不完全相同， 这里我们讨论将以 Server 为基准。监视场景需要在 Run 视图中设置然后， 出现添加计数器的对话框其他的操作就和控制面板“ 性能” 中添加性能计数器的操作一样， 这里不再详细说明。本章主要说明一下各个系统计数器的含义（ 数据库的计数器不做重点， 只是拿 SQL Server2000 作为例子进行说明。因为数据库各个版本之间差异比较大， 请参考您使用的数据库系统的帮助）。

6.1 Memory 相关

内存是第一个监视对象， 确定系统瓶颈的第一个步骤就是排除内存问题。内存短缺的问题可能会引起各种各样的问题。

作者：徐涛 msn： szxutao@hotmail.com

Object（ 对象）	Counters	Description（ 描述）	参考值
Memory	Available MBytes	物理内存的可用数（ 单位 Mbytes）。默认情况下 IIS5.0 使用 50%的可用物理内存， 作为 IIS 的文件缓存（file cache）。IIS 基本占用 2.5 MB，每个附加连接将在此基础上占用 10 KB 左右	至少要有 10%的物理
Memory	Page/sec Page Faults/sec Pages Input/sec Pages Input/sec Page Reads/sec Transition Faults/sec	物理内存的可用数（ 单位 Mbytes）。默认情况下 IIS5.0 使用 50%的可用物理内存， 作为 IIS 的文件缓存（file cache）。IIS 基本占用 2.5 MB，每个附加连接将在此基础上占用 10 KB 左右。至少要有 10% 的物理内存值当处理器向内存指定的位置请求一页（ 可能是数据或代码）出现错误时， 这就构成一个 Page Fault。如果该页在内存的其他位置， 该错误被称为软错误（ 用 Transition Fault/sec 数器衡量）； 如果该页必须从硬盘上重新读取时， 被称为硬错误。许多处理器可以在有大软错误的情况下继续操作。但是， 硬错误可以导致明显的拖延。Page Faults/sec 是处理器每秒钟处理的错误页（ 包括软错误和硬错误）。Pages Input/sec 是为了解决硬错误页， 从硬盘上读取的页数， 而 Page Reads/sec 是为了解决硬错误，从硬盘读取的次数。如果 Page Reads/Sec 比率持续保持为 5， 表示可能内存不足。Pages/sec 是指为解析硬页错误从磁盘读取或写入磁盘的页数。	Page/sec 推荐 00-20（ 如果服务器没有足够的内存处理其工作负荷， 此数值将一直很高。如果大于 80，表示有问题）。这些计数器的值比较低，说明 Web 服务器响应请求比较快，否则可能是服务器系统内存短缺引起（ 也可能是缓存太大， 导致系统内存太少）。Page Input/sec 的值可以衡量出硬错误页发生的速率， 通常它的值会于或者等于 Page Reads/sec 。Memory Cache Bytes
Memory	Cache Bytes	文件系统缓存（File System Cache）	默默认情况下认情况下为 50%的可用物理内存。如为 50%的可 IIS5.0 运行内存不够时， 它会自动

			整理用物理内存缓存。需要关注该计数器的趋势变化
Internet File Cache Hits %		File Cache Hits %是文件缓存命中全部（ 对于一个 Information File Cache 缓存需求的比例， 反映了 IIS 的文件缓大部分是静 Services Flushes 存设置的工作情况。而 File Cache Hits 态网页组成 Global File Cache Hits 是文件缓存命中的具体值，File Cache 的网站）File Flushes 是自服务器启动之后文件缓存 Cache Hits% 刷新次数，如果刷新太慢， 会浪费内存； 如果刷新太快， 缓存中的对象会太频繁属于非常好！ 的丢弃生成， 起不到缓存的作用。通过 File Cache Hits 和 File Cache Flushes 可以得到一个适当的刷新值（ 参考 IIS 的设置 ObjectTTL 、MemCacheSize 、MaxCacheFileSize）	
Memory	PoolPaged BytesPool Nonpaged Bytes	Pool Paged Bytes Pool Nonpaged Bytes 这两个计数器监视服务器上各个进程的分页池字节数和非分页池字节数。	在访问数比较固定的情况下， Pool Nonpaged Bytes 是比较定的， 如果访问数逐步增加， 该值会缓慢的增加
Process	Virtual Bytes Working Set 计数器	Virtual Bytes(实 Virtual Bytes 数器监视 IIS5.0 保留的例 inetinfo 、虚地址空间的数量，实例化为 inetinfo dllhost) Working Set（ 实例进程(IIS 运行的核心)和 Dllhost 进程（ 隔离 / 连接池的应用程序必需的）。inetinfo 、dllhost） Working Set 计数器反映了每个进程使 Dllhost#n 进程都用的内存页的数量。系统的内存页（pool 要添加计数器 Page）	

		只能由操作系统的核心模块直接访问，用户进程不能访问。运行 IIS5.0 的服务器上， 负责 web 连接的线程以及它需要的一些对象都保存在未分页的池中（nonpaged pool）， 比如文件句柄和 socket 连接	
Process	Private Bytes	指这个处理不能与其他处理共享的、已分配的当前字节数	
Memory	Committed Bytes	是指以字节表示的确认虚拟内存。(确认内存是指为磁盘分页文件在磁盘上保留的空间以便在需推荐不超过物理内存的 75% 要将其写回磁盘时使用)	推荐部超过物理内存的 75%

内存问题主要检查应用程序是否存在内存泄漏。如果发生了内存泄漏，Process\Private Bytes 计数器和 Process\Working Set 计数器的值往往会升高， 同时 Available Bytes 的值会降低。内存泄漏应该通过一个长时间的， 用来研究分析当所有内存都耗尽时， 应用程序反应情况的测试来检验。

6.2 Processor 相关

Object（ 对象）	Counters	Description（ 描述）	参考值
Sytem	Processor Queue Length	Processor Queue Length 是指处理列队中的线程数。即使在有多个处理器的计算机上处理器时间也会有一个单列队。不象磁盘计数器， 这个计数器仅计数就绪的线程，而不计数运行中的线程。如果处理器列队中总是有两个以上的线程通常表示处理器堵塞	小于 2。显示在由 Web 服务器所有处理器共享的队列中等待执行的线程数。处理器瓶颈会导致该值持续大于 2
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	小于 75%。排除内存因素， 如果该计数器的值比较大， 而同时网卡和硬盘的值比较低， 那么可以定 CPU 瓶颈

		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	
System	Context Switches/sec	Context Switches/sec 指计算机上的所有处理器全都从一个线程转换到另一个线程的综合速率。当正在运行的线程自动放弃处理器时出现上下文转换， 由一个有更高优先就绪的线程占先或在用户模式和特权(内核)模式之间转换以使用执行或分系统服务。它是在计算机上的所有处理器上运行的所有线程的 Thread: Context Switches/sec 的总数并且用转换数量衡量。在系统和线程对象上有上下文转换计数器	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而不是处理 ASP 脚本
Processo	%Privileged Time	% Privileged Time 是在特权模式下处理线程执行代码所花时间的百分比。当调用 Windows 系统服务时， 此服务经常在特权模式运行， 以便获取对系统专有数据的访问。在用户模式执行的线程无法访问这些数据。对系统的调用可以是直接的(explicit)或间接的(implicit)， 例如页面错误或中断。不像某些早期的操作系统，Windows 除了使用用户和特权模式的传统保护模式之外，还使用处理边界作为分系统保护。某些由 Windows 为您的应用程序所做的操作除了出现	

		在处理的特权时间内，还可能在其他子系统处理出现	
Time	Switches/sec （实例化 inetinfo 和 dllhost	如果你决定要增加线程字节池的大小，你应该监视这三个计数器（包括上面的一个）。增加线数可能会增加上下文切换次数，这样性能不会上升反而会下降。如果十个实例的上下文切换值非常高，就应该减小线程字节池的大小	
Processor	Interrupts/sec %DPC Time	Time 这两个计数器能够反映处理器用在处理中断以及推迟处理调用的时间。如果处理器使用率超过 Interrupts/sec 指处理器每秒钟接收并维 90% 且 硬件中断的平均值。正常的线程操作在中断时悬停。大多数的系统时钟每 Interrupt Time 大于隔 10 毫秒中断处理器一次，形成了间 15%，则处理隔活动的后台	如果处理器使用率超过 90％，且 Interrupts/sec time 大于 15％则处理器可能负载过重，并发生中断

Processor Interrupts/sec %DPC Time 这两个计数器能够反映处理器用在处理中断以及推迟处理调用的时间。如果处理器使用率超过 Interrupts/sec 指处理器每秒钟接收并维 90% 且 硬件中断的平均值。正常的线程操作在中断时悬停。大多数的系统时钟每 Interrupt Time 大于隔 10 毫秒中断处理器一次，形成了间 15%，则处理隔活动的后台。器可能负荷过重，并发生中断。判断应用程序是否存在处理器瓶颈的方法：如果 Processor Queue Length 显示的队列长度保持不变（>=2）个并且处理器的利用率%Processor Time 超过 90%，那么很有可能存在处理器瓶颈。

如果发现 Processor Queue Length 显示的队列长度超过 2，而处理器的利用率却一直很低，那么或许更应该去解决处理器阻塞问题，这里处理器一般不是瓶颈。如果系统由于应用程序代码效率低下或者系统结构设计有缺陷而导致大量的上下文切换（Context Switches/sec 显示的上下文切换次数比较大），那么就会占用大量的系统资源。如果系统的

吞吐量降低并且 CPU 的使用率很高，并且此现象发生时切换水平在 15000 以上， 那么意味着上下文切换次数过高同时还可以比较 Context Switches/sec 和%Privileged Time 来判断上下文切换是否过量。如果后者的值超过 40%， 且上下文切换的速率也很高， 那么应该检查为什么会产生这样高的上下文切换。

6.3 网络吞吐量以及带宽

Object	Counter	Description	参考值
Network Interface	Bytes Total/se	Bytes Total/sec 为发送和接收字节的速率， 包括帧字符在内。判断网络连接速该计数器的值和目前网度是否是瓶颈， 可以用该计数器的值和络的带宽相目前网络的带宽比较	改计数器的值和目前网络带宽相除，结果应该小于 50%
Web Servic	Maximum Maximum Connections	Maximum Maximum Connections ：“ 最大连接数” Attempts Total Connection Attempts ：“ 连接尝试总数”是从服务启动时利用 Web 服务尝试连接的总数。该计数器应用于全部所列的实例。	

6.4 磁盘相关

Object（ 对象） Counters（ 计数器名称） Description（ 描述） 参考值

Object	Counters	Description	参考值
Network	Bytes Total/sec	Bytes Total/sec 为发送和接收字节的速 Interface 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	

作者：徐涛 msn： szxutao@hotmail.com

Processo	%Processor Time % Privileged Time	CPU 使用率该计数器对应于处理器执行 Windows. 2000 内核命令（如处理 SQL Server I/O 请求） 所用时间的百分比。如果 Physical Disk 计数器的值很高时该计数器的值也一直很高， 则考虑使用速度更快或效率更高的磁盘子系统。	
PhysicalDisk	%Disk Time	% Disk Time 指所选磁盘驱动器忙于为读或写入请求提供服务所用的时间的百分比。如果三个计数器都比较大， 那么硬盘不是瓶颈。如果只有%Disk Time 比较大，另外两个都比较适中，硬盘可能会是瓶颈。在记录该计数器之前， 请在 Windows 2000 的命令行窗口中运行 diskperf -yD 。若数值持续超过 80%， 则可能内存泄漏。	
PhysicalDisk	AverageDisk Queue Length	指读取和写入请求(为所选磁盘在实例间隔中列队的)的平均数。	
PhysicalDisk	PhysicalDisk	指在此盘上读取操作的速率	
PhysicalDisk	Disk Writes/sec	指在此盘上写入操作的速率	

判断磁盘瓶颈的方法是通过以下公式来计算：

每磁盘的 I/O 数 = [读次数 + （4 * 写次数）] / 磁盘个数

如果计算出的每磁盘的 I/O 数大于磁盘的处理能力， 那么磁盘存在瓶颈。

6.5 Web 应用程序

这里以 ASP.NET 开发的 Web 应用程序为例进行说明。

Object	Counters	Description	参考值
ASP.NET Applications	Request/Sec Request Executing	每秒执行的请求数。	如果 Request/Sec ApplicationsRequest Executing 当前执行的请求数。的值比较小， 你的 Web 程序可能是瓶颈
ASP.NET	ASP.NETRequestWait Time Request Executing Time	最近的请求在队列中等待的毫秒数。执行最近的请求所用的毫秒数。Queued 在理想状况下应该接近 0， Request Queued 等候处理的请求数。该计数器应保持接近 0。超过 IIS 队列长度会出如果这两个值太大，那么需要重现“服务器太忙”错误	

6.6 SQL Server

这里针对 SQL Server2000， 而且只是列出比较关键的几个。更加详细的信息可以参考 SQL Server 的联机文档。

Object（	Counters	Description	参考值
Processor	%Processor time	CPU 使用率	
SQL Server: Logins/sec	这是每秒登录到 SQL Server 的计数		
SQLServer:CacheManage	Cache Hit Ratio (all instances)	显示在高速缓存中找到数据的命中率。如果数值持续小于 85%， 则表示内存有问题。	
SQL Server General Statistics	User Connections	显示当前 SQL 用户数。与 Active Server Pages：Requests/Sec 计数器进行比较， 可帮助了解	

		脚本对 SQL Server 的影响程度。如果差别过大， 则表示测试脚本不能有效地对 SQL Server 进行应力测试。	
SQLServer:Locks	Lock Waits/sec	显示在当前进程完成之前强制其他进程等待的每秒锁定请求的数量。如果该值始终大于 0， 则表示事务有问题。	
SQLServer: BuffeManage	Buffer Manager Hit Ratio	计数器值依应用程序而定， 但比率最好为 90%或更高。增加内存直到这一数值持续高于 90%，表示 90% 以上的数据请求可以从数据缓冲区中获得所需数据。	
SQLServer SQL Statistics	Batch Requests/sec	每秒收的 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 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。	
SQL Server: Buffer Manager	Lazy Writes/sec	每秒被缓冲区管理器的惰性写入器写入的缓冲区数。惰性写入器是一个系统进程， 其主要任务是刷新成批的老化的脏缓冲区（ 指包含更改的缓冲区， 这些更改必须写回磁盘， 才能使该缓冲区由其它页重新使用)， 并使之可由用户进程使用。惰性写入器消除了为创建可用缓冲区而频繁执行检查点的需要。	

SQL Server: Buffer Manager	Page Reads/sec	每秒发出的物理数据库页读取数。这一统计信息显示的是在所有数据库间的物理页读取总数。由于物理 I/O 的开销大，可以通过使用更大的数据高速缓存、智能索引、更高效的查询或者改变数据库设计等方法，使开销减到最小。	
SQL Server:Databases	Transactions/sec	每秒为数据库启动的事务数	

这里针对 SQL Server2000， 而且只是列出比较关键的几个。更加详细的信息可以参考 SQL Server 的联机文档。

6.7 Network Delay

如果要监视的两台计算机在同一个局域网络内， 建议不要使用 Network Delay Monitor。因为在同一局域网内，Network Delay 会非常的小， 网络监视器会有足够的时间在每秒钟内发送成百上千的请求， 这样会导致源计算机（source machine） 的 CPU 和内存超负荷工作。

默认情况下“Enable display of network nodes by DNS names” 选择是没有选中的， 因为选中它会明显的降低该监视器的速度。

7 分析实时监视图表

这一章仅仅介绍几个最重要的图表。

Q1 事务响应时间是否在可接受的时间内？ 哪个事务用的时间最长？

看 Transaction Response Time 图， 可以判断每个事务完成用的时间， 从而可以判断出那个事务用的时间最长， 那些事务用的时间超出预定的可接受时间。

Q2 网络带宽是否足够?

“Throughput”图显示在场景运行期间的每一秒钟， 从 Web Server 上接受到的数据量的值。

拿这个值和网络带宽比较， 可以确定目前的网络带宽是否是瓶颈。

如果该图的曲线随着用户数的增加， 没有随着增加， 而是呈比较平的直线， 说明目前的网络速度不能够满足目前的系统流量。

Q3 硬件和操作系统能否处理高负载?

“Windows Resources” 图实时地显示了 Web Server 系统资源的使用情况。利用该图提供的数据， 可以把瓶颈定位到特定机器的某个部件。

8 经常遇到的问题

8.1 VuGen 的问题

在使用 VuGen 中经常会遇到的问题。

8.2 Controller 的问题

在使用 Controller 中经常会遇到的问题。

1. 在添加完 Load Generators 机器时， 连接老是失败； 添加的机器明明已经安装了 loadrunner， 并且网络通讯正常。

解决方法： 在安装 loadrunner 的第七步骤， 应该选择第 2 项， 如果选择了第一项，就会有这种问题。重新安装一下即可。

2. 在 VuGen 中运行良好的脚本， 到 Controller 中运行却出问题。

这种问题可能会遇到。为了确定问题出在 Controller 中的场景，而不是脚本的问题，你应该在所有的 Load Generators 机器上使用 VuGen 运行测试脚本， 确保都能够运行正确。因为 VuGen 和 Controller 运行的机制不一样。在 VuGen 中运行时使用的是完整的浏览器， 而在 Controller 中运行时使用的只是浏览器的基本的部分。

8.3 计数器的问题

在使用性能计数器中经常会遇到的问题。

1. 添加了 Windows Resources 计数器后， 却看不到实时的数据。

解决方法： 要得到监视的数据， 必须要在被监视的服务器（Web Server） 上获得管理员权限。最简单的方法是在“ 网络邻居”中以 administrator 身份登陆 Web Server。

作者：徐涛 msn： szxutao@hotmail.com

当然使用下面的控制台命令也可以：net use \\< 机器名> 然后登陆用户名和密码即可。（登陆的用户名必须具有管理员权限）

2. 添加了一些默认的性能计数器后， 出现了错误。

解决方法： 可能是一些 LoadRunner 默认的计数器在 WebServer 上已经不存在的原因， 尤其是数据库的计数器方面。简单的解决方法， 就是删除有问题的计数器， 添加比较接近的计数器（ 可能需要参考 Windows 帮助或者数据库的帮助）

9. 结果分析

根据不同的场景设计，配置脚本后进行测试得到如下结果

测试环境

LMM:

CPU：4x2.7G RAM:4G

Websphere 5.0 + IBM Http Server

线程池：100

JDBC 连接池：100

会话超时：30 分钟

DS:

CPU：4x2.2 RAM:4G

Websphere 5.0 + IBM Http Server

线程池：100

JDBC 连接池：100

会话超时：30 分钟

DB&LDAP:

CPU：2x2.2G RAM:4G

Oralce 8.1.7 + LDAP

测试工具：Load Runner 7.8

作者：徐涛 msn： szxutao@hotmail.com

用户数据：用户名 test1ï test100； 口令与用户名相同。

测试用例 **1**

测试场景描述

用户登录的 lmm 模块，总共登陆 24 个用户，所有用户都同时并发操作。

用户点击“登记的教程”

用户点击“启动”，进行课程学习，进入 DS 模块

在 DS 模块中进行学习，过程包括：首先，点击一次课程结构树；然后，进行课程内容的学习。

点击“返回 LMSò 按钮，返回到 lmm 模块

点击“退出”按钮，退出系统

测试结果

LMM 与 DS 模块 CPU 平均利用率在 10%以下。LMM 服务器 CPU 利用率峰值为 20%，其阶段为 LMM 处理多个用户同时的登录请求与点击“已登记教程”的学习课程查询。DS 服务器 CPU 利用率峰值为 100%（持续时间为 7 秒），其阶段为 DS 处理多个用户单一登录验证和同时对课程结构树查询。用户平均操作响应时间不超过 5 秒，所有交易成功。

测试用例 **2**

测试场景描述

用户登陆 lmm 模块，总共登录 48 个用户，每 1 秒登录 1 个用户

用户点击“已登记教程”

用户点击“启动”，进行课程学习，进入 DS 模块

在 DS 模块中进行学习，过程包括：首先，点击一次课程结构树；然后，进行课程内容的学习；

点击“返回 LMSò 按钮，返回到 lmm 模块

点击“退出”按钮，退出系统

测试结果

LMM 与 DS 模块 CPU 平均利用率在 5%以下。LMM 服务器 CPU 利用率峰值为 10%，其阶段为 LMM 处理多个用户同时的登录请求与点击“已登记教程”的学习课程查询。DS 服务器 CPU 利用率峰值为 8%，其阶段为 DS 处理多个用户单一登录验证和同时对课程结构

树查询。用户操作响应时间不超过 3 秒，所有交易成功。

测试用例 **3**

测试场景描述

用户登录的 lmm 模块，总共登陆 48 个用户，所有用户都同时并发操作。

用户点击“登记的教程”

用户点击“启动”，进行课程学习，进入 DS 模块

在 DS 模块中进行学习，过程包括：首先，点击一次课程结构树；然后，进行课程内容的学习。

点击“返回 LMSò 按钮，返回到 lmm 模块

点击“退出”按钮，退出系统

测试结果

LMM 与 DS 模块 CPU 平均利用率在 20%以下。LMM 服务器 CPU 利用率峰值为 40%，其阶段为 LMM 处理多个用户同时的登录请求与点击“已登记教程”的学习课程查询。DS 服务器 CPU 利用率峰值为 100%（持续时间为 10 秒），其阶段为 DS 处理多个用户单一登录验证和同时对课程结构树查询。用户平均操作响应时间不超过 10 秒，所有交易成功。

测试用例 **4**

测试场景描述

用户登录的 lmm 模块，总共登陆 48 个用户，每秒同时登录 10 个用户。

用户点击“登记的教程”

用户点击“启动”，进行课程学习，进入 DS 模块

在 DS 模块中进行学习，过程包括：首先，点击一次课程结构树；然后，进行课程内容的学习。

点击“返回 LMSò 按钮，返回到 lmm 模块

点击“退出”按钮，退出系统

测试结果

LMM 与 DS 模块 CPU 平均利用率在 10%以下。LMM 服务器 CPU 利用率峰值为 10%，其阶段为 LMM 处理多个用户同时的登录请求与点击“已登记教程”的学习课程查询。DS 服务器 CPU 利用率峰值为 100%（持续时间为 2 秒），其阶段为 DS 处理多个用户单一登录验证和同时对课程结构树查询。用户平均操作响应时间不超过 5 秒，所有交易成功。

测试用例 **5**

测试场景描述

用户登录的 lmm 模块，总共登录 100 个用户，每 1 秒登录一个用户。

用户点击“登记的教程”

用户点击“启动”，进行课程学习，进入 DS 模块

在 DS 模块中进行学习，过程包括：首先，点击一次课程结构树；然后，进行课程内容的学习。

点击“返回 LMSò按钮，返回到 lmm 模块

点击“退出”按钮，退出系统

测试结果

LMM 与 DS 模块 CPU 平均利用率在 20%以下。LMM 服务器 CPU 利用率峰值为 10%，其阶段为 LMM 处理多个用户同时的登录请求与点击“已登记教程”的学习课程查询。DS 服务器 CPU 利用率峰值为 100%（持续时间为 2ò20 分钟），其阶段为 DS 处理多个用户单一登录验证和同时对课程结构树查询。用户最大操作响应时间 30 秒，所有交易成功。

测试用例 **6**

测试场景描述

用户登录的 lmm 模块，总共登陆 100 个用户，所有用户同时并发操作。

用户点击“登记的教程”

用户点击“启动”，进行课程学习，进入 DS 模块

在 DS 模块中进行学习，过程包括：首先，点击一次课程结构树；然后，进行课程内容的学习。

点击“返回 LMSò按钮，返回到 lmm 模块

点击“退出”按钮，退出系统

测试结果

LMM 与 DS 模块 CPU 平均利用率在 20%以下。LMM 服务器 CPU 利用率峰值为 40%，其阶段为 LMM 处理多个用户同时的登录请求与点击“已登记教程”的学习课程查询。DS 服务器 CPU 利用率峰值为 100%（持续时间为 3 分钟），其阶段为 DS 处理多个用户单一登录验证和同时对课程结构树查询。用户超时 1 个。

测试用例 **7**

测试场景描述

用户登录的 lmm 模块，总共登陆 200 个用户，所有用户同时并发操作。

用户点击“登记的教程”

用户点击“启动”，进行课程学习，进入 DS 模块

在 DS 模块中进行学习，过程包括：首先，点击一次课程结构树；然后，进行课程内容的学习。

点击“返回 LMSò 按钮，返回到 lmm 模块

点击“退出”按钮，退出系统

测试结果

LMM CPU 平均利用率在 20%以下。LMM 服务器 CPU 利用率峰值为 40%，其阶段为 LMM 处理多个用户同时的登录请求与点击“已登记教程”的学习课程查询。DS 服务器 CPU 利用率峰值为 100%（持续时间为 5 分钟），其阶段为 DS 处理多个用户单一登录验证和同时对课程结构树查询。用户超时 108 个。

10 参考文献

LoadRunner 中文使用手册（完全版） ---作者：huior

LoadRunner 7.8 联机帮助