**LoadRunner**

**一、基本概念**

1、定义：LoadRunner是一种预测系统行为和性能的负载测试工具。它通过模拟实际用户的操作行为和实行实时性能监测，来帮助测试人员更快的查找和发现问题。LoadRunner适用于各种体系架构的自动负载测试，能支持广泛的协议和技术，为测试的特殊环境提供特殊的解决方案。通过使用LoadRunner，企业能最大限度地缩短测试时间，优化性能并加速应用系统的发布周期。默认C语言。

2、4大模块： Virtual User Generator —— 用于创建、编辑性能测试脚本 Vugen

LoadRunner Controller —— 用于创建、监控场景和运行 Controller

LoadRunner Analysis —— 用于分析性能测试结果 Alalysis

Load Generator —— 生产虚拟用户，并向服务器发起请求 LG

3、测试流程：规划测试 ：业务建模，确定性能测试测什么，怎么测。

创建脚本 ：使用Virtual User Generator录制、编辑和完善测试脚本。

定义场景： 使用LoadRunner Controller 设置测试场景。

运行场景： 使用LoadRunner Controller 驱动、管理并监控场景的运行。

分析结果：使用LoadRunner Analysis 生成报告和图表并评估性能。

4、脚本开发步骤：确定文件的存储结构 🡪录制基本脚本🡪增强/编辑脚本🡪配置运行时设置🡪试运行脚本

5、存储结构：在一个父目录存储所有当次性能测试的文件，比如这个目录名可以为“ProjectName\_Perf”，然后在“ProjectName\_Perf”下新建

5个目录，分别存放脚本、参数文件、场景、场景运行结果、分析报告，名字分别为Test\_Script、Test\_Parameter\_List、Test\_Scenario、Test\_Scenario\_Result、Test\_Report

6、启动LoadRunner

1）选择开始🡪程序🡪 HPLoadRunner🡪LoadRunner，打开HP LoadRunner11窗口。

2）打开VuGen，在LoadRunner Launcher窗格中，单击Create/Edit Scripts，链接启动Virtual user Generator起始页。

3）选择File🡪New菜单，打开New Virtual User对话框，显示可供选择脚本的协议。

B/S结构，选择Web（HTTP/HTML）协议。C/S结构，根据后端数据库类型来选择，如MS SQL Server协议用于测试后台数据库为SQL Server的应用；对于没有数据库的WINDOWS应用，可以选择Windows Sockets协议。

**二、规划测试**

1、用户数：根据用户需求文档、软件前期运行统计数据或者友盟等第三方数据服务平台确定 pv—浏览次数 ； uv—独立访客

2、并发用户数：并发用户数 = 用户数 \* 10%

3、测试时长：并发用户加载时长、运行时长、退出时长

4、测试点：需要测试的地方

5、性能指标：相当于功能测试用例中的预期结果，用来衡量测试点是否达标

**三、录制基本脚本**

1、录制前的设置

1）在LR启动程序中，点击创建/编辑脚本

2）在vugen起始页中，点击新建脚本

3）在新建虚拟用户弹出框中选择新建单协议脚本----Web（HTTP/HTML），点击创建按钮，打开Start Recording开始录制对话框，填写信息。

选择协议不同，打开的窗口就会不同，以Web录制的对话框为例，包括application type应用程序类型、program to record要录制的程序、URL address地址、working directory工作目录、record into录制到操作（Vuser\_init，Action，Vuser\_end）。

VuGen脚本分三部分：Vuser\_init，Action，Vuser\_end。其中Vuser\_init和Vuser\_end都只能存在一个，而Action可分成无数多个部分，可以通过点击旁边的【new】按钮来创建Action。在迭代执行测试脚本时，Vuser\_init和Vuser\_end中的内容只会执行一次，迭代的是Action部分。

4）在Start Recording对话框，点击Options选项按钮，进入录制选项设置。一般要设置以下选项：

Recording录制🡪HTTP/HTML level级别中设置脚本的显示形式：URL\_based script基于URL的脚本。

基于浏览器的应用程序推荐使用HTML-based script；不是基于浏览器的应用程序推荐使用URL-based script。

基于浏览器的应用程序中包含了JavaScript，且该脚本向服务器发送了请求，如DataGrid的分页按钮等，推荐使用URL-based script；

基于浏览器的应用程序中使用了HTTPS安全协议，建议使用URL-based script。

Advanced高级🡪Support charset支持字符集中设置编码格式：UTF-8；

提示：录制Web脚本时，生成的脚本中存在乱码该如何解决？

新建脚本--->选择协议(Http)-->选项-->高级-->选择“支持字符集”并点选“UTF-8”。

在回放脚本之前：Vuser-->运行时设置-->浏览器-->浏览器仿真-->更改-->使用浏览器-->语言下来选择 “中文(中国)”。

2、录制 ：在Start Recording对话框，点击OK按钮，自动启动要录制的程序，开始录制；点击停止，录制停止，产生生成日志。

在录制的过程中，屏幕上有一个悬浮的录制工具栏，是脚本录制过程中测试人员和VuGen交互的主要平台。 包括以下按钮：

暂停、停止、选择活动操作（init、action、end）、插入开始和结束事务、插入集合点、插入注释、插入文本检查

**四、增强/编辑脚本**

创建Vuser脚本—脚本调试CheckList

1、是否做参数化 4、是否添加检查点

2、是否做关联 5、是否添加集合点

3、是否添加事务 6、是否设置了Run Time Setting(\*)

1、插入事务 lr\_start\_transaction("name") ； lr\_end\_transaction("name",LR\_AUTO)

1）插入事务是用于把功能分为若干部分，在统计性能指标时，可以对不同的事务分别作统计；不包含思考时间；插在集合点后面

2）在录制时插入：录制某个功能开始前🡪单击插入事务的起始点🡪录制，直到某个功能结束🡪插入事务的结束点。

3）脚本中手动插入：单击某个功能起始前的空白处🡪右键鼠标🡪insert🡪start transaction🡪命名🡪OK。

系统自动在脚本语句中插入如下语句: lr\_start\_transaction("登录");

单击某个功能结束后的空白处🡪右键鼠标🡪insert🡪end transaction🡪命名（默认）🡪OK。

系统自动在脚本语句中插入如下语句: lr\_end\_transaction("登录",LR\_AUTO);

2、参数化 {name}

（1）作用：在进行场景执行时，每个不同的虚拟用户可以按照参数的读取策略读取到参数值，以模拟不同用户在提交或者读取不同的数据。

（2）参数化的方法和实施过程：

1）确定需参数化常量：每个用户在界面上读取和提交的信息都不太相同，因此一般都需要参数化，其它与输入信息对应的比如用户id之类的信息也需要参数化；

另外，录制环境绝大多数情况下与执行环境不一致，因此一般需要对IP、端口或者域名做参数化。

2）准备数据：可在Excel中编辑好，一般参数表中只显示100个值，要注意参数取值、个数和格式

3）对脚本进行参数化：选中常量🡪右击鼠标🡪 replace with a parameter替换为参数 🡪命名🡪properties属性 （见后面）

4）如果其它地方也用到这个相同的参数，替换方法如下：

方法一：选中需要被替换的常量🡪右键鼠标🡪use existing parameter 使用现有参数🡪选择已经存在的参数名。

方法二：选中已设置好的参数，右键鼠标🡪replace more occurrences替换其他出现位置🡪逐个或全部替换

5）还原初始字符串

方法为：在脚本视图中，右键单击该参数并选择“Restore original value()”。

\*parameter type参数类型：参数类型不同，相应属性不同。

Data/Time：使用当前日期/时间替换所选常量。

Group Name：使用Vuser组的名称替换所选常量。

Load Generator Name：使用Vuser脚本的负载发生器名替换所选常量。

Iteration Number：使用当前的迭代编号替换所选常量。

Random Number：使用一个随机生成的整数替换所选常量，可以通过参数属性设定参数的范围。

Unique Number：使用一个唯一编号替换所选常量，可以通过参数属性设定参数的第一个值和递增的规则。

Vuser ID：使用运行脚本的虚拟用户ID来代替选择的常量。

File：采用外部的数据来代替，可以使用单独的文件，也可以使用现成的数据库中获取数据。

User Defined Function：从用户开发的dll文件中获取数据。

以file参数类型的属性设置为例：

\*文件路径：参数化结束后，脚本保存的根目录下会自动生成以参数名称命名的参数文件；也可直接选择已备好的参数文件。

\*添加行、添加列

\*用记事本编辑（在记事本中添加参数值）、数据向导、模拟参数

\*Select Column选择列： By number按编号；By name按名称

\*file format文件格式： 列分隔符：逗号、空格、Tab； First data第一个数据行：从第几行读取数据。

\*Select next row选择下一行：

Sequential：顺序的分配Vuser参数值。当正在运行的Vuser访问数据表格时，它将会提取下一个可用的数据行。

Random：当脚本开始运行时，“随机”的为每个Vuser分配一个数据表格中的随机值。

Unique：为Vuser的参数分配一个“唯一”的顺序值。注意，参数数量一定要大于等于“Vuser量\*迭代数量”。

\*Update value on更新值的时间：

Each iteration：脚本每次迭代都顺序的使用数据表格中的下一个新值。

Each occurrence：在迭代中只要遇到该参数就重新取值。

Once：在所有的迭代中都使用同一个值。

\*When out of values当超出值时：（选择数据为unique时才可用到）

Abort Vuser：中止。

Continue in a cyclic manner：继续循环取值。

Continue with last value：取最后一个值。

3、插入集合点 lr\_rendezvous("name")

（1）插入集合点是一种增大并发压力的方式。脚本运行时，只有当到达集合点的虚拟用户数满足设置要求时，才会继续往下运行。

需要注意的是，集合点只能插入Action部分的脚本中，不能插入vuser\_init和vuser\_end两部分脚本中。

（2）在录制时插入操作为：录制某个功能开始前🡪单击插入集合点🡪命名🡪OK

在脚本的插入方式为：单击需要插入集合点的功能开始前的空白处🡪右键鼠标🡪insert🡪 rendezvous🡪命名🡪OK。

系统自动在脚本语句中插入如下语句: lr\_rendezvous("rnd\_qianghongbao")

4、插入关联 web\_reg\_save\_param(“param\_name”, ”LB= ”, ”RB= ”, LAST)

（1）在脚本回放过程中，客户端发出请求，通过关联函数所定义的左右边界值（也就是关联规则），在服务器所响应的内容中查找，得到相应的值，以变量的形式替换录制时的静态值，从而向服务器发出正确的请求。最典型的是用于sessionID。

（2）录制前关联： Start Recording对话框🡪Options录制选项🡪correlation关联，勾选LR已有的关联规则，或新建规则；

录制后扫描关联：在脚本录制完成后，回放脚本，然后在脚本的菜单的vuser🡪scan script for correlations扫描脚本中的关联，进行设置。

通过回放脚本和扫描关联，系统尝试找到录制与回放时服务器响应的差异部分，找到需要关联的数据，并建立关联。

录制后手动关联：1）录制两份脚本，保证业务流程和使用的数据相同。

1. 使用WinTiff工具比较两份脚本，对两份脚本中不同的地方进行判断，找到需要关联的数据。------方法二？

------------Script View 可以查看全部录制的脚本代码， Tree View 可以查看每个URL获取来的页面

3）在生成日志中搜索需要关联的数据，从而确定左边界和右边界字符串

4）在回放日志中搜索需要关联的数据，从而定位到脚本中插入关联函数的位置

5）右键鼠标🡪insert🡪web\_reg\_save\_param🡪命名🡪左右边界🡪勾选实例、找不到、搜索范围

6）用关联函数中定义的参数替换脚本中需要关联的数据

5、插入注释 //

在脚本中插入注释，可以清晰找到需要修改的位置，增强脚本的可读性。插入一行注释用 // ，插入多行用 /\* \*/

6、插入检查点 web\_reg\_find(“Fail=NotFound”, ”Search=Body”, ”SaveCount=”, ”Text=userID”,LAST)

1）设置检查点函数，将返回值的结果反映在Controller的状态面板上和Analysis统计结果中，由此可以判断数据传递的正确性。

2）步骤：单击需插入集合点的功能开始前的空白处🡪右键🡪insert🡪 web\_reg\_find🡪搜索特定文本(如userID)🡪勾选搜索范围🡪勾选保存计数并命名🡪勾选失败条件🡪OK。

**五、配置运行时设置**

选择Vuser🡪Run-time Settings🡪

1）Iteration Count (迭代次数) 迭代次数、运行逻辑，插入、删除、上下移动操作和块设置。

2）步 开始新迭代的方式设置 在前一个迭代结束后立即开始or延迟、间隔\*\*秒再开始

3）日志 消息发送和消息记录级别设置 设置记录参数替换、服务器返回数据等

4）Think Time (思考时间)

Ignore think time 脚本回放时，将不执行lr\_think\_time()函数，这样会给服务器产生更大的压力。

Replay think time 按录制时获取的值回放；按录制时获取值的整数倍回放；制定一个最大和最小比例，按之间的随机值回放

Limit think time to 用于限制think time的最大值，脚本回放过程中，如果发现有超过这个值的，用这个最大值替代。

5）Error Handling(错误处理)

Continue on error 遇到错误时继续运行。

Fail open transactions on lr\_error\_message 执行到事务中调用的lr\_error\_message()函数时将事务的结果置为Failed。

Generate snapshot on error 对错误进行快照。

6）Multithreading(运行方式) 设定脚本是以多线程方式运行Run Vuser as a process还是以多进程方式运行Run Vuser as a thread。

7）自动事务 设置将每个操作定义为一个事务还是将每步定义为一个事务

8）网络速度模拟 设置网络速度是使用对打带宽还是使用带宽选项还是使用自定义带宽

9）浏览器模拟 设置浏览器属性

**六、试运行脚本**

1）试运行脚本：按F5键，或点击菜单中的运行按钮，可以试运行脚本。回放过程中VuGen在下方同步打印回放日志。

2）设置日志：在vuser🡪run\_time-settings🡪log选择不同的项，回放脚本时将打印不同级别的日志。

3）查看运行结果：通过View视图🡪Test Results测试结果，查看系统给出的回放结果。

4）调试脚本遵循以下原则： 1、单用户单循环 ---- 解决关联的问题

2、单用户多循环 ---- 参数化

3、多用户单循环 ---- 一般采用添加集合点后的并发测试，以检查程序是否有线程锁，资源争用等情况。

4、多用户多循环 ---- 检查在多用户情况下，参数取值是否正确。通常不用运行场景，在参数列表中就可以检查

**七、定义场景**

1、设计场景

面向目标场景（Goal Oriented Scenario） 测试性能是否能达到预期的目标，在能力规划和能力验证的测试过程中经常使用。

手动设计场景（Manual Scenario）能够更灵活地按照需求来设计场景模型，使场景能更好地接近用户的真实使用。步骤如下：

（1）VuGen 🡪tools🡪create controller scenario 从程序中打开，然后选择脚本；也可从VuGen中直接连接到该脚本控制场景

（2）在弹出窗口选择虚拟用户数、运行结果保存目录

按照事先约定选择目录，结果文件的命名最好包含用户数/加压方式/场景名、负载产生的负载机所在地。

（3）在Create Scenario窗口中点击OK，链接启动LoadRunner Controller。

默认显示用户组模式，可通过菜单Scenario🡪Convert Scenario to the Percentage Mode切换为百分比模式。

在Controller的Scenario Schedule中，可以设置场景的各项计划，如虚拟用户的加载方式、释放策略等。

1）设置场景的基本信息

Schedule Name：设置场景名称。

Schedule by：选择按场景计划或按用户组计划。

Run Mode： real-world schedule 是真实场景模式，可以通过增加Action来增加多个用户。

basic schedule 是我们以前用的‘经典模式’，只能设置一次负载的上升和下降。

2）设置场景的各类参数 （双击Global Schedule中的对应行，可以设置schedule的各类参数。）

Initialize：初始化是指运行脚本中的Vuser\_init操作，为测试准备Vuser和Load Generator。

Start Vusers：设置场景Vuser加载方式，瞬时加压与逐步加压。

Duration：设置场景持续运行的情况。 一般不低于15min

Stop Vusers：设置场景执行完成后虚拟用户释放的策略。

Start Time：设置场景启动时间。

2、设置集合点

（1）LR允许测试人员根据项目需要自己设定集合点的并发策略。Controller🡪 Scenario🡪Rendezvous，可以查看场景中所有的集合点名称、所属脚本、当前状态和相关的虚拟用户列表信息等，根据系统需求，还可以针对集合点的执行进行设定。

（2）集合点设置

单击Disable/Enable Rendezvous按钮可以选定集合点是否启用。

单击Disable/Enable VUser按钮可以设定一个用户是否参与到集合点中。

单击Policy按钮可以设定集合点执行策略：

第一项：当前所有用户数的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：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：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。

Timeout between Vusers：设定一个超时时间，当第一个用户到达集合点时，系统开始计时，如果在这个设定的时间内没有达到指定的用户数，系统就不再等待，释放用户让场景继续执行。

（3）手动释放Vuser

在场景运行过程中，点击Status Information列表的Release按钮，可以手动释放等待中的虚拟用户。

3、设置负载均衡

（1）为了避免一台测试机器模拟的虚拟用户数过多，他本身性能的下降直接影响测试效果，LR允许使用多台机器运行场景来均衡测试机器的负荷。只要一台机器安装了Load Generator并启动了LoadRunner Agent Process进程，就可以被Controller统一调度来运行场景，Controller负载收集统一的测试信息和执行结果。

（2）Scenario🡪Load Generator，可以查看到所有已经加载的Load Generator信息。

（3）设置：先选择Scenario🡪Convert Scenario to the Percentage Mode，将场景模式由用户组切换为百分比，然后在已经添加好的Load Generators机器列表中选择需要的机器。这样就可以保证负载机均匀地对服务器施压。

4、设置Windows资源监控器

（1）通过添加Windows资源监控器，可监控负载对server的CPU、内存、磁盘等的影响。在监视server之前确保监视连接成功步骤：

（2）添加步骤如下：

1）准备工作：被监视主机的访问模式改为：经典-本地用户以自己的身份验证；且必须设置密码。

被监视系统开启以下三个服务：Remote ProcedureCall（RPC）和Remote Procedure Call（RPC）、Remote Registry。

确认安装Controller的机器可以连接到被监视的机器。

确认并打开共享文件C$。

2）设置步骤：\*点击Controller界面下方的运行Run选项卡，打开Run视图。

\*Monitors监控器🡪Add Measurements添加度量，或在Windows Resources图中右键选择Add Measurements，打开Windows Resources窗口。

\*点击Monitored Server Machines下的Add按钮，在Add Machine中填写计算机IP，选择计算机平台，点击OK。

\*默认的Windows Resources度量将列出在Resource Measurement on文本框中，点击OK激活监控器。

5、定义场景—场景定义CheckList

1、脚本是否正确添加

2、是否设置场景中的Run Time Setting

3、场景是否选对(手工场景，目标场景)

4、用户数是否正确，用户加载和场景运行时间是否正确

5、集合点策略是否正确

6、Load Genarater是否连接成功

7、SLA是否设置(可选)

8、各项监控是否开启

**八、运行场景**

1）场景设计完成后，单击Controller界面下方的Run选项卡，可以进入场景的执行界面。这个界面用于控制场景的执行，包括启动停止执行场景，观察执行时是否出错及出错信息、执行时用户情况、相关性能数据。

2）单击Start Scenario按钮，场景开始运行。一些即时的数据（比如用户数，等待数，成功事务数，失败事务数等）以及性能数据的折线图，会在Run的过程中显示。

3）执行完成后，执行结果以事先的命名默认保存在建立场景时设置的保存目录。如果涉及到调优，需要多次执行同一个场景，建议每次运行前先调整菜单的Results🡪Results Settings，场景结果保存的名字建议包含重要调优参数值。调优参数比较多样，可以在具体的项目用附件约定。

4）测试期间，可以使用LoadRunner的联机监控器观察Web服务器在负载下的运行情况。特别是可以看到，负载的增加如何影响服务器对用户操作的响应时间（事务响应时间），以及如何引起错误的产生。

**九、分析结果**

1、启动：Analysis分析器可以从程序中打开，然后选择保存好的结果文件；也可以从Controller中直接连接到该脚本的控制场景。

Controller🡪Results🡪Analysis Results，链接启动Analysis，将上一次运行的场景结果生成报告。

2、图形合并

理论上性能测试的所有数据都是需要去关注的，一个最基本的关注点是在概要报表的Transaction Summary中被列出的用户响应时间。其对应的折线图为Average Transaction Response Time。通常测试报告需要给出“虚拟用户—用户响应时间”的折线图，这个折线图可以通过合并报表的形式生成，过程如下：选中Average Transaction Response Time报表，单击菜单栏的View🡪Merge Graphs🡪然后选择与Running Vuser图合并，生成的折线图即为“虚拟用户—用户响应时间”。

3、测试结果需要合并的图表

1）Runner Vuser & Hits per Second & HttP Respone per Second

-- Hits per Second & HttP Respone per Second 随着用户数的变化而变化，且两者的曲率大概一致。

2）Runner Vuser & Total Transactions per Second -- Total Transactions per Second随着用户数的变化而变化。

3）Throughput 网络吞吐量，如果测试是在生产环境上进行，要查看吞吐量是不是接近生产环境上的带宽，如果达到生产环境上带宽的80%以上，则表明带宽存在瓶颈。 1M的带宽，理论上能提供125K/S的下行速度

网络容量单位换算方法： 1G=1024M 1M=1024K 1K=1024B

4、测试结果分析思路

1）首先查看结果概况，了解大概情况，查看有没有大面积失败的事务，有没有大量Error的数，虚拟用户数有没有正常加载，点击率和HTTP响应曲率是否一致，如果都正常，表明测试结果可信，可以继续分析其他性能指标；反之，测试结果不可信，要分析异常的原因，修改后重新测试；

2）如果某指标不符合要求，比如某事务的平均响应时间不达标，则在平均事务响应图上选中该事务，右键，选中”Web Page Diagnostics for xxxx”，进入该事务的网页细分图，选中”Time To First Buffer(Over Time)”，查看事务所消耗的时间主要在网路传输还是服务器

3）生成报告：Analysis可以生成HTML或Microsoft Word两种方式的测试报告。

HTML报告：1. Reports🡪HTML Report。

2.为HTML报告选择文件名和保存路径，点击【保存】按钮。Analysis将创建报告，并将其显示在web浏览器中。

Microsoft Word报告：1.点击菜单Reports🡪 New Report打开右图所示的窗口。

2.设置General、Format、Conten选项卡内容后，点击Generate按钮，即可生成

5、分析结果—测试结果需要关注的图表

1、Runner Vuser

2、Hits per Second

3、Throughput

4、Average Transactions Response Time

5、Total Transactions per Second

6、Transaction Response Time(Pe rcentile)

7、Http Respones per Second