

LoadRunner 性能测试基本步骤

作者：森林一木

前言

本文旨在指导初学者使用 LoadRunner 进行基础的性能测试。

我们在接到一个性能测试任务的时候，需要从以下几点考虑：我们的测试对象是什么，测试要求是什么，测试环境怎么部署的，业务规模如何，哪些业务点是客户最关注的等等，下面将从性能测试启动开始讲解基本的测试流程。

1、测试脚本录制

在使用 loadrunner 工具前，需确定哪些业务需要使用该工具进行测试，不需要的时候坚决不用，不要认为这个工具万能。以本次测试中的综合查询（预付费综合业务信息查询）为例进行讲解。

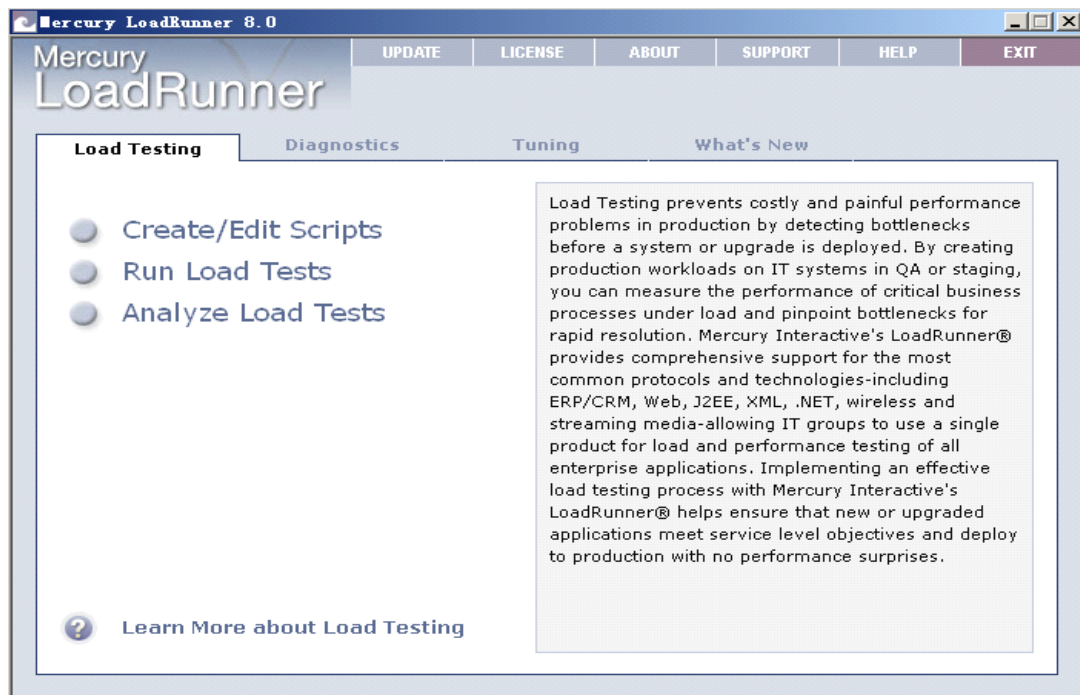
1.1 录制前准备工作

在录制脚本前需检查压测环境的整体功能是否正确，待测部分的功能是否正确，只有确保功能正确后才可进行压测。如本次测试，可先验证 50 环境是否正常，CICS 服务器（49）是否正常，/var/cics_regions 目录的使用率是否过高等等，一切确定 OK 后，开始验证功能，这些都保证没有问题后，检查一下测试工具 loadrunner 是否正常使用，可简单的点点用用，确保工具 OK。

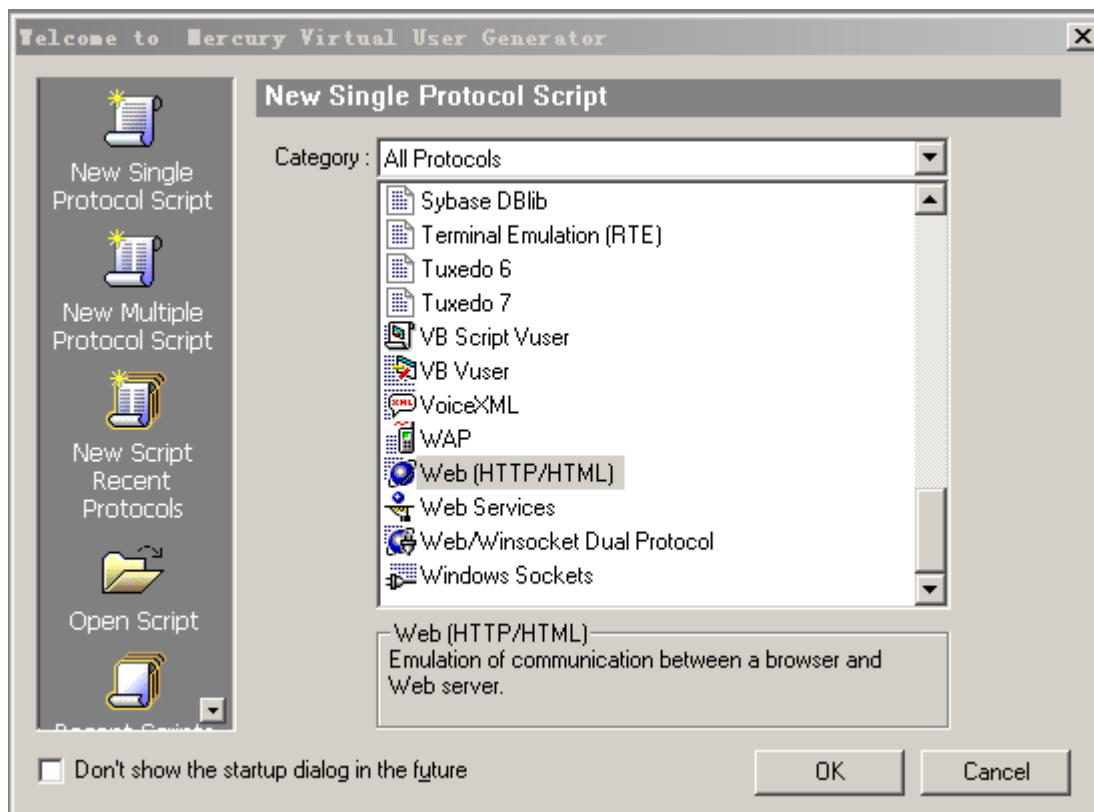
1.2 录制及调试脚本

在准备工作 OK 后，进行脚本的录制，具体过程如下：

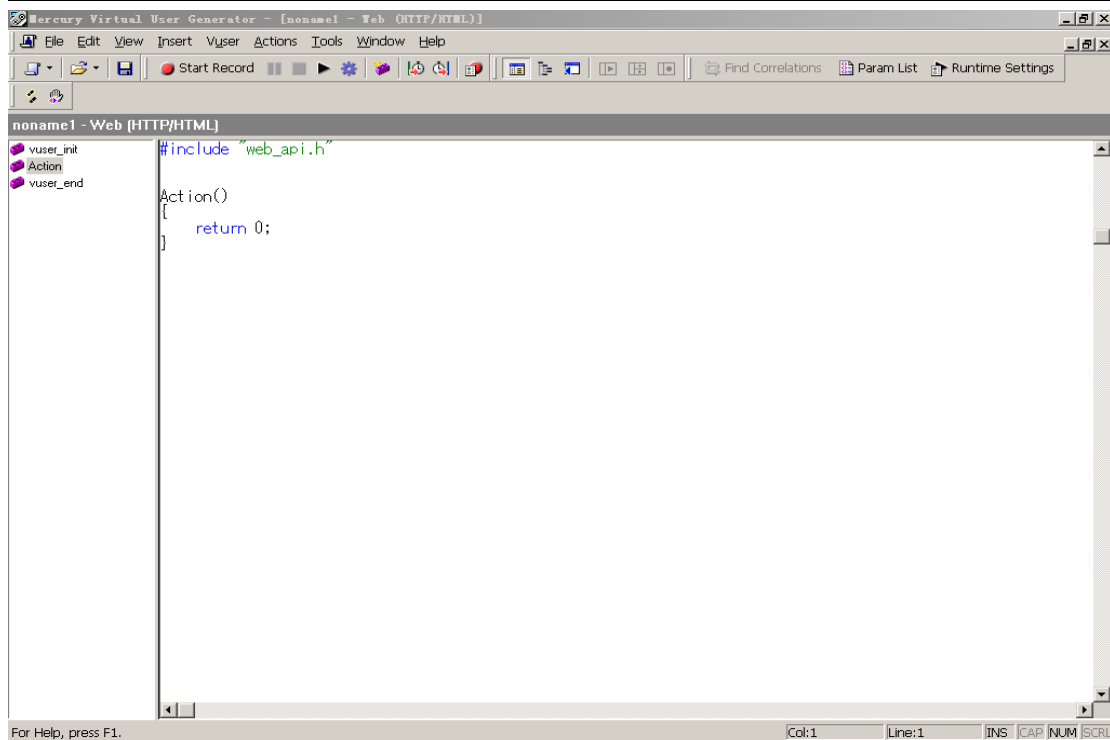
1、打开“开始->程序->Mercury LoadRunner->Mercury LoadRunner”出现下图



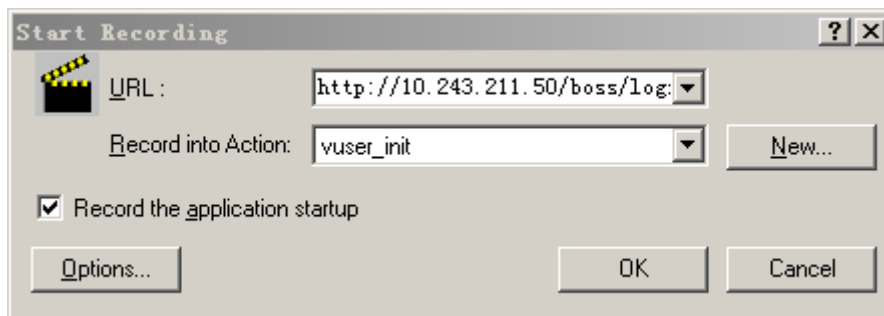
2、点击“Create/Edir Scripts”,出现下图,如果没有出现,则可在“File”下选择 New 新建。






3、出现这个界面后,选择 Web(HTTP/HTML)协议,我们测试的是 B/S 模式,采用的是 Web 协议。选择后,点【OK】按钮。出现下图:



4、点击界面中的 ，这个表示开始录制脚本，点这个按钮后，出现下图：



图中的 URL 输入待测的网址，如本次测试网址：<http://10.243.211.50/boss/loginauthservlet> 在 Record into Action 中选择 vuser_init，把登录部分放在 vuser_init 中，vuser_init 与 vuser_end 在测试过程中仅执行一次，这里解释一下，Action 的作用是讲测试功能主体放在里面执行，举例，假如做产品转换，我们讲登陆的部分放在 vuser_init 中，具体业务操作放在 Action 中，退出部分放在 vuser_end。这样，我们将压力集中在业务操作上，而不是登陆退出上。同时，可以创建多个 Action，将业务操作分成多个部分，比如用户鉴权放在 Confirm 中，将选择产品放在 Select_Prod 中，将业务分开放在多个 Action 的好处是可以统计这个操作的处理时间，处理速度等，便于定位问题。

Action 的增加、修改、删除：Action 可以在录制前增加，具体方法是选中界面左边的  vuser_init  Action  vuser_end 部分，然后点右键，可以看到有增加 Action 的按钮（Create New Action），也可进行删除、重命名。在测试前可以根据需要将业务分为几个操作部分，建立

对应的 Action，名称最好能清晰操作部分的功能。录制脚本的时候，可以将对应的操作放在对应的 Action 中。

这里我们假设综合查询需要以下几个步骤：

第一、登陆



第二、进入菜单

第三、输入测试号码、提交查询

则可设计 Action 为这几个：vuser_init（这个默认有）、IntoMenu、SubQue

5、设置好后，点【OK】，进行录制，在录制前，如果已经打开待测页面的话，建议关闭该页面。点【OK】后，这时会出现待测页面，如 <http://10.243.211.50/boss/loginauthservlet>，同


时会出现 ，这表示现在已经开始录制，可根据需要将业务放在一个 Action 中，也可以分成多个，放在多个 Action

中，具体方法是在进行下一个业务操作前，点上图中的 ，选择对应的 Action，如果事先没有创建 Action 的话，则可点击  增加新的 Action。

在页面中输入用户名后，登陆到系统，待页面都加载完毕后，将 vuser_init 改为 IntoMenu，点击相应的菜单，如查询统计 -> 营业受理查询 -> 预付费查询 -> 预付费综合业务信息查询，页面加载完毕后，将 IntoMenu 改为 SubQue，在“服务号码：”中输入号码 13539300000（测试号码），点击【查询】，待页面返回查询结果后，将 SubQue 改为 vuser_end，退出系统。


注：页面加载完毕可以参考网页左下角有个信息提示“完毕”。

所有操作完成后，点击 

中停止按钮 ，停止录制，页面将自动关闭，返回到 loadrunner 录制界面，将在界面中显示录制脚本代码，保存录制的脚本。

6、调试代码并进行参数化

录制后的代码需要进行调试才可用于压测，调试的办法就是进行回放操作，如果回放过程无错误，运行结果也正确的话，则可用于压测。具体调试步骤如下：

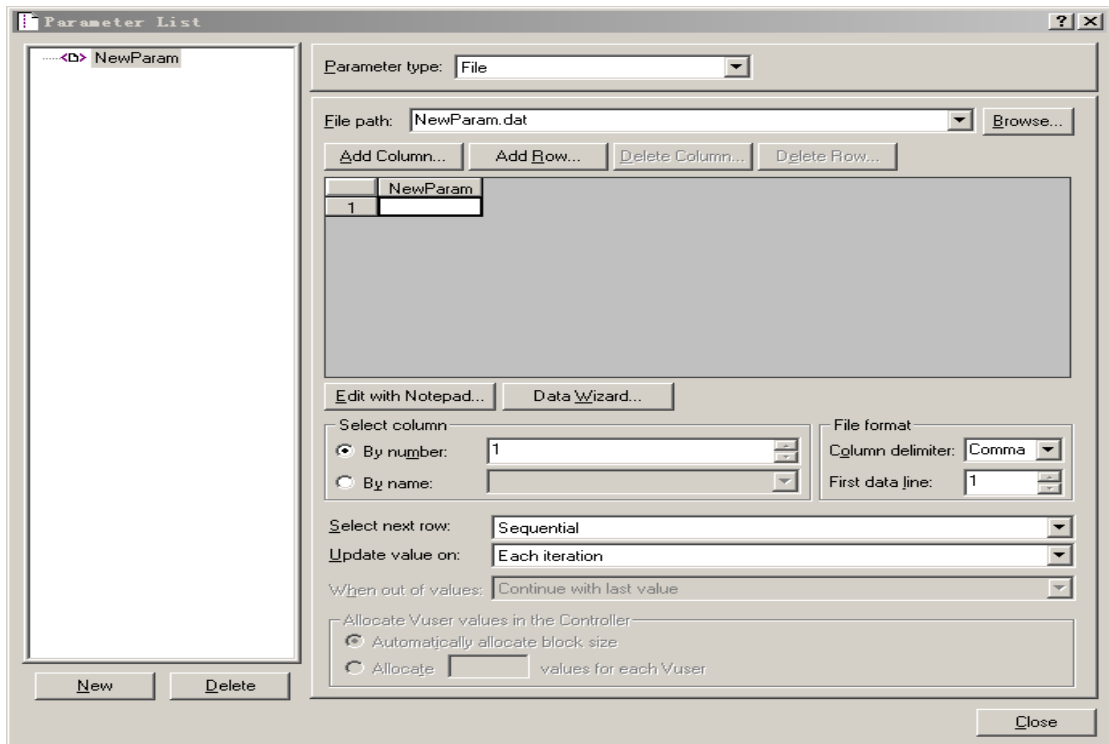
点击界面中的 ，进行单次运行调试，运行后，会弹出运行预览的一个窗口，可以看到每一个 Action 的执行过程，运行结束后，会出现一个结果报告，如果有错误，会在报告中以红色叉标志显示出来，同时在 Execution log 中也会打出错误信息，可以根据这些错误信息进行调试。如果无错误，则可进行插入事务、参数化设置等其他操作。现假设调试无错误，进行参数化设置。

在测试过程中，有可能需要不同的测试号码，如果产品转换，首次激活等，如果有同样的号码将导致测试失败，因为相同的号码不能做同样的业务操作多次，所以需要大量不同的测试号码，这个就需要用到参数化设置。我们在编写测试方案的时候，已经得出要准备多少测试号码，在测试工作准备的时候，已经准备好测试号码，那么可以利用这些准备的号码进行参数化设置。参数化设置的意思就是将需要用其他数据代替的地方设置为一个参数，在运

行时读到这个参数，就使用其他的值代替，在这个例子中，我们需要设置参数的地方是服务号码。这样，我们需要先创建一个参数，步骤如下：

先准备好号码，可在数据中导出，存放在 txt 文本中，格式为：测试号码，一行一个号码，最后一行要为行，如果文件名为 test_num.txt

a、点击界面中  Param List，出现下面界面



在这个界面中，点击左边的 New，创建一个新的参数，在界面的右边，Parameter type 选择 File，File path 选择存放号码的 txt 文件路径，选定文件后，会在下面的表格中列出测试号码，我们在 Select next row 中选择 Unique，这个表示整个测试过程仅使用唯一的号码，保证号码不重复，这样就要号码资源足够多，同时测试时间也需要控制好，否则会报错。

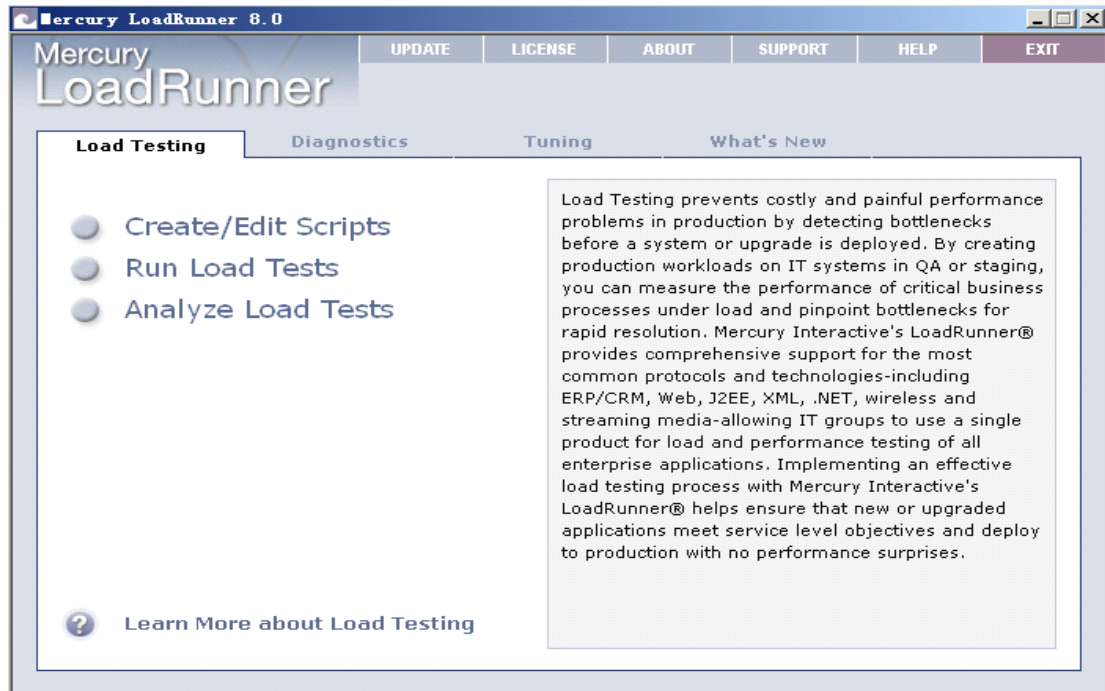
创建好参数后，返回到刚才录制的脚本中，找到对应的 Action，如 SubQue 中服务号码字段，选择该号码，右键选择“Replace with a parameter”，在 Parameter name 的下拉列表中选择需要替换的参数，选定后点击 OK。

设置 OK 后，可进行调试，如无问题，则可以进行场景的设置。这里有个注意点要说明一下，参数化也可以直接在脚本中选中需替换的地方，点右键，选择“Replace with a parameter”，改更 Properties 进行设置，但这样做经常出问题，不容易调试，不建议这样做。

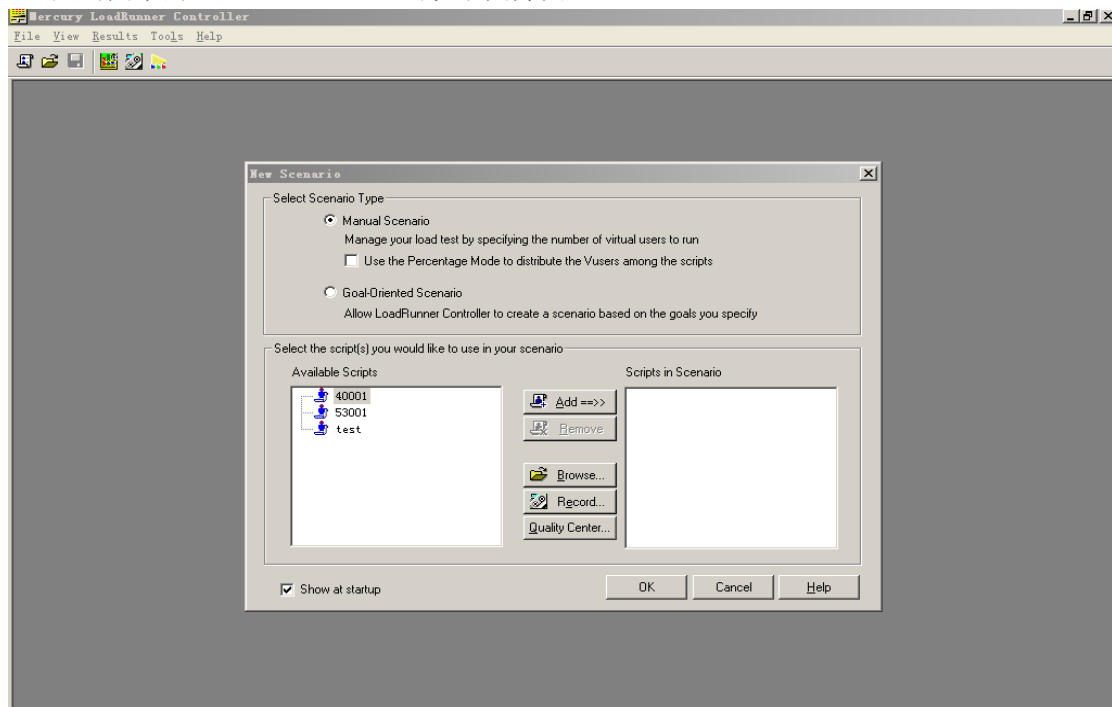
2、设计测试场景

在脚本录制完成，调试通过后，可以进行测试场景的设计。具体步骤如下：

1、打开“开始—>程序—>Mercury LoadRunner—>Mercury LoadRunner”出现下图



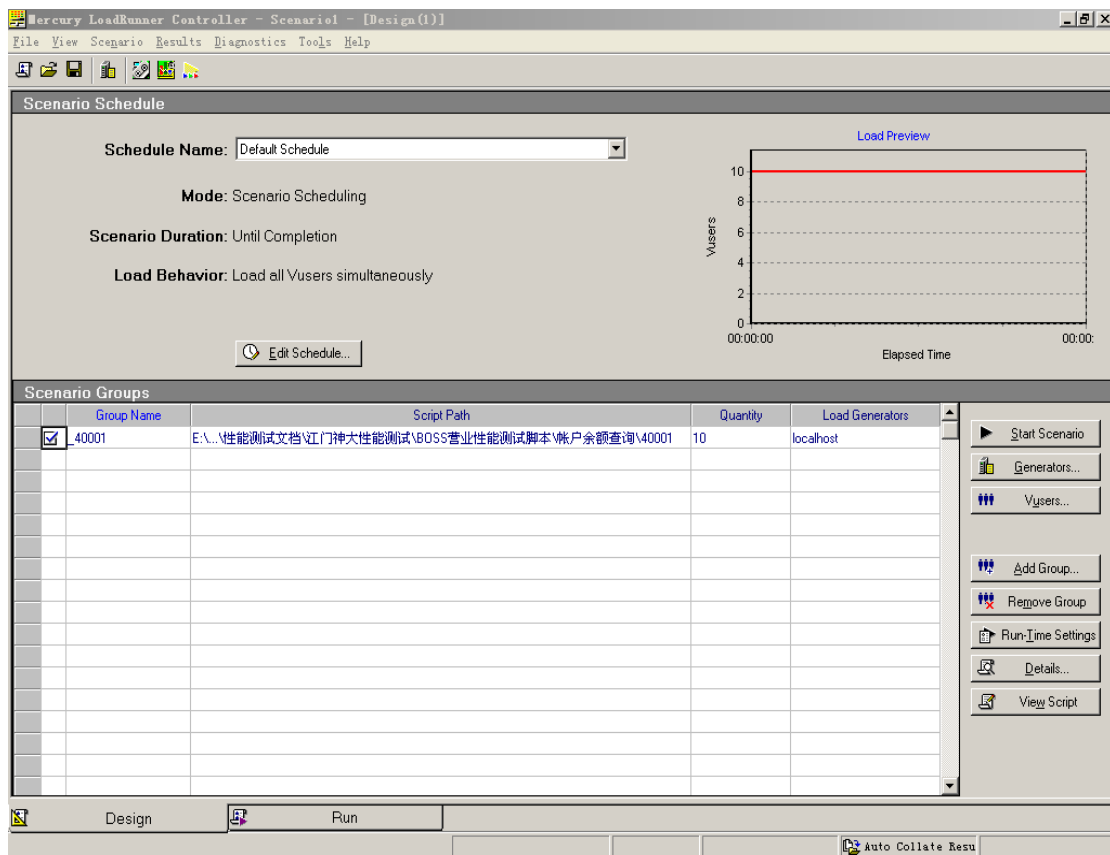
2、点击图中的 Run Load Tests，出现下图界面



在新建场景的窗口，选择一种场景类型。下面对三种类型进行简单的说明。

- **Manual Scenario:** 该项要完全手动的设置场景。
- **Manual Scenario with Percentage Mode:** 该项只有在“Manual Scenario”选中的情况下才能选择。选择该项后，在场景中我们需要定义要使用的虚拟用户的总数，Load Generator machine 机器集，然后我们为每一个脚本分配要运行的虚拟用户的百分比。
- **2 Goal—Oriented Scenario:** 在测试计划中，一般都包括性能测试要达到的目标。选择该项后，LoadRunner 基于这个目标，自动为你创建一个场景。在场景中，我们只要定义好我们的目标即可。

3、在上图中出现的 Available scripts，选择要进行场景设计的脚本，若没有出现需要对应的脚本，可点击 Browse 查找后添加进来，选择好脚本后，点 Add，则可加入到右边的窗口中，然后点【OK】，出现下图



4、上图中的 ScenarioGroups，显示的是脚本的路径与并发数个数，根据测试方案中的并发数可更改此处的并发数，在上图中点击 Edit Schedule，出现下图

Schedule Name: Default Schedule

Schedule Definition

Schedule by Scenario Schedule by Group

Ramp Up | Duration | Ramp Down

Load Settings

- Load all Vusers simultaneously **同时加载所有用户**
- Start 2 Vusers every 00:00:15 (HH:MM:SS) **15秒内增加2个用户，成阶梯上升，直至达到并发用户数最大值**

Initialize all Vusers before Run **必须选上**
(selecting this option means that running begins only after all Vusers reach the Ready state)

Load Preview

Y-axis: Vusers (0 to 10)
X-axis: Elapsed Time (00:00:00 to 01:10)

Schedule Name: Default Schedule

Schedule Definition

Schedule by Scenario Schedule by Group

Ramp Up | Duration | Ramp Down

Run Settings

- Run until completion **所有并发用户只执行一遍脚本，然后停止场景执行**
- Run for 00:05:00 (HH:MM:SS) **加载完所有用户后，场景持续运行5分钟**
- Run indefinitely **一直运行，不停止，除非人工干预**

Initialize all Vusers before Run
(selecting this option means that running begins only after all Vusers reach the Ready state)

Load Preview

Y-axis: Vusers (0 to 10)
X-axis: Elapsed Time (00:00 to 00:06)

Schedule Name: Default Schedule

Schedule Definition

Schedule by Scenario Schedule by Group

Ramp Up | Duration | Ramp Down

Stop Settings

- Stop all Vusers simultaneously **持续运行时间结束后，立即停止所有并发数**
- Stop 5 Vusers every 00:00:30 (HH:MM:SS) **持续运行时间结束后，每隔30秒停止5个并发数，直到所有并发数停止**

NOTE: These settings are relevant only if you specified a limited duration in the Duration tab

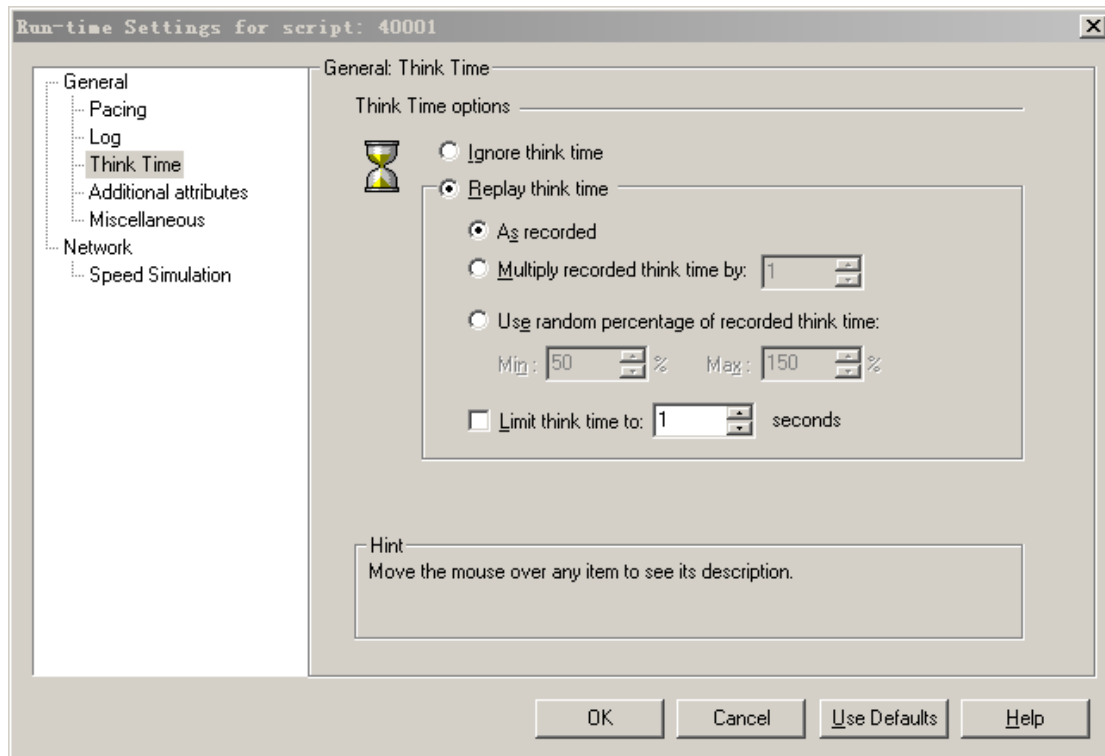
Initialize all Vusers before Run
(selecting this option means that running begins only after all Vusers reach the Ready state)

Load Preview

Y-axis: Vusers (0 to 10)
X-axis: Elapsed Time (00:00 to 00:07)


举个例子，假如我们设计的场景是每 15 秒增加 2 个，所有并发数增加完后持续运行 5 分钟，5 分钟运行结束后，每 30 秒减少 5 个并发，则上面三张图设置了就行了，注意那个 Initialize... 必须勾选上。

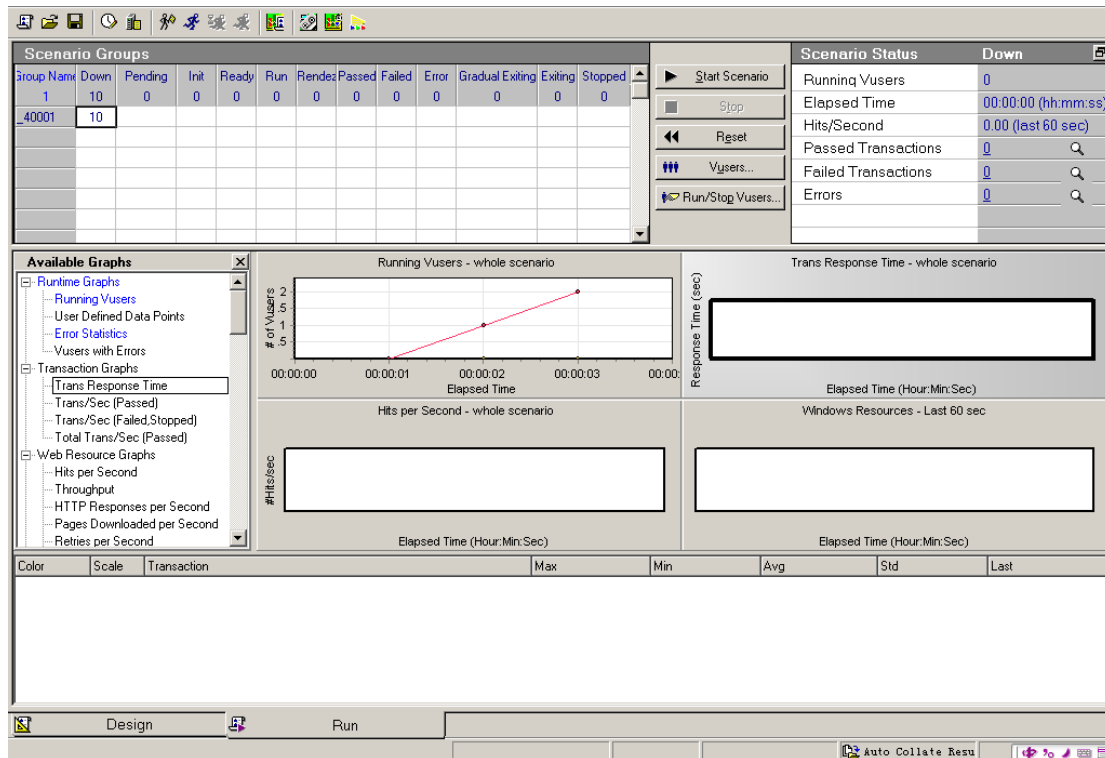
5、再点击页面右下角的“Run-time Settings”，出现下图



选择图中的 Think Time，在右边选择 Replay think time，再勾选中 Limit think time to: 1 seconds，表示即使脚本 think time 时间可能超过 1 秒，也用 1 秒替换，可以自行调整这个时间。这样做的目的是在测试过程中使得每个业务操作里加上 think time，表示用户在操作的时候，有个时间延迟，真实的模拟用户的操作，比如用户在做产品转换的时候，可能在选择产品的时候，有个停顿思考的时间，这样 loadrunner 会记录下来。如果选择 Ignore think time，这样对服务器造成的压力是最大的，在运行时，会没有停顿的，持续对服务器加压，不太符合实际使用情况。

设置好 Think Time 后，选择 Miscellaneous，在出现的窗口中勾选 Continue on error，表示在遇到错误的时候，继续执行场景，直到场景运行结束。

6、一切设置 OK 后，点击 ，运行测试场景，如下图



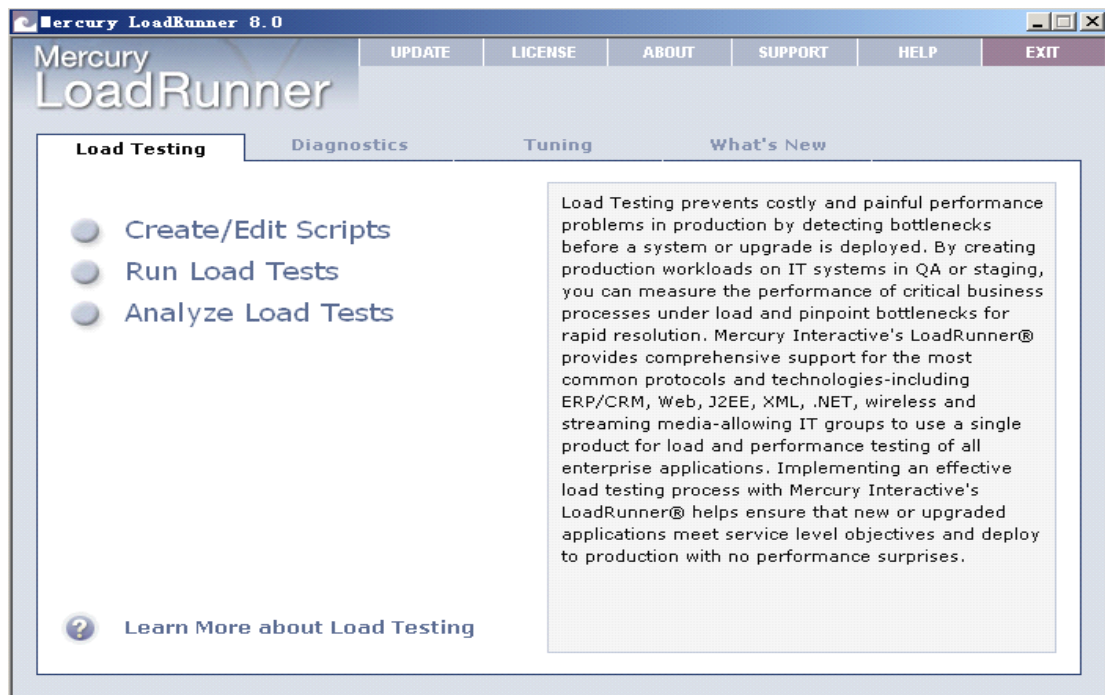
在图中的左边可以查看并发用户数的运行情况，右边可以查看通过的事务数、失败的事务等，如果运行过程中有错误出现，则可以点击 Errors 右边的放大镜，查看详细错误信息。窗口下面是各种监控窗口，Running Vusers 展示的是目前并发用户数的运行情况，Trans Response Time 表示的是事务的响应时间，即每个事务处理的时间是多少秒。

3、测试结果分析

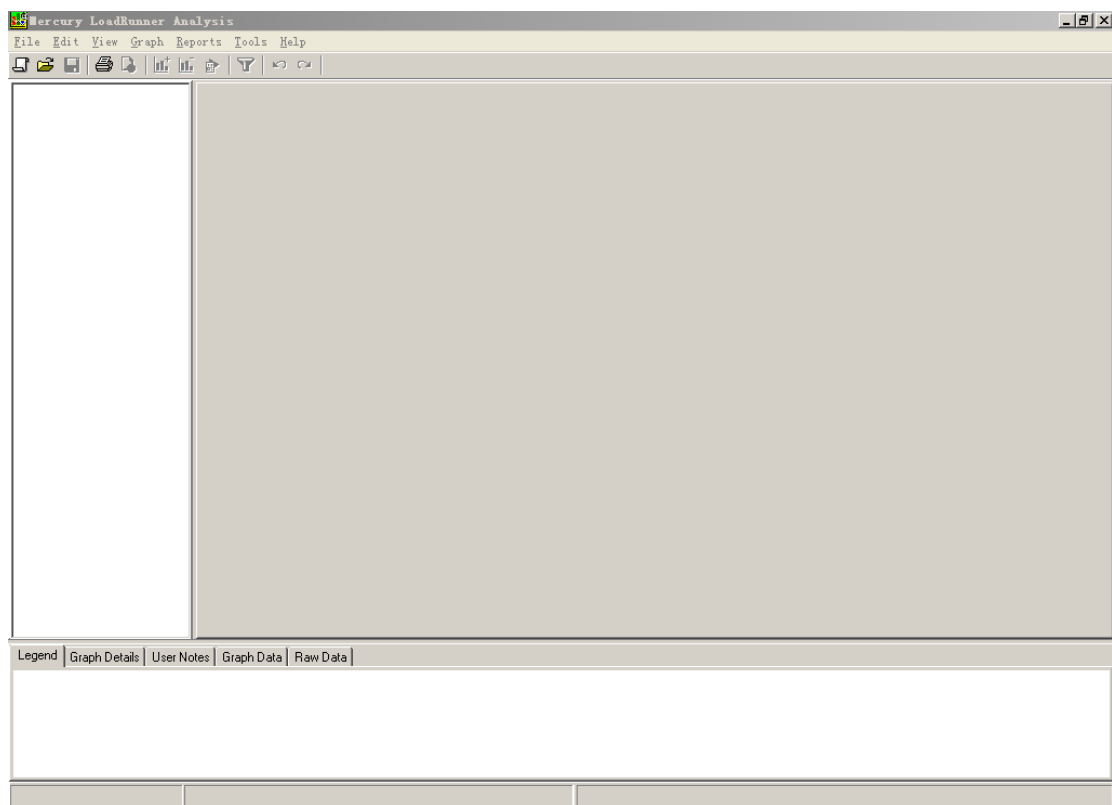
场景执行结束后，可以使用 loadrunner 自带的分析工具进行结果分析，这里我们主要考察两个地方，第一是平均事务响应时间 Average Transaction Response Time，第二是并发数运行情况 Running Vusers，这两个显示了场景运行过程中并发数的执行情况与每笔事务的处理时间。还有其他几个考察点，做简要解释。


注：事务数概念解释：事务就是脚本中定义每个 Action。

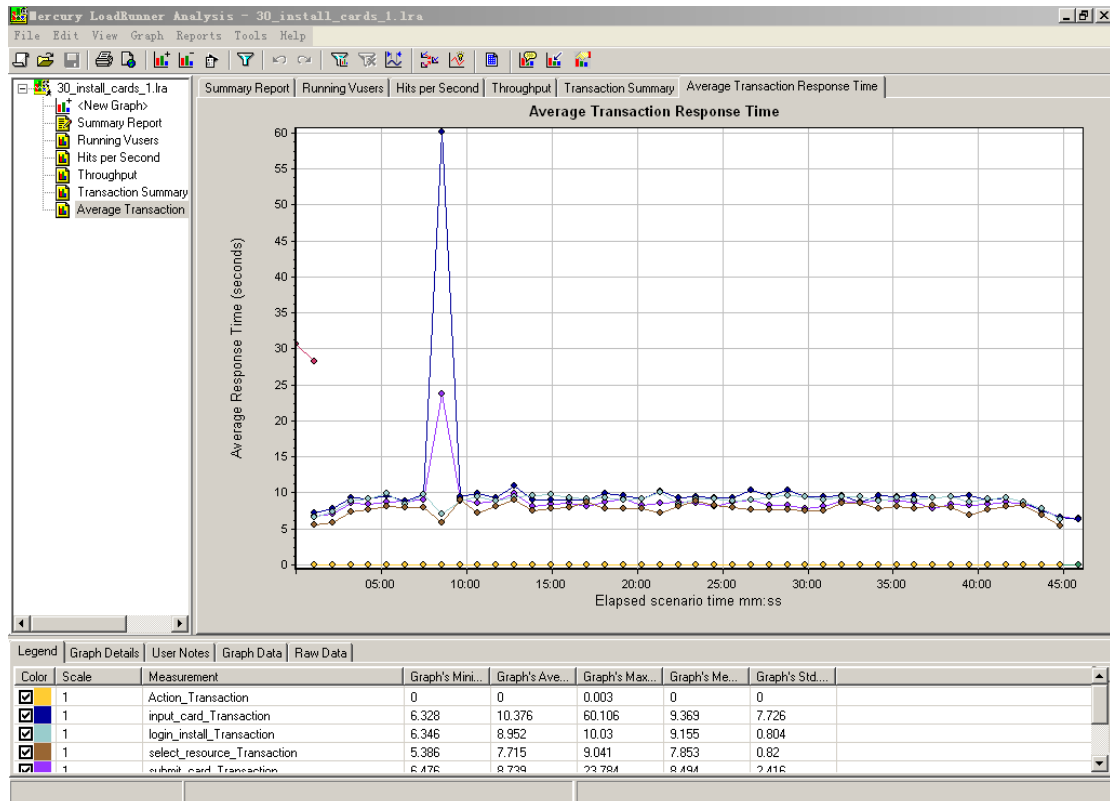
具体分析步骤是先打开“开始—>程序—>Mercury LoadRunner—>Mercury LoadRunner”出现下图



点击 Analyze Load Tests，出现下图



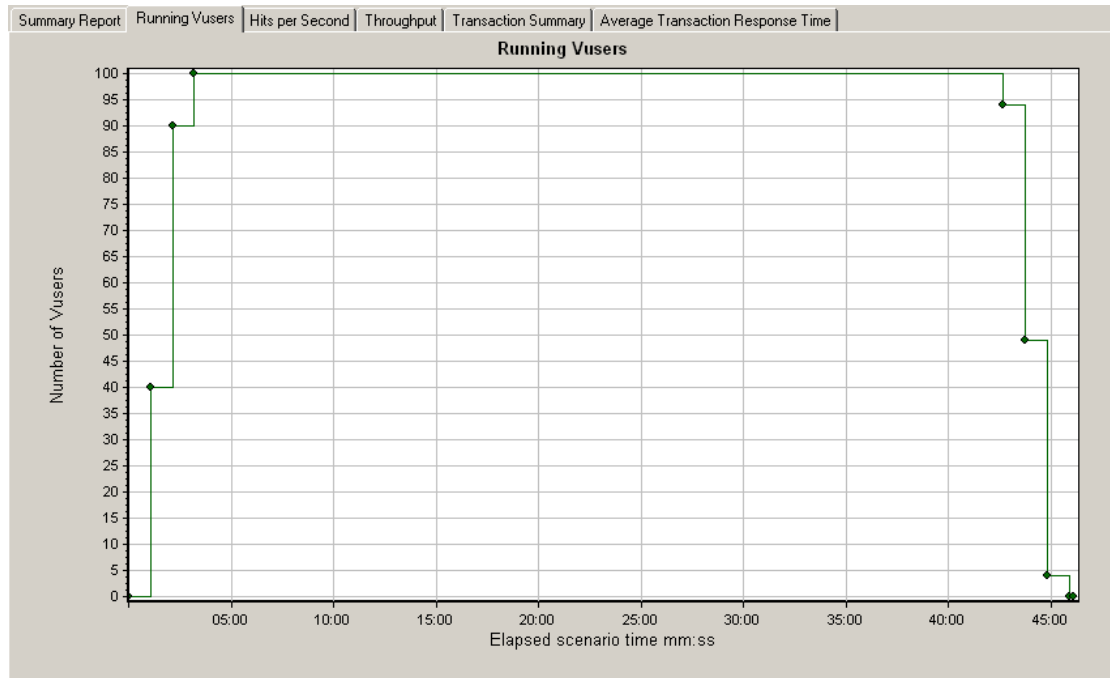
在图中的菜单栏中选择打开，找到要分析的场景执行结果，点【打开】即可，还可以直接在场景运行结束后，点击 Controller 菜单栏的 ，直接收集场景运行结果进行分析。
打开场景执行结果后的界面如下：



界面左边是各个指标的列表，右边是图形的显示，下面是指标对应的采集数据。接下来将重点选择几个指标做解释。

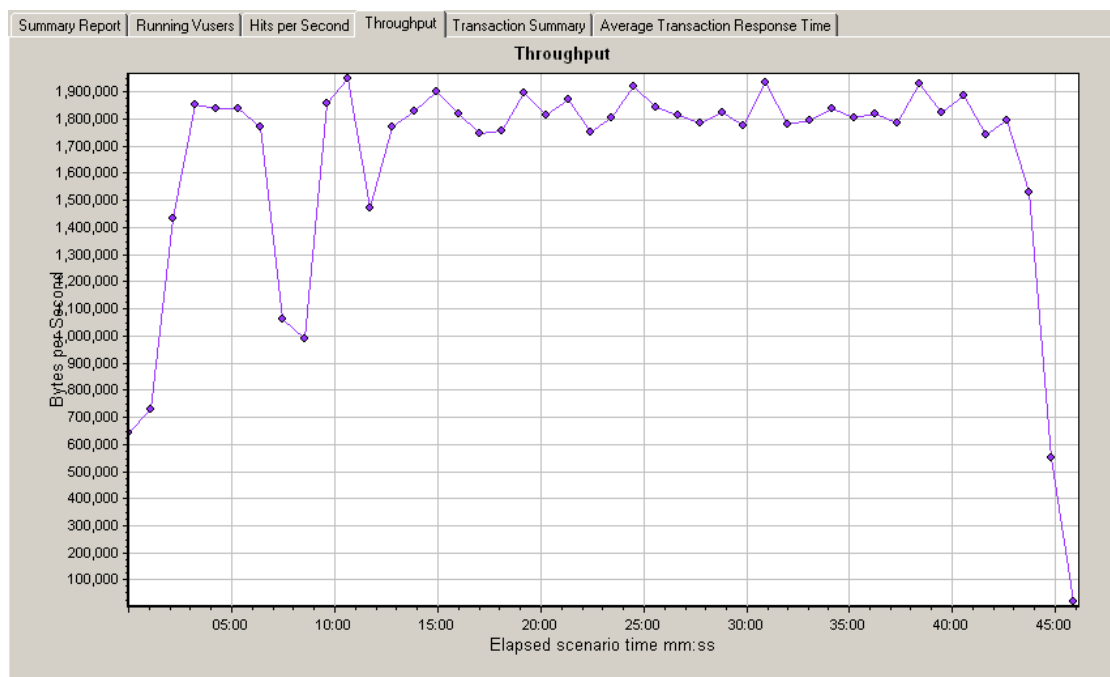
3.1 并发数执行情况 (Running Vusers)

并发数执行情况反映了在场景执行过程中各个并发数的运行情况，成功了多少，失败了多少，是否按照既定的场景执行计划运行，是否达到预期的执行效果，如果在某个时间，执行失败了，或者存在异常，那么并发数的图表将是波动，可以从图中直观的看出来，同样根据场景中 Error 的信息，定位在何时出现了错误，此时执行的并发数是多少。并发数的图表如下：



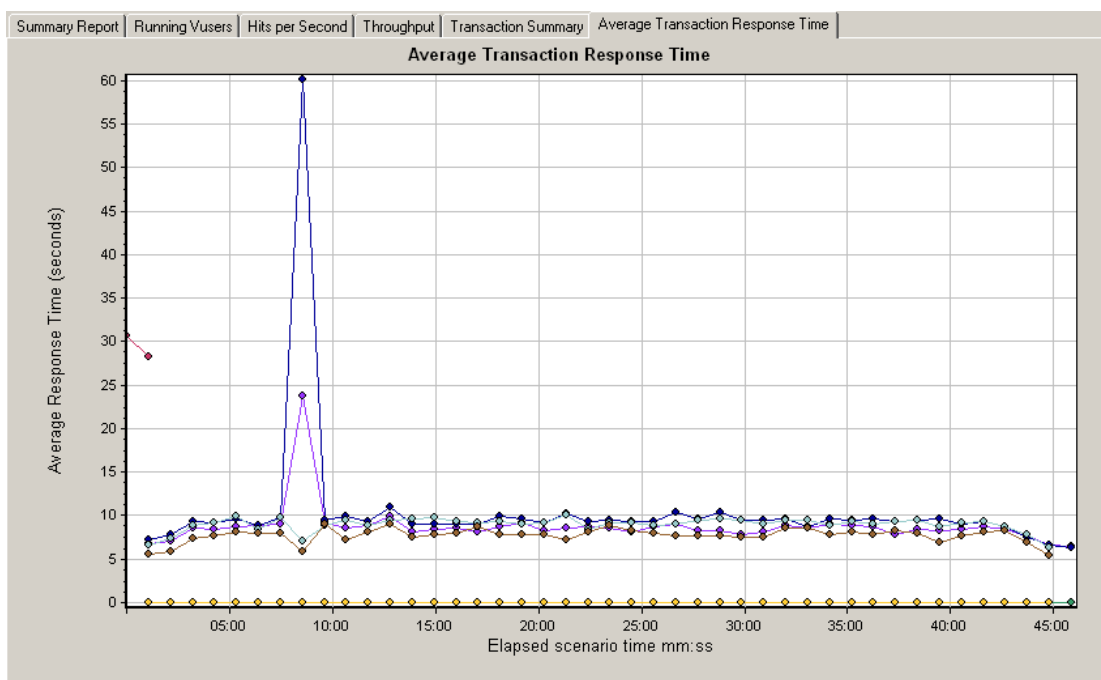
3.2 事务通过数 (Throughput)

事务通过数的意思是每个特定时间间隔内通过的事务数，如果随着场景执行时间的推延，通过事务数曲线应该是平缓的，如果突然上升，则可能是服务不稳定，或者网络因素导致的，如果持续下降，则表示服务的处理能力在下降，此时可以查看服务器段是否有进程堵塞，或者服务器的连接数不够，也可能是网络带宽不够。事务通过数的图表如下



3.3 平均事务响应时间 (Average Transaction Response Time)

平均事务响应时间在整个测试过程中是一个很重要的参考指标，他能清晰的反映出场景执行过程中，每个事务的执行时长，整个业务中哪个操作最耗时等等。场景执行结束后，可以根据这个图表分析出在一定响应时间要求下，系统可支持的并发数，假如我们要求在查询的时候要求这个返回结果的时候不超过 5 秒，那么可以在场景中找到对应的 SubQue 事务在处理时间为 5 秒左右的时间点，再从 Running Vuser 图中找到对应的时间点，查看对应的并发用户数。同样，在整个执行过程中，当并发数达到一定数值后，平均事务响应时间曲线应该是平缓的，如果出现突然升高或者降低的情况，则表示系统存在异常，这样我们可以找到这个时间点去查看服务器端的运行日志，查找原因。平均事务响应时间的图表如下：



3.4 服务器资源分析

服务器资源监控利用的是 nmon 工具，在得出分析结果后，可以查看对应的图表进行分析。

4、总结

整个测试过程是有序的，需要事先计划好，在录制脚本，调试脚本的时候需要注意的是要单个单个号码的试，不可一次把号码都跑完，这样不经济。这里仅仅介绍了简单的测试流

程，更深刻更精妙的功能未能列出，请自行学习。本文档重点介绍的 LoadRunner 的使用。