LoadRunner

进行 Web 测试时

吞吐量和点击量深入研究

一起测试网 ： 软件质量专家

2006 年 7 月 9 日

版权声明：

主要作者：王玉亭。联系方式：MSN: wangyuting66@hotmail.com

目　录

1 前言

很多从事软件测试工作的网友在使用 Mercury 的性能测试工具 LoadRunner 做性能测试的时候，常常遇到一个问题，这就是对测试结果的分析无从下手。

在性能测试项目中，最困难的部分就是对性能测试结果的分析。这需要测试人员对被测试系统有着深刻的理解。但是这又是大部分性能测试人员所缺乏的。

本文以一个小实验为手段，深入研究 LoadRunner 在 Web 系统性能测试中两个重要指标 Total Throughput(bytes)和 Total Hits 的真正含义，引导读者研究如何深入分析测试结果。

本文只是一个小小的砖头，希望提供一个研究的手段，帮助大家深入研究测试结果。

Let's go!

2 关于“一起测试网”

“一起测试网”（wwww.17testing.com，以下简称 17testing 或一起测试网）是苏州市软件评测中心有限公司推出的关于软件测试方面的大型技术门户网站。“一起测试网”将本着服务软件测试人员，推动软件测试产业发展的目的，为广大测试人员提供更多更好的技术资料。

3 本文档所涉及的软件

在我们开始下面的旅行之前，我们要准备好做实验所需要的软件。这些软件都可以从www.17testing.com上下载。

性能测试工具 LoadRunner 8.0 试用版：

http://www.17testing.com/zyxz/csgjzy/6.html

4 一次简单访问主页的真相

为了深入理解测试结果，我们要深入理解被测试的系统在表面的背后到底发生了什么。下面我们精心构造一个小实验，来深究一下浏览器访问一个简单的页面背后到底发生了什么。这样的实验有助于我们了解最基本的 HTTP 通讯的真相。

我们使用的工具是我编写的一个小工具，一个简单的 Proxy 程序，可以帮助我们捕获一次通讯的全部过程。这个小工具的下载地址是：

http://www.17testing.com/bbs/viewthread.php?tid=5&extra=page%3D1

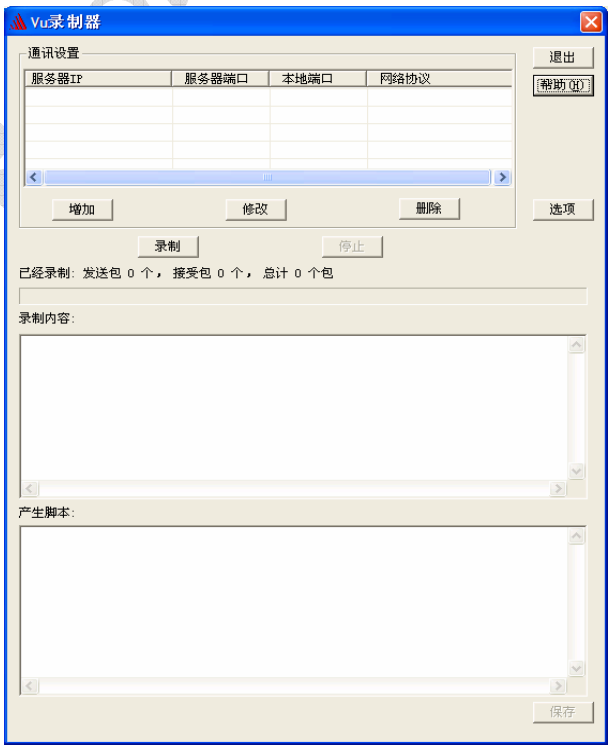
下载 VuGen.zip，把它解开。这个程序可以在 VC6 下面编译。如果你找不到 VC6，可以在一起测试网的如下地址找到：

VC6: http://www.17testing.com/bbs/viewthread.php?tid=806&extra=page%3D1

MSDN: http://www.17testing.com/bbs/viewthread.php?tid=831&extra=page%3D1

为了帮助大家学习 VC 的编程。我把 MSDN 也放上来了。

如果大家不想编译，可以直接运行 VuGen\Release\VuGen.exe。如下图所示：



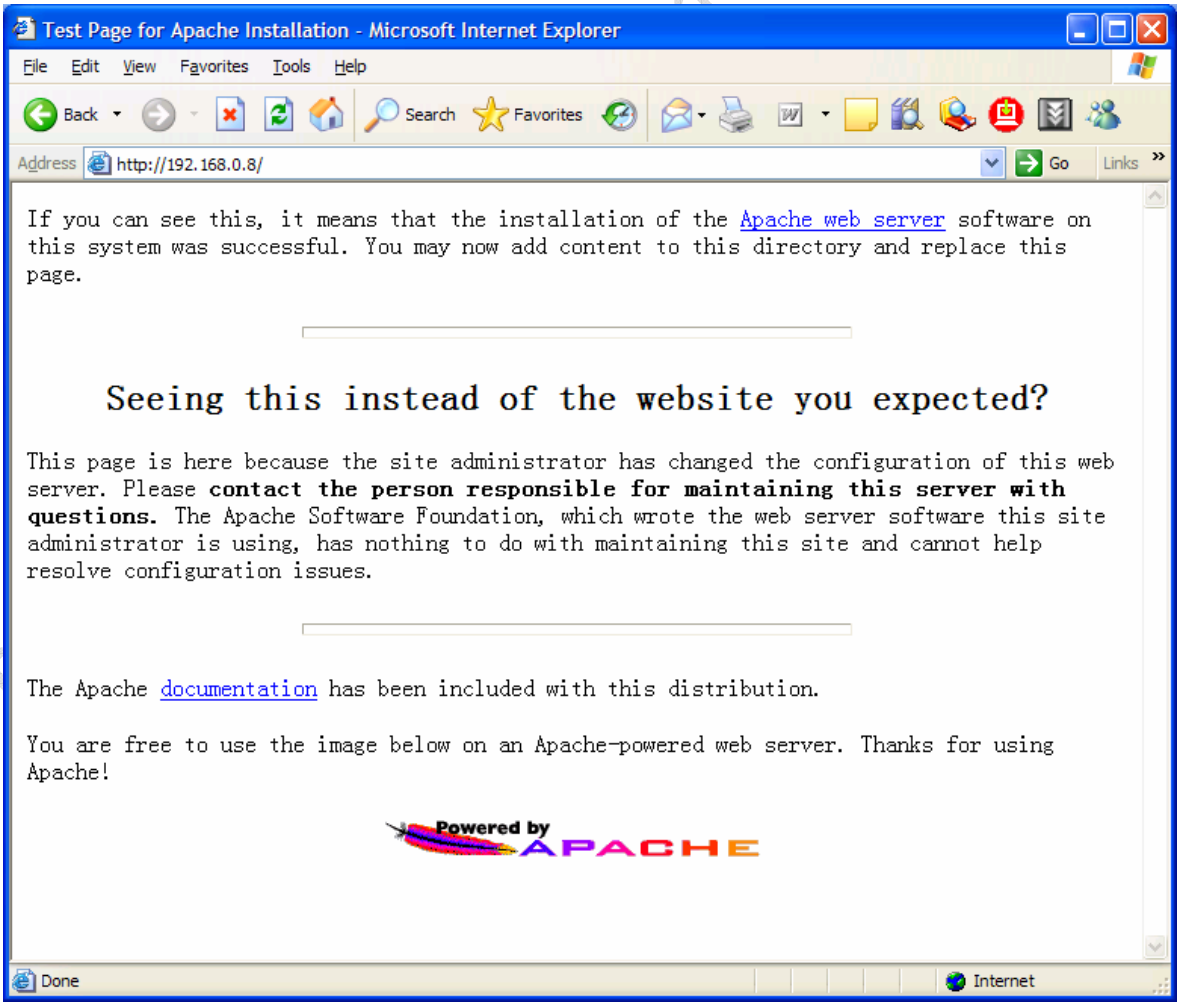
我们先搭建一个 Apache 服务器，并启动起来。关于如何搭建 Apache，请参考网上的相关文章。最省事的方法是到 Apache 网站上下载一个 Windows 的安装程序，鼠标双击，就完成了安装任务。

缺省配置下，Apache 的 Web 文档主目录在$APACHE_HOME/htdocs 目录下。其中$APACHE_HOME 指的是 Apache 的安装目录。我们在这个目录下只留两个文件

index.html.en 和 apache_pb.gif。把 index.html.en 重新命名为 index.html。这样，在 htdocs 目录下只有两个文件：

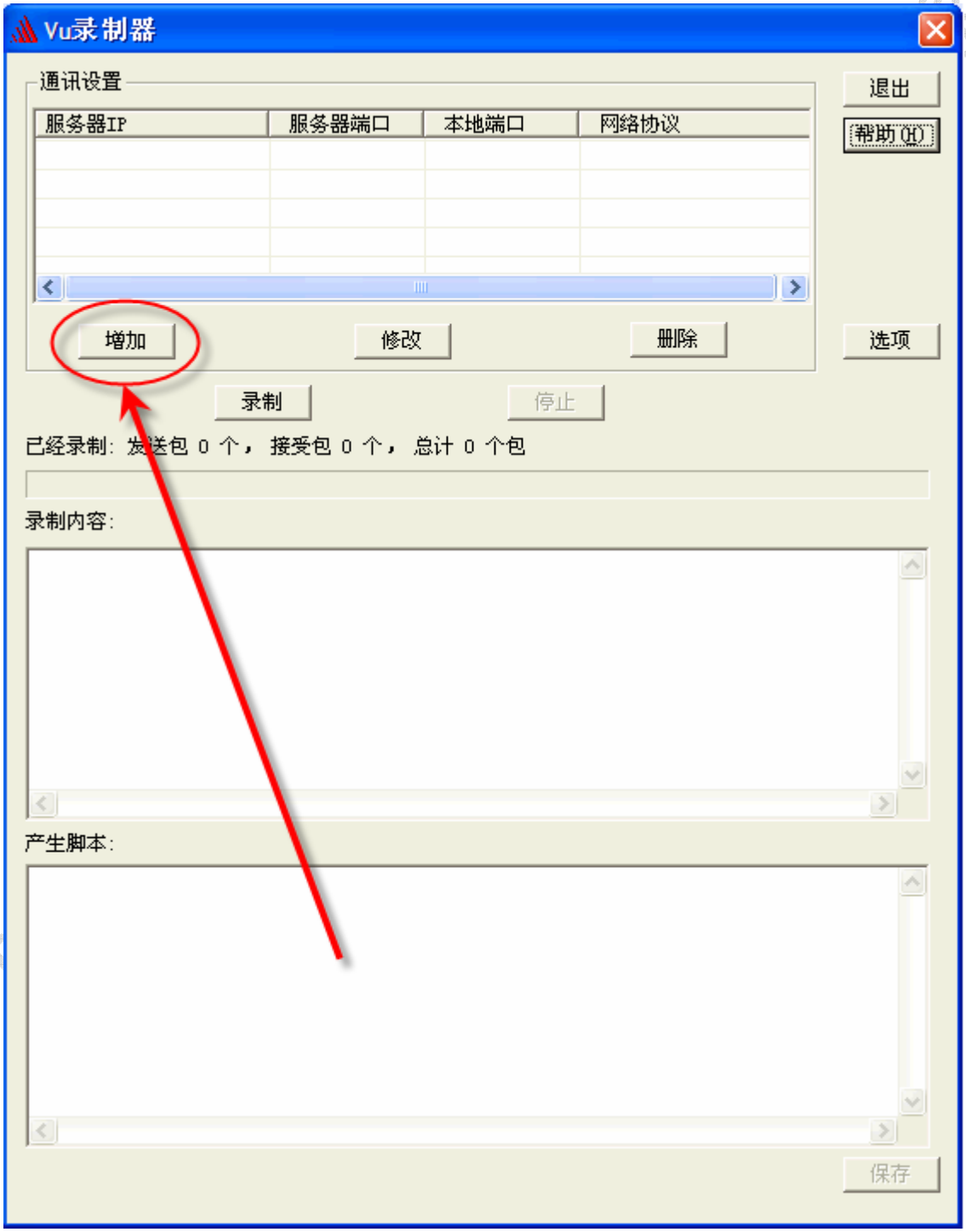
- index.html，大小是 1456 字节
- apache_pb.gif，大小是 2326 字节

启动 Apache 后，我们通过浏览器访问该服务器，得到如下界面：

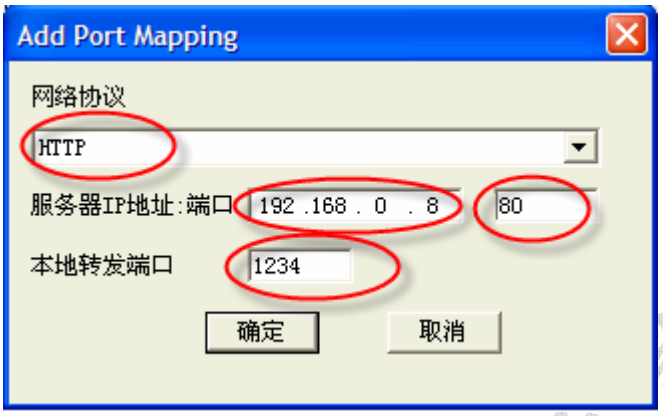


如果你安装过 Apache 软件，你应该熟悉这个界面。它实质上是 Apache 的测试页。我们通过刚才的小程序研究一下访问这个简单的页面，在网络通讯上发生了什么。

我使用的 Apache 服务器的 IP 地址是 192.168.0.8。那么我首先运行 VuGen.exe。选择“增加”按钮，如下图所示：

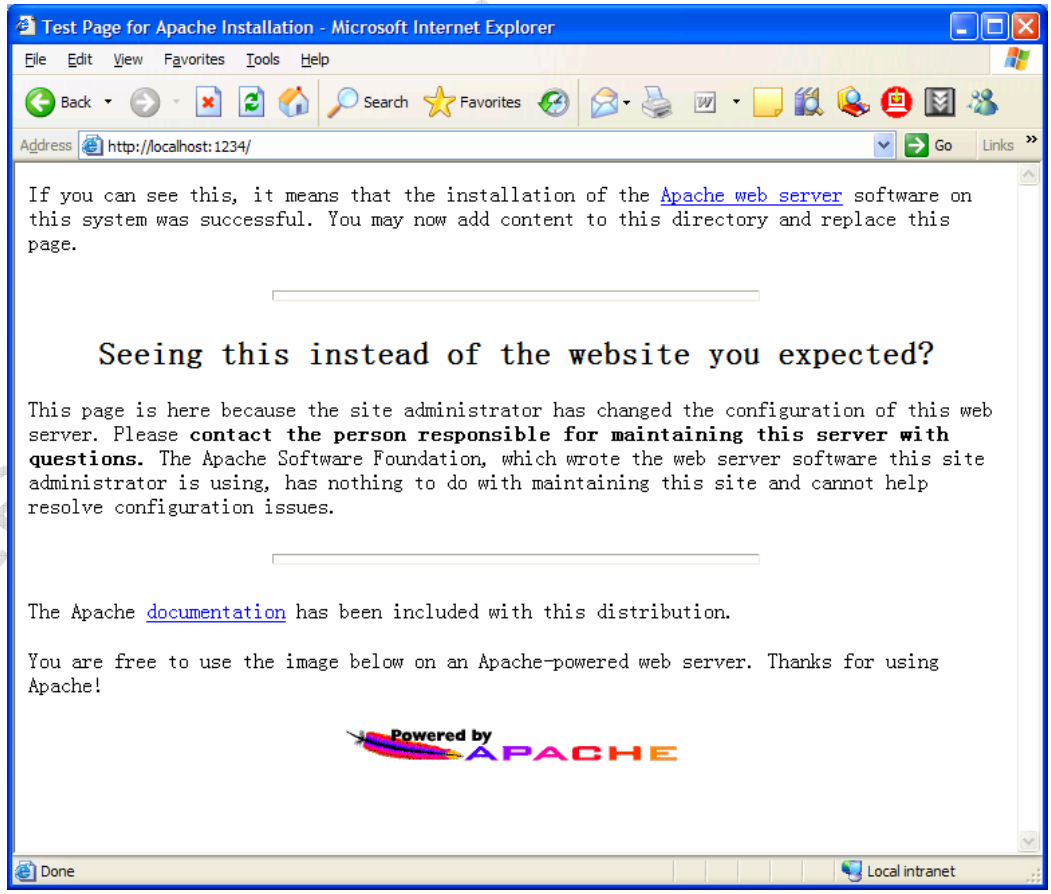


在随后出现的对话框中按照如下方式设置。VuGen 是一个 proxy 代理程序。这种设置的意思是：把转发到本地 1234 端口的 http 请求转发到 192.168.0.8 的 80 端口。

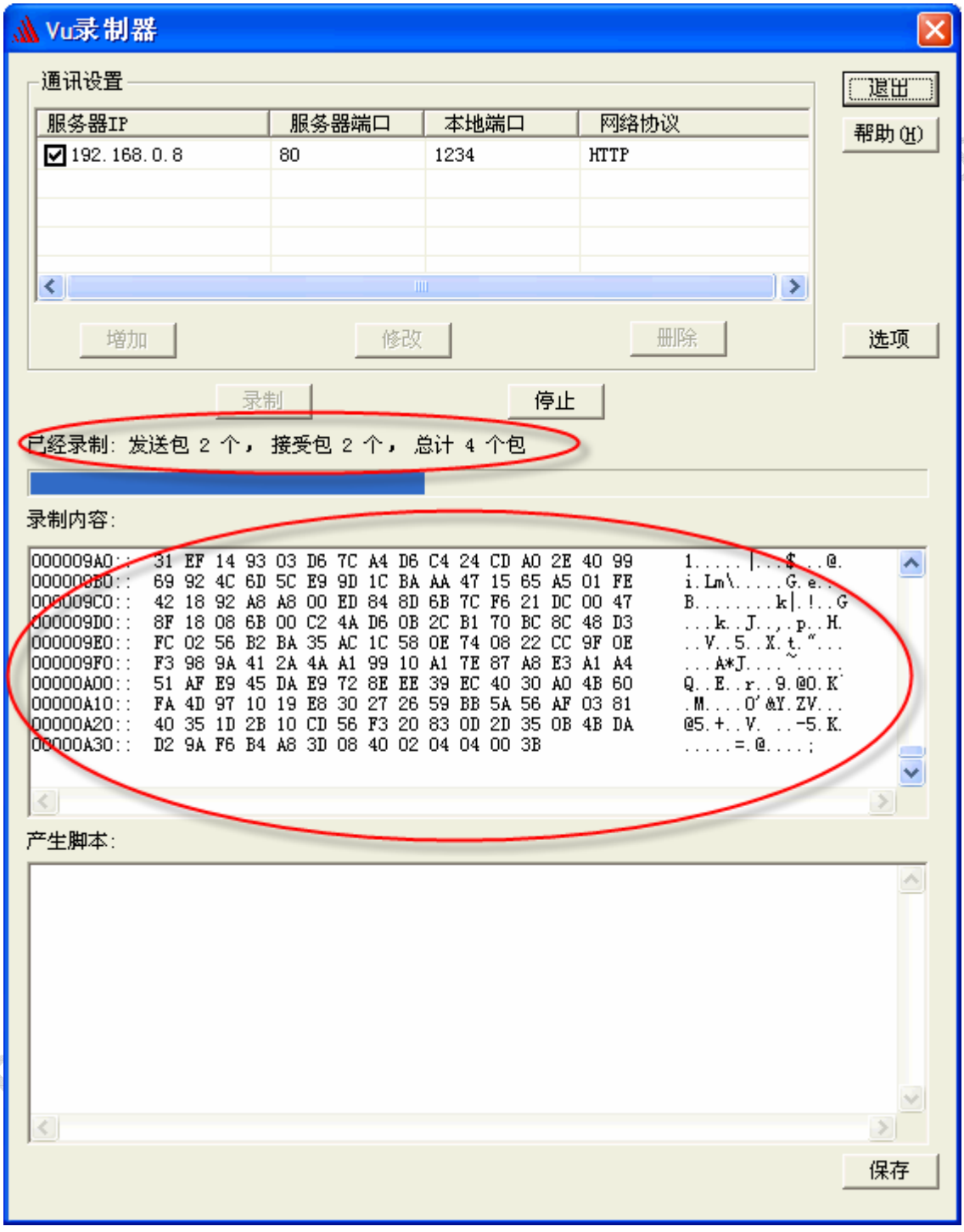


由于 80 端口是 Web 服务器的标准端口。本程序实际上把通过 proxy 截获发往 192.168.0.8 的 http 请求。

一起准备就绪，我们选择VuGen主界面上的“录制”按钮。VuGen开始进入录制状态。然后我们打开浏览器，输入*http://localhost:1234*，出现下图界面。



这表明 VuGen.exe 把发往本机 1234 端口的 http 请求正确地转发到 192.168.0.8 的 80 端口，并把结果返回给浏览器。我们在 VuGen 的主界面上发现如下信息：

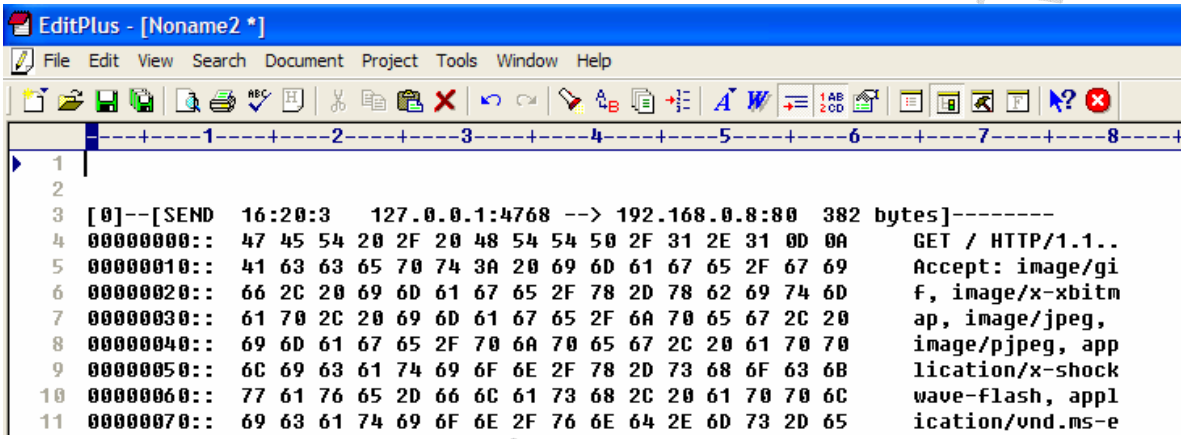


VuGen 向我们报告说：浏览器向 Web 服务器发送了两个请求包，接受了两个请求包，并且把全部的通讯数据都录制下来了。这个是 VuGen 的基本功能。

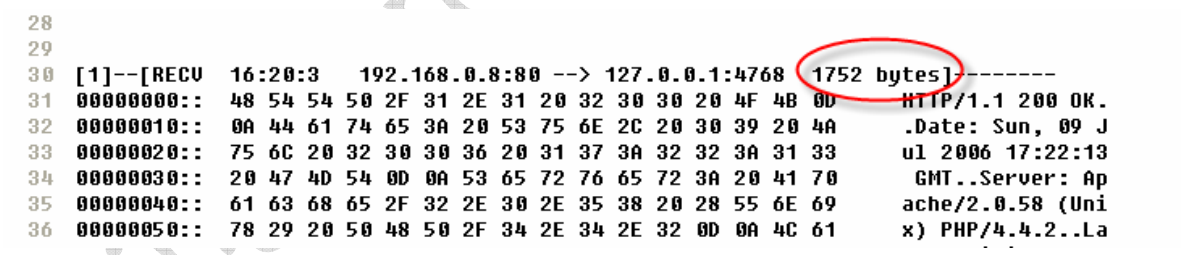
我们把通讯数据包拷贝到一个记事本里面。仔细观察这次HTTP通讯的全部记录。我们发现了 http 通讯的全部轨迹和真相。由于我们要访问的主页是两部分组成的，一部分是文本文件 index.html，一部分是图片文件 apache_pb.gif。所以在一次访问过程中，实际上存在着两次请求动作。对应的通讯如下面所示：

浏览器向 Apache 服务器发送请求 *GET / HTTP/1.1*，Apache 返回的是 index.html。

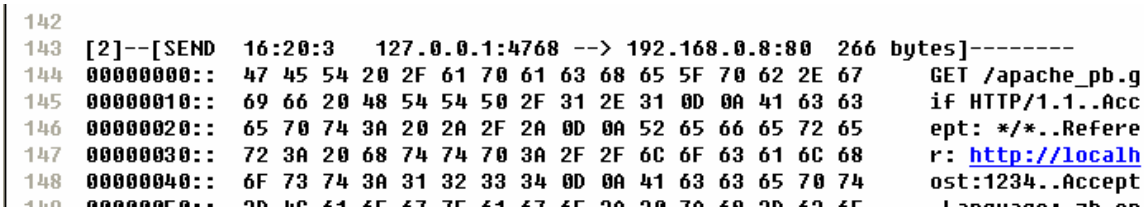
发送请求，要求获取 index.html：



接受 index.html



浏览器接收到 index.html 后，内置的 HTML 语法分析器开始解析 index.html，发现里面还有一个图片链接 apache_pb.gif，于是浏览器再次发送请求 *GET /apche_pb.gif*，要求获得图片文件 apache_pb.gif。如下图所示：

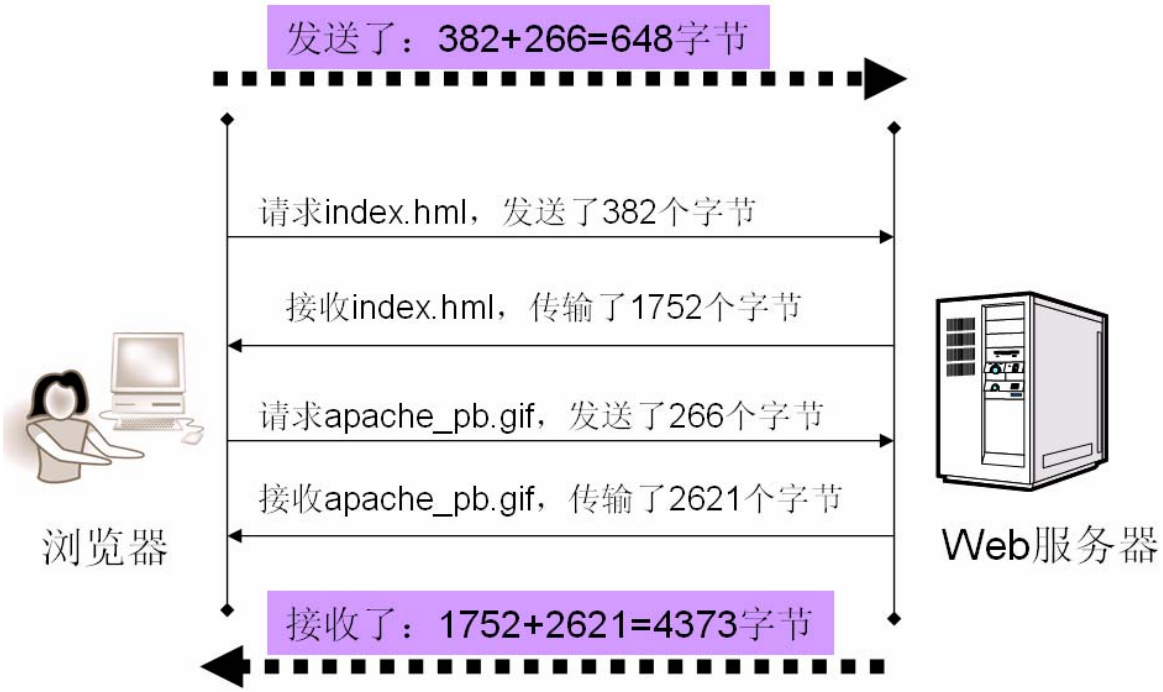


接收的通讯包里面包含了 apache_pb.gif 的全部内容，如下图所示：

```
161
162
163  [3]--[RECV  16:20:3   192.168.0.8:80 --> 127.0.0.1:4768  2621 bytes]--------
164  00000000::  48 54 54 50 2F 31 2E 31 20 32 30 30 20 4F 4B 0D     HTTP/1.1 200 OK.
165  00000010::  0A 44 61 74 65 3A 20 53 75 6E 2C 20 30 39 20 4A     .Date: Sun, 09 J
166  00000020::  75 6C 20 32 30 30 36 20 31 37 3A 32 32 3A 31 33     ul 2006 17:22:13
167  00000030::  20 47 4D 54 0D 0A 53 65 72 76 65 72 3A 20 41 70      GMT..Server: Ap
168  00000040::  61 63 68 65 2F 32 2E 30 2E 35 38 20 28 55 6E 69     ache/2.0.58 (Uni
```

通过这个小小的实验，我们明白了浏览器的工作原理：**当我们访问某个主页 html 文件时，浏览器首先把该 html 文件拿到，然后进行语法分析。如果这个 html 文件还包含图片、视频等信息，浏览器会再次访问后台 Web 服务器，依次获取这些图像、视频文件，然后把 html 和图像、视频文件组装起来，显示在屏幕上。**这个过程和我们的直观印象很吻合。

我们归纳一下这次简单的访问，发生的通讯过程：

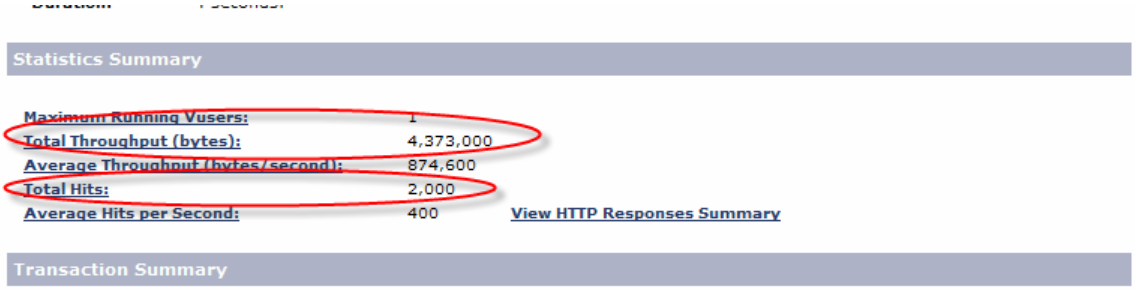


这个就是一次简单的 http 通讯的全部真相。读者还可以仔细阅读 VuGen.exe 捕获下来的全部通讯包，会更加深入地了解 http 协议的一些基本概念。建议读者结合 http 的官方文件来学习。官方文件是 RFC2616，可以从如下网址下载

http://www.17testing.com/bbs/viewthread.php?tid=970&extra=page%3D1

做完这个小实验后，我们提出了一个重要的问题：在 LoadRunner 分析报告中，

有 Total Throughtput(bytes)和 Total Hits 两个重要的指标。我们简称为**吞吐量**和**点击量**。那么这两个指标的真正含义是什么，是怎么计算出来的呢？

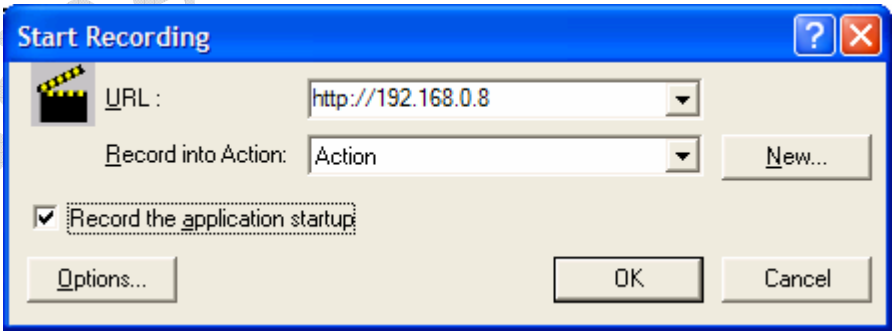


以本次实验为例，用户每访问一次主页，底层通讯发生了两次“请求-接收”，发送了 648 个字节，接收了 4373 个字节。那么在一次访问中，吞吐量是 4373 字节呢？还是 4373+648 字节呢？点击量是一次呢？还是两次？

下面的实验就是利用 LoadRunner 录制上面的访问行为，进行压力测试，看看分析报告中给出的数据到底是多少。

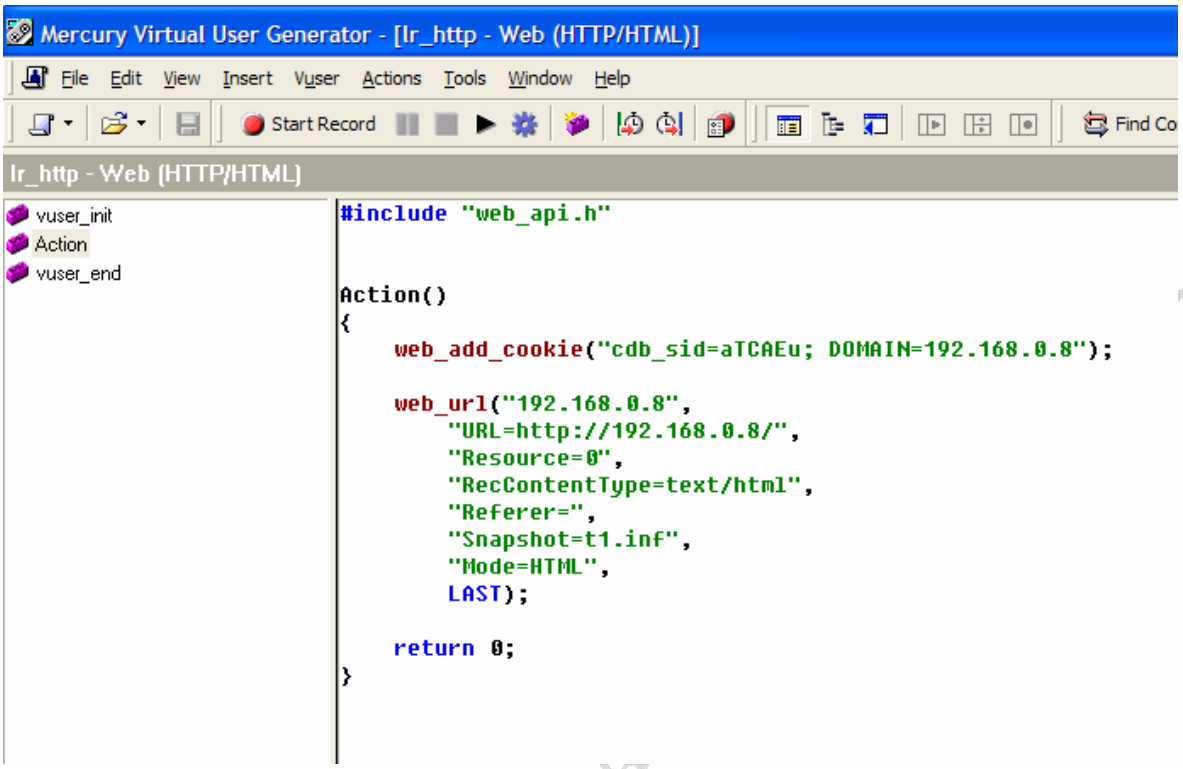
5 吞吐量和点击量深入研究

我们利用LoadRunner录制脚本，运行LoadRunner，选择录制脚本。在录制协议中选择http。由于这样的录制是非常简单的，其过程不在赘述。在开始录制对话框中输入我们要访问的网址：http://192.168.0.8。

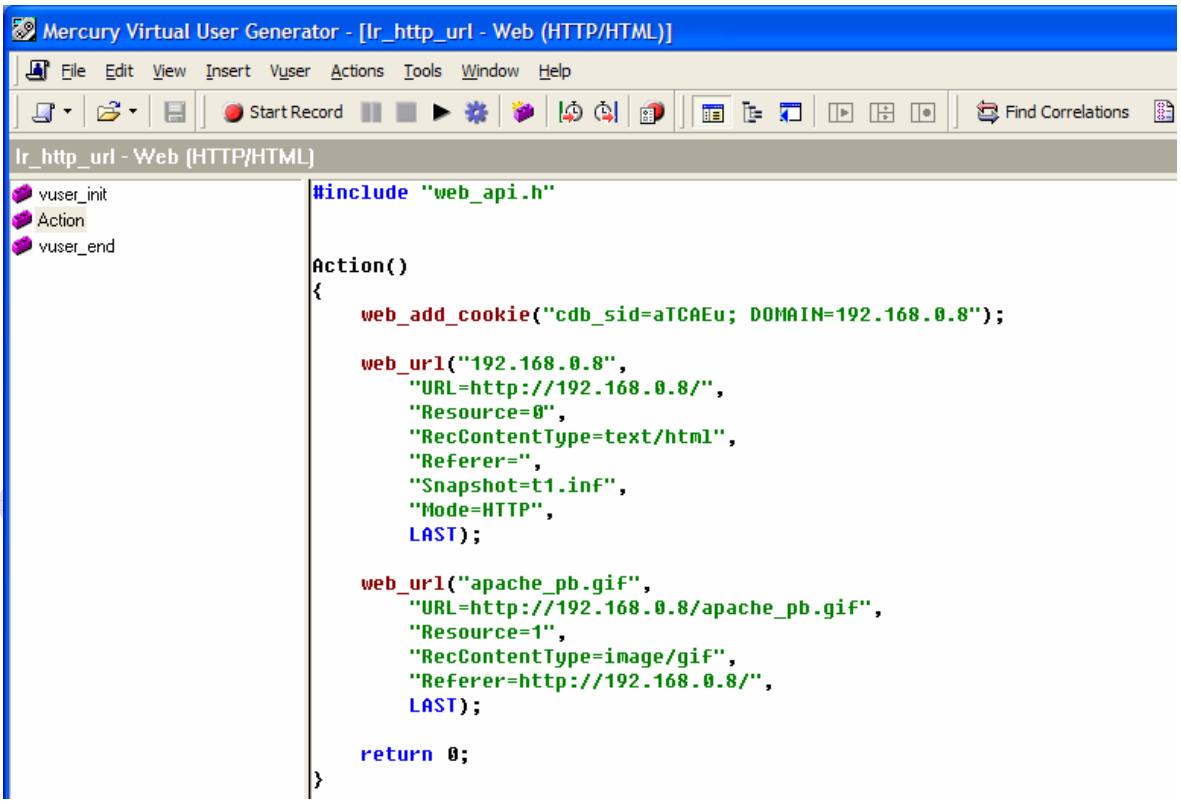


LoadRunner 录制 Web 脚本有 HTML 模式和 URL 模式。我们可以分别录制这两种模式的脚本：

HTML 模式录制的脚本如下图：



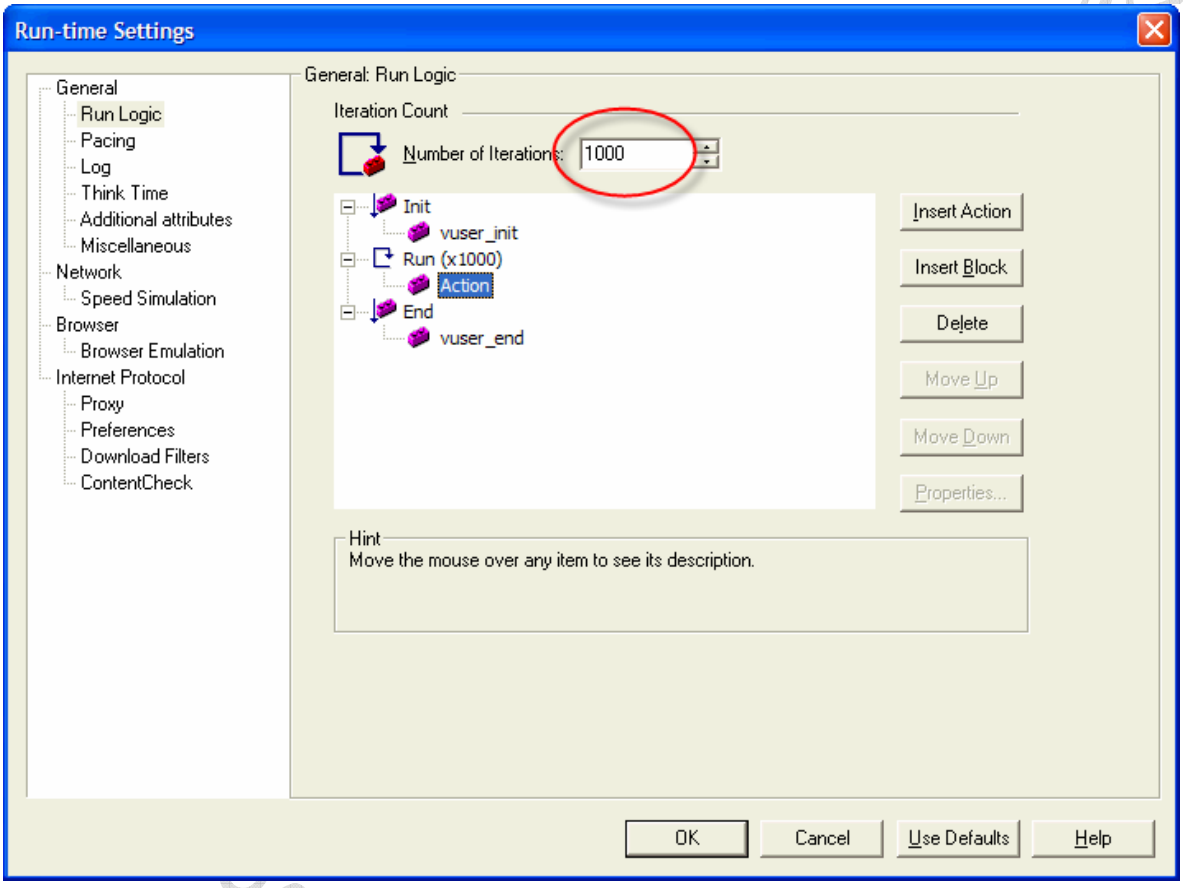
URL 模式录制的脚本如下图所示：



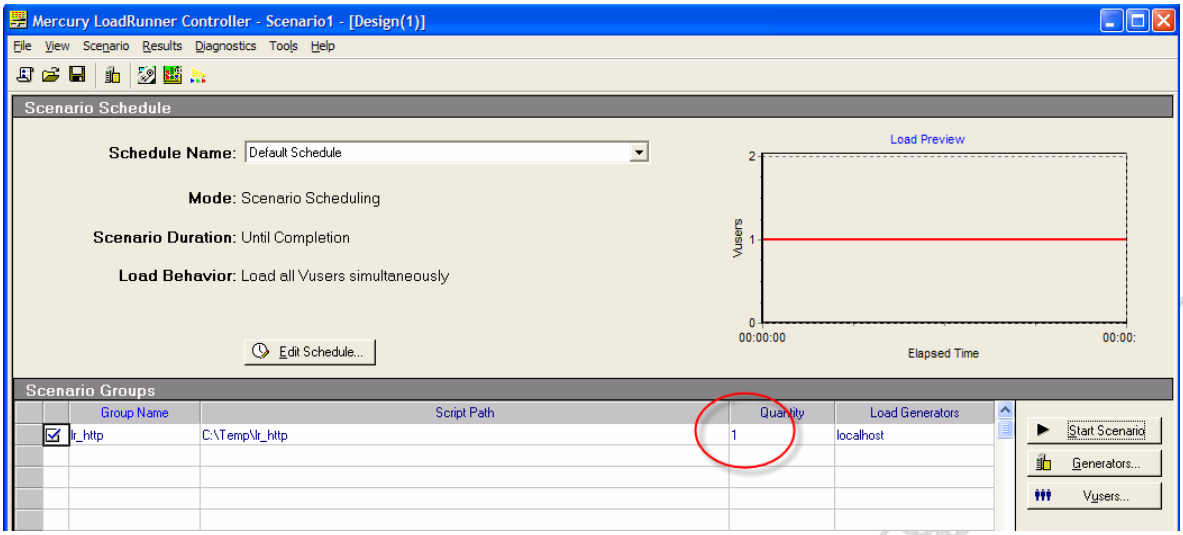
从这两个脚本的区别中，我们可以看出，URL 模式是显示地把一次通讯过程按照

真实发射功能的状况划分成了两个函数。而 HTML 模式把这个过程用一个语句包含了。

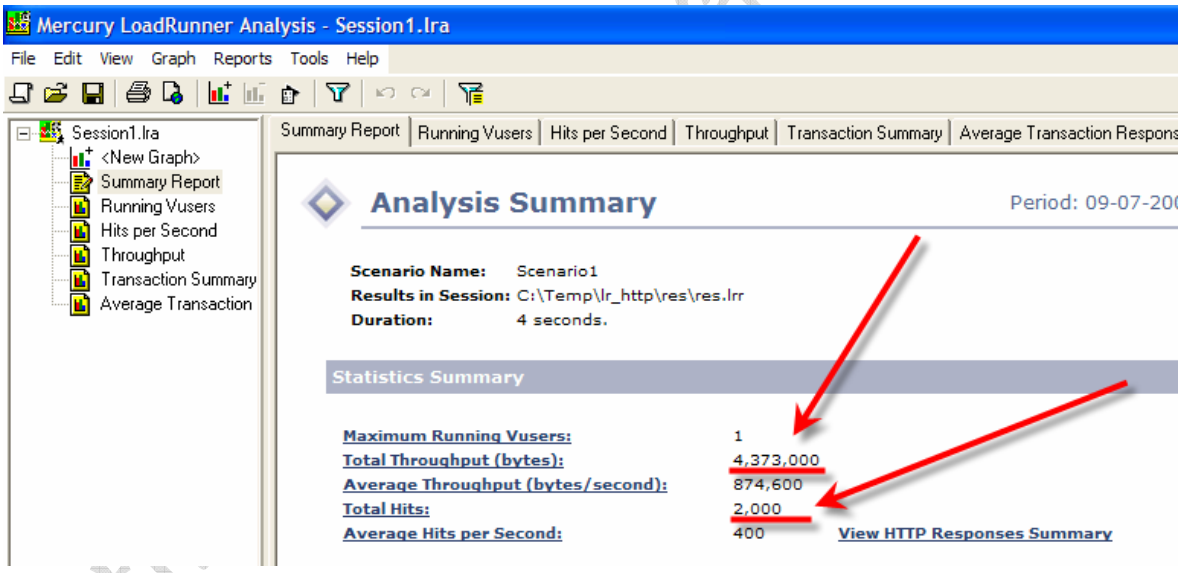
为了让性能测试的数据更加准确，我们决定把测试时间延长一下，每个脚本跑 1000 遍。在 RunTime Setting 里面做如下设置：



然后我们启动 LoadRunner 的 Controller，把脚本放进去，只跑一个虚拟用户，跑一次。由于脚本里面设置了跑 1000 遍，所以一次测试，LoadRunner 将向后台发送请求 1000 次。



然后我们启动压力测试，每次测试完毕，保存结果。然后我们用分析器打开测试结果。在测试结果中，我们发现了如下信息：



我们发现，运行 1000 次后的吞吐量是 4373000 字节，点击量是 2000 次。很显然，每次访问时的吞吐量是 4373000/1000=4373 字节，而每次访问时的点击量是 2000/1000=2 次。用 html 模式的脚本和 url 模式的脚本运行，html 模式的每次访问时的吞吐量是 4374，而 url 模式的每次访问时的吞吐量是 4373。为什么多一个字节，我也不是很清楚。这个问题也许只有 Mercury 才能够回答。

不管怎么样，我们从这个实验中可以得到如下结论：

在 LoadRunner 的报告中，Total Throughput(bytes)的含义是：在整个测试过程中，从服务器返回给客户端的所有字节数量。

Total Hits 的计算不是按照用户的鼠标点击次数计算，而是按照客户端向后台发起了多少次请求计算。譬如：在访问一次页面中，假设该页面里面包含 5 个图片，那么，用户只用点击鼠标一次就可以访问该页面，而 LoadRunner 视该词访问的点击量为 1+5=6 次。

理解了吞吐量和点击量，那么吞吐率和点击率我们就容易理解了。这就是：

吞吐率 = 吞吐量 / 测试时间

点击率 = 点击量 / 测试时间

但是这个测试时间是如何规定的呢？这个我们就不清楚了。

到此为止，我们解决了一些问题，但是我们还有更深的疑惑。这个问题的根源是 LoadRunner 是一个黑盒子，我们无从知道性能测试指标的具体算法。如果 LoadRunner 是一个开源的性能测试工具，我们就很容易知道性能测试指标是如何计算出来的。

6 结束语

本文通过一个小小的实验，深入研究了 LoadRunner 在做 Web 测试过程中两个重要的指标的真正含义。性能测试中最重要的性能结果分析部分是整个性能测试过程中最难的部分。它需要测试人员对被测试系统有深刻的理解。深入地搞性能测试能帮助我们深刻理解计算机技术本身很多原理性的东西。

本文只是作者在搞性能测试过程中一点点的研究成果。作者希望以本文抛砖引玉，能帮助更多的测试人员深入分析测试结果。

—THE END—