**确定性能接收标准**：

确定响应时间、吞吐量、资源利用目标和约束。

在这个系统中，目前可以参考的响应时间性能指标如下：

Excellent (<1s);

Good (1s-2s);

Acceptable (2s-5s), 这个指标作为性能验收标准;

Fair (5s-10s);

Poor (>10s), in which Acceptable (2s-5s).

我们目前的性能验收标准如下，更多的测试结果可以参考：

在并发用户正常工作负载下，(CPU, memory, Disk IO, etc..)的资源利用率应该在80%以下，90%的请求的相应时间应该达到性能验收标准或更好的水平。

没有关键活动小于验收标准中的poor等级。

此外，确认项目成功标准，如果使用性能测试的测试结果超出了验收标准，可以结合配置的设置来评估测试结果，这样可以得到更理想的性能标准。它帮助我们配置测试环境的设置，以使测试结果达到验收标准。

**测试计划**：

确定关键路径，确定代表用户的差异，以及如何模拟差异、定义测试数据和确立应该被收集的度量值。整合这些信息到一个或多个模型实现系统使用，执行，和分析。它包括关键场景和导航路径/用户延迟。

在测试之前，需要为团队制定一个性能需求，可以更好的了解用户应用程序的数量，数据库包含的数据量、关键业务场景的总体分布……等等，在这里是按照下面这个模板来进行性能需求调查的。



**关键场景和导航路径**：

下面的文档描述了关键情形和工作负载特性用于评估Argus定价选择系统的性能。更准确的用户建模、性能测试结果将更可靠。在性能需求基础之上，提供工作负载评估如下：



**时间延迟**：

一个经常被忽视的方面准确的用户建模是用户延迟的建模。

以下是一些建模用户延迟,在实践中我们在性能测试中使用3项:

1. 真正的延迟时间:在理论上，根据用户生产实际行为可以更准确的使用真实的延迟时间,但很难通过生产日志收集这些信息，
2. 生成的延迟时间:大部分的记录工具可以根据在生成记录的时候用户实际等待时间生成延迟时间。它会参考记录延迟时间自动生成延迟时间。如果你想要每个线程暂停请求相同的时间,可以使用这个定时器。
3. 高斯随机定时：这个定时器暂停每个线程请求一个随机的时间,大部分的时间间隔于附近的一个特定的值。周围的变化不断抵消高斯曲线分布。它是更可靠的现实世界。(你可以得到特定的值通过方法2))。

**确定测试数据完成关键场景**：

一旦有了一系列的关键场景,就需要确定测试数据为用户实际完成的与这些场景相关的任务或活动。更准确的测试数据进行建模,将更可靠的性能测试结果。根据性能需求为每个场景设计测试数据。不要忽视准确的用户建模另一个方面：测试数据的建模。或者你会得到乐观或悲观的结果。

|  |  |  |  |
| --- | --- | --- | --- |
| Application | Scenario | Data size(general) | Data size(Max) |
| Drug List | Drug list search result |  |  |
| Drug list entry search result |  |  |
| Drug list details -- entries tab | 20 | 100 |
| Drug list mass change search result |  |  |
| Drug list mass change -- Entries tab |  |  |
| Drug list mass change -- Criteria tab |  |  |

如上所述,一种药物列表组件通常有20项,所以设计测试药物列表组件大约20个条目通常会得到适当的测试结果。在这种情况下,如果您设计测试数据与2或3项,你可能会得到比较乐观的结果

一种药物列表组件最多可以有100个条目,所以设计几个有100条目的组件,将用于压力测试。

在实践中,注意以下提示：

1. 创建一个新的客户,将仅用于应用程序的性能测试。

2、为每个场景创建独立的组件,查看/更新/删除/

3、当录制脚本时,为每个场景记录只选择一个数据项。为了避免影响，不要使用相同的数据。

4、Argus项目,只有添加/删除脚本成功运行时测试数据包含相同数量的条目。所以使用删除条目的操作记录添加条目操作,以确保条目的数量保持原来的时间相同。我可能有很多原因。一个是1条目添加成功后添加/删除/ save按钮的id会改变,另一个是验证如果1条目添加成功后不能安装添加另一个条目。所以使用删除条目的操作记录添加条目,确保操作的条目数量保持原来的时间相同。以防你的脚本可以添加/删除操作无论有多少条目包含,跳过此通知。

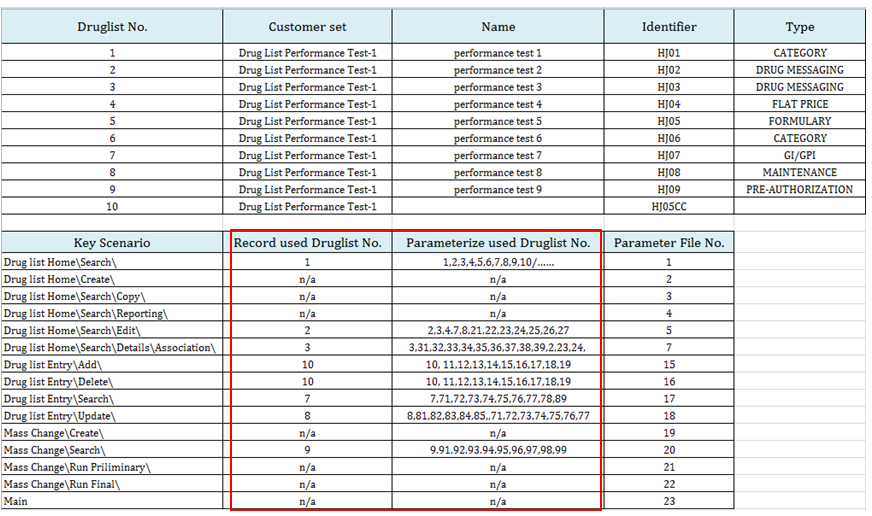
5、使用DB sql来添加和删除测试数据。因为测试数据会因为错误/调试/并发性变成无用的数据。所有这一切可能会使您的脚本在接下来的测试中失败。所以测试数据应该在测试前插入，在测试后删除。

6、在性能测试期间,使用动态测试数据,因为相同的数据反复频繁会导致无效的性能测试结果。这意味着要准备不止一个的测试数据来完成相同的任务或活动相关的相同的关键场景。这也意味着在测试中使用随机数据作为参数。参考常用的参数化函数,使用动态数据。并确保您的脚本成功运行所有的测试数据。

7、确保一些数据用于多个场景。现实主义是一个用户搜索毒品而另一个用户列表编辑它。它是用来在相同的数据测试并发性能问题。

8、一些场景不需要独特的测试数据,比如创建/复制。但是你需要随机数据,以避免重复错误.

例如：



**1.4配置测试环境**

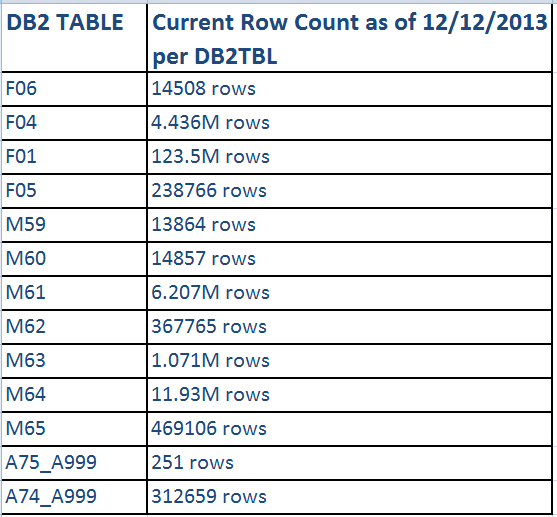
准备测试环境,所需工具和必要资源来执行 战略作为测试功能和组件可用。确保测试环境是必要的资源监控检测。

**准备测试环境**

According to [Identify the Test Environment](#_Identify_the_Test)

**准备目标系统维护数据**

它通常包括在数据库或文件服务器的数据。数据集的量是一个关键因素,影响到性能测试的结果,插入的数据应该有业务意义。数据的记录数量应该和生产环境相似。例如：



注意：

1. “一到多”的关系应该反映真实的业务需求。例如:在Drug list应用程序中,一种drug list一般或最多包含多少药物条目影响系统性能。在benefits 应用程序中,一般或最多有多少个数据范围影响着系统性能。此信息可以计算一般从上面的图表或从性能需求的反馈中得到。
2. 尤其是在设计和调试性能测试时,测试数据库可以显著超载数据。定期检查数据库是否存储大量不符合实际情况的数据，而这些数据的情况正式想要模拟的。

**准备工具**：

Jmeter工具,请参考[Jmeter User Manual](#_Generate_Script_with)

工具的监控关键指标,参考[Monitor Key Metrics](#_Monitor_Key_Metrics)

**1.5实现测试设计**

开发性能测试按照测试设计。我们使用jmt实现测试设计,参考[Jmeter User Manual](#_Generate_Script_with)

**1.6执行测试**

验证和运行测试

1. 验证测试环境和测试脚本

验证测试环境中通过一些冒烟测试,以确保没有发现系统误差。

验证测试脚本,以检查收集的指标是否正确,测试脚本设计是否正确模拟负载特性。

1. 运行测试

性能测试是一个通用术语,可以引用许多不同类型的相关测试。性能测试解决共同类型和种类的性能相关测试，每种性能测试解决特定的问题，但是也有着他们自己各自的优点、风险和挑战。耐力测试、负载测试、压力测试

1. 生成结果

请参阅:

JMeter结果

JMeter-插件结果

监控关键指标

1. 创建基线或基准

**1.7分析结果,报告,重测**

合并和共享结果数据。分析数据得到有用信息,性能分析和调优。性能测试技术解决几种常见类型或类别的绩效测试。对于每个测试,它介绍了如何分析结果。如果发现任何潜在的问题,重新测试会收集到更多有用的信息。

注意：

关键性能测试人员的数学原则:软件开发团队的成员,开发人员、测试人员、管理员和经理都需要知道如何应用数学和解释统计数据,以有效地完成他们的工作。它包括：

平均值、百分比、中位数、正常标准值、偏差值、均匀分布、正态分布、性能测试报告基本面(TODO later)

**1.8 总结**：

现在你有一个通用的性能测试活动,下面将分别详细描述性能测试,并专注于如何使用Jmeter的工具实现每一个片段。

在第二章Jmeter用户手册,介绍了实现与Jmeter测试设计工具,如何执行测试,如何分析结果。在第三章性能测试技术,它介绍了如何执行性能测试在不同的目的和目标详细活动6和7。这一章解决几种常见类型或类别的绩效测试。这是一个通用术语,可以引用许多不同类型的绩效测试:耐力测试,负载测试,压力测试。他们中的每一个地址一个特定问题域和提供自己的利益,风险,和挑战。

**2 Jmeter用户手册**

**2.1介绍**

Apache JMeter是100%纯Java桌面应用程序设计为负载测试客户机/服务器软件(例如web应用程序)。它可用于测试性能等静态和动态资源静态文件,Java servlet,CGI脚本,Java对象,数据库,FTP服务器,等等。JMeter可以用来模拟一个沉重的负载服务器,网络或对象来测试其强度或分析总体性能在不同的负载类型。此外,JMeter可以帮助你回归测试您的应用程序,允许您创建测试脚本与断言验证您的应用程序返回你期望的结果。最大的灵活性,JMeter允许您创建这些断言使用正则表达式。但是请注意,JMeter不是一个浏览器。

Jmeter下载

这个项目是一个Apache Jmeterсustom组插件,不隶属于Apache软件基金会。(http://code.google.com/p/jmeter-plugins/)

Jmeter插件

需要运行Jmeter 2.8或以上JRE 1.6或以上使用插件。

2.2测试计划的要素

测试计划对象有一个名为“功能测试”的复选框。如果选中,它将导致JMeter记录为每个样本从服务器返回的数据。如果你选择了一个文件在您的测试听众,这些数据将被写入文件。这可能是有用的,如果你正在做一个小运行,确保正确配置JMeter,服务器返回预期的结果。结果是,文件将巨大的快速增长,和JMeter的性能就会受到影响。如果你正在做的这个选项应该压力测试(它默认是关闭的)。如果你不记录数据文件,这个选项没有区别。您还可以使用上一个侦听器配置按钮决定什么字段保存。

线程组

测试片段

事务控制器

监听器

定时器

断言

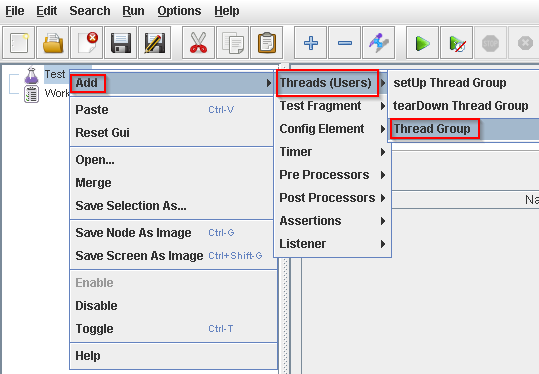
BeanShell取样器

Http代理服务器

**2.3构建一个Web测试计划**

1、增加线程组

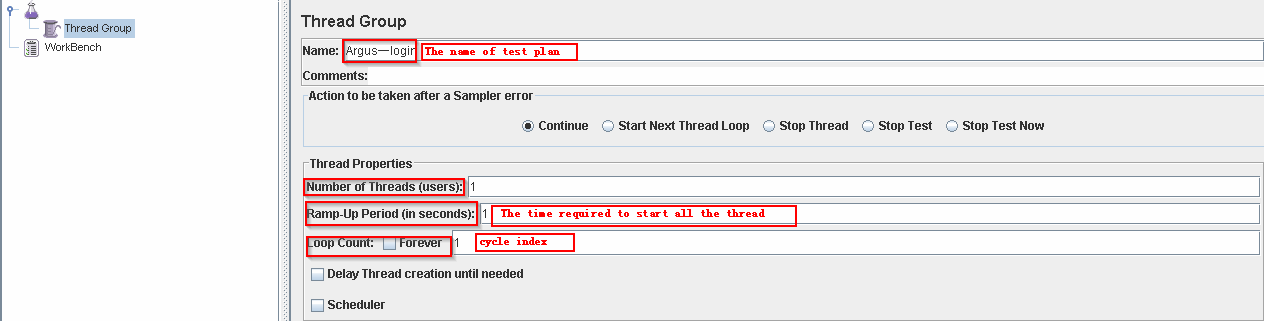
你想做第一步每JMeter测试计划是添加一个线程组元素。线程组告诉JMeter的用户数量你想模拟,用户应该多久发送请求,他们应该发送的请求的数量。继续添加ThreadGroup元素首先选择测试计划,点击你的鼠标右键添加菜单,然后选择add - - > ThreadGroup。您现在应该看到线程组元素在测试计划。如果您没有看到的元素,那么“扩大”测试计划树通过单击测试计划元素



2、线程组的参数设置

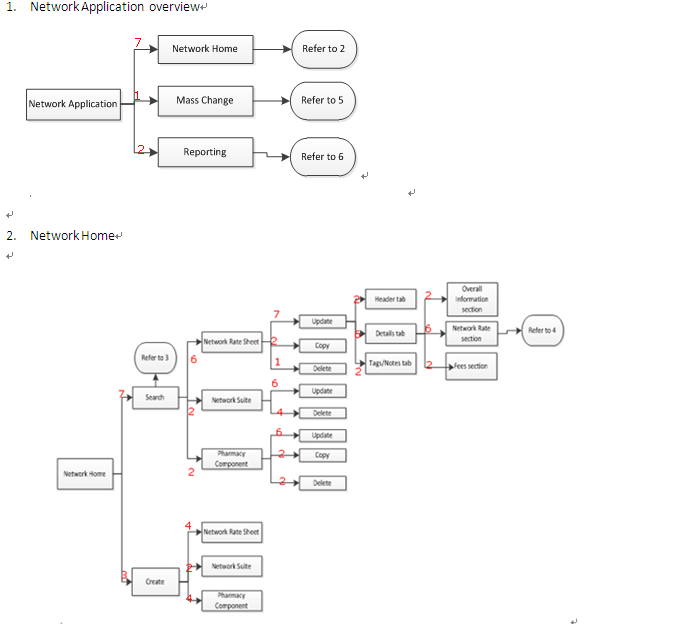
首先提供一个更具描述性的名字我们的线程组。在name字段中,输入JMeter用户。接下来,增加用户(称为线程)的数量为1。下一个字段的过渡时期,离开的默认值1秒。这个属性告诉JMeter多久开始每个用户之间的延迟。例如,如果你输入一个5秒的过渡时期,JMeter将完成你所有的用户开始在前5秒结束。所以,如果我们有5用户和一个5秒的过渡时期,然后开始用户之间的延迟将1秒(5用户/用户每秒5秒= 1)。如果你将值设置为0,那么JMeter将立即开始你所有的用户。

最后在循环计数字段中输入值为2。这个属性告诉JMeter多少次重复测试。如果你输入一个值1循环计数,然后JMeter将只运行您的测试一次。选择永远复选框，JMeter会反复运行测试计划。

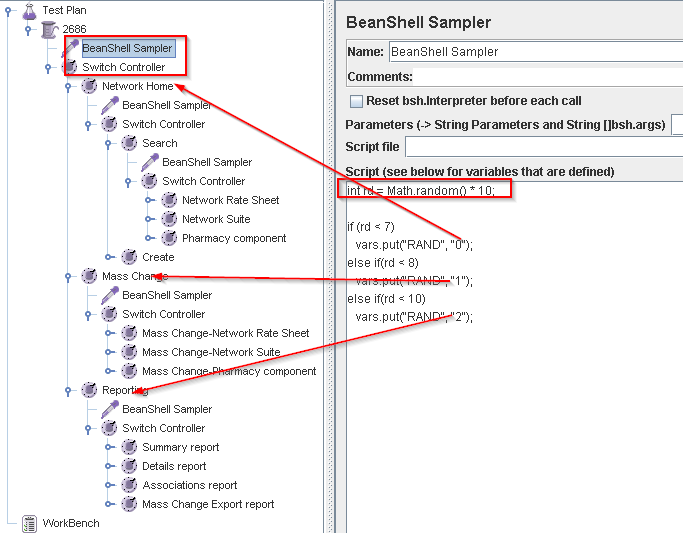


注意:(在大多数应用程序中,您必须手动接受改变你在控制面板。然而,在JMeter,控制面板,你把它们自动接受您的更改。如果你改变一个元素的名称,这棵树将被更新为你离开后控制面板后的最新的文本(例如,在选择另一个树元素)。

1. 生产情况和工作负载

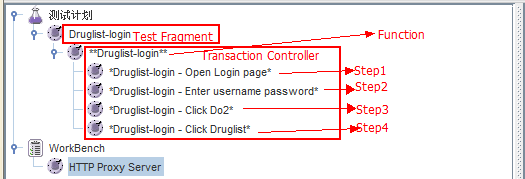


使用BeanShell取样器和开关控制器来模拟用户的活动,首先使用Math.random()1 - 10之间随机生成的数据,如果数量< 7,然后执行网络,如果< 8,然后执行质量变化,如果< 10日执行报告。



4、确保脚本的实施

在录制之前,你应该确保监视和有多少步骤的测试用例应该分成。事务控制器用于分隔不同的步骤,这将使它容易记录。注意:为一个典型的名称为每个步骤使报告数据清楚地看到像下图:【项目名称】(函数名)——(步骤名称)。这是非常重要的测试和分析报告期间。



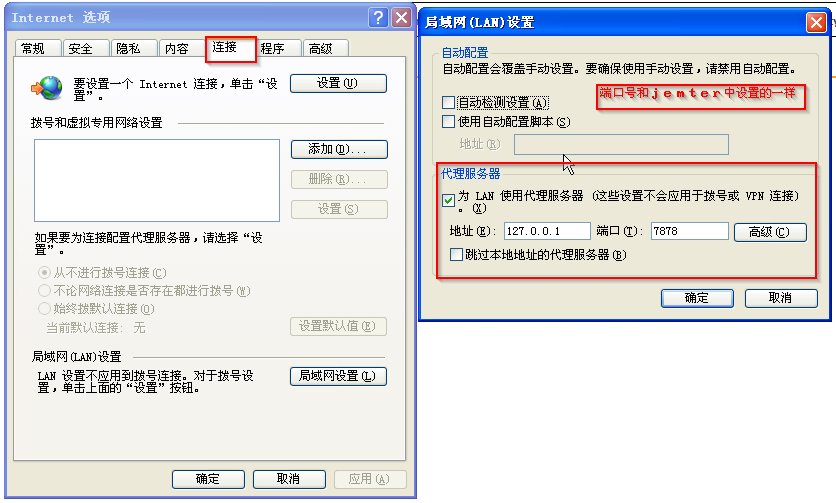
注意:改变事务控制器默认配置下,不花时间包括思考时间到一步。



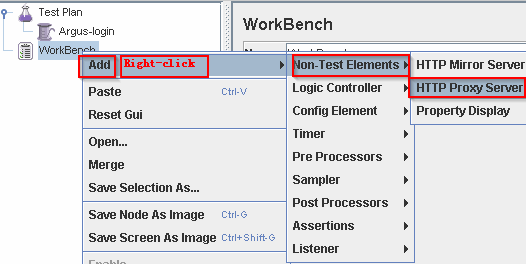
1. 录制脚本

设置浏览器代理服务器

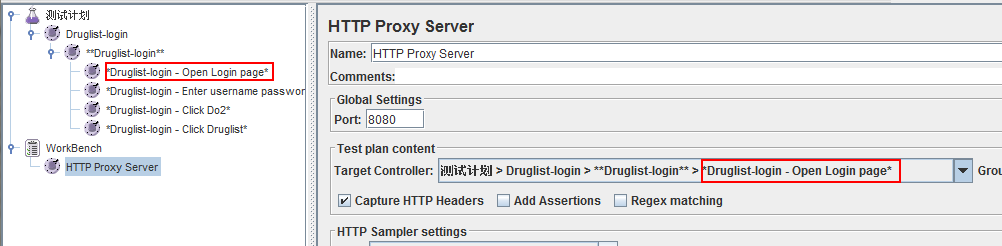
打开浏览器,开放的互联网选项,设置代理服务器的局域网(LAN)设置:localhost,端口设置代理服务器的端口:7878。



WorkBench-非测试元素-http代理服务器



端口:代理服务器监听这个端口,设置成7878，目标控制器



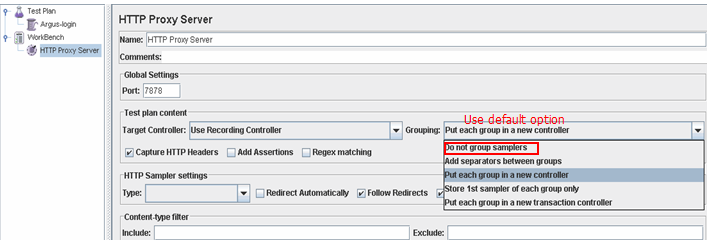
分组:请求数据包。“分组”的概念总结包的请求数量,我们作为url请求和组一个详尽的清单的所有请求。

“Do not group samplers”:一个详尽的清单的所有请求组；

“Add separators between groups”:不同的操作添加一个虚拟的分界线；

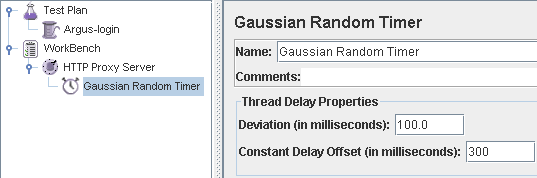
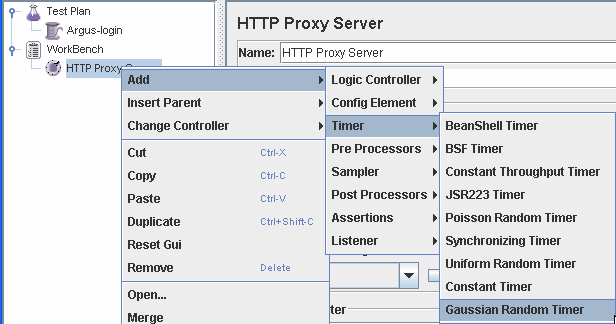
“Put each group in a new controller”:根据控制器的输出在执行。

“Store 1st sampler of each group only”:这个选项有利于一个url请求,实际的http请求很多次,因为我们经常不关心这些请求的后者。

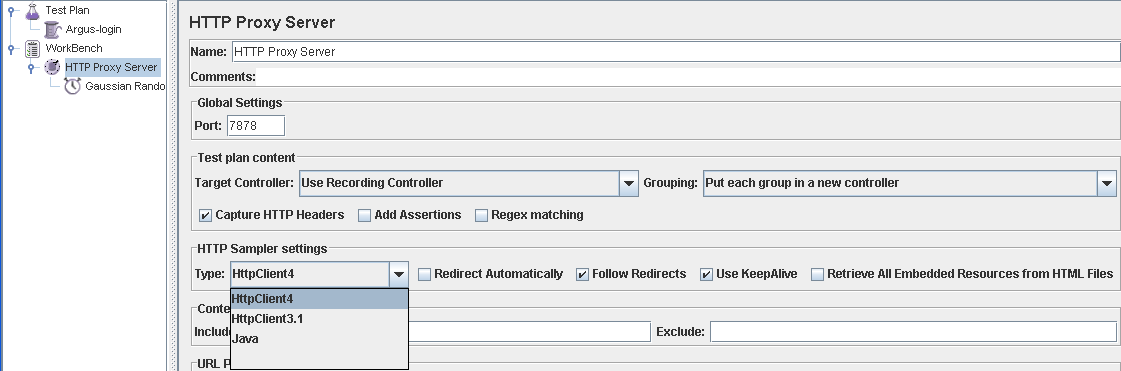


等待时间-添加高斯随机计时器

高斯随机定时添加到HTTP代理服务器(用于通知Jmeter添加一个计时器时自动生成HTTP请求)。计时器将导致取样器被推迟。规则延迟计时器的影响在一个访问请求响应延迟指定的时间,下一个抽样访问请求将被发送。如果高斯随机定时添加到HTTP代理服务器,它应该在一个恒定的延迟补偿(恒定延迟抵消)添加$ { T }(自动引用记录延迟时间)。

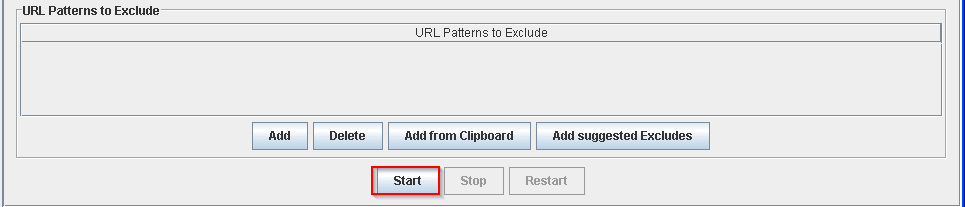


设置HTTP取样器设置



启动HTTP代理服务器

HTTP访问代理服务器配置后,单击“开始”,代理服务器将开始记录接受请求。



浏览操作

在浏览器地址栏中输入地址和相关操作记录,录音完成后,停止HTTP代理服务器,右键单击组件将被记录在记录控制器组件保存到文件中供以后重用也不忘了恢复浏览器的代理服务器设置。

当录制Https

原则:我们记录了代理充当中间人,所以获得双方的信任:也就是说,浏览器和服务器。

服务器,我们需要设置信任库,服务器证书导入到密钥存储库服务器端系统的密钥存储库。房地产和JMeter。属性集,代理将服务器端提供的默认证书。

然后是代理需要向浏览器提供一个证书,需要包含公钥、私钥和证书密钥存储库。JMeterproxy。jks这个密钥库来满足这个要求。密钥存储库相关属性这个密钥库的位置。

注:以上使用两个不同的密钥存储库需要设置不同的属性。

打开JMeter代理,这个时间浏览其他HTTPS站点JMeter将为我们提供一个证书(通过查看证书可以找到),我把Hengtian邮箱招商银行网站测试,并可以记录脚本。

疑问:虽然录制脚本,但控制台将抛出一个异常,这种异常我们记录恒天邮箱和招商银行网上银行会面时,但结果可以被记录下来。仍然不确定除了脚本记录情况可能JMeter HTTPS代理不好保持连接到远程主机的状态会主动断开(是否为了防止中间人攻击的人吗?),远程断开浏览器将断开,因为在会话JMeter需要re-handshake不一致。

这种情况下记录有影响力的过程中,影响只记录将抛出异常。重播应该没问题,因为只有两方在回放过程中,只需要信任服务器很好,不需要欺骗客户。

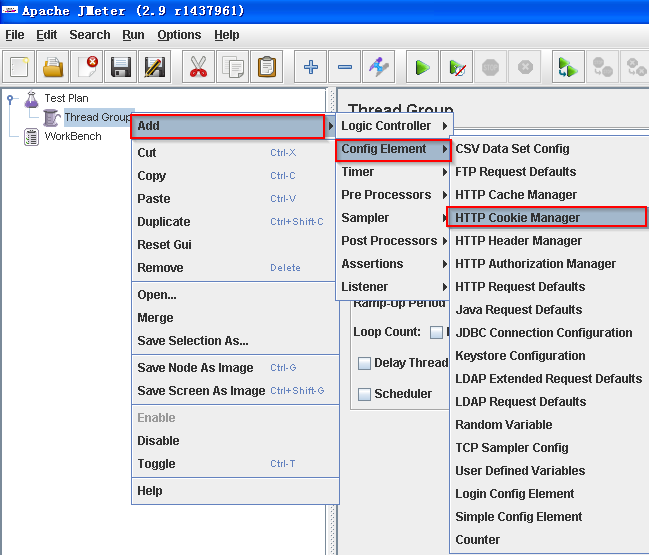
2.4参数化

第一的主要目标是确保你的脚本可以运行成功,Argus应用程序中有一些来自响应的动态数据,如果你不做参数化工作,脚本不能运行成功,所以你必须做参数化的工作,以确保脚本运行。

三个主要问题

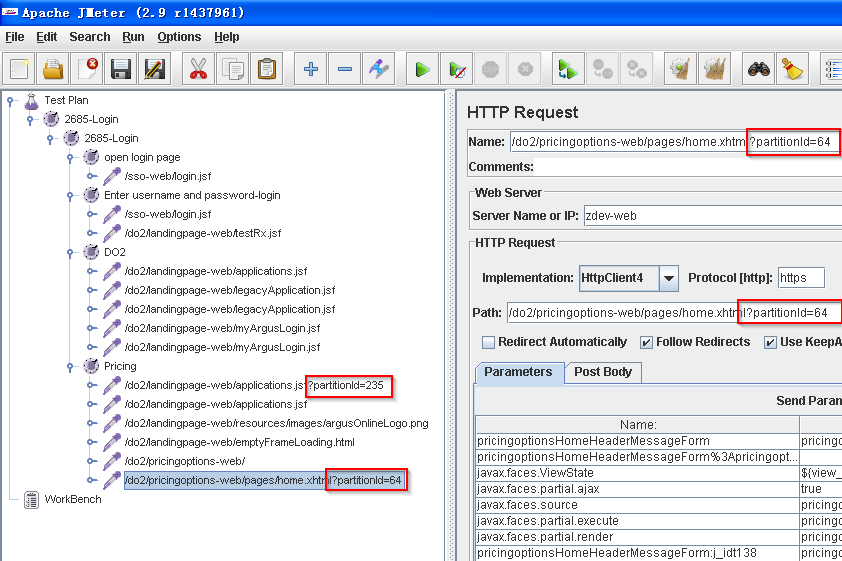
JSESSIONID

识别JSESSIONID,你只需要添加一个HTTP Cookie管理器,请确保HTTP Cookie管理器在您想要运行的脚本中是在父级水平的。

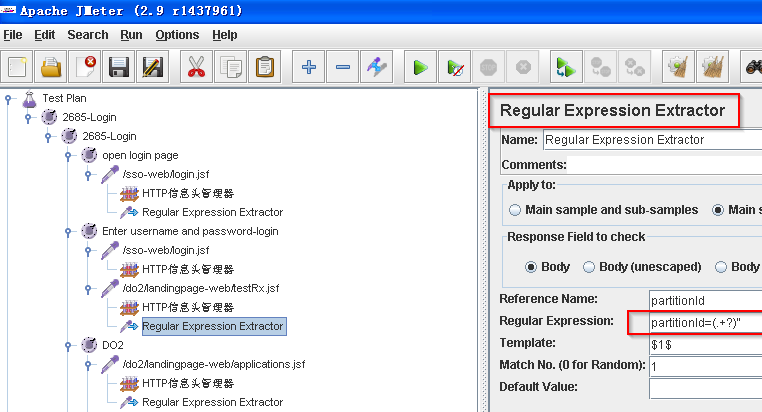


PartitionID

PartitionID 是JSESSIONID的变形,不同操作有不同的PartitionID,它确保不同的操作彼此独立的,如图。

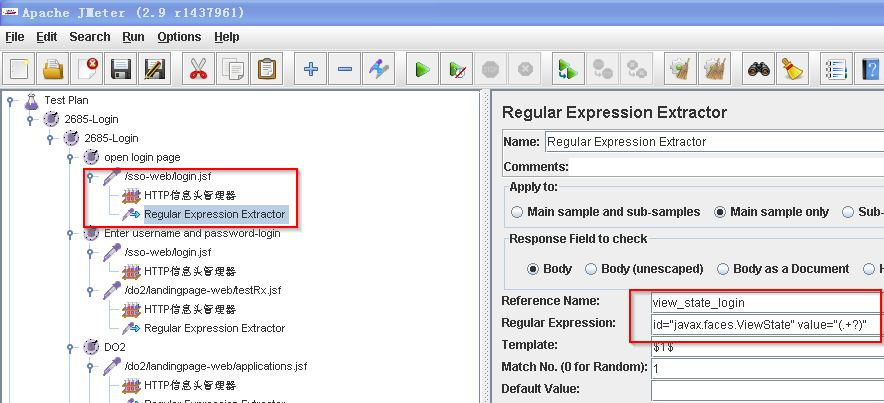


partitionID来自响应,需要一个正则表达式提取器来得到它,partitionID让你在同一个会话。下面的图片显示了一个脚本,添加正则表达式提取器。



ViewStatus

登录的状态,不同的页面有不同ViewStatus。你必须得到的值ViewStatus从不同的响应,并设置以下请求ViewStatus这将确保你的身份登录。



Duplication of data

错误出现在Argus系统当你填写信息在数据库中存在一些文本框,例如现有的ID或名称。但JMeter在这种情况下不会给任何错误消息。你必须确保数据填写不重复。请确保您得到正确的响应是否去看结果树的详细信息视图。

Associated date range

如果数据范围相关联,这是无法通过系统验证,如果你不够小心的话,你可能会犯这个错误。请确保您得到正确的响应是否去看结果树的详细信息视图。

参数化数据:用户名/密码(登录),编辑数据,搜索条件…独特的数据:组件名称,标识符…不应重复。

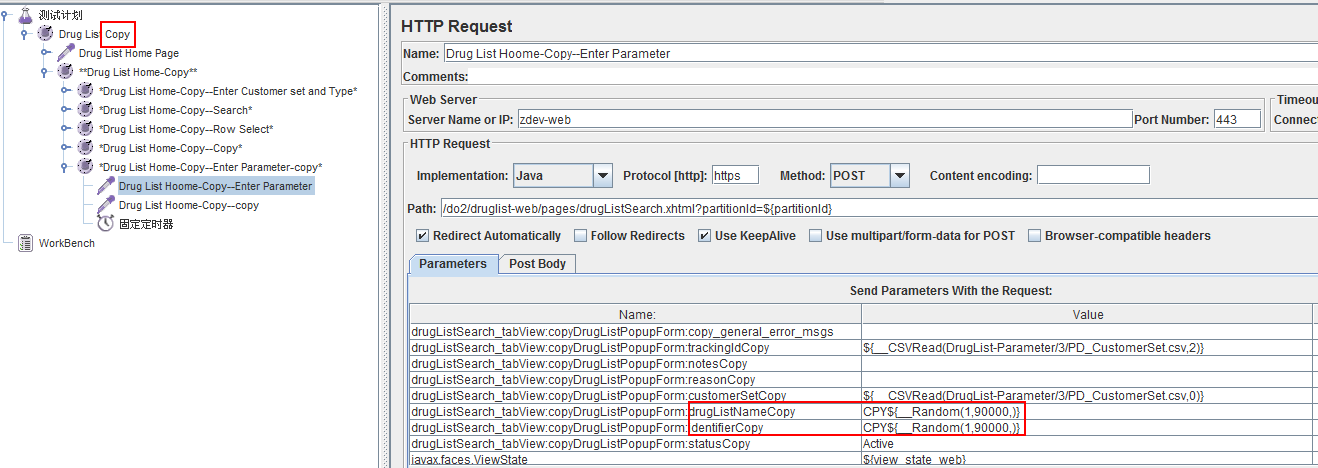
使用Jmeter函数获取参数值,具体调用方法如下：

*${\_\_Random(,,)}，${\_\_threadNum}，${\_\_CSVRead(,)}，${\_\_StringFromFile(,,,)}.*引用jmeter使用手册,通过菜单“选项”- >“功能帮助对话框”,可以发现在jmeter”功能帮助”弹出框的功能。

第一个参数$ { \_\_Random(,,)}是随机的下限，第二个参数是随机的上限，第三个参数存储随机变量的名称。第一个参数$ { \_\_CSVRead(,)}是文件名，第二个参数是文件列表(从0开始列号)，第一个参数$ { \_\_StringFromFile(,,,)}是文件名，$ { \_\_StringFromFile(,,,})方法没有指定的参数列在文件中，所以$ { \_\_StringFromFile(,,,)}可以阅读只包含一个文件列表。

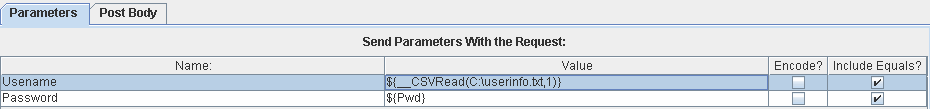
${\_\_Random(0,10,)}

例如:在复制/创建场景中,组件名称、标识符不应重复,使用随机数据,如下图：



${\_\_CSVRead(c:\userInfo.txt,1)}

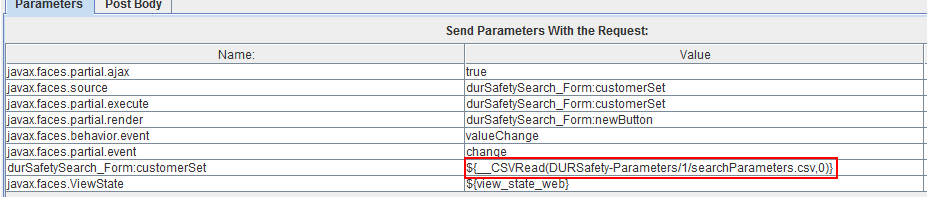
从txt文件中读取数据，



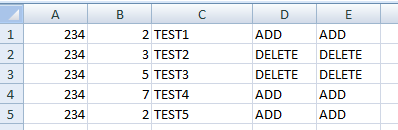
保存用户名和密码进入用户信息。txt如下,确保它有多个行使用动态数据：

oriana,123456  
admin,admin  
dandan,123456

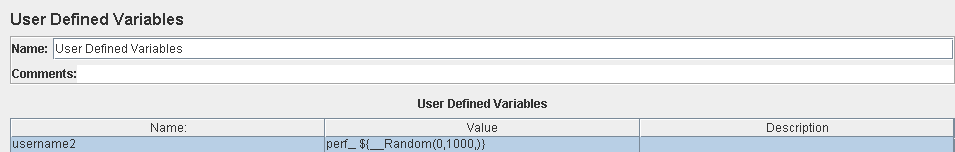
${\_\_CSVRead(searchParameters.csv,0)}



从searchParameters读取数据。csv如下,所有行读取默认序列和循环。确保它有多个行使用动态数据。



添加“配置元素”- >“用户定义的变量”



文件名——设置文件路径

文件编码- file编译方法,一般是空的

变量名称————文件中的所有列项目参数;每个参数之间用逗号;参数项的名称应按照HTTP请求中的参数项。

使用逗号分隔符——如果文件,填写逗号;如果使用标签,填写\ t;

回收利用EOF吗?——True=当读取文件结束,然后开始阅读该文件；FALSE=阅读文件结束时,停止阅读该文件.

停止线程EOF吗?——当EOF回收吗?是假的,TRUE=当读取文件结尾巴,停止这一过程,

添加监听器

http是无状况的和谈，客户每次读取web页面时，办事器都打开新的会话，并且办事器也不会主动保护客户的高低文信息，那么要怎么才干实现网上市廛中的购物车呢，session就是一种保存高低文信息的机制，它是针对每一个用户的，变量的值保存在办事器端，经由过程SessionID来区分不合的客户，session是以cookie或URL重写为根蒂根基的，默认应用cookie来实现，体系会发明一个名为JSESSIONID的输出cookie，我们叫做session cookie，以差别persistent cookies，也就是我们凡是所说的cookie，重视session cookie是存储于浏览器内存中的，并不是写到硬盘上的，这也就是我们刚才看到的JSESSIONID，我们凡是情是看不到JSESSIONID的，然则把浏览器的cookie禁止后，web办事器会采取URL重写的体式格式传递Sessionid，我们就可以在地址栏看到sessionid=KWJHUG6JJM65HS2K6之类的字符串。  
熟悉打听了道理，我们就可以很轻易的辨别出persistent cookies和session cookie的差别了，网上那些关于两者安然性的评论辩论也就一目了然了，session cookie针对某一次会话而言，会话停止session cookie也就跟着消散了，而persistent cookie只是存在于客户端硬盘上的一段文本（凡是是加密的），并且可能会遭到cookie诳骗以及针对cookie的跨站脚本进击，天然不如session cookie安然了。  
凡是session cookie是不克不及跨窗口应用的，当你新开了一个浏览器窗口进入雷同页面时，体系会付与你一个新的sessionid，如许我们信息共享的目标就达不到了，此时我们可以先把sessionid保存在persistent cookie中，然后在新窗口中读出来，就可以获得上一个窗口SessionID了，如许经由过程session cookie和persistent cookie的连络我们就实现了跨窗口的session tracking（会话跟踪）。  
在一些web开辟的书中，往往只是简单的把Session和cookie作为两种并列的http传送信息的体式格式，session cookies位于办事器端，persistent cookie位于客户端，可是session又是以cookie为根蒂根基的，熟悉打听的两者之间的接洽和差别，我们就不难选择合适的技巧来开辟web service了。