

Jmeter 用户手册

整理人：大傻

2008-12-12

1 介绍

Apache JMeter 是 100% 纯 JAVA 桌面应用程序，被设计为用于测试 CS 结构的软件。同时也可以用来测试静态和动态资源的性能，例如：静态文件，Java Servlets，CGI Scripts，Java Object，数据库和 FTP 服务器等等。JMeter 可用于模拟大量负载来测试一台服务器，网络或者对象的健壮性或者分析不同负载下的整体性能。

同时，JMeter 可以帮助你对你应用程序进行回归测试。通过你创建的测试脚本和 assertions 来验证你的程序返回了所期待的值。为了更高的适应性，JMeter 允许你使用常规表达式来创建这些 assertions。

2 开始

2.1 安装

下载安装最新的版本，将会包含所有你需要建立和运行 Web，FTP，JDBC 和 JNDI 测试的文件。如果你想进行 JDBC 测试，你需要使用合适的 JDBC driver。JMeter 中不包含 JDBC drivers。其他可能需要下载的软件：

Bean Shell

Java Activation Framework – needed for Java Mail

Java Mail – needed for mail visualiser and SOAP test

JMS – for JMS samples

General Java Kit

接下来一步，就是浏览“建立测试计划”一章来熟悉 JMeter 的基本操作，例如：添加和删除元素。最后，浏览适当的章节来学习如何建立不同种类的测试计划。例如，如果你对 Web 应用程序的测试比较感兴趣，就浏览“建立一个 Web 测试计划”章节。其他种类的测试计划建立的章节包括：JDBC，FTP 和 JNDI。

一旦你熟悉了建立和运行 JMeter 测试计划，就可以开始熟悉各种配置参数，例如：timers，listeners，assertions 和其他的参数，来帮助你更好地控制你的测试计划。

2.2 系统要求

JMeter 1.8 以上的版本需要 JDK1.4 以上的版本支持运行。JMeter 可以运行在如下操作系统上：Unix，Windows 和 Open VMS。

I 安装 JDK：

I JAVA 的环境变量设置

1. 使用鼠标右击“我的电脑”->属性->高级->环境变量
2. 系统变量->新建->变量名：JAVA_HOME 变量值：d:\jdk
3. 系统变量->编辑->变量名：Path 在变量值的最前面加上：%JAVA_HOME%\bin;
4. 系统变量->新建->变量名：CLASSPATH 变量值：。；

I 安装 Tomcat

CATALINA_HOME=d:\tomcat

User Manual

2.3 可选项目

如果你打算进行 JMeter 开发或者使用 Sun 的 Java 标准扩展包，你需要如下所列的可选包：

2.3.1 Java Compiler

需要 1.4 以上的 JDK 版本

2.3.2 SAX XML Parser

JMeter 默认使用 Apache 的 Xerces XML parser。如果你想使用一个不同的 XML parser，将第三方的 parser 包放入 JMeter 的 classpath 中，然后使用设置的 parser 包的全名更新文件 jmeter.properties

2.3.3 Email Support

JMeter 拥有有限的邮件功能来发送测试结果到指定邮箱中。应用邮箱功能，需要将 Sun 的 JavaMail 包放入 JMeter 的 classpath 中并激活。

2.3.4 SSL Encryption

为使用 HTTPS 测试 web 服务器，JMeter 需要设立 SSL 机制。将所需要的 SSL 类包放入 JMeter 的 classpath 中，更新 jmeter.properties 来注册 SSL Provider。可以使用 SSL Manager 用来管理这些证书。注意：在 JDK 1.4 以上版本，SSL 已经作为标准库加入。

2.3.5 JDBC Driver

你需要将你的数据库供应商提供的 JDBC drivers 添加到 classpath 中进行 JDBC 测试。

2.3.6 Apache SOAP

Apache SOAP 测试需要 mail.jar 和 activation.jar。只需要将下载的两个 jar 文件复制到 JMeter 下的 lib 目录中，JMeter 会自动的识别和使用它们。

2.4 运行

运行 JMeter，使用 jmeter.bat (windows) 或 jmeter (UNIX)。

2.4.1 JMeter 的 classpath

JMeter 自动在 /lib 和 /lib/ext 目录中的 jar 中的类。如果你想添加其他 JAR 文件到 JMeter 的 classpath 中，你必须复制他们到 lib 目录中。如果你开发了一个 JMeter 特定组件，生成 jar 包后复制到 lib 目录下的 ext 目录中。而且，你也可以将有效的 jar 文件安装到 \$JAVA_HOME/jre/lib/ext 目录中。

值得注意的是设置 CLASSPATH 环境变量没有任何作用。这是因为 JMeter 使用 "java -

jar”开始，java 命令自动忽略 CLASSPATH 值，同时-jar 选项的使用将屏蔽掉-classpath 和-cp 选项。

2.4.2 使用代理服务器

如果你在防火墙或者代理服务器后进行测试，你需要给 JMeter 提供防火墙和代理服务器的 hostname 和 port 号。在命令行运行 jmeter.bat 或 jmeter 命令并使用如下参数：

- n** -H 代理服务器的 hostname 或者 IP 地址
- n** -P 代理服务器的 port 号
- n** -u 代理服务器验证使用的用户名
- n** -a 代理服务器验证使用的用户密码

【示例】

```
jmeter -H yahoo -P 8080 -u test -a password
```

2.4.3 非 GUI 模式

对于那些非交互的测试，你可以使用非 GUI 的模式运行 JMeter。使用如下命令：

- n** -n 指定的 JMeter 运行在 non-GUI 模式下
- n** -t 包含测试计划的 JMX 文件的名称
- n** -l 用于记录简单结果的 JTL 文件名称。
- n** -r 运行所有的在 jmeter.properties 中指定的远程主机。(或在命令行中提供的覆盖属性提供的远程主机名。

也可以同时提供防火墙或者代理服务器的信息

- n** -H 服务器名或 IP 地址
- n** -P 端口号

【示例】

```
jmeter -n -t my.jmx -l log.jtl -H yahoo -P 8800
```

2.4.4 服务器模式

为了对分布式测试，可以运行在服务器模式下，然后通过 GUI 来控制每个服务器。运行 jmeter-server/jmeter-server.bat，附加下列可选项：

- n** -H 代理服务器的 hostname 或 IP 地址
- n** -P 端口号

【示例】

```
jmeter-server -H yahoo -P 8000
```

2.4.5 通过命令行来重载 Properties

Java system properties, JMeter properties 和 logging properties 能够直接通过命令行来直接重载而不用修改 jmeter.properties 文件。使用以下参数项：

- n** -D[prop name]=[value] 定义一个 java system property 值
- n** -J[prop name]=[value] 重载一个 JMeter property
- n** -L[category]=[priority] 重载一个 logging setting, 设置一个指定的类别名给指定的优先级
- n** -L 标志位可以不指定 category, 将其设为 root 级别

【示例】

```
jmeter -Duser.dir=/home/mstover/jmeter_stuff \ -Jremote_hosts=127.0.0.1
    -Ljmeter.engine=DEBUG
jmeter -LDEBUG
```

【注意】

命令行输入的 properties 在启动时先处理，但是晚于 logging system 的建立。所以尝试使用-J 标志去更新 log_level 或者 log_file 的 properties 将没有任何效果。

2.4.6 Logging 和 error 信息

如果 JMeter 检查到一个错误，一个信息就被写道 log 文件中。Log 文件的名字是在 jmeter.properties 文件中定义的。默认值位 jmeter.log，在 JMeter 的开始目录中可以看到，例如 bin 目录中。

除了记录错误信息，jmeter.log 文件中同时记录了一些测试运行的信息。例如：

```
n 10/17/2003 12:19:20 PM INFO - jmeter.JMeter: Version 1.9.20031002
n 10/17/2003 12:19:45 PM INFO - jmeter.gui.action.Load: Loading file:
c:\mytestfiles\BSH.jmx
n 10/17/2003 12:19:52 PM INFO - jmeter.engine.StandardJMeterEngine: Running the
test!
n 10/17/2003 12:19:52 PM INFO - jmeter.engine.StandardJMeterEngine: Starting 1
threads for group BSH. Ramp up = 1.
n 10/17/2003 12:19:52 PM INFO - jmeter.engine.StandardJMeterEngine: Continue on
error
n 10/17/2003 12:19:52 PM INFO - jmeter.threads.JMeterThread: Thread BSH1-1 started
n 10/17/2003 12:19:52 PM INFO - jmeter.threads.JMeterThread: Thread BSH1-1 is
done
n 10/17/2003 12:19:52 PM INFO - jmeter.engine.StandardJMeterEngine: Test has ended
Log 文件有助于定位错误出现的原因，因为它并未打断测试来显示一个 error 窗口。
```

2.5 配置 JMeter

如果你想修改 JMeter 运行使用的 properties，你可以通过两种方式：修改 bin 目录中的文件 jmeter.properties 或者创建你自己的 properties 文件使用命令行来指定该文件。

Parameters

Attribute	Description	Required
ssl.provider	你可以指定一个类支持 SSL. 如果你使用 sun 的 JSSE，那么是：com.sun.net.ssl.internal.ssl.Provider。JMeter 默认的 https 支持在 JDK1.4 中包含。	No
xml.parser	你可以指定一个你自己的 XML parser. 默认值为：org.apache.xerces.parsers.SAXParser	No
user.dir	JMeter 用于保存和装载测试用例的目录。	No
remote_hosts	使用逗号分割的远程 JMeter 主机。如果你运行在分布式环境下，列出你拥有的运行的 JMeter 远程主机. 允许你通过 GUI 去控制这些机器。	No
not_in_me	一个组件的列表, 都是你不想在 JMeter 的菜单中看到的. 由于	No

nu	JMeter 添加了越来越多的组建, 你可能希望自己定制你的 JMeter 来仅仅显示那些你感兴趣的组建. 你可以列出他们类名或者类的标签(在 JMeter 界面中显示的字符串)在这里, 他们将不会被显示。	
----	---	--

3 建立测试计划

一个测试计划描述了一系列 JMeter 在运行中要执行的步骤。一个完整的测试计划包含一个或多个 Thread Groups, logic controllers, sample generating controllers, listeners, timers, assertions 和 configuration elements。

3.1 添加删除 Elements

添加 elements 到测试计划, 可以通过在 tree 中 element 上点击右键, 然后从 add 列表中选择一个新的 element。同样, elements 也可以通过 open 选项从一个文件中载入。

删除一个 element, 确定该 element 被选定, 右击选择删除选项。

3.2 载入和存储 elements

载入文件中的 elements, 在已有的 tree 中点击右键, 然后选择 open 选项。选择 elements 存储的文件, JMeter 将载入文件中的所有 elements 到 tree 中。

存储 tree 的 elements, 选择一个 element 然后右击, 选择 save 选项。JMeter 会存储选定的 element, 以及所有的子 elements。这样你就可以存储测试树的一段, 单独的 element 或者整个测试计划。

3.3 配置 tree 的 elements

任何一个一个测试树中的 element 都可以在 JMeter 的右边框架显示。这样允许你来配置该测试 element 的属性。能够配置什么属性取决于你选定的 element 的类型。

3.4 运行测试计划

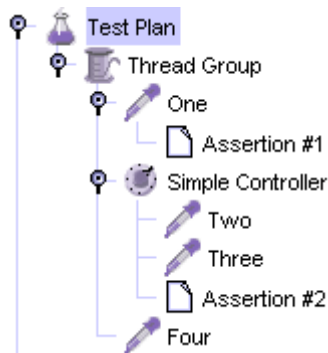
在 run 菜单中选择开始来运行测试计划。停止测试计划, 从菜单中选择停止。JMeter 不会自动的在运行测试计划时有任何表现。一些 listeners 使得 JMeter 运行表现出来。但是唯一的方法是检查 run 菜单中的 start 选项, 如果是 disable 的而且 stop 是 enabled, 那么 JMeter 就在运行你的测试计划, 至少表面上看来如此。

3.5 Scoping Rules

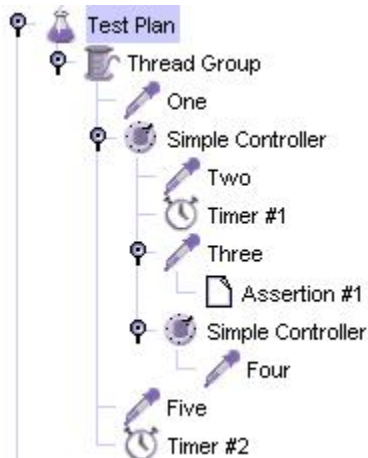
JMeter 测试树中包含的 elements 是分级和有次序的。一些 elements 在测试中有严格的等级要求(Listeners, Config Elements, Post-Processors, Pre-Processors, Assertions, Timers), 而其他一些有 primarily ordered 的要求(controllers, samplers)。当你创建测试计划时, 你需要建立一个有次序的 sample 请求(通过 sampler), 这些反映了一组需要执行的测试步骤。这些请求经常在有序的 controllers 中组织。例图 1



请求的次序将为： one, two, three, four。有些 controllers 会影响他们 sub-elements，你可以在 component reference 中查看这些。一些其他的 elements 是有级别的。例如，Assertion 在测试树中是有级别的。如果他的父亲节点是一个 request，那么它将仅仅影响该 request。如果父节点是一个 controller，那么它将影响该 controller 的所有后代的请求。例如：



Assertion #1 仅仅应用到 Request One 上。而 Assertion#2 应用到 Request Two 和 Three 上。另外一个关于 Timers 的【示例】



在这个例子中，request 的名字反映了他们在测试中应该执行的次序。Timer#1 将应用到 Request Two, Three 和 Four 上(注意对于分等级的 elements 如何排序毫不相关)。Aserrrtion#1 将仅仅应用到 Request Three 上。Timer#2 将影响所有的请求。

如果你想任何 request 通过树的路径被传递，到它的父节点，在到它的父节点的父节点，以此类推，在每一级的父节点上集合所有的 configuration elements，那么你就知道它是如何工作的了。

【注意】

对于 Configuration elements 中的 Header Manager, Cookie Manager 和 Authorization manager 来说，与其他的 Configuration Default elements 不同。Configuration Default elements 被整合成一个 Sampler 访问的集合。而 Manager 的设置不会被整合。如果超过一个的 Manger

在 Sampler 的范围内，只有一个 Manager 被使用，但是目前还无法判断那个被使用。

4 测试计划的 elements

测试计划对象现在拥有了一个新的 checkbox 选项，叫做“功能测试”。如果选择，JMeter 将记录所有 server 为每个 sample 返回的值。如果你已经在测试 listeners 中选定了文件，那么这些数据将被写入到文件中。这个有助于你通过一个小型的测试来检验 JMeter 是否配置正确，而你的服务器返回正确的结果。值得注意的是这个文件的大小将增长的很快使得 JMeter 的性能降低。这个选项在做压力测试时需要关闭掉(默认为关闭)。如果你不将数据写入文件中，那么这个选项没有任何不同。

4.1 Thread Group

Thread Group elements 是任何一个测试计划的开始点。所有的测试计划中的 elements 都要在一个 Thread Group 中。如名字显示那样，thread group element 控制了一组线程，JMeter 使用这些线程来执行你的测试。你可以做如下操作来控制 thread group:

- | 设置线程的数量
- | 设置 ramp-up 周期
- | 设置测试执行的次数

每一个线程都会完全和独立的执行测试计划而不影响其他线程。多线程可以用于模拟到你的服务器程序的并发连接。

Ramp-up period 告诉 JMeter 需要多长时间来“ramp-up”去装载全部的线程。如果 10 个线程被使用，而 ramp-up period 为 100 秒，那么 JMeter 会花 100 秒来使得所有的 10 个线程运行。每个线程将在上个线程开始后 10 秒开始。又如，如果有 30 个线程而 ramp-up 为 120 秒。那么每个线程在延迟 4 秒后启动。延迟时间=ramp-up period 时间数/线程数

默认的，一个 thread group 被设置为不限定的循环。你可以设置 thread group 循环的次数。如果设置为一次，那么 JMeter 将执行测试一次然后停止。Version1.9 将允许你预定时间去运行测试。选定 Scheduler 的 checkbox，你就可以设定时间。

4.2 Controllers

JMeter 有两种 controller: Samplers 和 Logical Controllers。

Samplers 告知 JMeter 发送请求到服务器。例如，如果你想 JMeter 发送一个 HTTP 请求，就添加一个 HTTP Request Sampler。当然你也可以定制一个请求，在 Sampler 中添加一个或多个 Configuration Elements。

Logical Controllers 使你可以定制 JMeter 发送请求的逻辑。例如，你可以添加一个 Interleave Logic Controller 来控制交替使用两个 HTTP Request Samplers。同样，一个特定的 Logic controller，作为 Modification Manager，可以让你修改请求的结果。

4.2.1 Samplers

Sampler 告知 JMeter 发送请求到 server 端。JMeter 现在有如下的 samplers:

- | FTP Request
- | HTTP Request
- | JDBC Request
- | Java Object Request

- l LDAP Request
- l SOAP/XML-RPC Request
- l Web Service (SOAP) Request (Alpha Code)

每个 sampler 都有一些可以设置的 properties。你也可以在一个 sampler 中添加 Configuration Elements 来做更多的设置。当然，值得注意的是 JMeter 按照 request 在 tree 中添加的次序来发送请求。

如果你想同时发送多个并发的同一种类的 request，例如 HTTP request，到同样一台服务器。可以考虑使用一个 Defaults Configuration Element。每个 Controller 拥有一个或多个默认元素。当然不要忘记添加一个 Listener 到你的 Thread Group 中来察看和存储你的测试结果。

如果你对 JMeter 对你的 request 返回的 response 进行简单的校验很感兴趣，你可以在 Request Controller 中添加一个 Assertion。例如，在一个 web 程序的压力测试中，服务器可能会返回多个成功的” HTTP Response”，但是页面也许包含了一些错误或者丢失的部分。你可以添加 assertions 来检查特定的 HTML 标签，常见的错误字符串，等等诸如此类的数据。JMeter 允许你使用常规表达式来创建这些 assertions。

4.2.2 Logic Controllers

Logic Controller 使得用户可以定制 JMeter 发送请求的时机。

Logic Controllers 可以拥有如下的任何一个作为子 elements: Samplers(requests), Configuration Element 和其他的 Logic Controllers。Logic Controllers 可以改变子 elements 的请求次序。他们可以自己修改 requests，使得 JMeter 可以重复发送请求等等。

为了便于理解 Logic Controllers 在测试计划中所起的作用，举例如下：

- l Test Plan
 - n Thread Group
 - u Once Only Controller
 - l Logic Request (an HTTP request)
 - u Load Search Page (HTTP sampler)
 - u Interleave Controller
 - l Search “A” (HTTP Sampler)
 - l Search “B” (HTTP Sampler)
 - l HTTP default request (Configuration Element)
 - u HTTP default request (Configuration Element)
 - u Cookie Manager (Configuration Element)

首先表明了 login 请求只会在第一次运行时执行。其余的请求将自动跳过它。达到这个效果的原因就是因为 Once Only Controller 的影响左右。

在 login 之后，下一个 Sampler 加载了 search page (想象一下，一个在用户登录后就转入 search 页面来做搜索的 web 程序)。这个仅仅是一个简单的请求，并未通过任何 Logic Controller 的筛选过程。

在 search 页面加载后，我们想做一些搜索。实际上，我们要做两个不同的搜索。然而，我们想在每个独立的搜索中重新加载 search 页面。我们可以用四个简单的 HTTP request elements 来达到目的(载入 search 页面，搜索” A”，载入 search 页面，搜索” B”)。而实际上，我们使用了另一个方法，我们使用 Interleave Controller，它将每次运行一个完整的测试只使用一个子 element 的 request。而且能够保持这些子 elements 的次序来运行。

【注意】

HTTP Request Defaults 是属于 Interleave Controller 的。想象一下” search A” 和” Search

B” 分享同一个” PATH” 信息(一个指定的 HTTP 请求包括 domain, port, method, protocol, path 和 arguments 等, 还有其他额外的信息)。这导致: 两者都是 search 的请求, 点击同一个后台搜索引擎。与其给两者配置相同 PATH 信息, 不如将信息抽象出来给一个 Configuration Element。

当 Interleave Controller 传递”Search A”和”Search B”的请求时, 将自动使用 HTTP Default Request Configuration Element 的值来取代空白的 PATH 域。所以, 我们将这些请求的 PATH 域空置, 然后将这些信息放入到 Configuration Element 中。

树中接下来的 element 是另外一个 HTTP Default Request, 这次是给 Thread Group 自己使用的。Thread Group 拥有一个内置的 Logic Controller, 因此, 它象上面的介绍的那样一样, 来使用这个 Configuration Element。它将填入到任何一个通过的 Request 的空白中。当你做 web 测试时, 这是非常有用的。你可以将所有的 HTTP Sampler elements 的 DOMAIN 域空白, 然后使用 Thread Group 中添加的 HTTP Default Request 的信息补充这些空白域。这样, 你可以仅仅改变一个域就能将整个测试计划应用到另外的 server 上。否则你需要一个一个更改 Sampler。

最后一个 element 是 HTTP Cookie Manager。所有的 web 测试中都应该添加该 element, 否则 JMeter 将忽略所有的 cookie。在 Thread Group 级添加这个元素, 可以保证所有 HTTP request 将共享同样的一个 cookie。

Logic Controller 可以被组合起来从而获得多种不同的结果。

4.3 Listeners

Listeners 提供了获取在 JMeter 运行过程中搜集到的信息的访问方式。最简单的 Listener, Graph Results Listener 图示出了响应时间。Listeners 提供了一种图型化的方式来查看 JMeter 运行过程中产生的数据。

另外, Listeners 能够直接将搜集到的数据存入到文件中以备后用。任何一个 listener 都拥有一个设置该文件存储地址的域。

Listener 能够加到测试中的任何位置。他们将仅仅收集同级别和所有低级别的 elements 产生的数据。

4.4 Timers

默认的, JMeter 线程发送 requests 时之间没有任何停顿。我们建议你指定一个延迟时间。这可以通过添加一个有效的 Timer 到你的 Thread Group 中实现。如果你没有添加一个延迟时间, JMeter 可能会在极短时间内发送大量的请求而引起 server 崩溃。

Timer 可以使得 JMeter 在线程发送每个请求时有一个延迟。如果你添加了多个 timer 到一个 Thread Group 中时, JMeter 将使用累计的延迟时间。

4.5 Assertions

Assertions 允许你对在测试中从 server 端返回的回复中验证某些数据。使用 assertion, 你可以实际地测试你的程序返回的值是否是期望值。

例如, 你可以检验回复字符串中包含一些特殊的文本。你指定的文本可以是一个 Perl-style 的常规表达式, 然后指定回复中是否包含该文本, 或者整个回复都应该相符合。

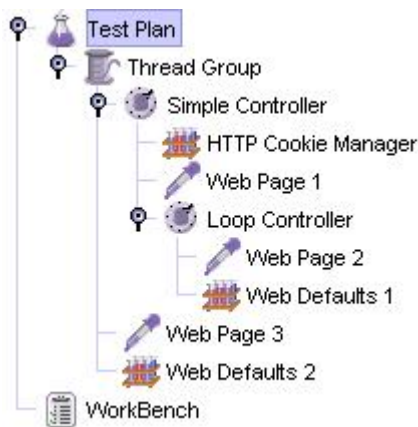
你可以给任何一个 Sampler 添加一个 assertion。例如, 你可以添加一个 assertion 到一个 HTTP Request 来检查文本” yahoo”。JMeter 就会在返回的回复中察看该文本。如果 JMeter 不能发现该文本, 那么将标志该请求是个失败的请求。

为了查看 assertion 的结果，你需要添加一个 Assertion Listener 到 Thread Group 中。

4.6 Configuration Elements

Configuration Element 与 Sampler 紧密联系在一起工作。尽管它不能发送请求，但是它可以被添加到或者去修改请求(除了 HTTP Proxy Server)之外。

Configuration Element 只能被所在的 tree branch 访问。例如，你在一个 Simple Logic Controller 中设置了一个 HTTP Cookie Manager，那么这个 Cookie Manager 将只能被设置在 Simple Logic Controller 中的 HTTP Request Controllers 访问。如下图中所示，Cookie Manager 能够被 HTTP request “Web Page1” 和 “Web Page2” 访问，但是不能被 “Web Page3” 访问。并且，一个 tree branch 中的 configuration element 将拥有更高的优先级，与同一个 parent branch 中的同样的元素相比。例如，我们定义了两个 HTTP Request Defaults elements, “Web Defaults 1” 和 “Web Defaults 2”。因为我们将 “Web Defaults 1” 放入到一个 Loop Controller 中，所以只有 “Web Page 2” 能访问它。而其他的 HTTP requests 只能使用我们在 “Thread Group” (所有其他 branch 的 “parent”) 中设置的 “Web Defaults 2”。



4.7 Pre-Processor Elements

Pre-Processor 在 Sampler Request 被创建前执行一些操作。如果一个 Pre-Processor 被附加到一个 Sampler element 上，那么它将先于 Sampler Element 运行。Pre-Processor 最主要用于在 Sampler 运行前修改一些设置，或者更新一些无法从 response 文本中获取的变量。

4.8 Post-Processor Elements

Post-Processor 在 Sampler Request 被创建后执行一些操作。如果一个 Post-Processor 被附加到一个 Sampler element 上，那么将紧接着 sampler element 运行后运行。Post-Processor 主要用于处理回复数据，常常用来从其中获取某些值。

5 建立一个 web 测试计划

在本章中，你将学习到如何创建一个基本的测试计划用来测试一个 web 站点。你创立 5 个用户向 Jakarta 网站的两个页面发送请求。同时，你将设置这些用户运行他们的测试两遍。所以，总共的 request 数是： $5(\text{user}) * 2(\text{request}) * 2(\text{repeat 2 times}) = 20$ HTTP Requests。为建立

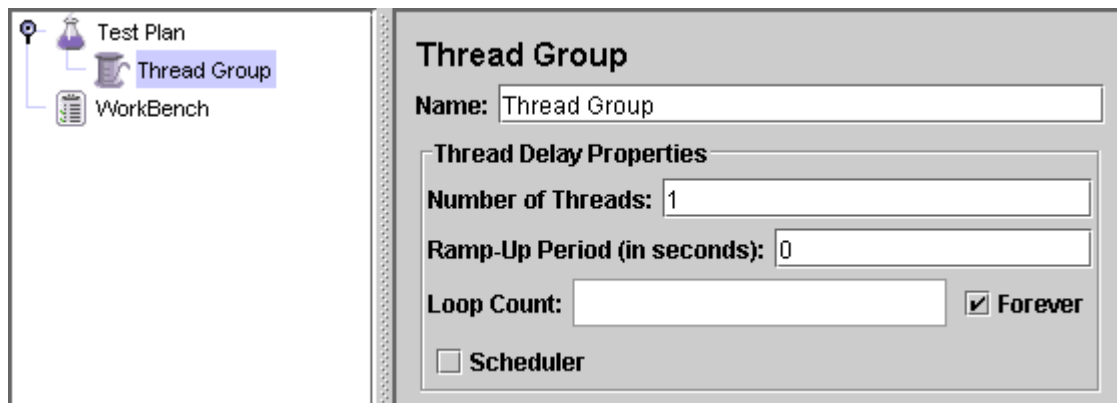
测试计划，你将使用如下 elements： Thread Group, HTTP Request, HTTP Request Defaults 和 Graph Results。

5.1 Adding Users 添加用户

每个 JMeter 测试计划的第一步都是要添加一个 Thread Group element。Thread Group 将告诉 JMeter 你要设置的用户数目，发送请求的频率和请求的次数。

选择测试计划，点击右键，选择添加 Thread Group。这时你就应该在 Test Plan 下看到 Thread Group element 了。如果你没有看到，那就点击 Test Plan 展开树。

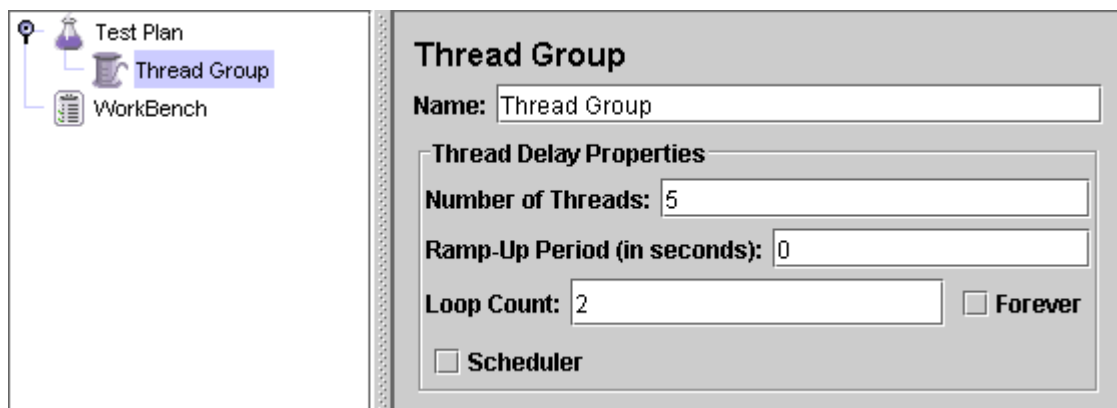
然后，你需要修改那些默认的 properties。选择树中的 Thread Group element，在 JMeter 左侧的窗口中应该显示出 Thread Group 的 Control Panel。



第一步，给我们的 Thread Group 添加一个更详细的名称。在名字域中，输入 Jakarta Users。然后增加用户的数目到 5 (调用的线程数)。

在下一个域中，Ramp-Up Period，使用默认值 0 秒。这个 properties 告诉 JMeter 两个 User 开始发送请求之间的延迟时间。例如，你输入 5 秒，JMeter 将在 5 秒后启动所有的用户。所以，我们有五个用户和 5 秒的 Ramp-up period，那么两个用户之间开始时间的延迟是 1 秒。 $(5 \text{ user}/5=1)$ 。如果值为 0，那么 JMeter 将立刻开始你所有的用户操作。

最后，清除“Forever”的 checkbox，输入 2 在 Loop Count 域中。这个 property 告诉 JMeter 要重复运行你的测试几次。如果你输入的循环值为 1，那么 JMeter 将只运行一次你的测试。如果要 JMeter 不停运行你的测试，选择 Forever。

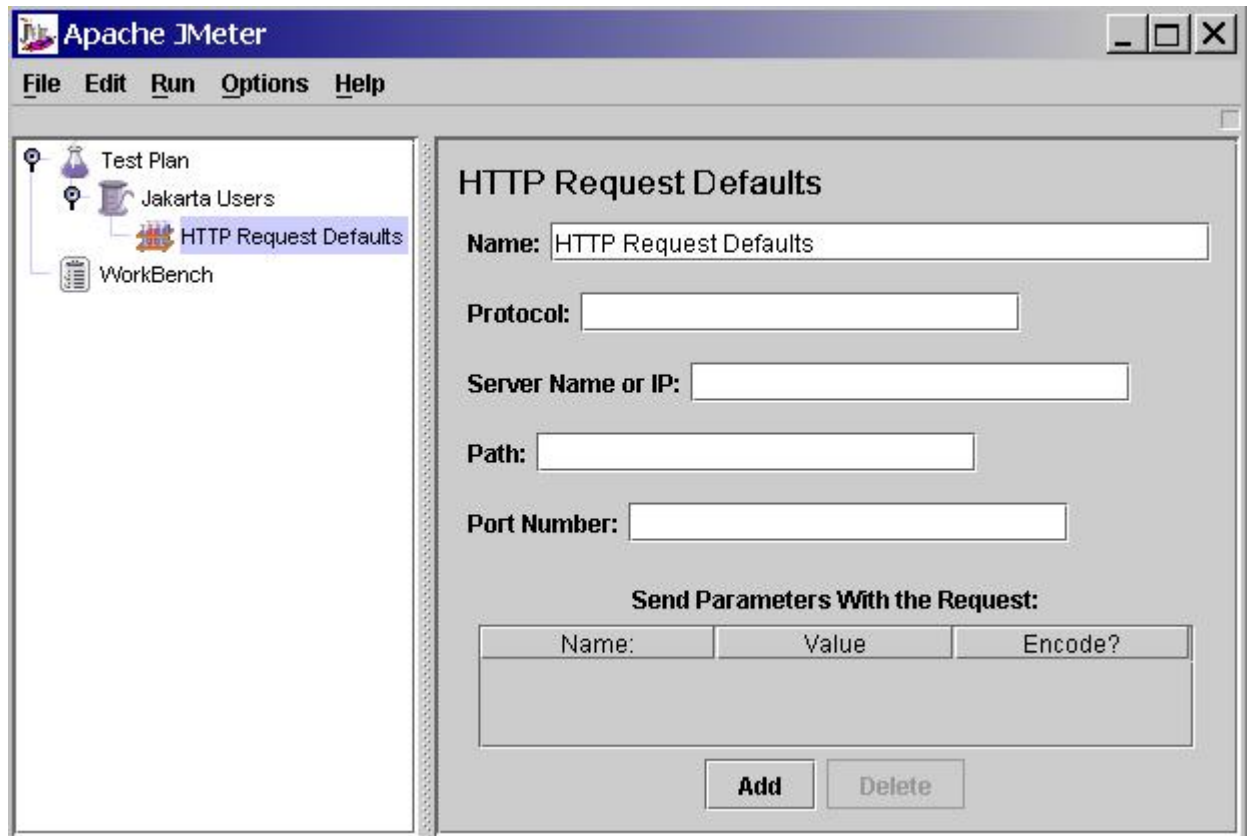


在大部分程序中，你不得不手工方式来更新你在控制面板中的更改。然而，在 JMeter 中，控制面板将自动的接受你的更改。例如，你改变了一个 element 的名称，树将自动更新为最新的文本在你离开控制面板后(例如，选择了另一个 element)。

5.2 添加 Default HTTP Request Properties

现在我们已经定义了我们的用户，是时间来定义他们要执行的任务了。在这节，你将指定你的 HTTP 请求的默认设置。然后，在 5.3 中，你将添加 HTTP Request elements 来使用你制定的默认值。

选择你的 Jakarta User element。右击选择添加 Config Element 中的 HTTP Request Defaults。然后选择这个新的元素查看他的控制面板。



象大多数的 JMeter 元素一样，HTTP Request Default 的控制面板拥有一个可修改的名称域。在本例中，使用默认值。

跳到下个域，Web Server's Name/IP。在本例中，所有的测试用例将发送到同一个 web 服务器上去： Jakarta.apache.org。将这个域名输入到该字段中。这是我们唯一指定的默认值，所有留下其他的域为空白值。

【注意】

HTTP Request Defaults elements 不会告诉 JMeter 去发送 HTTP 请求。它只是简单的定义了 HTTP request elements 会使用到的默认值。

5.3 添加 Cookie 支持

几乎所有的 web 测试要求使用 cookie，除非你的程序特别指明不使用 cookie。添加 cookie 支持，最简单的是添加 HTTP Cookie Manager 到每一个 Thread Group 中。这会保证每个线程都获得自己的 cookie，但是交叉共享所有的 HTTP Request 对象。

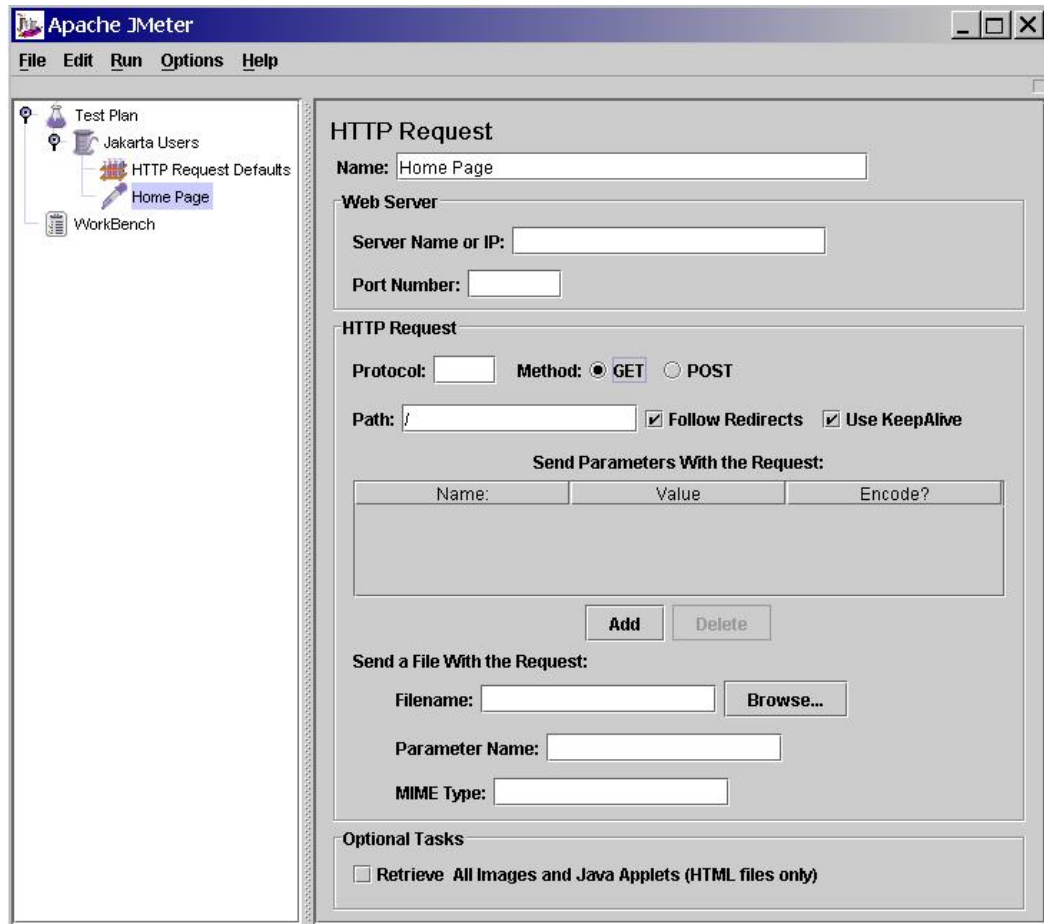
添加 HTTP Cookie Manager，点击 Thread Group，然后选择添加 HTTP Cookie Manager。

5.4 添加 HTTP Request

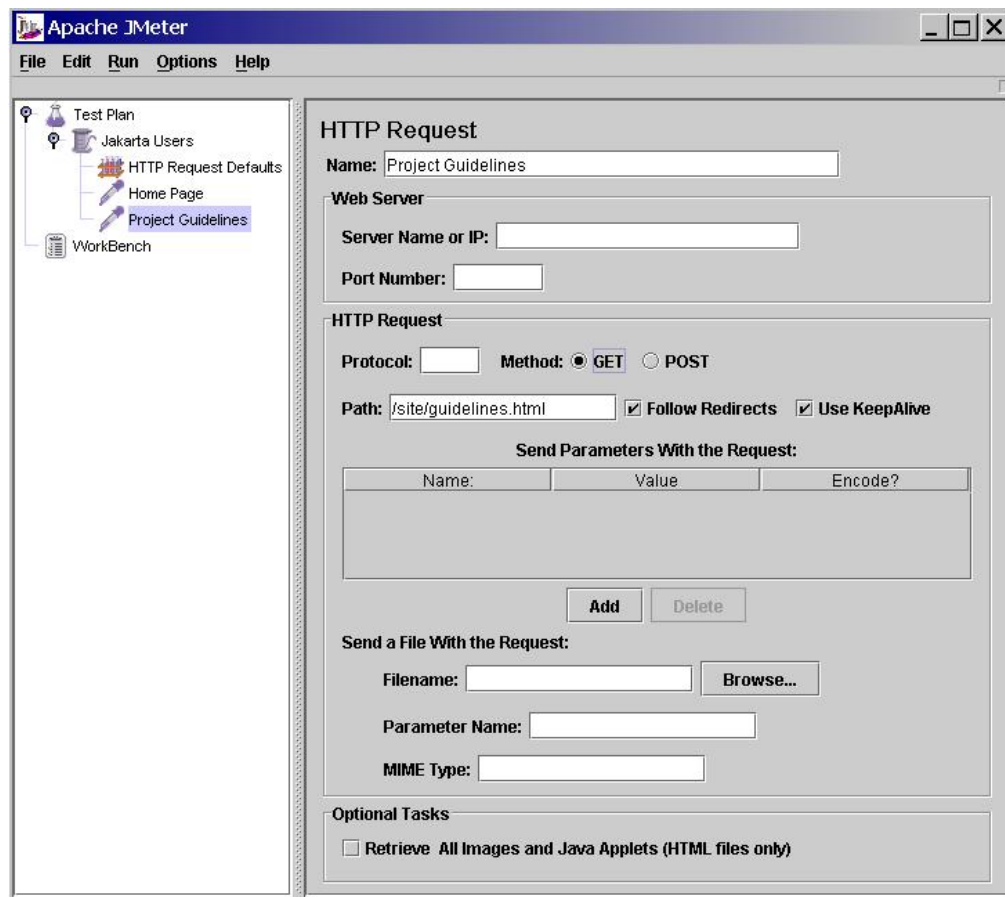
在我们的测试计划中,我们需要创建两个 HTTP Requests。第一个是访问 Jakarta 的主页, (<http://Jakarta.apache.org>)。第二个是访问 <http://jakarta.apache.org/site/guidelines.html>。

右击添加第一个 HTTP Request, 然后修改该元素的 properties

1. 将名字改为 HOME Page
2. 设置路径为 “/”。注意的是你不需要设置 server 的名称, 因为你再 HTTP Request Defaults element 中已经设置了。



3. 紧接着添加第二个 HTTP request。



5.5 添加一个 Listener 来查看和存储测试结果

最后一个要在测试计划中添加的元素是 Listener。这个元素用来存储所有的测试结果到一个文件和提供一个数据的视图模式。

添加 Graph Results listener 到 Thread Group 中。然后指定一个路径和文件名用于存储测试结果。你可以输入，也可以通过 browse 按钮来选择。



5.6 保存测试计划

尽管这不是一个必须的步骤，但是我们建议你将在运行前存入一个文件。

5.7 运行测试计划

在 Run 菜单中，选择运行，一旦 JMeter 完成了你的测试计划，选择停止。如果你选择将 listener 中的测试结果存储到一个文件中，那么你将拥有这个可以被任何浏览器查看的文件。不同的浏览器显示不同的风格。

6 建立一个高级的 web 测试计划

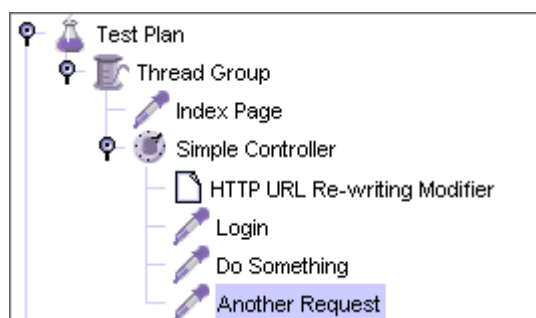
本节中，你将学习如何建立一个更高级的测试计划来测试 web 站点

6.1 使用 URL 重写技术处理用户的 session

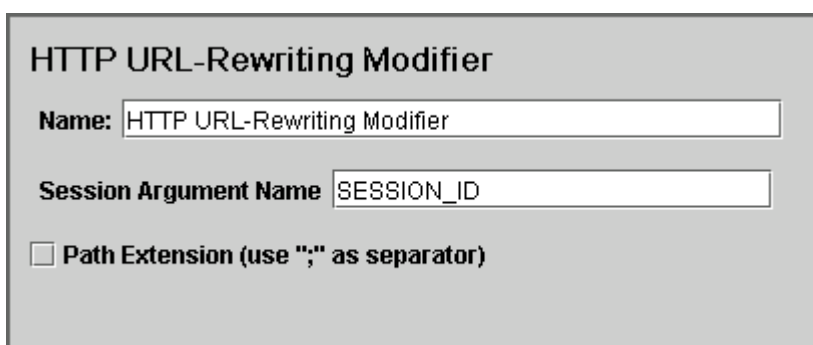
如果你的 web 程序使用 URL 重写而不是 cookie 来保存 session 的信息，那么你需要做一些额外的工作来测试你的站点。

为了正确的进行 URL 回写，JMeter 需要过滤从 server 端获的 HTML 来获得唯一的 session ID。使用合适 HTTP URL Re-Writing Modifier 来达到此目的。将你的 Session ID 作为参数写入到 modifier 中，它将自动添加 session 到每一个 request 中。如果 request 中已经有了该值，将直接覆盖。

【示例】



上图中显示了一个使用了 URL Re-writing 的测试计划。注意 modifier 被附加在 Thread Group 上，这样就保证了所有线程组的请求都会被应用到。



在上图中，我们看到的是 URL Re-Writing modifier 的 GUI。仅仅只有一个域给用户用于设置 session ID 参数的名字。

6.2 使用 Header Manager

HTTP Header Manager 可以使你定制 JMeter 发送的请求的报头。象 HTTP cookie Manager 一样，应该添加到合适的级别上。除非出于某种原因你希望你测试不同的 request 发送不同的报头。

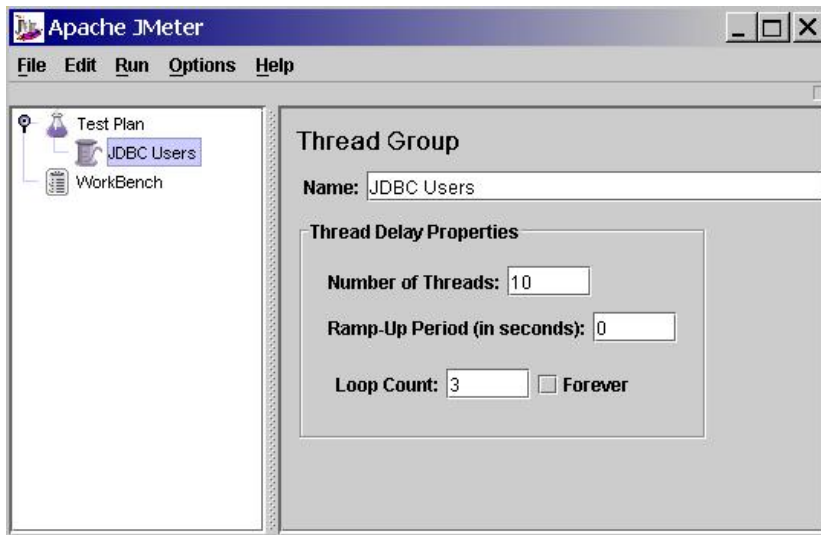
7 建立一个 database 测试计划

本节中，你将学习如果建立一个测试数据库的基本测试计划。你将创建一个有 10 个用户发送 2 个 SQL 请求到数据库 server 的测试。同时，你将告诉用户运行他们的测试三遍。所以，总共的请求次数是： $10 \text{ users} * 2 \text{ requests} * 3 \text{ times} = 60 \text{ JDBC requests}$ 。为了创建测试计划，你将使用元素： Thread Group, JDBC Request 和 Graph Results。

7.1 添加用户

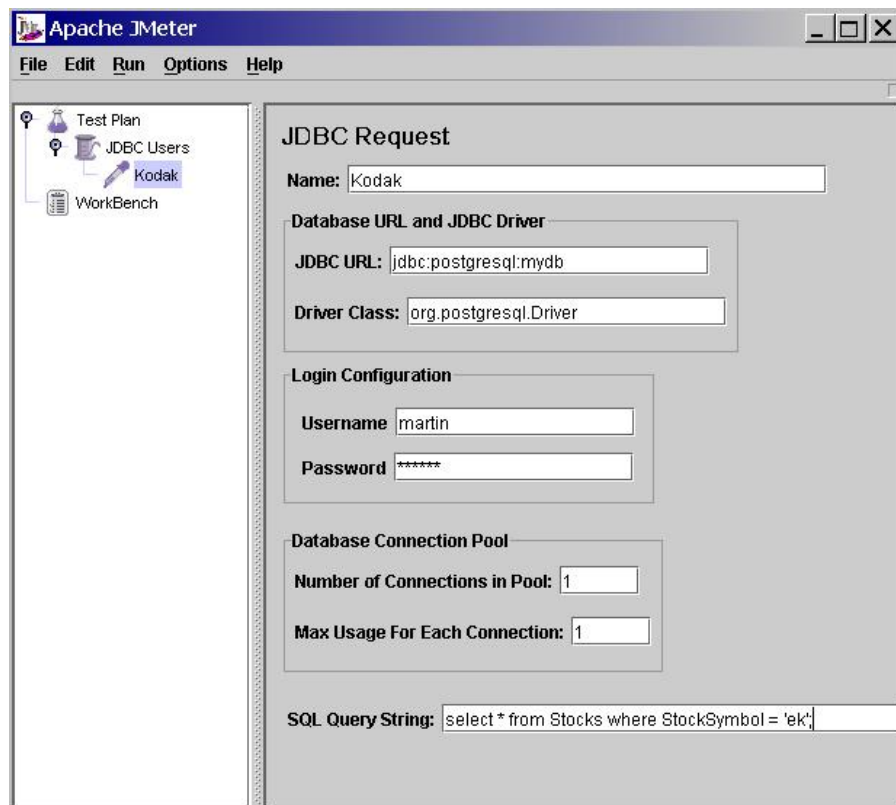
每个 JMeter 测试计划创建的第一步都是添加 Thread Group 元素。Thread Group 用于设置发送请求的用户数，频率和请求总数。

如下图，设置 10 个线程代表 10 个用户，用户的延迟时间为 0，一共循环 3 次。



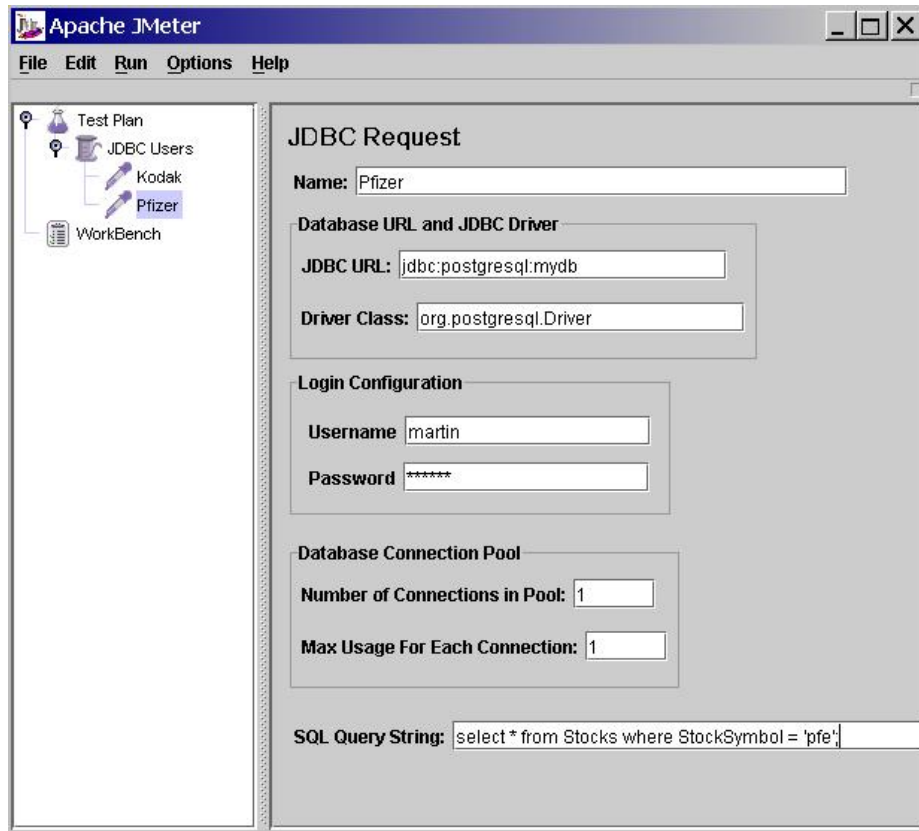
7.2 添加 JDBC 请求

尽管有 JDBC DATABASE Connection Pool Defaults, JDBC Database Login Defaults 和 JDBC SQL Query Defaults 组件可用,但是本例中并未涉及。我们要添加两个 JDBC Request。下图为请求一。



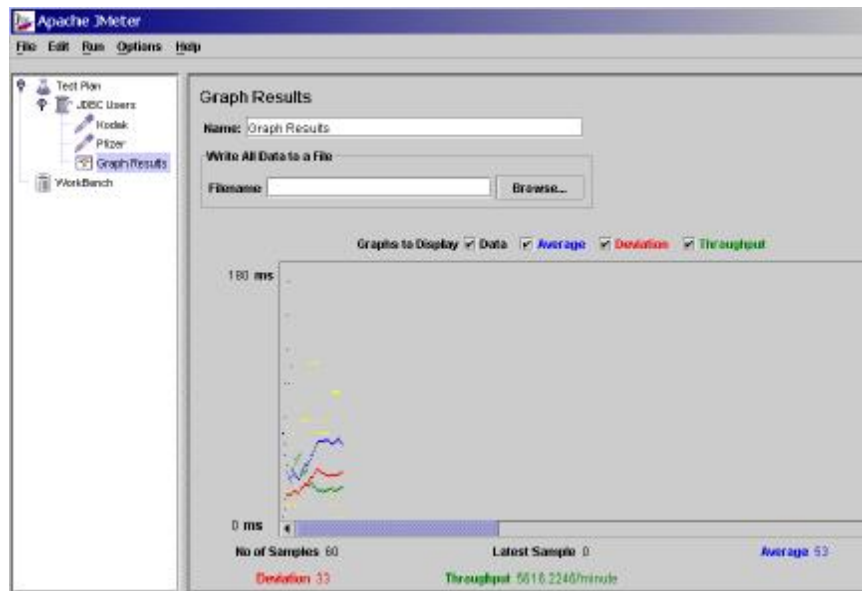
【注意】

需要将需要的 JDBC 的 drive 放入 Classpath 中。
下图为请求二。



7.3 添加 Listener 元素来监视和存储测试结果

如下图所示，添加了一个 Graph Result Listener。



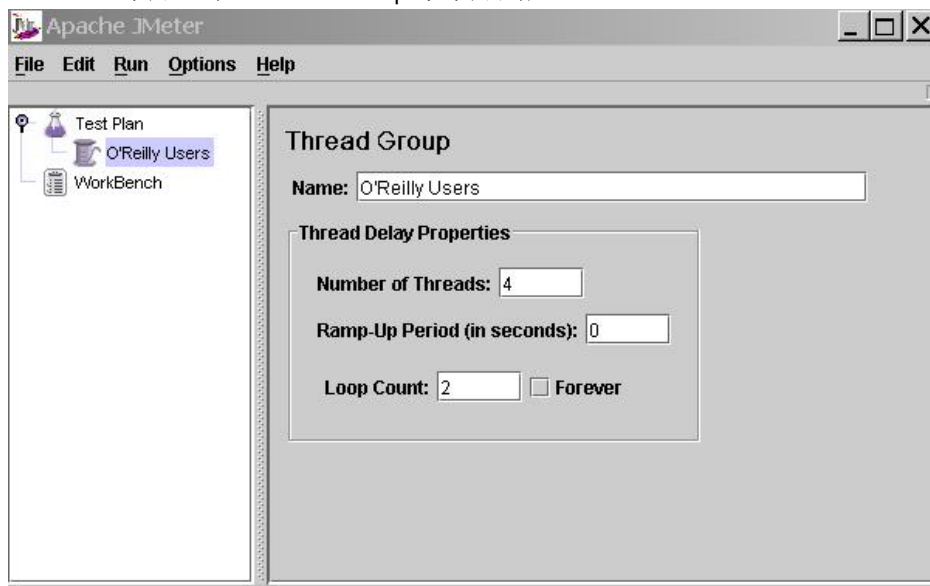
7.4 保存测试计划

7.5 运行测试计划

8 建立一个 FTP 测试计划

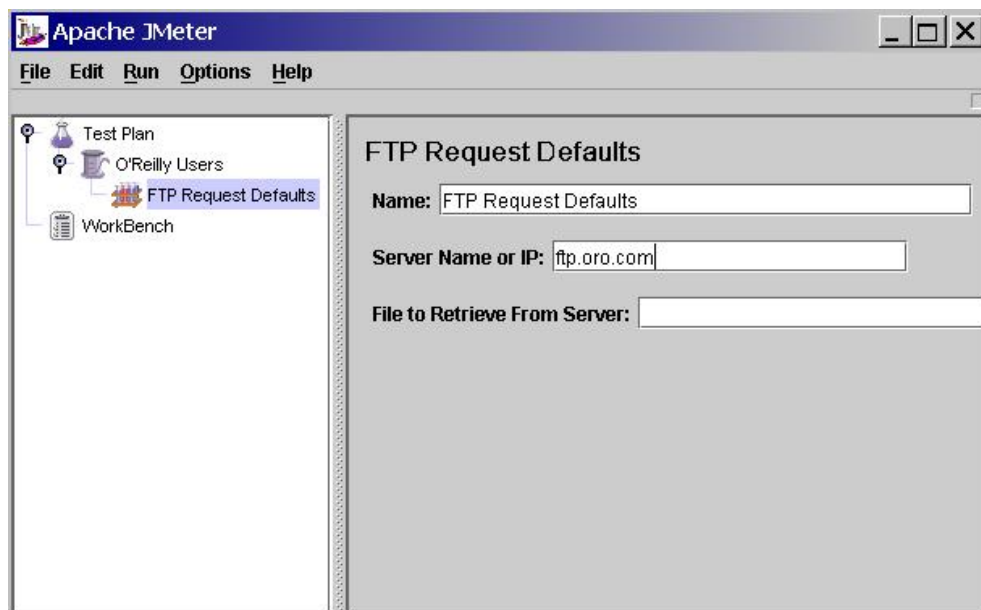
8.1 添加用户

通过添加一个 Thread Group 来添加用户



8.2 添加 Default FTP Request Properties

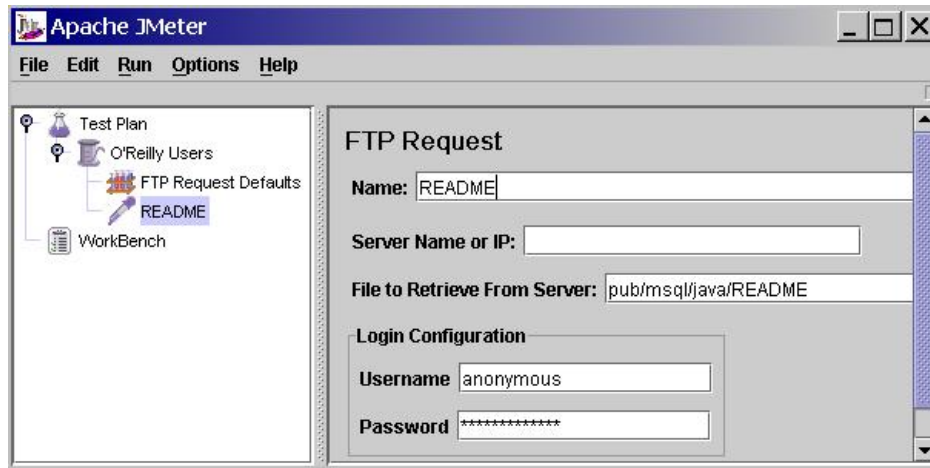
添加一个 config element 中的 FTP Request Default 用于所有的 FTP requests。



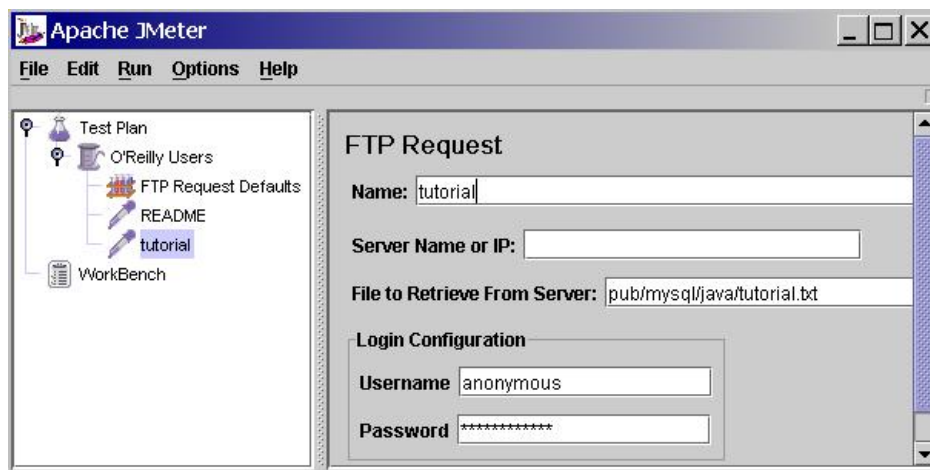
8.3 添加 FTP Requests

在我们示例中，要添加两个 FTP 请求。

n 请求一：

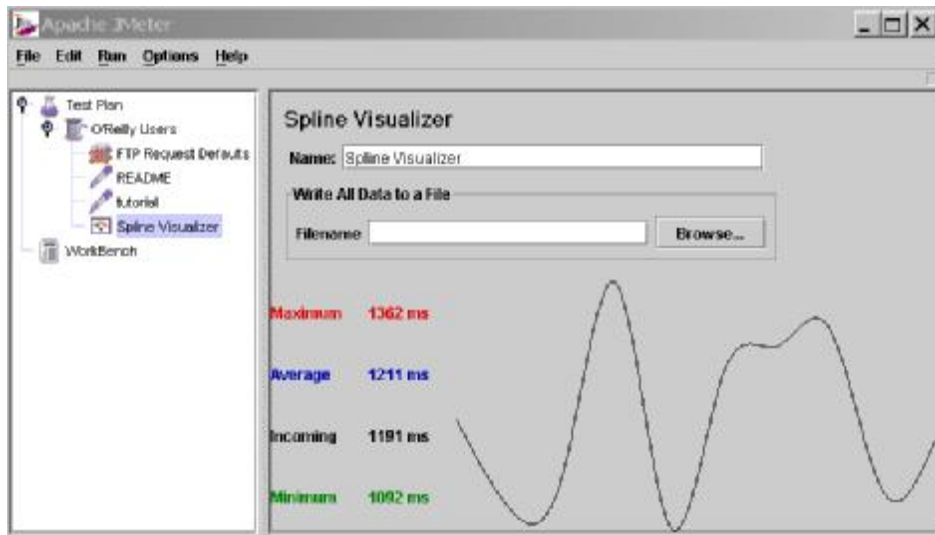


n 请求二：



8.4 添加一个 Listener 来查看和存储测试结果

最后，我们添加了一个 Spline Visualizer listener 用于观察和存储测试结果。



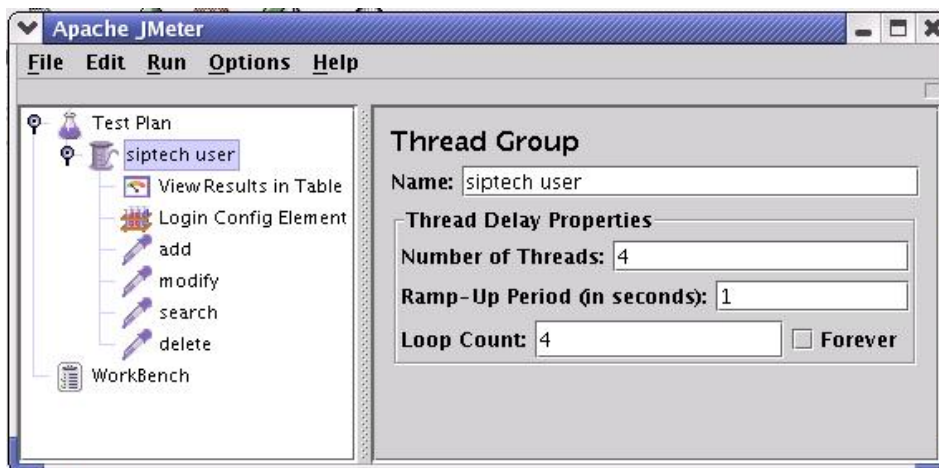
8.5 保存测试计划

8.6 运行测试计划

9 建立一个 LDAP 测试计划

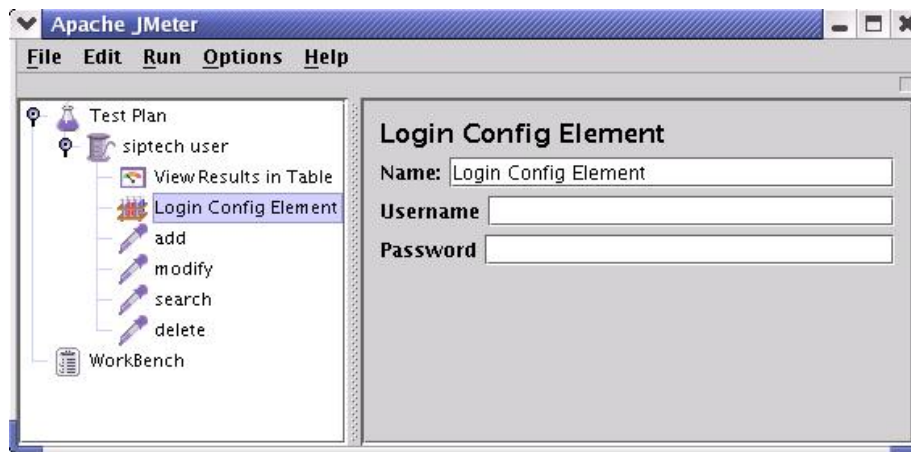
9.1 添加用户

创建一个 Thread Group 用于添加用户和设置请求次数和频率。



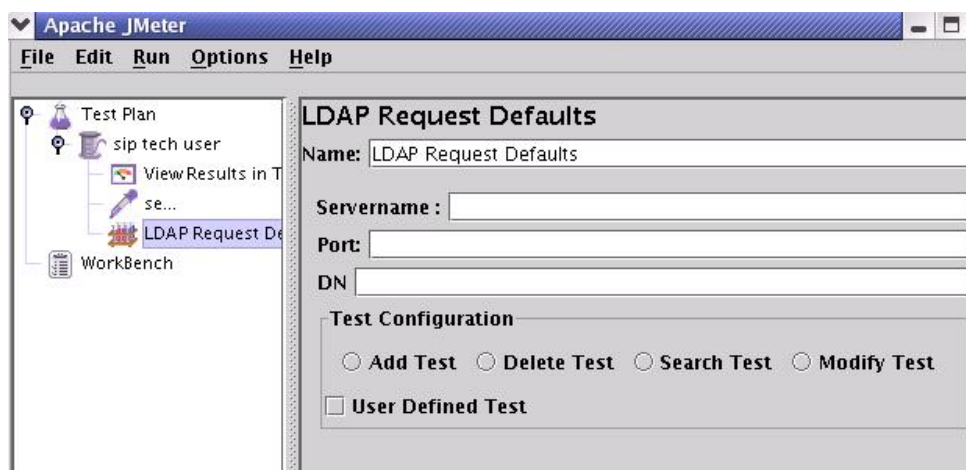
9.2 添加一个登录控制元素

可以添加一个 Config Element 的 Login Config Element 用于用户登录控制。



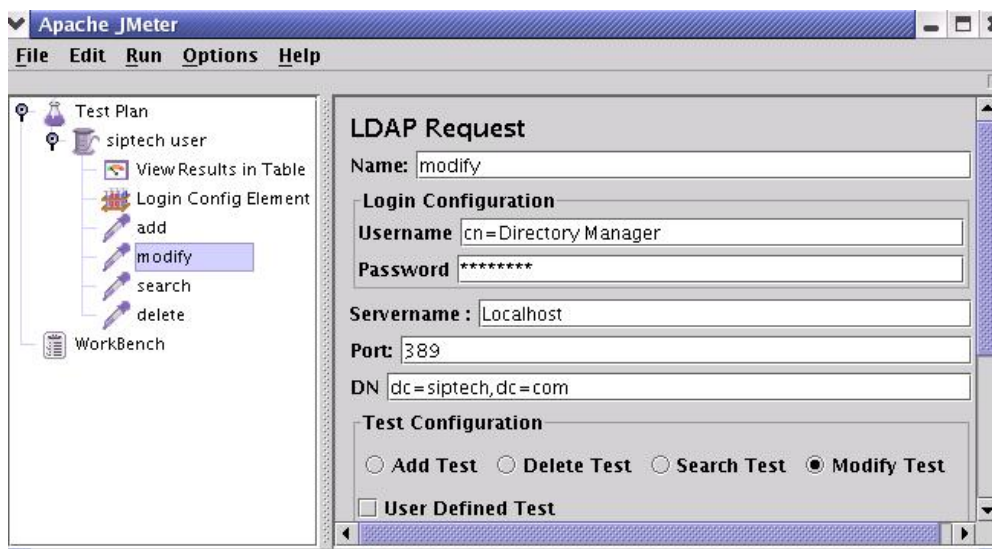
9.3 添加一个 LDAP Request Defaults

可以添加一个 config element 中的 LDAP Request Defaults 用于控制一些共享默认值。



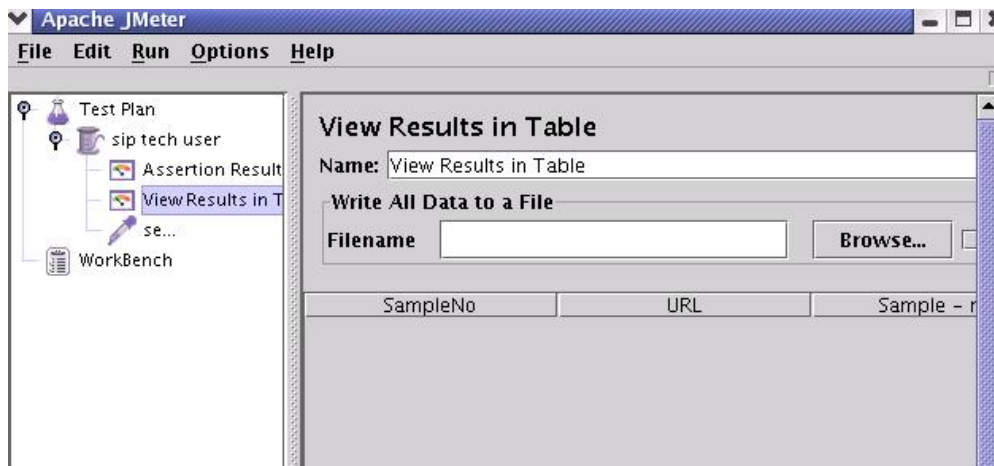
9.4 添加 LDAP Request

添加的 LDAP Request 通过选择不同的测试配置来完成四种操作：添加，删除，修改和查询，例如，修改：



9.5 添加一个 Listener 来查看和保存测试结果

添加了一个 View Results in Table 的 listener。



9.6 保存测试用例

9.7 运行测试用例

10 创建一个 Web Service 测试计划

对于 Web Service Sampler, 当前是使用 Apache Soap driver, 需要 activation.jar 和 mail.jar。由于 license 的限制, JMeter 中并未包含这些 jar 文件。

如果 sampler 显示从 web service 获得一个 error. 需要同时检查 SOAP 的信息来确认格式是正确的。特别要确认 xmlns 的 attributes 要和 WSDL 完全一致。如果 xml 的 namespace 不同, 那么 web service 将很可能返回一个错误。Xmethods 中包括了一个公共 service 的列表。

10.1 添加用户

10.2 添加 web service 请求

10.3 添加 listener 用于查看和存储测试结果.

10.4 保存测试计划

10.5 运行测试计划

11 创建 JMS Point-to-Point 测试计划

需要如下 jar 文件的支持

I activation.jar	I geronimo-spec-j2ee-management-1.0-rc4.jar
I activeio-1.0-SNAPSHOT.jar	I geronimo-spec-jms-1.1-rc4.jar
I activemq-3.0.jar	I geronimo-spec-jta-1.0.1B-rc4.jar
I activemq-core-3.0.jar	I jms.jar
I commons-logging-1.0.3.jar	I jndi.jar
I concurrent-1.3.4.jar	I log4j-1.2.8.jar
I geronimo-spec-j2ee-jacc-1.0-rc4.jar	I spring-1.1.jar

11.1 添加用户

11.2 添加 JMS 请求

11.3 添加 listener 用于查看和存储测试结果.

11.4 保存测试计划

11.5 运行测试计划

Name	Value	Description
JMS Resources		
QueueConnectionFactory	ConnectionFactory	This is the default JNDI entry for the connection factory within active mq.
JNDI Name Request Queue	Q.REQQ	This is equal to the JNDI name defined in the JNDI properties.
JNDI Name Reply Queue	Q.RPL	This is equal to the JNDI name defined in the JNDI properties.
Message Properties		
Communication Style	Request Response	This means that you need at least a service that responds to the requests.
Content	test	This is just the content of the message.
JMS Properties		Nothing needed for active mq.
JNDI Properties		
InitialContextFactory	org.activemq.jndi.ActiveMQInitialContextFactory	The standard InitialContextFactory for Active MQ
Properties		
brokerURL	tcp://localhost:61616	This defines the url of the active mq messaging system.
queue.Q.REQ	example.Q.REQ	This defines a JNDI name Q.REQ for the request queue that points to the queue example.Q.REQ.
queue.Q.RPL	example.Q.RPL	This defines a JNDI name Q.RPL for the reply queue that points to the queue example.Q.RPL.

12 建立 JMS Topic 测试计划

12.1 添加用户

12.2 添加 JMS Subscriber 和 Publisher

12.3 添加 listener 用于查看和存储测试结果.

12.4 保存测试计划

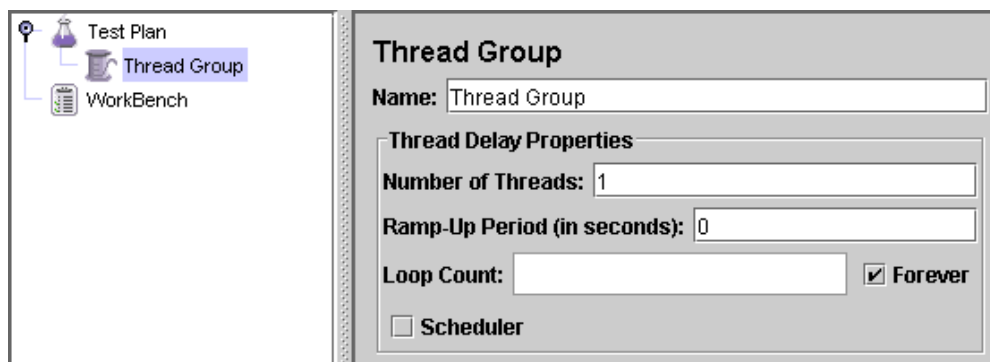
13 建立一个 Monitor 测试计划

Monitors 对压力测试和系统管理非常有效。在压力测试中使用，monitor 提供关于 server 性能的额外信息。能够使得在用户端查看 server 得性能和响应时间变得很简单。作为系统管理工具，monitor 提供了一种非常简单得方法来在同一个 console 中监视多个 server。Monitor 被设计和 Tomcat5 中的 status servlet 一起使用。在理论上来说，任何一个 servlet container 能够支持 JMX 的话，都可以提供相同的信息。

如果有人想使用 monitor 和其他 servlet 或者 EJB container 一起工作的话，tomcat 的 status servlet 应该能够和其他 container 一起工作来监视 memory statistic 而不用做任何修改。为了获得线程信息，你需要修改 MBeanServer 来获取正确的 MBeans。

13.1 添加 server

通过添加一个 Thread Group 来告知 JMeter 你想要的线程数。如果我们就将 JMeter 作为 Monitor 使用的话，那么这个线程数只能为 1。这对于那些不熟悉 server 监控的人非常重要。作为一个惯用规则，在一个 server 上使用多个线程是错误的，会导致非常大的压力。



13.2 添加 HTTP Auth Manager

添加 HTTP Auth Manager 元素：

1. base URL 为空白
2. 输入用户名

3. 输入密码

13.3 添加 HTTP Request

添加一个 HTTP Request sample:

1. 名字改为 “server status”
2. 输入 IP 地址或者主机名
3. 输入端口号
4. 设置 Path 为: /manager/status
5. 添加参数 XML, 值为 true
6. 选择 “use as Monitor”

13.41 添加一个 Constant Timer

添加 Constant Timer, Thread Delay 为 5000 微秒。

13.5 添加一个 Listener

添加 Simple Data Writer Listener 来存储测试结果。

13.6 添加 Monitor Results

添加 Monitor Result Listener 来查看。

13.7 保持测试计划

13.8 运行测试计划

14 listeners

14.1 screen captures

可以将 listener 的测试结果保存为 PNG 或者 TIF 图片。使用 Edit 菜单中的 save as image。

15 Remote Testing 远程测试

在你的 JMeter 客户机不可用的情况下, 聪明的方法是: 为模拟足够多的用户来对你的 server 做压力测试, 有一个控制多个的选项存在, 远程控制多个 JMeter Engine 从一个单独的 JMeter GUI 中。远程运行 JMeter, 有可以复制一个测试到多个低端的机器上从而模拟一个非常大数量的用户负载到 server 上。一个 JMeter 的 GUI 客户端可以控制任何数目的远程 JMeter, 搜集他们的数据。这样提供了如下的特点:

- l 真实的性能测试而不存在网络瓶颈问题。
- l 快速响应的 GUI
- l 将测试结果存储到本地的一台机器上。
- l 使用一台机器管理多个 JMeter Engines。

【注意】

当你想在要测试的 server 上运行 JMeter Engine 时，你需要注意的是这样给 server 带来额外的处理量从而使得你的测试数据有些不准确。近似解决的方法是在同样一个网段中有一个或多个机器作为你的配置运行 JMeter Engine 的 server。这样会最低限度的减少网络对测试结果的影响，因为减少了对该 server 的性能的压榨。

第一步： start the servers

为使得 JMeter 运行在远程节点模式，在所有你想运行 JMeter 的机器上运行 JMeter 组件运行：

- ！ JMeter_HOME/bin/jmeter-server (UNIX 下)。

- ！ JMeter_HOME/bin/jmeter-server (Windows 下)

第二步： 添加 server 的 IP 地址到客户端的 Properties 文件中

在控制机上修改 bin/jmeter.properties 文件，找到 property 名字为 remote_hosts 的行，添加你运行 JMeter 的机器的 IP 地址，多个地址间使用逗号分隔。

【注意】

你可以使用 -J 的命令行参数来指定一个 host 来运行 JMeter。

例如： jmeter -Jremote-hostes=host1, 127.0.0.1, host2

第三步： start JMeter 客户端

现在就可以在控制 JMeter 的客户端上开始运行了。在 Run 菜单中选择 remote start 即可。

15.1 手动

在某些情况下，jmeter-server 脚本很可能无法使用。这是使用手工方式启动 JMeter 的方法。

JMeter 使用 Remote Method Invocation (RMI) 作为远程传输机制。因此，你需要运行一个 RMI Registry 程序 (rmiregistry)，它在 JDK 包中的 bin 目录中。在运行之前，确定下列的 jar 文件包含在你的系统路径中：

- ！ JMeter_HOME/lib/ext/ApacheJMeter_core.jar

- ！ JMeter_HOME/lib/jorphan.jar

- ！ JMeter_HOME/lib/logkit-1.2.jar

rmiregistry 程序需要访问特定的 JMeter 类。无参数运行 rmiregistry。程序默认的监听端口为 1099。一旦 RMI Registry 程序运行起来，就可以运行 JMeter Server 了。使用 -s 选项。即，jmeter -s

15.2 Tips

如果你运行在 Suse Linux 上，这些技巧对你很有帮助。默认的安装会启动防火墙，在这种情况下，远程测试将无法正常工作。如果你发现连接被拒绝后，通过下列项打开 debugging。

解决的方法是从 etc/hosts 文件中删除对 127.0.0.1 和 127.0.0.2 的 loopback。当 127.0.0.2 的 lookback 无效时，jmeter-server 无法连接到 rmiregistry。

替换：‘dirname \$0’ /jmeter -s “\$@” 为：

HOST= “-Djava.rmi.serve.hostname=[computer_name][computer_domain] -Djava.security。

policy= ‘dirname \$0’ /[policy_file]” ‘dirname \$0’ /jmeter \$HOST -s “\$@”

同时创建一个 policy 文件，添加 [computer_name][computer_domain] 行到 etc/hosts

16 Best Practices

16.1 Limit the number of Threads

你的硬件能力将限制 JMeter 能运行的线程数。同样也依赖于你的 server 速度有多快。速度快的机器使得 JMeter 工作困难，因为 server 很快的回复了 request。JMeter 工作越多，将降低时间信息的准确度。JMeter 的工作越多，每个线程等待使用 CPU 的时间越长，会越膨胀时间信息。如果你需要一个大数值的负载测试，要先考虑在多个机器上远程使用 JMeter。

16.2 何处放置 Cookie Manger

16.3 何处放置 Authorization Manger

16.4 使用代理服务器

使用代理服务器，最重要的事情是如何过滤你不想要的信息。例如，没有方法记录 image 的请求 (JMeter 可以被设置为下载页面上的所有图片)。这些会扰乱你的测试计划。在大多数情况下，都有共享的文件都有一个扩展名，例如，.jsp, .asp, .php, .html 等等。这些你应该在字段 “Include Pattern” 中输入 “*.jsp” 来引入。

同样的，你可以在 “Exclude Pattern” 中输入 “*.gif” 将其排除在外。依赖于你的系统，这些也许是也可能不是最好的方法。你同样可以使用同样的方法来拒绝 stylesheets, javascript 等等文件。测试你的设置来验证你记录的是你想要的。

代理服务器期待能够找到一个拥有 Recording Controller 的 Thread Group 来记录 HTTP 请求。这样就方便的将你的所有 samples 封装到一个 controller 下，并可以给定其一个名字用来描述测试用例。

现在，浏览一下一个测试用例的步骤。如果你没有预先定义的测试用例，使用 JMeter 去记录你的动作来定义你的测试用例。一旦你完成了一系列步骤的定义，存储整个测试用例到一个合适的命名文件中。然后，刷新开始记录一个新的测试用例。通过这种方式，你可以很快的录制大量的测试用例“粗稿”。

另外一个 proxy server 的用处是你可以抽象出一些特定的普通元素从记录的 samples 中。通过定义一些 user-defined variables 在测试计划级别或者在 User Defined Variables Elements 中，你可以使得 JMeter 自动在你录取的 samples 中替换这些值。例如，你测试在一个 server 的程序，xxx.yyy.com，你可以定义一个变量叫做 server，值为 xxx.yyy.com，然后在你录制的 samples 中的任何地方发现的该值，都会被替换为 “\${server}”。