火龙果·整理
uml.org.cn

JMeter 基础之一录制脚本

Jmeter 是一个非常流行的性能测试工具，虽然与 LoadRunner 相比有很多不足，比如：它结果分析能力没有 LoadRunner 详细；很它的优点也有很多：

- 开源，他是一款开源的免费软件，使用它你不需要支付任何费用，
- 小巧，相比 LR 的庞大（最新 LR11 将近 4GB），它非常小巧，不需要安装，但需要 JDK 环境，因为它是使用 java 开发的工具。
- 功能强大，jmeter 设计之初只是一个简单的 web 性能测试工具，但经过不段的更新扩展，现在可以完成数据库、FTP、LDAP、WebService 等方面的测试。因为它的开源性，当然你也可以根据自己的需求扩展它的功能。

我觉得它更像一个瑞士军刀，小巧，且功能齐全。初次认识 Jmeter 的时候，我觉得它不好，是因为相比 LR 来说，它没有脚本录制功能，也许不是没有，只是我不知道，因为文档上介绍的是这样，我要做一个 web 性能测试的话，就手动的一个个添加循环控制器、http 信息管理头、http 请求等等各种元件。如果测试的脚本较多时，这无疑是个体力活。

Badboy 是一款不错 web 自动化测试工具，利用它来录制脚本，并且录制的脚本可以直接保存为 JMeter 文件来使用。我无疑给我们带来了很大我方便。

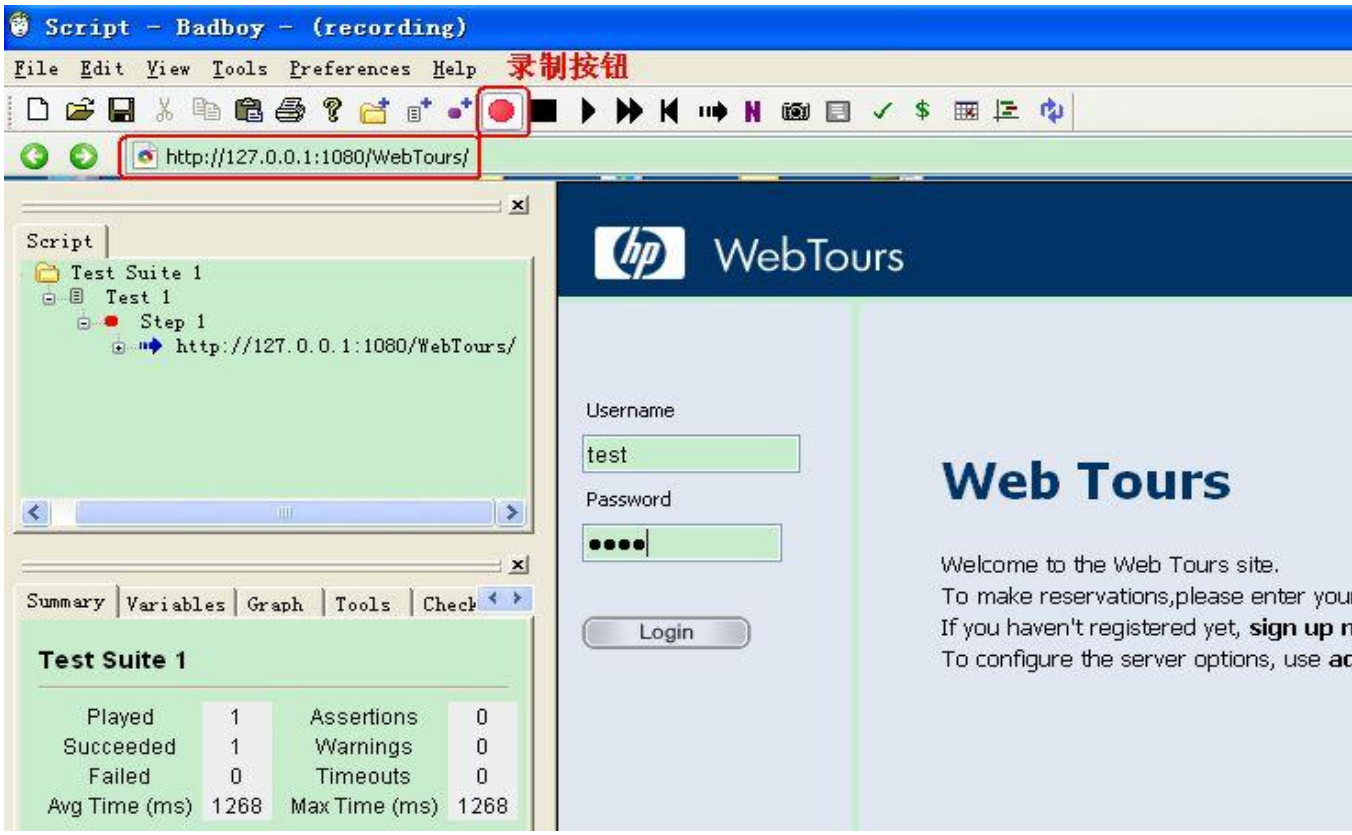
----------我的环境----------

Badboy version 2.1.1

Apache JMeter-2.3.4（需要 JDK 环境来运行）

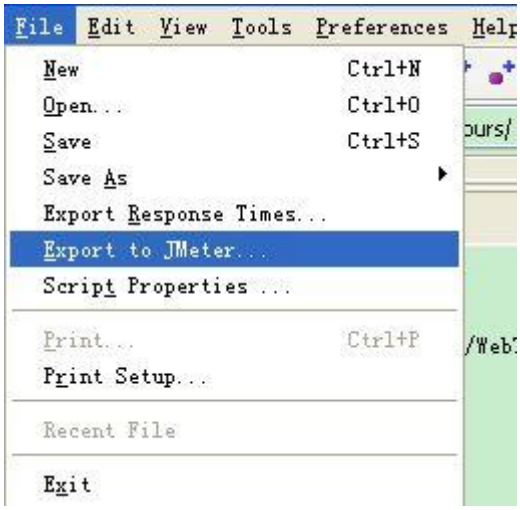
第一种方法：通过 bodboy 来录制脚本。

1.打开人 badboy 工具，点击工栏目上的红色圆形按钮，在地址栏目中输入被测试项目的地址。

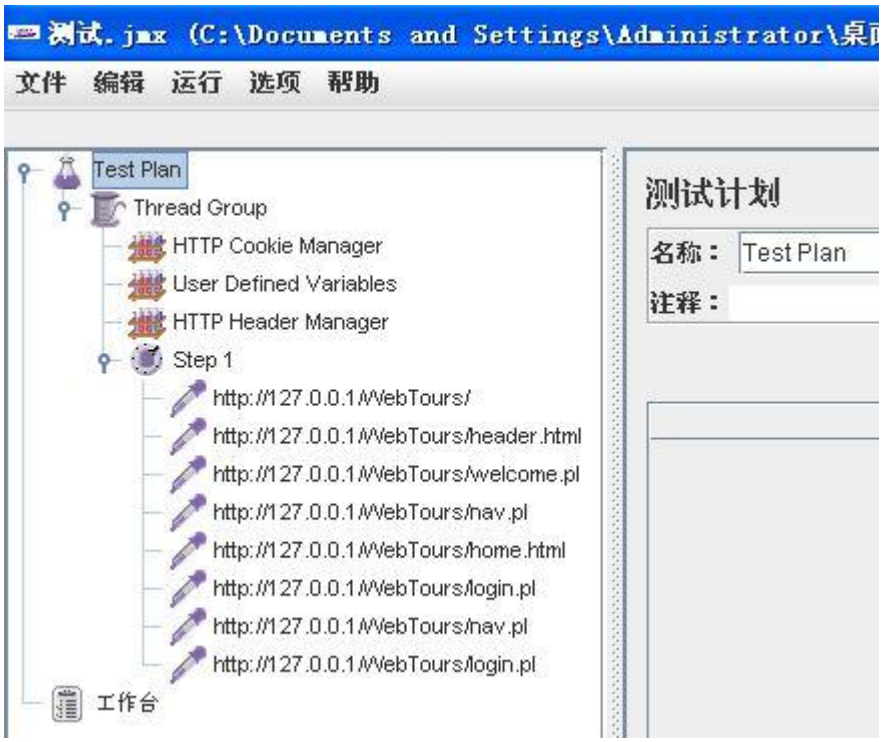


录制完成后，点击工具栏旁边黑色按钮，结束录制。

选择“文件”--→Export to Jmeter…



2. 打开 Jmeter 工具，选择“文件”--✇“打开”选择刚才保存的文件(.jmx 类型)，将文件导入进来了。



第二种方法，通过 JMeter 自身设置来录制脚本。

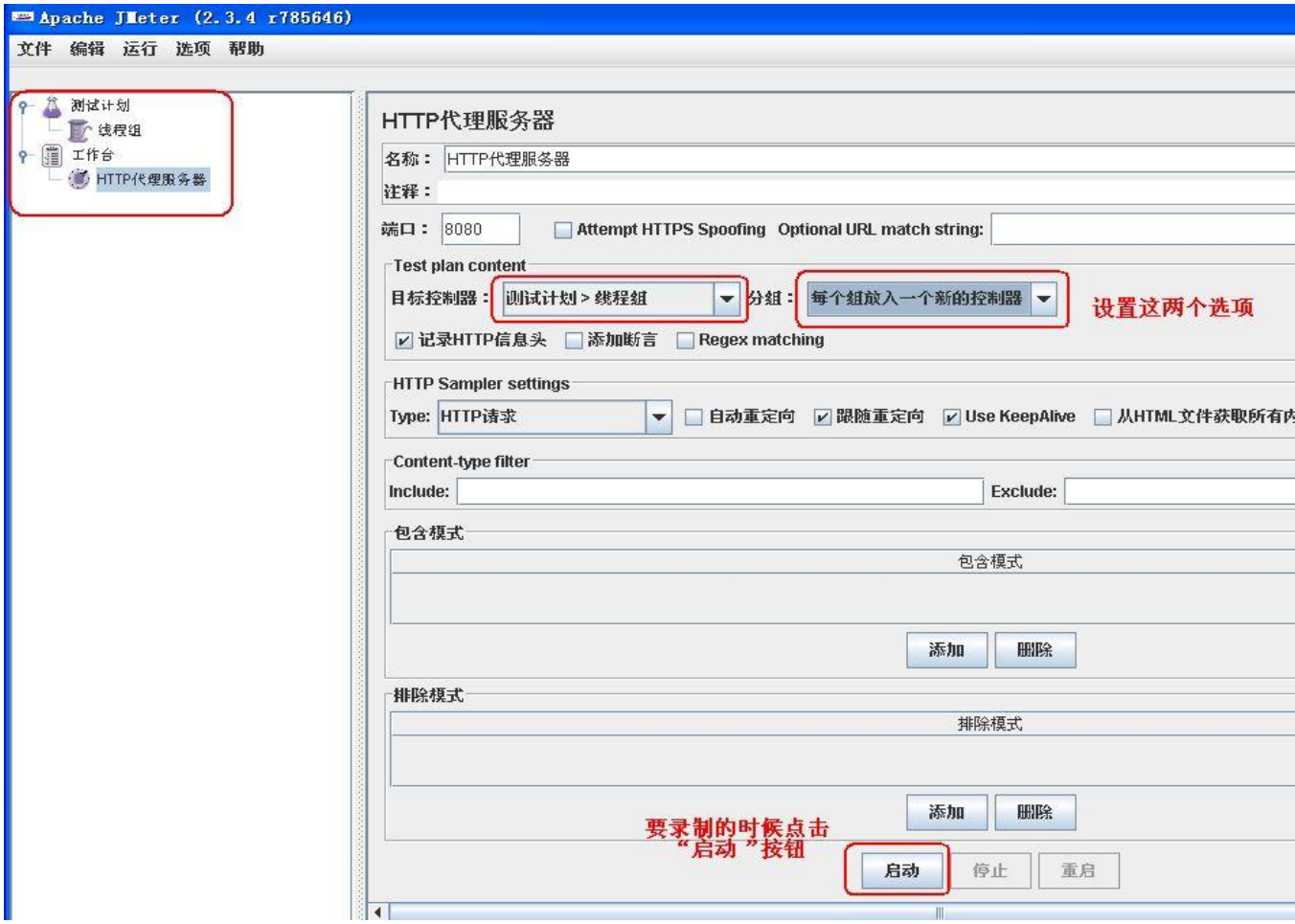
这种方法是我才发现的（鄙视一下自己的无知，嘻嘻~！），觉得方法比较简单。

1. 打开 JMeter 工具

创建一个线程组（右键点击“测试计划”--->“添加”---->“线程组”）

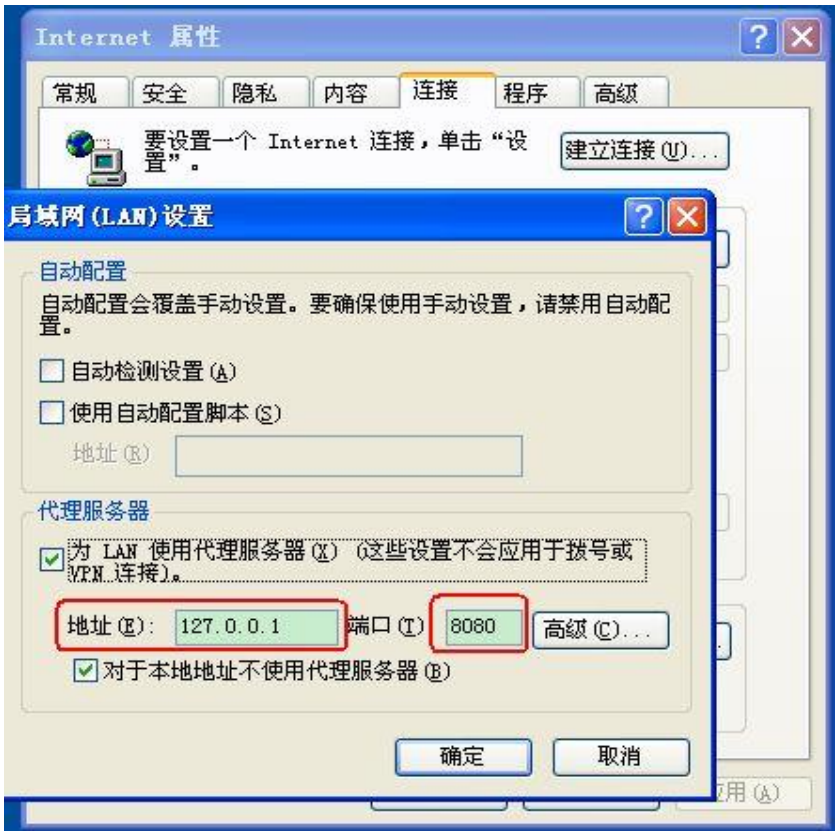
创建一个 http 代理服务器（右键点击“工作台”--->“添加”--->“非测试元件”--->“http 代理服务器”）

完整的设置参照下图：



2. 下面来设置一下 IE 浏览器

IE--->“internet 属性”--->“连接”--->“局域网设置”



设置为本机 IP 就可以了，注意端口号要与 Jmeter 上的端口号一致。默认都是 8080 端口。

3. 现在点击 jmeter 上的“启动”按钮，打开浏览器输入需要录制 web 项目地址，jmeter 会自动记录你 IE 所访问的页面。

PS:第二种方法是我刚才知道的一种，关于这两种方法哪个更好，现在还不知道，但第二方法有通过 IE 浏览器辅助的，我想可能只要 IE 能打开的，它都能记录，但它录制的脚本看上去比较乱（感觉上）

还就是 http 代理服务器的设置，（比如：分组：每一个组放入一个新的服务器---只有这一个选项才能正常录制），有时间再仔细比较一下两种方法的不同之处。

Jmeter 基础之---jmeter 基础概念

2012-12-21 19:15 by 虫师, 5245 阅读, 3 评论, 收藏, 编辑

JMeter 介绍： 一个非常优秀的开源的性能测试工具。

优点：你用着用着就会发现它的重多优点，当然不足点也会呈现出来。

从性能工具的原理划分：

Jmeter 工具和其他性能工具在原理上完全一致，工具包含 4 个部分：

（1）负载发生器：用于产生负载，通常以多线程或是多进程的方式模拟用户行为。

（2）用户运行器：通常是一个脚本运行引擎，用户运行器附加在线程或进程上，根据脚本要求模拟指定的用户行为。

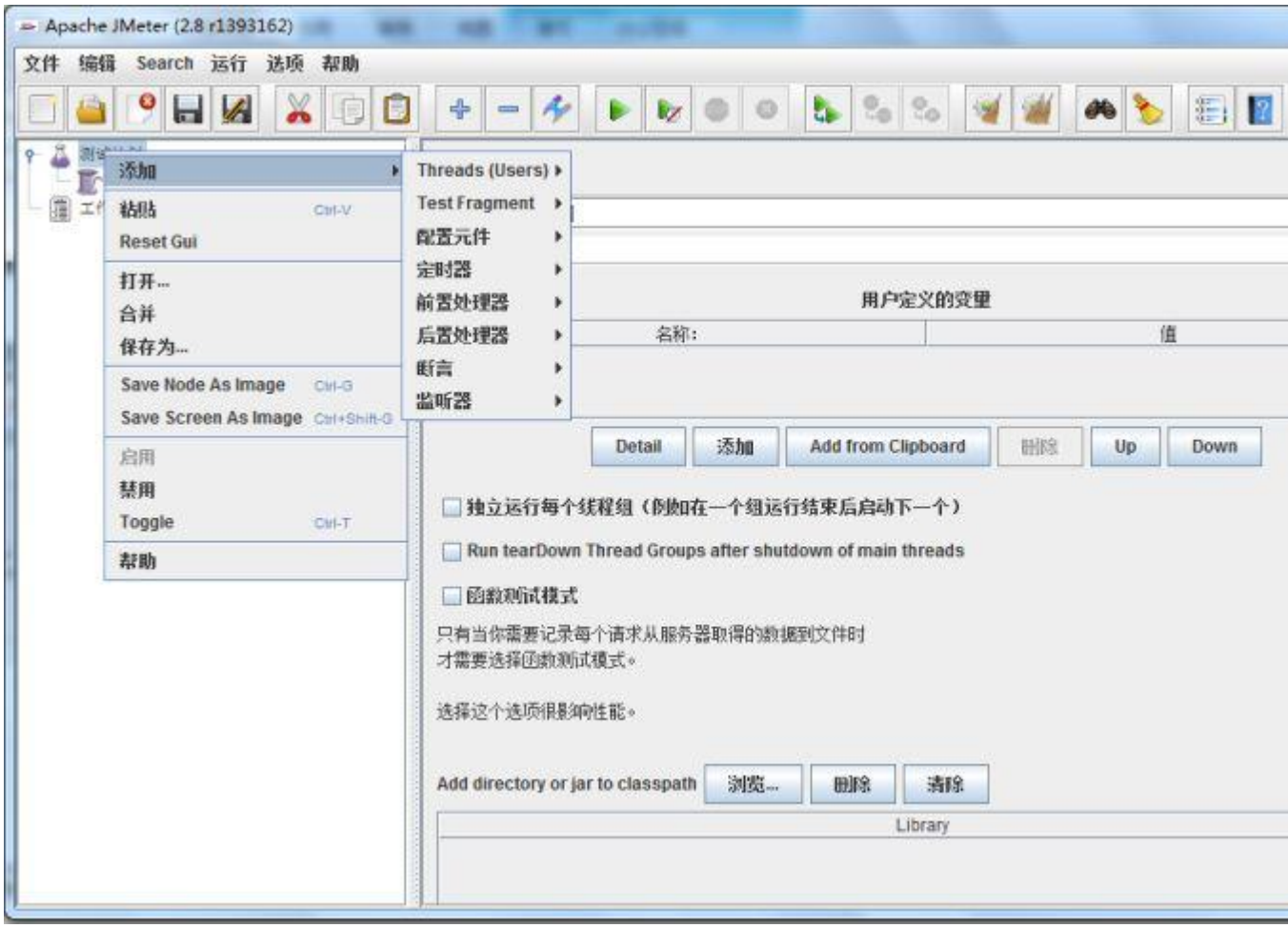
（3）资源生成器：用于生成测试过程中服务器、负载机的资源数据。

（4）报表生成器：根据测试中霍地的数据生成报表，提供可视化的数据显示方式。

测试计划元件

Test Plan (测试计划)：用来描述一个性能测试，包含与本次性能测试所有相关的功能。也就说本的性能测试的所有内容是于基于一个计划的。

下面看一下一个计划下面都有哪些主要的功能模块（右键单击“测试计划”弹出菜单）。



Threads （Users）线程 用户

虽然有三个添加线程组的选项，名字不一样， 创建之后，其界面是完全一样的。之前的版本只有一个线程组的名字。现在多一个 setUp theread Group 与 terDown Thread Group

1) setup thread group

一种特殊类型的 ThreadGroup 的，可用于执行预测试操作。这些线程的行为完全像一个正常的线程组元件。不同的是，这些类型的线程执行测试前进行定期线程组的执行。

2) teardown thread group.

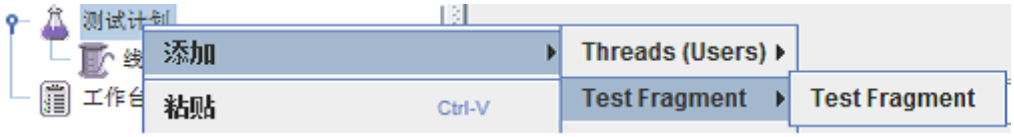
一种特殊类型的 ThreadGroup 的，可用于执行测试后动作。这些线程的行为完全像一个正常的线程组元件。不同的是，这些类型的线程执行测试结束后执行定期的线程组。

可能你还是不太理他们与普通的线程组有什么不同。 如果您用过 junit，想必你不会对 setup ，teardown 这 2 个字眼陌生。 即时每用过，也没关系。 熟悉 loadrunner 的应该知道，loadrunner 的脚本除了 action 里是真正的脚本核心内容，还有初始化“环境”的初始化脚本和测试完毕后对应的清除信息的脚本块。 那么这里 setup thread group 和 teardown thread group 就是分别指这两部分。 其实从本质上来看，他们并没有什么不同。

3) thread group(线程组).

这个就是我们通常添加运行的线程。通俗的讲一个线程组,，可以看做一个虚拟用户组，线程组中的每个线程都可以理解为一个虚拟用户。线程组中包含的线程数量在测试执行过程中是不会发生改变的。

测试片段（Test Fragment）



测试片段是在 2.5 版本之后新加的一个选项。

测试片段元素是控制器上的一个种特殊的线程组，它在测试树上与线程组处于一个层级。它与线程组有所不同，因为它不被执行，除非它是一个模块控制器或者是被控制器所引用时才会被执行。

控制器

JMeter 有两种类型的控制器：取样器（sample）和逻辑控制器（Logic Controller），用这些原件来驱动处理一个测试。

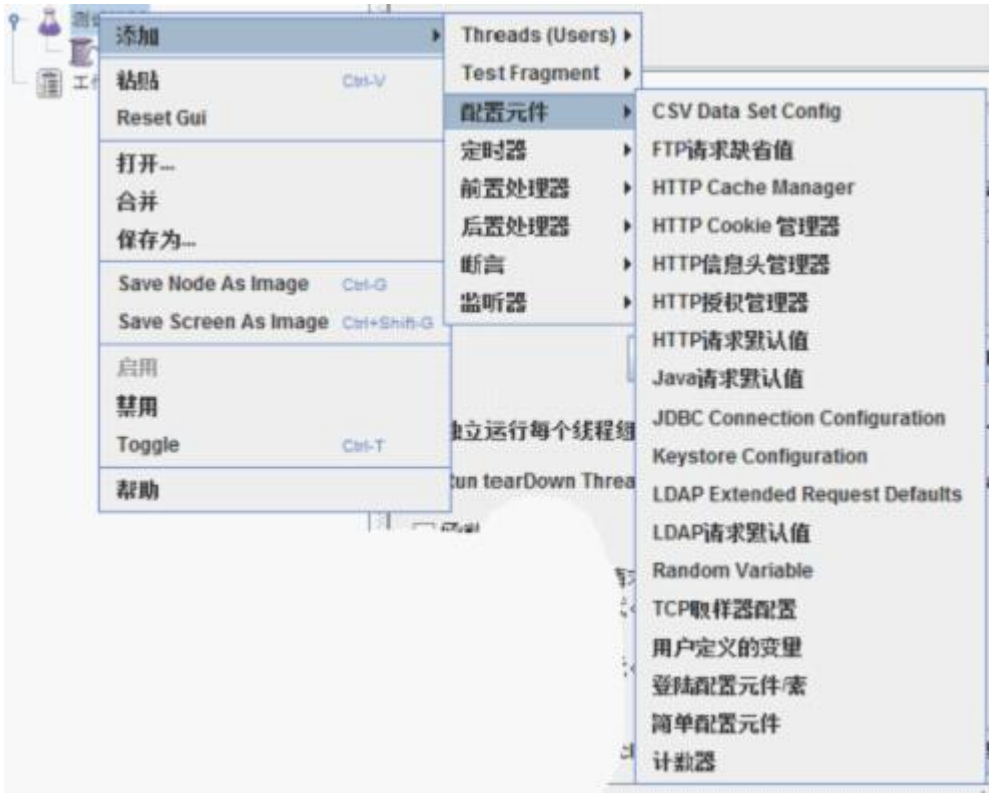
取样器（Sampler）

取样器（Sample）是性能测试中向服务器发送请求，记录响应信息，记录响应时间的最小单元，JMeter 原生支持多种不同的 sampler ，如 HTTP Request Sampler 、 FTP Request Sample 、TCP Request Sample 、JDBC Request Sampler 等，每一种不同类型的 sampler 可以根据设置的参数向服务器发出不同类型的请求。（在 jmeter 的所有 sampler 中，Java Request Sampler 和 Beanshell Request Sampler 是两种特殊的可定制的 Sampler ，后面会深入讨论。）

逻辑控制器（Logic Controller）

逻辑控制器，包括两类无件，一类是用于控制 test plan 中 sampler 节点发送请求的逻辑顺序的控制器，常用的有 如果（If）控制器 、switch Controller 、Runtime Controller、循环控制器等。另一类是用来组织可控制 sampler 来节点的，如 事务控制器、吞吐量控制器。

配置元件（Config Element）



配置元件（config element）用于提供对静态数据配置的支持。CSV Data Set config 可以将本地数据文件形成数据池（Data Pool），而对应于 HTTP Request Sampler 和 TCP Request Sampler 等类型的配制无件则可以修改 Sampler 的默认数据。（例如，HTTP Cookie Manager 可以用于对 HTTP Request Sampler 的 cookie 进行管理）

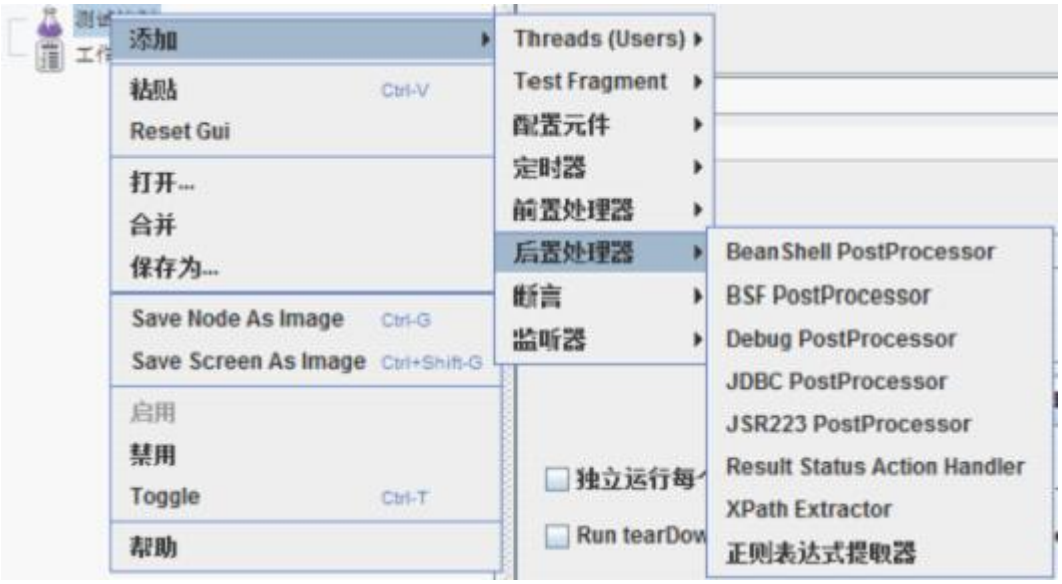
定时器（Timer）

定时器（Timer）用于操作之间设置等待时间，等待时间是性能测试中常用的控制客户端 QPS 的手端。类似于 LoadRunner 里面的“思考时间”。JMeter 定义了 Bean Shell Timer、Constant Throughput Timer、固定定时器等不同类型的 Timer。

前置处理器（Per Processors）

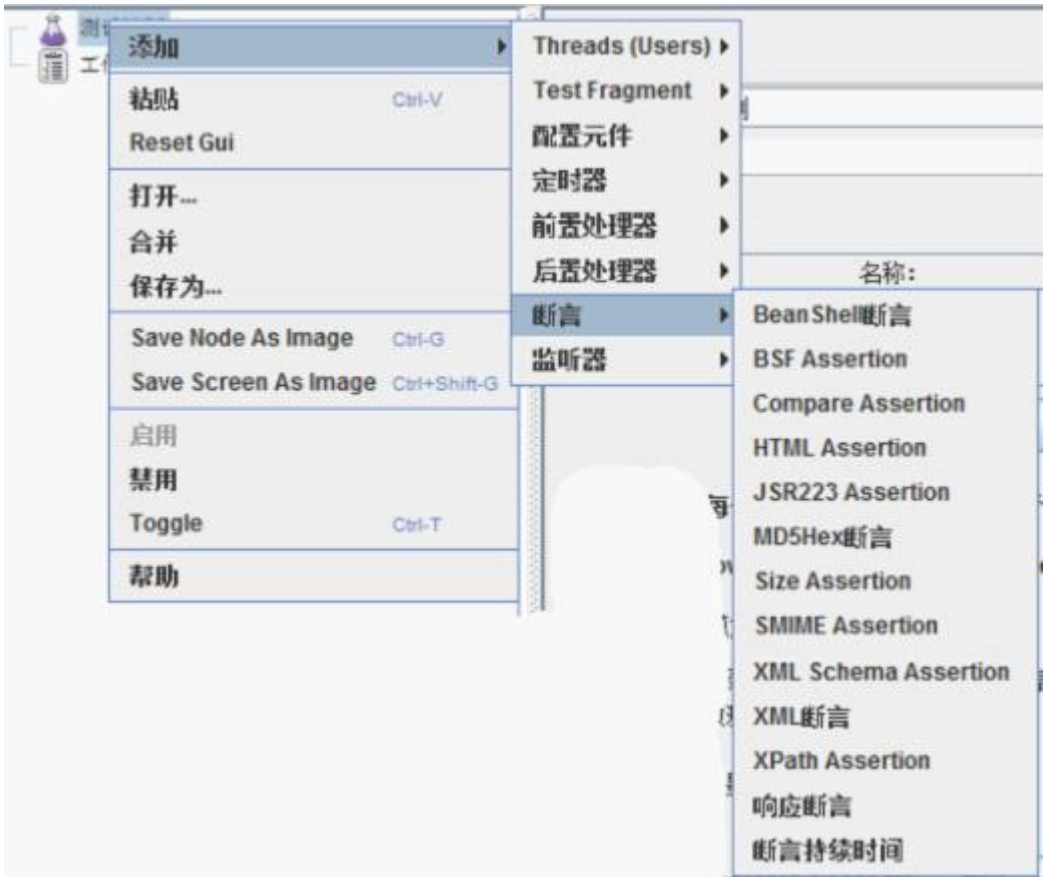
用于在实际的请求发出之前对即将发出的请求进行特殊处理。例如，HTTP URL 重写修复符则可以实现 URL 重写，当 RUL 中有 sessionID 一类的 session 信息时，可以通过该处理器填充发出请求的实际的 sessionID 。

后置处理器（Post Processors）



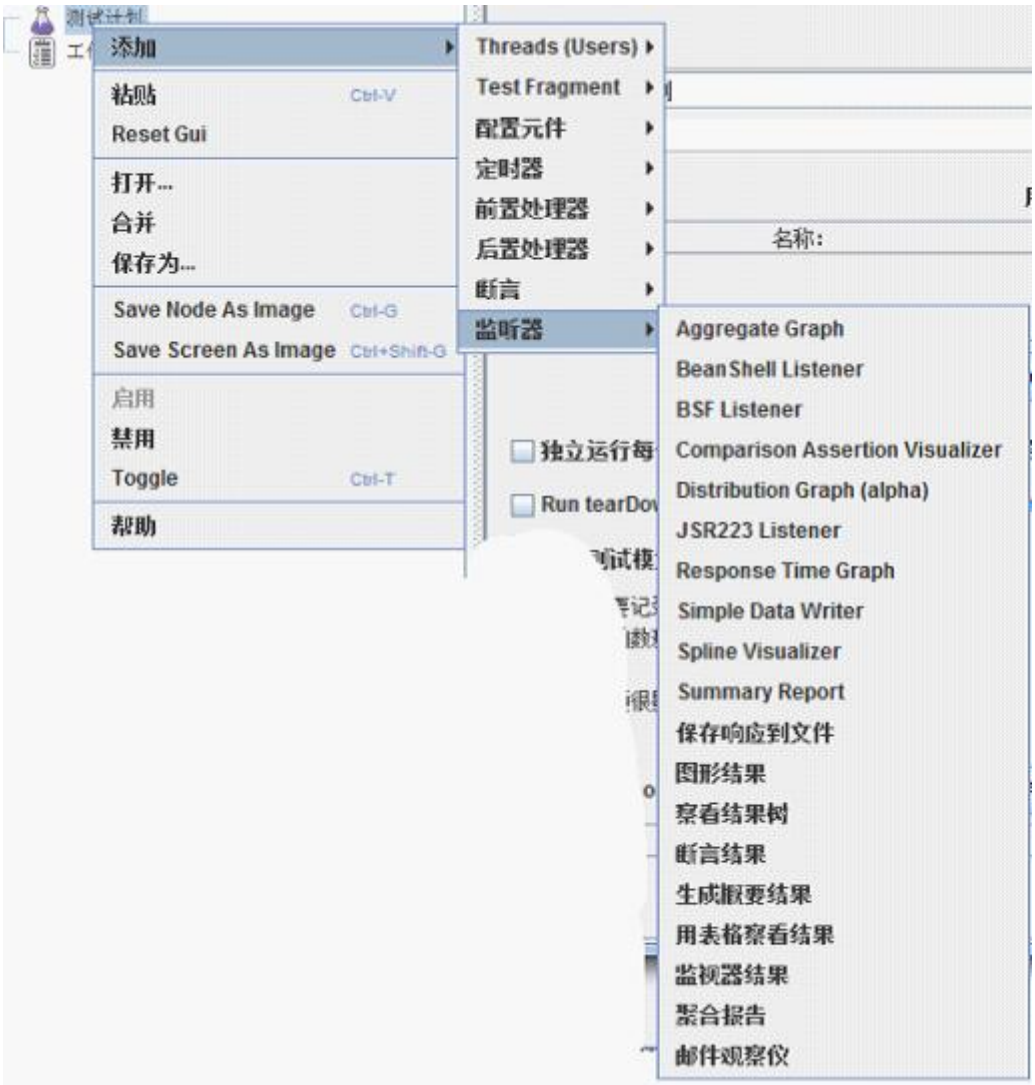
用于对 Sampler 发出请求后得到的服务器响应进行处理。一般用来提取响应中的特定数据（类似 LoadRunner 测试工具中的关联概念）。例如，XPath Extractor 则可以用于提取响应数据中通过给定 XPath 值获得的数据。

断言（Assertions）



断言用于检查测试中得到的相应数据等是否符合预期，断言一般用来设置检查点，用以保证性能测试过程中的数据交互是否与预期一致。

监听器（Listener）



这个监听器可不是用来监听系统资源的元件。它是用来对测试结果数据进行处理和可视化展示的一系列元件。 图行结果、查看结果树、聚合报告。都是我们经常用到的元件。

到此，我们已经简单了解了 jmeter 的基本组成原件，我们后序的性能测试工作也就是使用这些元件来完成测试任务。

JMeter 基础之--元件的作用域与执行顺序

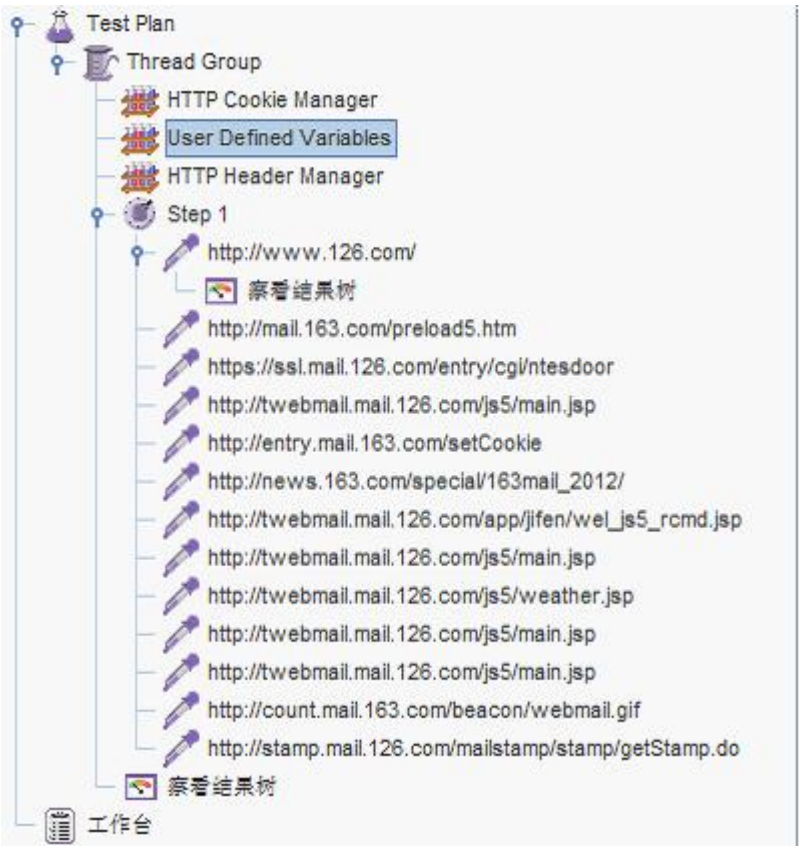
2012-12-27 22:20 by 虫师, 2609 阅读, 0 评论, 收藏, 编辑

前面有介绍过 jmeter 的元件类别，对于新手来说，jmeter 的元件是还是不少的，如果我们按照每一个元件的每一个参数的含义去学习，无疑会降低学习性能测试的热情，就算我们熟悉了所有元件以及元件上的参数了，我们也不知道如何将其组合成我们想要描述的性能测试。另一方面，我们点击每个节点上的元件都会出现类似或相同的菜单，他们的层级关系是什么样的。究竟这个元件放在哪个层级的哪个位置才达到我们意愿。

如何更快速的入门 jmeter，个人建议通过录制脚本的方式，快速的了解一个性能测试应该包括的元件以及它们的层级关系。

关于录制方式，请参考《JMeter 基础之—录制脚本》

如下，我通过 badboy 工具录制的一个脚本（脚本过程：登录 126 邮箱，给自己发一封邮件，祝自己圣诞快乐！），并可以方便的将脚本导出为 jmx 格式，这也是 jmeter 脚本的格式，所以可以被 jmeter 识别并导入。



脚本导入到 jmeter 后，我添加了两个监听器（查看结果树），因为我添加监听器的位置不同，他们的作用域也是不一样的。前一个只能监听到 www.126.com 页面的请求，后一个侧对所有的 http 请求进行监听。

元件的作用域

先来讨论一下元件有作用域。《Jmeter 基础之---jmeter 基础概念》一章中，我们介绍了 8 类可被执行的元件（测试计划与线程组不属于元件），这些元件中，取样器是典型的不与其它元件发生交互作用的元件，逻辑控制器只对其子节点的取样器有效，而其它元件（config elements 、timers 、post-processors、assertions、listeners、）需要与取样器（sampler）等元件交互。

配置元件（config elements ）

元件会影响其作用范围内的所有元件。

前置处理程序（Per-processors）

元件在其作用范围内的每一个 sampler 元件之前执行。

定时器（timers ）

元件对其作用范围内的每一个 sampler 有效

后置处理程序（Post-processors）

元件在其作用范围内的每一个 sampler 元件之后执行。

断言（Assertions）

元件对其作用范围内的每一个 sampler 元件执行后的结果执行校验。

监听器（Listeners）

元件收集其作用范围的每一个 sampler 元件的信息并呈现。

在 jmeter 中，元件的作用域是靠测试计划的的树型结构中元件的父子关系来确定的，作用域的原则是：

- 取样器（sampler）元件不和其它元件

相互作用，因此不存在作用域的问题。

-
- 逻辑控制器（Logic Controller）元件只对其子节点

中的取样器和逻辑控制器作用。

- 除取样器和逻辑控制器元件外，其他6类元件，如果是某

个sampler的子节点，则该元件公对其父子节点起作用。

-
- 除取样器和逻辑控制器元件外的其他6类

火龙果 · 整理
uml.org.cn

元件，如果其父节点不是sampler，则其作用域是该元件父节点下的其他所有后代节点（包括子节

点，子节点的子节点等）。

讲了这些，你可能迷糊了，到底是肿么个情况呀！？通过两个栗子（例子）来理解一下他们的作用域。

A（注：下图只是为了说明作用域，无法正常运行）

取样器（HTTP 请求 1 、FTP 请求 2 、TCP 取样器 3） 逻辑控制器（循环控制器） 监听器（图形结果 1、聚合报告 2）

* HTTP 请求 1 、FTP 请求 2 、TCP 取样器 3 元件没有作用域的概念。

* 循环控制器 元件作用域名是其子节点 FTP 请求 2 、TCP 取样器 3 。

* 图形结果 1 元件的作用域是是 FTP 请求 2 、TCP 取样器 3。

* 聚合报告 2 元作的作用域是 HTTP 请求 1 、FTP 请求 2 、TCP 取样器 3

B （注：下图只是为了说明作用域，无法正常运行）

这个例子稍微复杂一些，包含的元件较多。先来分分类。

取样器（HTTP 请求 、FTP 请求 、TCP 取样器、 JDBC Request ）逻辑控制器（循环控制器、随即控制器）定时器（固定定时器、Uniform Random Timer ）断言（响应断言、XML 断言）监听器（图形结果、聚合报告）

根据作用域原则，这些元件的作用域分别为：

* HTTP 请求 、FTP 请求 、TCP 取样器、 JDBC Request 元件没有作用域名概念

* 循环控制器 的作用域为 FTP 请求 、TCP 取样器和 随即控制器

* 固定定时器作用于 HTTP 请求 ， Uniform Random Timer 作用于所有取样器

* 响应断言作用于 JDBC Request ， XML 断言作用于 FTP 请求 、TCP 取样和 JDBC Request 。

* 图形结果作用于 FTP 请求 、TCP 取样和 JDBC Request ，聚合报告作用于作用于所有取样器。

其实，通过上面的分析，并没有你想象的那么复杂，我们从各个元件的层次结构就可以判断每个元件的作用域。

Jmeter 中的逻辑控制器（Config Elements）在其作用范围内的行为与其他元件相比稍有不同。认配置（HTTP 默认请求、FTP 默认请求等）和 管理（HTTP 头管理、HTTP cookie 管理等）。

其中默认配置（Configuration Defaults）元件中设置的值可以在作用域内叠加，例如，在一个测请求，其中第一个默认设置 Server name or IP 为 www.google.com ，第二个默认设置 Path 为/page-n 用域内的所有 HTTP 默认请求，其默认的 Server name or IP 和 Path 均为 Server name or IP 和 /page 我有一个笑话，我们一交换，两个人都分别拥有了一个故事加一个笑话。

管理（Manager）类逻辑控制器元件的效果则不能进行叠加。如果两个或两个以上相同的管理类作用域内的取样器元件只会随即受到其中一个的作用，这样会导致取样器行为的不确定性。因此，在 定要注意保证相同的管理类元件的作用域不发生重叠。

元件的执行顺序

了解了元件有作用域之后，来看看元件的执行顺序，元件执行顺序的规则很简单，在同一作用域名范围内，测试计划中的元件按照如下顺序执行。

（1）配置元件（config elements ）

（2）前置处理程序（Per-processors）

（3）定时器（timers ）

（4）取样器（Sampler）

（5）后置处理程序（Post-processors） （除非 Sampler 得到的返回结果为空）。

（6）断言（Assertions）（除非 Sampler 得到的返回结果为空）。

（7）监听器（Listeners）（除非 Sampler 得到的返回结果为空）。

关于执行顺序，有两点需要注意：

* 前置处理器、后置处理器和断言等元件公能对 取样器作用，因此，如果在它们的作用域内没有任何取样器，则不会被执行。

* 如果在同一作用域范围内有多个同一类型的元件，则这些元件按照它们在测试计划中的上下顺序一次执行。

注：本文件中提到的取样器（Sampler）在有些资料中翻译为“采样器”。

JMeter 基础之一 一个简单的性能测试

2012-12-22 21:04 by 虫师, 8377 阅读, 3 评论, 收藏, 编辑

上一节中，我们了解了 jmeter 的一此主要元件，那么这些元件如何使用到性能测试中呢。这一节创建一个简单的测试计划来使用这些元件。该计划对应的测试需求。

1）测试目标网站是 fnng.cnblogs.com 和 tt-topia.rhcloud.com

2）测试目的是该网站在负载达到 20 QPS 时的响应时间。

QPS 解释

QPS ：Query Per Second 每秒查询率。是一台查询服务器每秒能够处理的查询次数。在因特网上，作为域名系统服务器的机器的性能经常用每秒查询率来衡量。

为了达成预期的测目的，需要需要在 jmeter 中建立一个测试计划。因为本次测试仅要求完成对 fnng.cnblogs.com 和 tt-topia.rhcloud.com 两个博客首页请求，因此只需要使用 HTTP Request Sampler 即可。

建立测试计划

启动 jmeter 后，jmeter 会自动生成一个空的测试计划，用户可以基于该测试计划建立自己的测试计划。

添加线程组

一个性能测试请求负载是基于一个线程组完成的。一个测试计划必须有一个线程组。测试计划添加线程组非常简单。在测试计划右键弹出下拉菜单（添加-->Threads(Users)--->线程组）中选择线程组即可。

jmeter 中 每个测试计划至少需要包含一个线程组，当然也可以在一个计划中创建多个线程组，那么多个线程组之间又会怎样的顺序执行（串行还是并行）？在测试计划下面多个线程是并行执行的，也就是说这些线程组是同时被初始化并同时执行线程组下的 Sampler 的。

线程组主要包含三个参数：线程数、准备时长（Ramp-Up Period(in seconds)）、循环次数。

线程数：虚拟用户数。一个虚拟用户占用一个进程或线程。设置多少虚拟用户数在这里也就是设置多少个线程数。

准备时长： 设置的虚拟用户数需要多长时间全部启动。如果线程数为 20 ，准备时长为 10 ，那么需要 10 秒钟启动 20 个线程。也就是每秒钟启动 2 个线程。

循环次数：每个线程发送请求的次数。如果线程数为 20 ，循环次数为 100 ，那么每个线程发送 100 次请求。总请求数为 20*100=2000 。如果勾选了“永远”，那么所有线程会一直发送请求，一到选择停止运行脚本。

设置合理的线程数对于能否达到测试目标有决定性的影响。在本例中，要求得到网站首页在 20 QPS 负载情况下的响应时间，如果如果线程数量设置的过小，则很可能无法达到设定的 QPS 要求。另外，设置合理的循环次数也很重要，除了上面介绍的固定循环次数与永远外；也可以灵活的选择设定测试运行时间。勾选“调度器”，进行调度器配置。

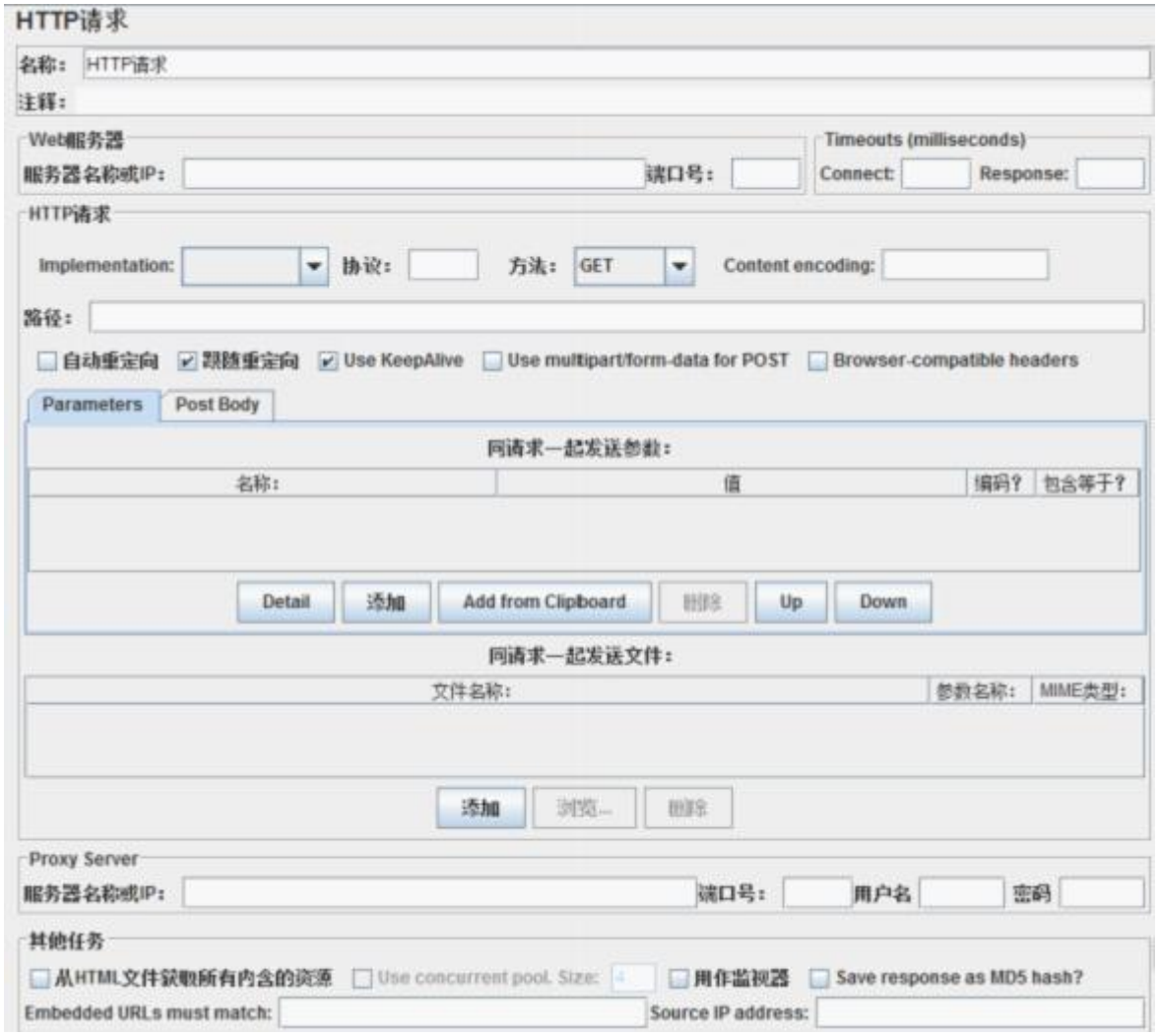
添加 HTTP 请求

添加完成线程组后，在线程组上右键菜单（添加--->Sampler--->HTTP 请求）选择 HTTP 请求。对于 jmeter 来说，取样器（Sampler）是与服务器进行交互的单元。一个取样器通常进行三部分的工作：

向服务器发送请求

记录服务器的响应数据

记录相应时间信息



一个 HTTP 请求有着许多的配置参数，下面将详细介绍：

名称：本属性用于标识一个取样器，建议使用一个有意义的名称。

注释：对于测试没有任何作用，仅用户记录用户可读的注释信息。

服务器名称或 IP ：HTTP 请求发送的目标服务器名称或 IP 地址。

端口号：目标服务器的端口号，默认值为 80 。

协议：向目标服务器发送 HTTP 请求时的协议，可以是 http 或者是 https ，默认值为 http 。

方法：发送 HTTP 请求的方法，可用方法包括 GET、POST、HEAD、PUT、OPTIONS、TRACE、DELETE 等。

Content encoding ：内容的编码方式，默认值为 iso8859

路径：目标 URL 路径（不包括服务器地址和端口）

自动重定向：如果选中该选项，当发送 HTTP 请求后得到的响应是 302/301 时，JMeter 自动重定向到新的页面。

Use keep Alive ： 当该选项被选中时，jmeter 和目标服务器之间使用 Keep-Alive 方式进行 HTTP 通信，默认选中。

Use multipart/from-data for HTTP POST ：当发送 HTTP POST 请求时，使用 Use multipart/from-data 方法发送，默认不选中。

同请求一起发送参数 ： 在请求中发送 URL 参数，对于带参数的 URL ，jmeter 提供了一个简单的对参数化的方法。用户可以将 URL 中所有参数设置在本表中，表中的每一行是一个参数值对（对应 RUL 中的 名称 1=值 1）。

同请求一起发送文件：在请求中发送文件，通常，HTTP 文件上传行为可以通过这种方式模拟。

从 HTML 文件获取所有有内含的资源：当该选项被选中时，jmeter 在发出 HTTP 请求并获得响应的 HTML 文件内容后，还对该 HTML 进行 Parse 并获取 HTML 中包含的所有资源（图片、flash 等），默认不选中，如果用户只希望获取页面中的特定资源，可以在下方的 Embedded URLs must match 文本框中填入需要下载的特定资源表达式，这样，只有能匹配指定正则表达式的 URL 指向资源会被下载。

用作监视器：此取样器被当成监视器，在 Monitor Results Listener 中可以直接看到基于该取样器的图形化统计信息。默认为不选中。

Save response as MD5 hash? ：选中该项，在执行时仅记录服务端响应数据的 MD5 值，而不记录完整的响应数据。在需要进行数据量非常大的测试时，建议选中该项以减少取样器记录响应数据的开销。

在这里我们添加两个 HTTP 请求，分别用于对 fnng.cnblogs.com 和 tt-topia.rhcloud.com 发送请求。

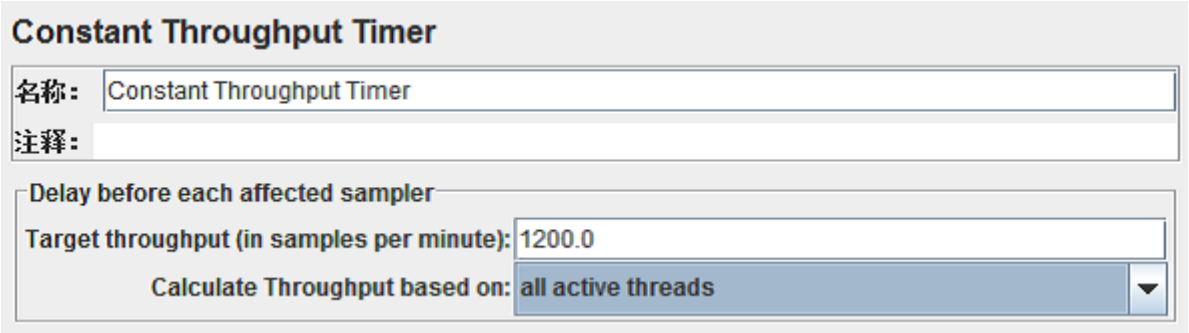
设置 QPS 限制

本次性能测试的需求中提到测试的目的是“了解博客的首页在负载达到 20 QPS 时的响应时间”，因此需要控制向博客首页发送请求的负载为 20QPS。

一种可行的方法是逐步调整测试计划中的线程计算的数量以及为取样器（Sampler）添加定时器（Timer），以使 HTTP 取样器发出的请求的 QPS 保持在 20 个左右。但这种方法耗时耗力，需要经过多次尝试才能达到；另一方法，完全通过设置定时器来控制 QPS，一旦取样器的响应时间发生改变（网络环境发生改变），就需要重新调整定时器的等待时间。

Jmeter 提供了一个非常有用的定时器，称为 Constant Throughput Timer（常数吞吐量定时器），该定时器可以方便地控制给定的取样器发送请求的吞吐量。

右键点击 fnng.cnblogs.com ，弹出菜单（添加--->定时器--->Constant Throughput Timer）选择 Constant Throughput Timer



Constant Throughput Timer 的主要属性介绍：

名称 ：定时器的名称

Target throughput（in samples per minute）：目标吞吐量。注意这里是每分钟发送的请求数，因此，对应测试需求中所要求的 20 QPS ，这里的值应该是 1200 。

Calculate Throughput based on ：有 5 个选项，分别是：

This thread only ：控制每个线程的吞吐量，选择这种模式时，总的吞吐量为设置的 target Throughput 乘以矣线程的数量。

All active threads ： 设置的 target Throughput 将分配在每个活跃线程上，每个活跃线程在上一次运行结束后等待合理的时间后再次运行。活跃线程指同一时刻同时运行的线程。

All active threads in current thread group ：设置的 target Throughput 将分配在当前线程组的每一个活跃线程上，当测试计划中只有一个线程组时，该选项和 All active threads 选项的效果完全相同。

All active threads （shared ）：与 All active threads 的选项基本相同，唯一的区别是，每个活跃线程都会在所有活跃线程上一次运行结束后等待合理的时间后再次运行。

All cative threads in current thread group （shared ）：与 All active threads in current thread group 基本相同，唯一的区别是，每个活跃线程都会在所有活跃线程的上一次运行结束后等待合理的时间后再次运行。

如上图，该元件仅作用于 fnng.cnblogs.com ，设置定时器的 Target throughput 为 1200/分钟（20 QPS），设置 Calculate Throughput based on 的值为 All active threads 。

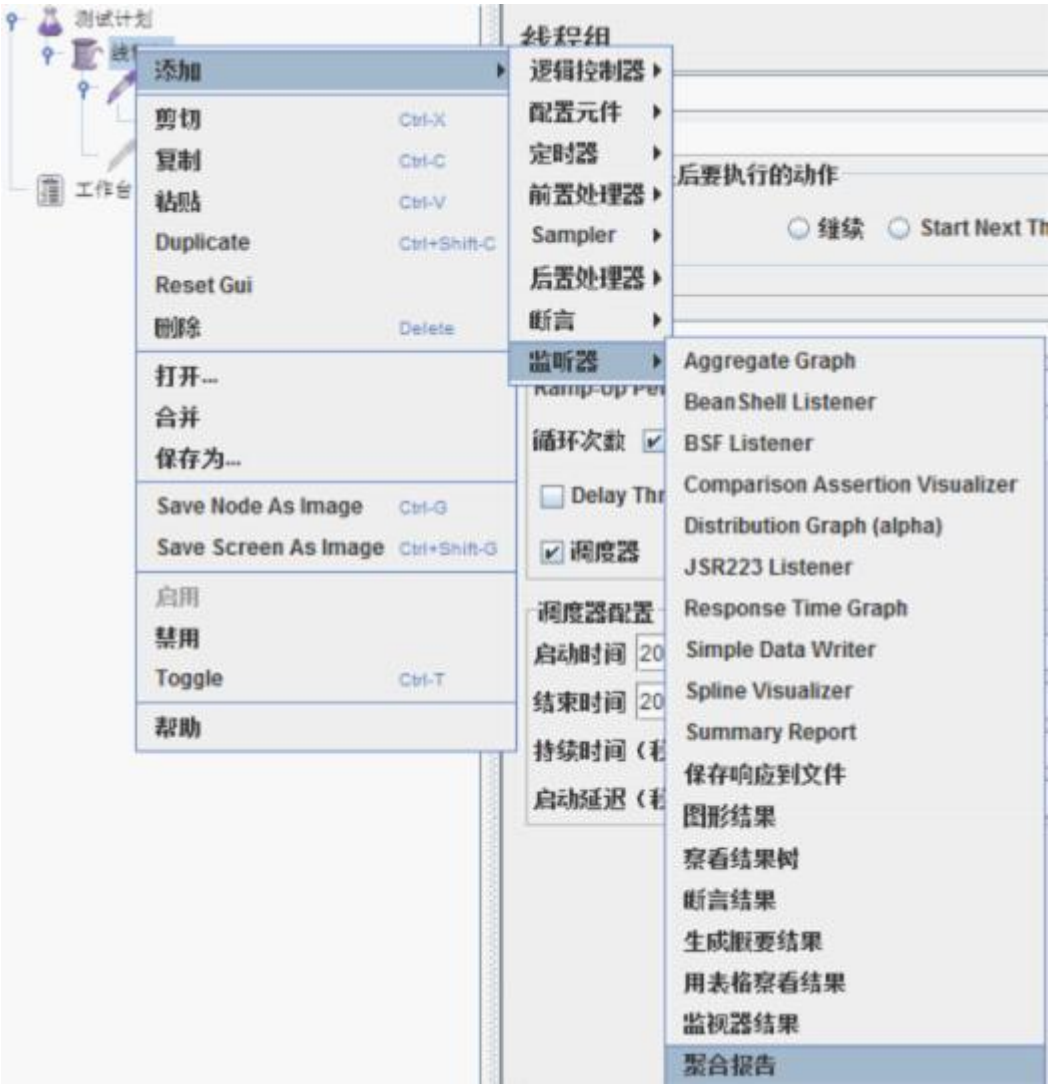
当然，Constant Throughput Timer 只有在线程组中的线程产生足够多的 request 的情况下才有意义，因此，即使设置了 Constant Throughput Timer 的值，也可能由于线程组中的线程数量不够，或是定时器设置不合理等原因导致总体的 QPS 不能达到预期目标。

添加监听器（Listener）

脚本的主要部分设置完成后，需要通过某种方式获得性能测试中的测试结果，在本例中，我们关心的是请求的响应时间。

Jmeter 中使用监听器元件收集取样器记录的数据并以可视化的方式来呈现。Jmeter 有各种不同的监听器类型，因为上 HTTP 请求，我们可在添加聚合报告，更为直观的查看测试结果。

添加聚合报告，右键点击线程组，在弹的菜单（添加--->监听器--->聚合报告）中选择聚合报告。



运行脚本

添加完成聚合报告后，我们来运行脚本，稍后介绍聚合报告的参数。

在运脚本之前，我们来查看一下，各个元件的参数设置：

线程组：

线程数：20

准备时长： 10

循环次数：10

HTTP 请求：

名称：fnng.cnblogs.com。

服务器名称或 IP ： fnng.cnblogs.com

端口号：80

Implementation ： java

协议： http

方法： GET

路径：/

常数吞吐量定时器：

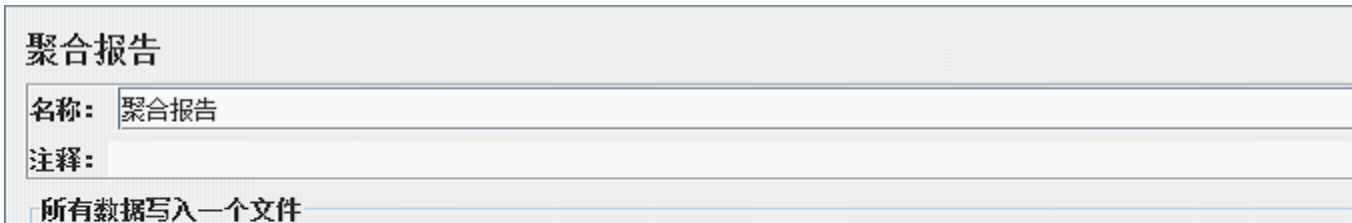
Target throughput（in samples per minute）：1200.0

Calculate Throughput based on ：All active threads

点击工具栏上的运行按钮，或者点击菜单栏“ 运行--->启动 ” 或者使用快捷键 ctrl +r 来运行程序。

聚合报告分析

查看聚合报告的运行结果：



Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throu
fnng.cnblog...	200	129	116	159	92	736	0.00%	1
总体	200	129	116	159	92	736	0.00%	1

Term	Definition
Label	每个 JMeter 的 element(例如 HTTP Request）都有一个 Name 属性，这里显示的就是 Name 属性的值
#Samples	表示你这次测试中一共发出了多少个请求，如果模拟 10 个用户，每个用户迭代 10 次，那么这里显示 100
Average	平均响应时间——默认情况下是单个 Request 的平均响应时间，当使用了 Transaction Controller 时，也可以以 Transaction 为单位显示平均响应时间。
Median	中位数，也就是 50% 用户的响应时间。
90%Line	90% 用户的响应时间。
Min	最小响应时间。
Max	最大响应时间 。
Error%	本次测试中出现错误的请求的数量/请求的总数。
Throughput	吞吐量——默认情况下表示每秒完成的请求数（Request per Second），当使用了 Transaction Controller 时，也可以表示类似 LoadRunner 的 Transaction per Second 数
KB/sec	每秒从服务器端接收到的数据量，相当于 LoadRunner 中的 Throughput/Sec

OK! 到此一次完整的性能测试结束，如果你从中有所收获，推荐一记~！

JMeter 使用技巧

2012-12-29 23:17 by 虫师, 5023 阅读, 0 评论, 收藏, 编辑

在这此对新版本 jmeter 的学习+温习的过程，发现了一些以前不知道的功能，所以，整理出来与大分享。本文内容如下。

1. 如何使用英文界面的 **jmeter**
2. 如何使用镜像服务器
3. **Jmeter** 分布式测试
4. 启动 **Debug** 日志记录
5. 搜索功能
6. 线程之间传递变量

如何使用英文界面的 JMeter

Jmeter 启动时会自动判断操作系统的 locale 并选择合适的语言启动，所以，我们启动 jmeter 后，其会出现一个倍感亲切的中文界面。但由于 jmeter 本身的汉化工作做得不好，你会看到有未被汉化的选项及元件的参数。而且部分翻译并不准确，因此对于英文比较好的牛人来说更喜欢纯正的英文界面。

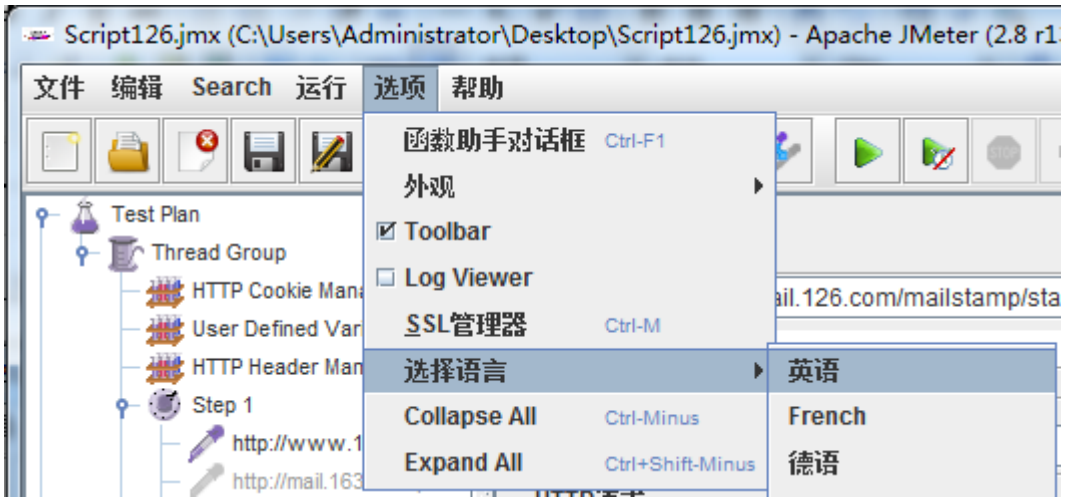
强制以英文方式启动 jmeter 的方法如下：

在 windows 环境下，打开 jmeter 解压目录，bin 目录下的 jmeter.bat 文件，也就是 jmeter 程序的启动文件，选择记事本方式打开。做以下修改：

```
.................
set  HEAP=-Xms512m  -Xmx512m
set  NEW=-XX:NewSize=128m  -XX:MaxNewSize=128m
set  SURVIVOR=-XX:SurvivorRatio=8  -XX:TargetSurvivorRatio=50%
set  TENURING=-XX:MaxTenuringThreshold=2
set  RMIGC=-Dsun.rmi.dgc.client.gcInterval=600000  -Dsun.rmi.dgc.server.gcInterval=6
00000
set  PERM=-XX:PermSize=64m  -XX:MaxPermSize=64m
set  LOCALE=-Duser.language=en    -Duser.region=rem
set  DEBUG=-verbose:gc  -XX:+PrintTenuringDistribution

.........
rem  Server  mode
rem  Collect  the  settings  defined  above
set  ARGS=%DUMP%    %HEAP%    %NEW%    %SURVIVOR%    %TENURING%
  %RMIGC%    %PERM%    %DDRAW%    %LOCALE%
.............
```

晕死，当我上面介绍了那么多后，在最新的 2.8 版本，我无意中发现了这个功能。



如何使用镜像服务器

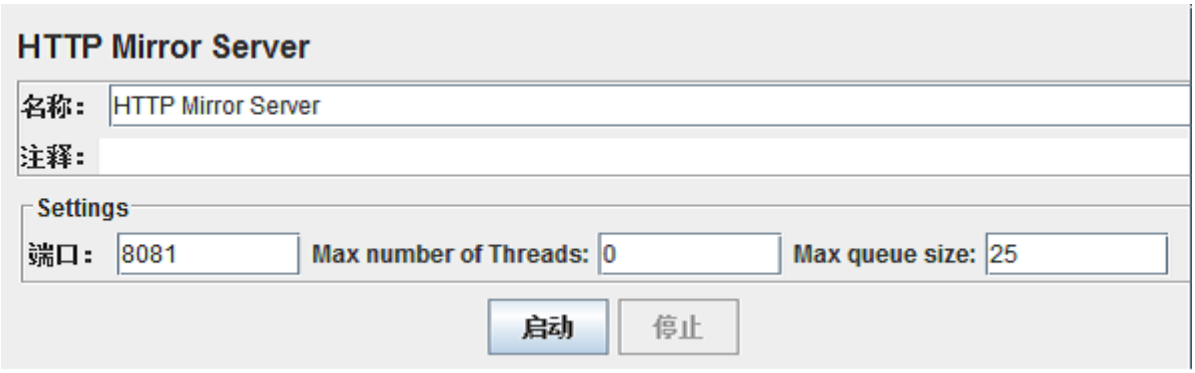
在调试和修改测试计划的过程中，通常会为采样器增加一些额外的设置，例何设置额外的 HTTP 头、cookie 管理器或认证管理器等，但当设置了这些内容后，sampler 发出的请求是否就与预期的完全一支呢？

当然用户可以通过添加监听器来看查采样器发出的 HTTP 请求，但如果调试过程中并不想真正地把请求发送给被测应用，如何解决这个问题呢？

Jmeter 提供了一个名叫 HTTP Mirror Server 的组件，HTTP Mirror Server 可以启动一个镜像的服务器，该服务器把所有接收到的请求原封不动地返回，这样就可以看到发出请求的具体内容了。

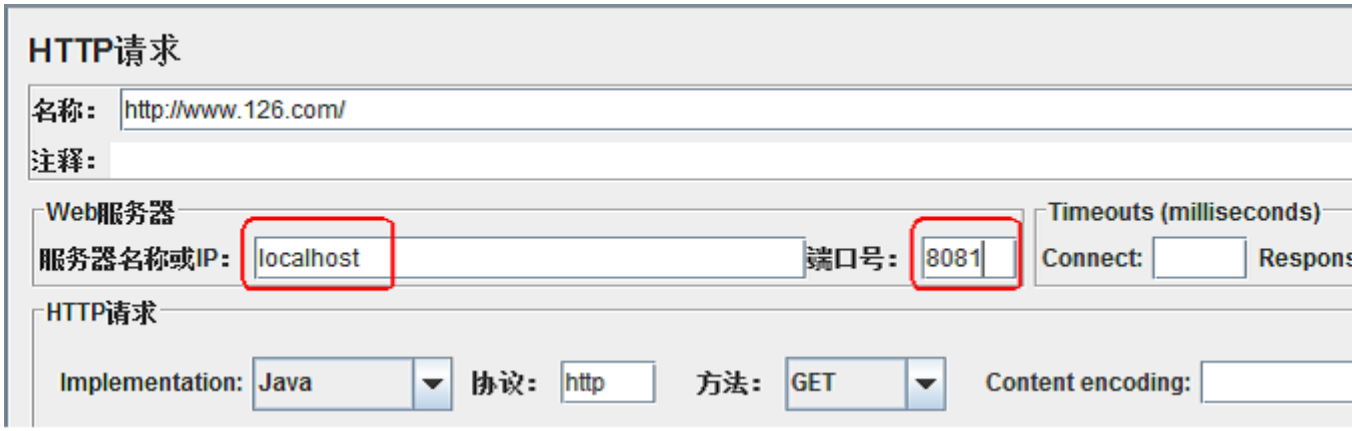
添加 HTTP Mirror Server 的方式：

右键点击“工作台”--->非测试元件--->HTTP Mirror Server

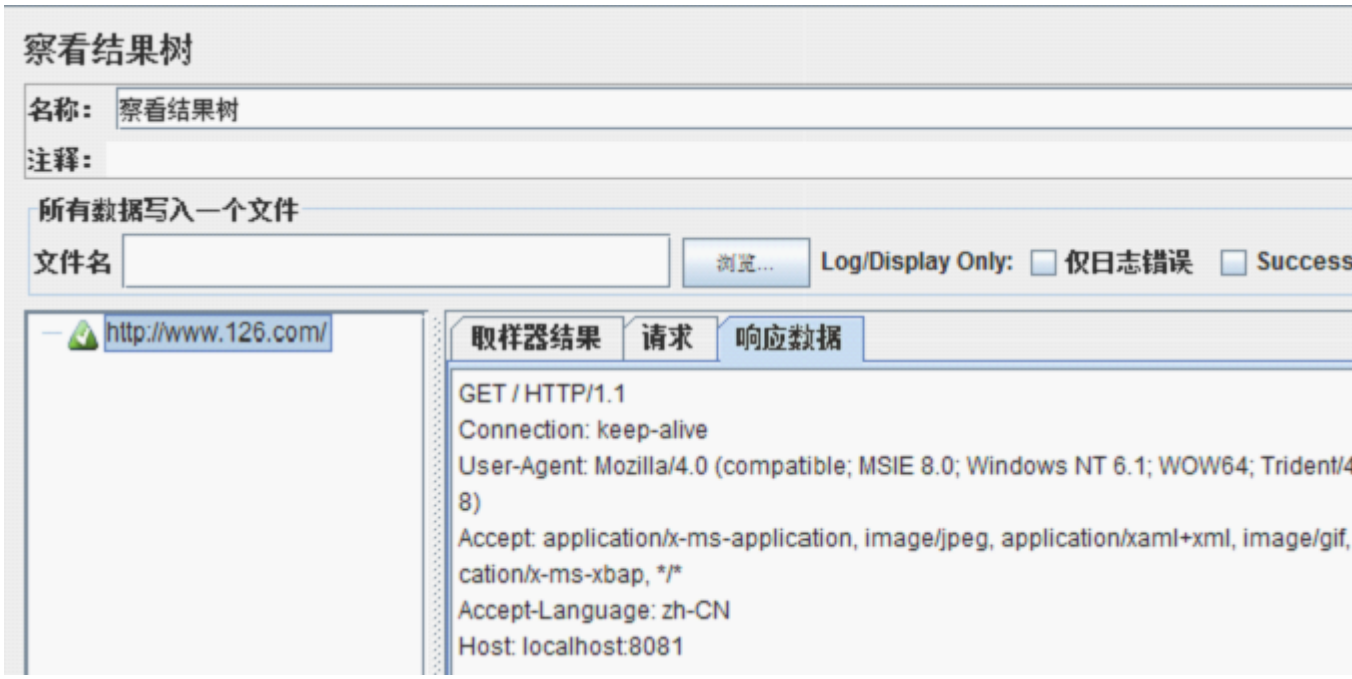


如果有必要的话需要修改端口号，点击“启动”按钮来启动 Server 。

接下来修改采样器，使其将 HTTP 请求 发送到 localhost:8081（也就是 Mirror Server 启动的位置）



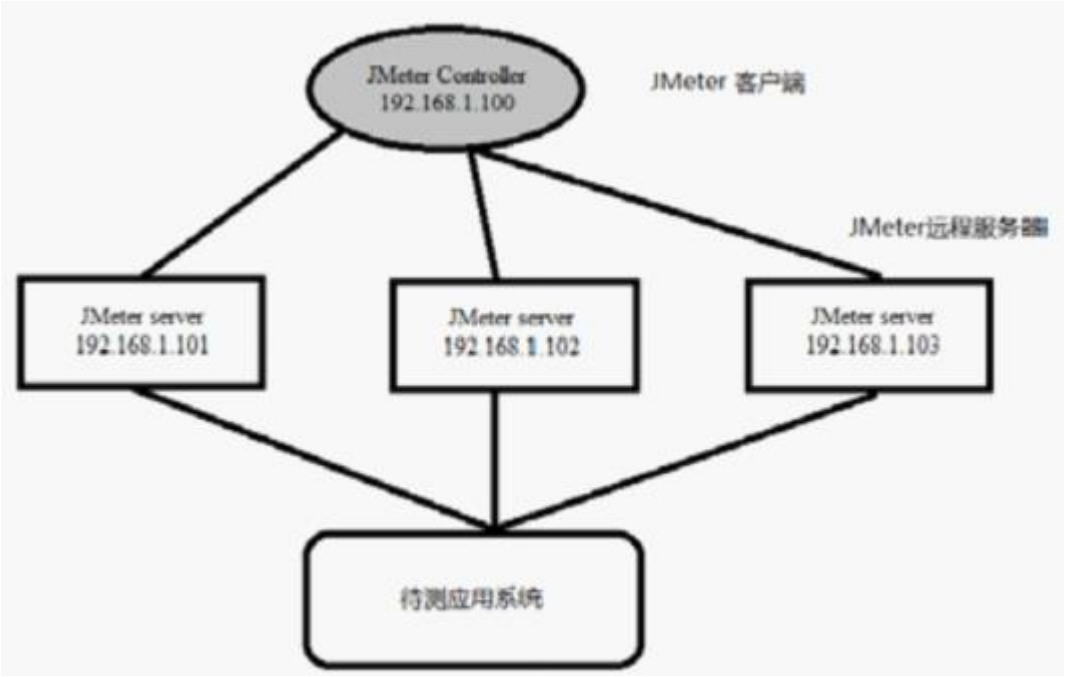
然后，运行测试计划，则可以从监听器“查看结果树”中看到响应数据。



JMeter 分布式测试

Jmeter 是 java 应用，对于 CPU 和内存的消耗比较大，因此，当需要模拟数以千计的并发用户时，使用单台机器模拟所有的并发用户就有些力不从心，甚至会引起 JAVA 内存溢出错误。为了让 jmeter 工具提供更大的负载能力，jmeter 短小精悍一有了使用多台机器同时产生负载的机制。

那么，是如何实现多台负载机同时运行的呢？当然不会多个人坐在多台负载机面前，一喊开始，大家同时启动 jmeter。这种方式很笨，也很难达到真正的同步。其实，我们通过单个 jmeter 客户端就可以控制多个远程的 jmeter 服务器，使它们同步的对服务器进行压力测试。



通过远程运行 jmeter，测试人员可以跨越多台低端计算机复制测试，这样就可以模拟一个比较大的服务器压力，一个 jmeter 客户端实例，理论上可以控制任意多的远程 jmeter 实例，并通过他们收集测试数据。这样一样，就有了如下特性：

* 保存测试采样数据到本地机器

* 通过单台机器管理多个 jmeter 执行引擎。

* 没有必要将测试计划复制到每一台机器，jmeter GUI 客户端会将它发往每一台 jmeter 服务器。

* 每一台 jmeter 远程服务器都执行相同的测试计划，jmeter 不会在执行期间做负载均衡，每一台服务器都会完整地运行测试计划。

在 1.4G Hz～3GHz 的 CPU 、1GB 内存的 JMeter 客户端上，可以处理线程 100～300。但是 Web Service 例外。XML 处理是 CPU 运算密集的，会迅速消耗掉所有的 CPU 。一般来说，以 XML 技术为核心的应用系统，其性能将是普通 Web 应用的 10%～25% 。另外，如果所有负载由一台机器产生，网卡和交换机端口都可能产生瓶颈，所以一个 JMeter 客户端线程数不应超过 10 0 。

采用 JMeter 远程模式并不会比独立运行相同数目的非 GUI 测试更耗费资源。但是，如果使用大量的 JMeter 远程服务器，可能会导致客户端过载，或者网络连接发生拥塞。

使用多台机器产生负载的操作步骤如下：

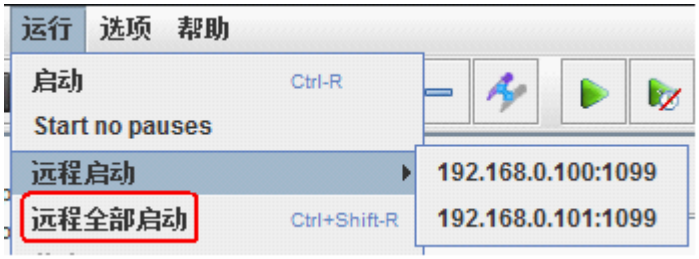
（1）在所有期望运行 jmeter 作为 负载生成器的机器上安装 jmeter， 并确定其中一台机器作为 controller ，其他的的机器作为 agent 。然后运行所有 agent 机器上的 jmeter-server 文件（假定使用两台机器 192.168.0.100 和 192.168.0.101 作为 agent）

（2）在 controller 机器的 jmeter 的 bin 目录下，找到 jmeter.properties 文件，编辑该文件：

查找：
remote_hosts=127.0.0.1
修改为：
remote_hosts=192.168.0.100:1099,192.168.0.101:1099

这里要特别注意端口后，有些资料说明端口 1644 为 jmeter 的 controller 和 agent 之间进行通信的默认 RMI 端口号，但是在测试时发现，设置为 1644 运行不成功，改成 1099 后运行通过。另外还要留意 agent 的机子是否开启了防火墙等。

（3）启动 controller 机子上的 jmeter 应用，选择菜单“运行”--->“远程启动”，来分别启动 agent ，也可以直接选择“远程全部启动”来将所有的 agent 启动。

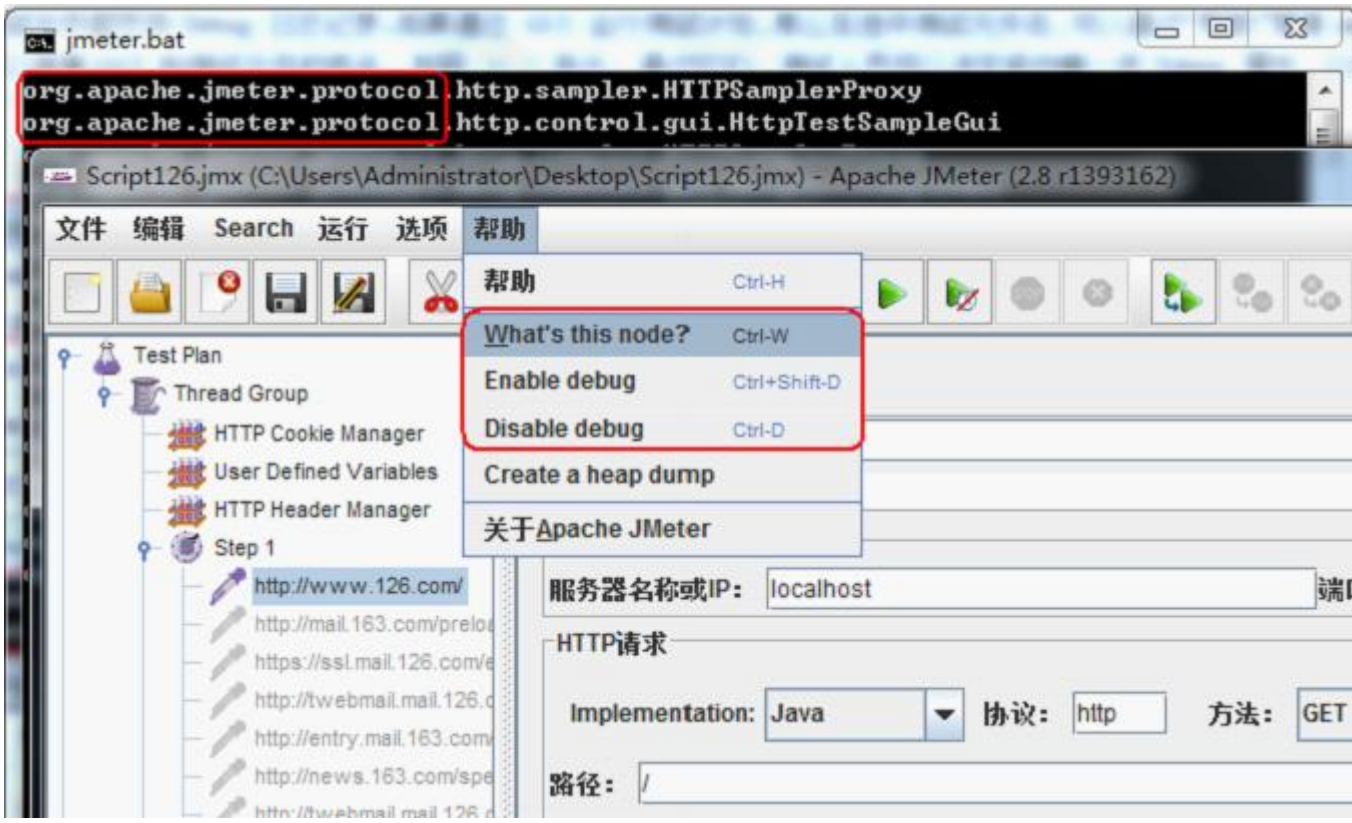


启动 Debug 日志记录

大多数测试元件都支持 Debug 日志记录。如果通过 GUI 运行测试计划，那么在选中测试元件后，可以通过“帮助”菜单 enable debug 或者 disable debug。在“帮助”菜单 中有一个选项“What’s this node? ”，

通过它可以查看 GUI 和测试元件的类名，如图 11 -7 所示。通过它们，测试人员可以决定修改哪一项 JMeter 属性，以便修改日志级别。

例如：我们可以点击一个 HTTP 请求，选择菜单栏“帮助”--->what's this node ？



在 jmeter 的 bin\目录下，找到 jmeter.properties 文件，关于日志级别的属性如下：

```
#Logging levels for the logging categories in JMeter. Correct values are FATAL_ERROR, ERROR, WARN, INFO, and DEBUG
# To set the log level for a package or individual class, use:
# log_level.[package_name].[classname]=[PRIORITY_LEVEL]
# But omit "org.apache" from the package name. The classname is optional. Further examples below.

log_level.jmeter=INFO
log_level.jmeter.junit=DEBUG
#log_level.jmeter.control=DEBUG
#log_level.jmeter.testbeans=DEBUG
#log_level.jmeter.engine=DEBUG
#log_level.jmeter.threads=DEBUG
#log_level.jmeter.gui=WARN
#log_level.jmeter.testelement=DEBUG
#log_level.jmeter.util=WARN
#log_level.jmeter.util.classfinder=WARN
#log_level.jmeter.test=DEBUG
#log_level.jmeter.protocol.http=DEBUG
# For CookieManager, AuthManager etc:
#log_level.jmeter.protocol.http.control=DEBUG
```

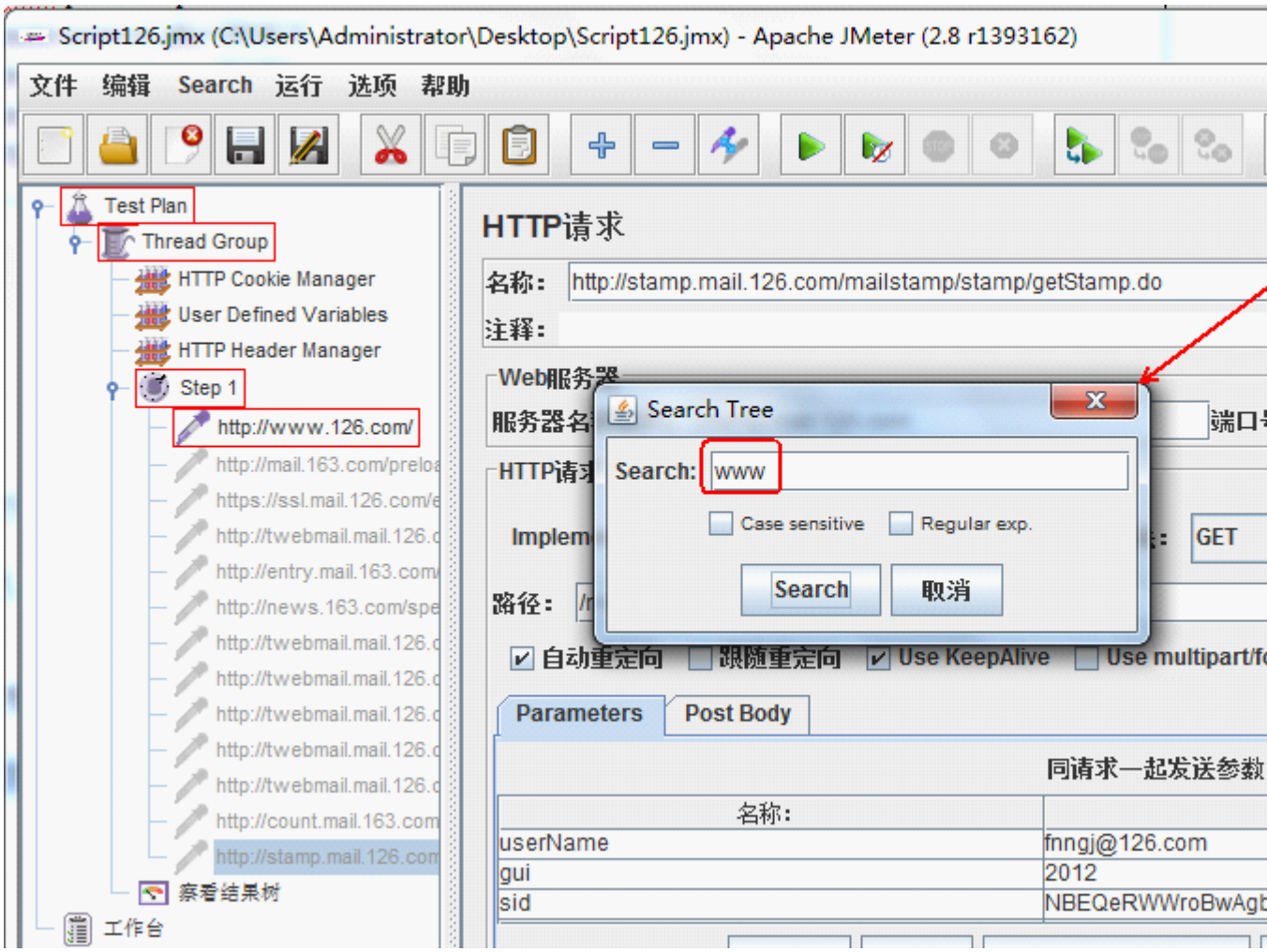
```
#log_level.jmeter.protocol.ftp=WARN
#log_level.jmeter.protocol.jdbc=DEBUG
#log_level.jmeter.protocol.java=WARN
#log_level.jmeter.testelements.property=DEBUG
log_level.jorphan=INFO
```

搜索功能

这是在使用一个变量或含有一定的 URL 或参数测试计划树和元素有时很难找到。现在一个新的特点是从 2.6 开始，你可以访问它的菜单搜索。它提供了搜索与下列选项：

* 使搜索区分大小写区分大小写：

* 正则表达式是一个正则表达式搜索文本，如果有的话将被搜索的正则表达式树的组件，例如“\ BTEST \ b”将匹配任何组件，包含测试组件的搜索元素



线程之间传递变量

JMeter 变量作用域局限于所属线程。这样设计是经过深思熟虑的，目的是让测试线程能够独立运转。有时候用户可能需要在不同线程间（可能属于同一个线程组，也可能不属于同一个线程组）传递变量。

其中一种方法就是使用属性。属性为所有 JMeter 线程所共享，因此当某个线程设置一个属性后，其他线程就可以读取更新后的值。

如果存在大量数据需要在线程间传递，那么可以考虑使用文件。例如，测试人员可以在一个线程中使用监听器，保存响应到文件（Save Responses to a file ）或者 BeanShell PostProcessor 。而在另外一个线程中使用 HTTP 采样器的“file: ”协议来读取文件，接着使用一个后置处理器或者 BeanShell 测试元件提取信息。

如果在测试启动前测试人员就能获得测试数据，那么最好将数据保存到文件中，使用 CSV Dataset 读取。

这最后一个技巧，操作较为麻烦，暂时不给详细的例子。算是提供个思路吧！知道有这当子事儿就行了。日后有机会再实践^_^

使用 JMeter 创建 FTP 测试计划

2011-07-01 16:59 by 虫师, 1932 阅读, 0 评论, 收藏, 编辑

FTP 服务主要提供上传和下载功能。有时间需要我们测试服务器上传和下载的性能。在这里我通过 JMeter 做一个 FTP 测试计划的例子。

当然，JMeter 官方网站的用户手册也有例子，但由于版本较早，我也算是对自己学习的一个总结，所以再整理一个。

* 本人使用的是 JMeter2.4 版本。

* 测试的服务器是 IP：124.205.228.54 (由于找不到 FTP 站点，所以在“主机屋网站 http://www.zhujiwu.com 申请了一个免费的 FTP 空间”)

1.创建一个线程组

2.线程组--->添加--->配置元件--->FTP 请求缺省值。

3.线程组--->添加--->Sampler--->FTP 请求

说明：

IP 为你 FTP 服务的 IP

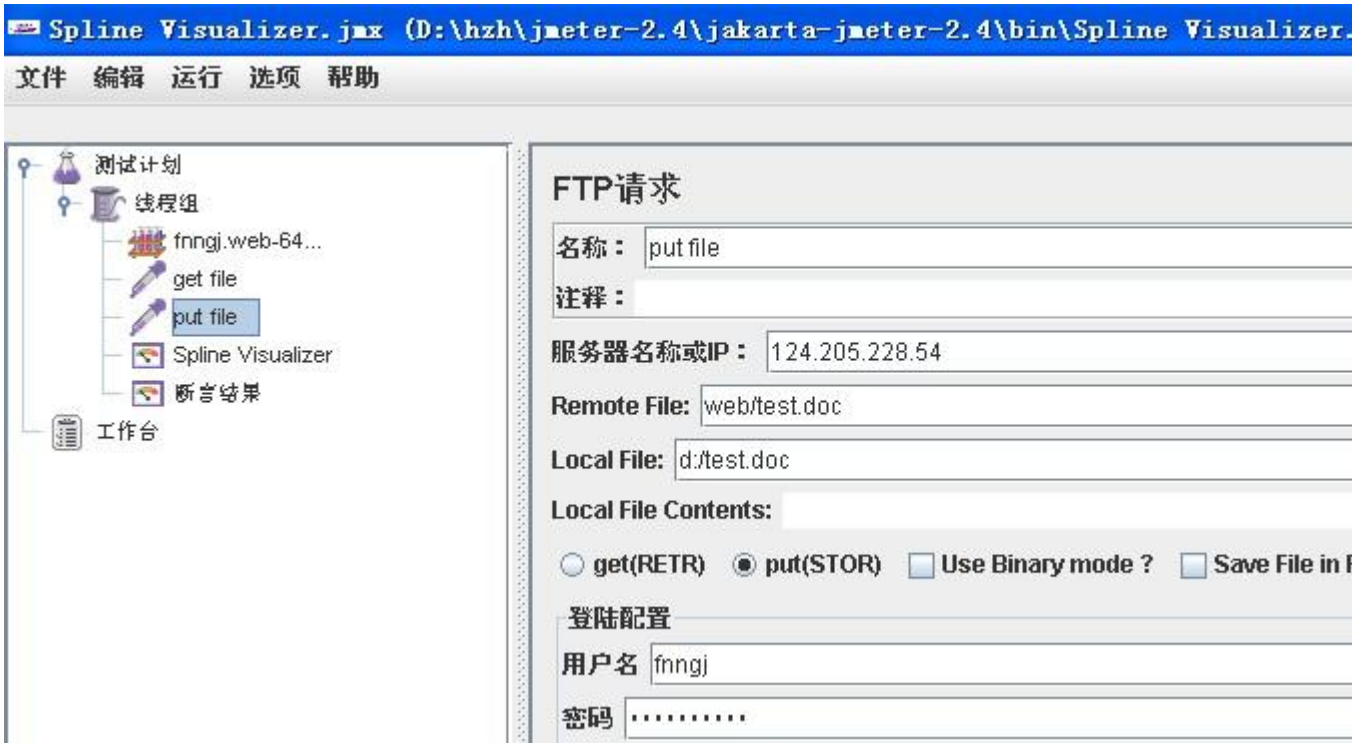
Remote file 为你 FTP 服务器上的一个文件。

local file 为本地你存放到本机上的路径。

选择 get(RETR) 为下载方式。

填写你的 FTP 服务器的用户名密码。

3.按照第二步的方式再添加一个“FTP 请求”。



说明：

IP 为你 FTP 服务的 IP

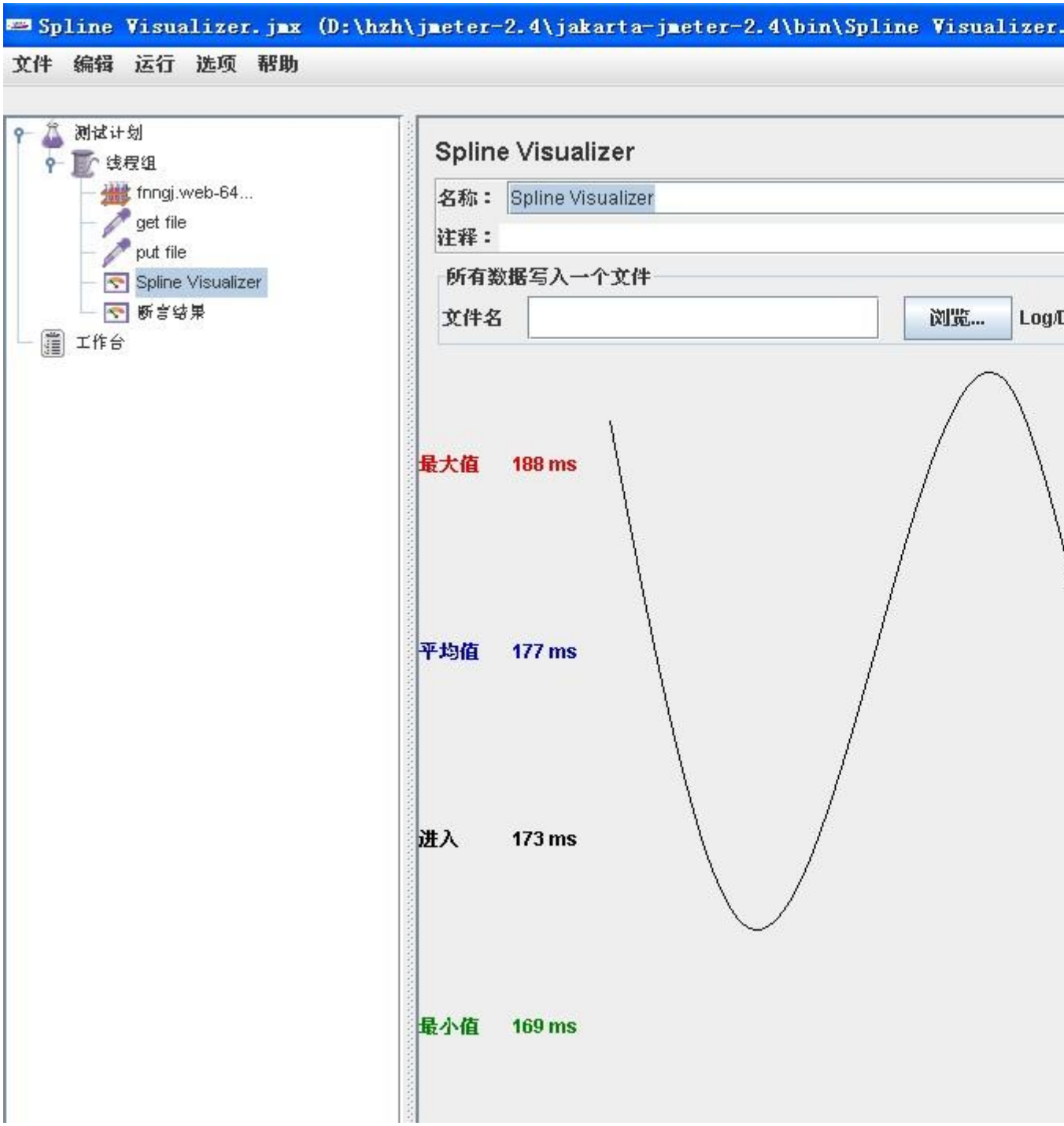
Remote file 为你要上传到 FTP 服务器上的文件路径及文件名

local file 为本地你要上传的文件的路径。

选择 put(RETR) 为上传方式。

填写你的 FTP 服务器的用户名密码。

4.添加一个监控器：线程组--->添加--->监控器--->Spline Visualizer



一个 FTP 计划创建成功 :)

使用 JMeter 创建数据库（Mysql）测试

2011-07-26 19:54 by 虫师, 6991 阅读, 2 评论, 收藏, 编辑

好吧！我一直觉得我不聪明，所以，我用最详细，最明了的方式来书写这个文章。我相信，我能明白的，你们一定能明白。

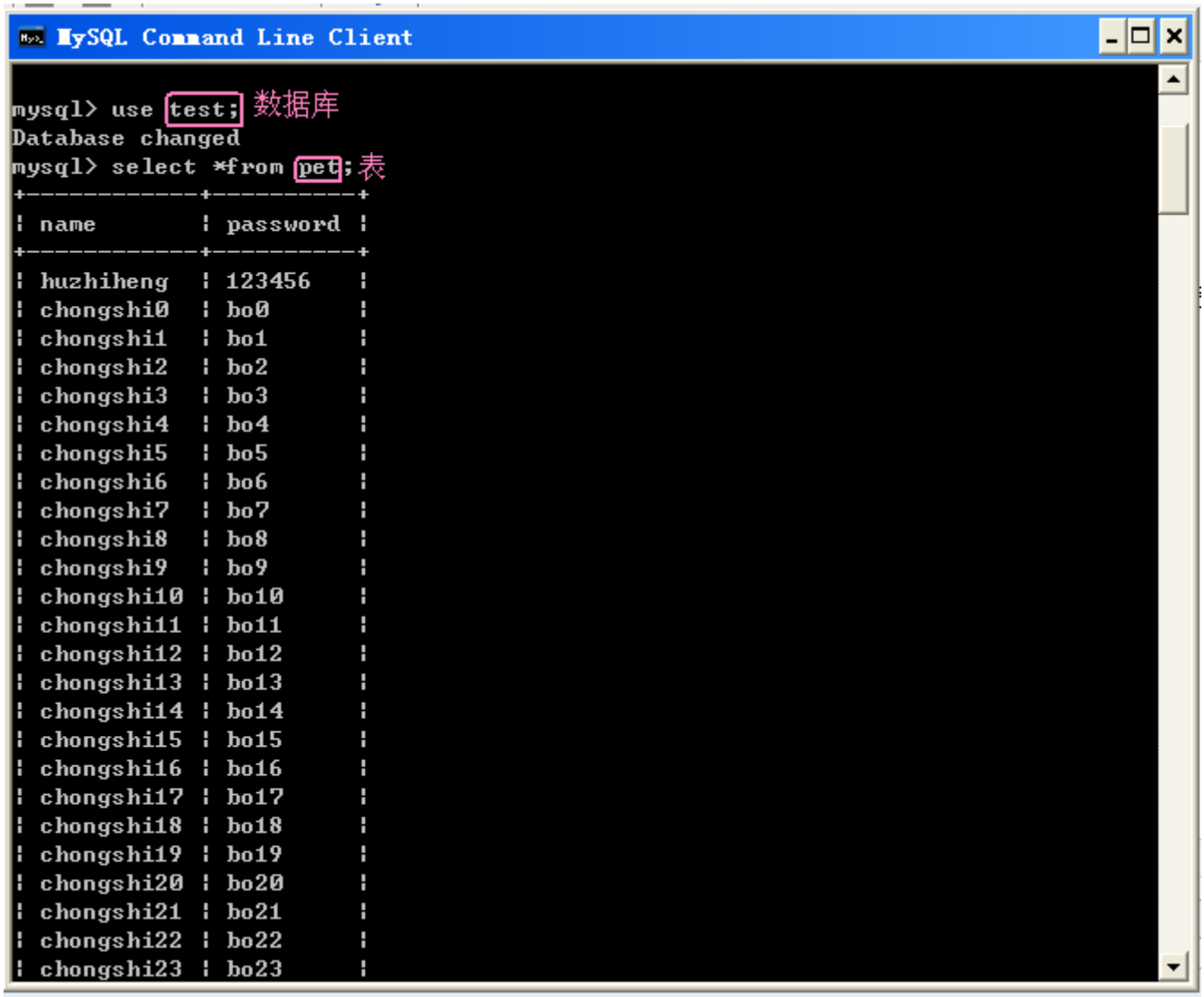
我的环境：MySQL：mysql-essential-5.1.51-win32

jdbc 驱动：我已经上传到 csdn 上一个：http://download.csdn.net/source/3451945

JMeter：jmeter-2.4 任意版本都行。

1.首先我们要有一个可以做测试的数据库，当然，里面要有数据，不然怎么测呢？我的上一篇文章《**Eclipse 连接 MySQL 数据库（傻瓜篇）**》，里面教你如何通过程序批量插入数据。

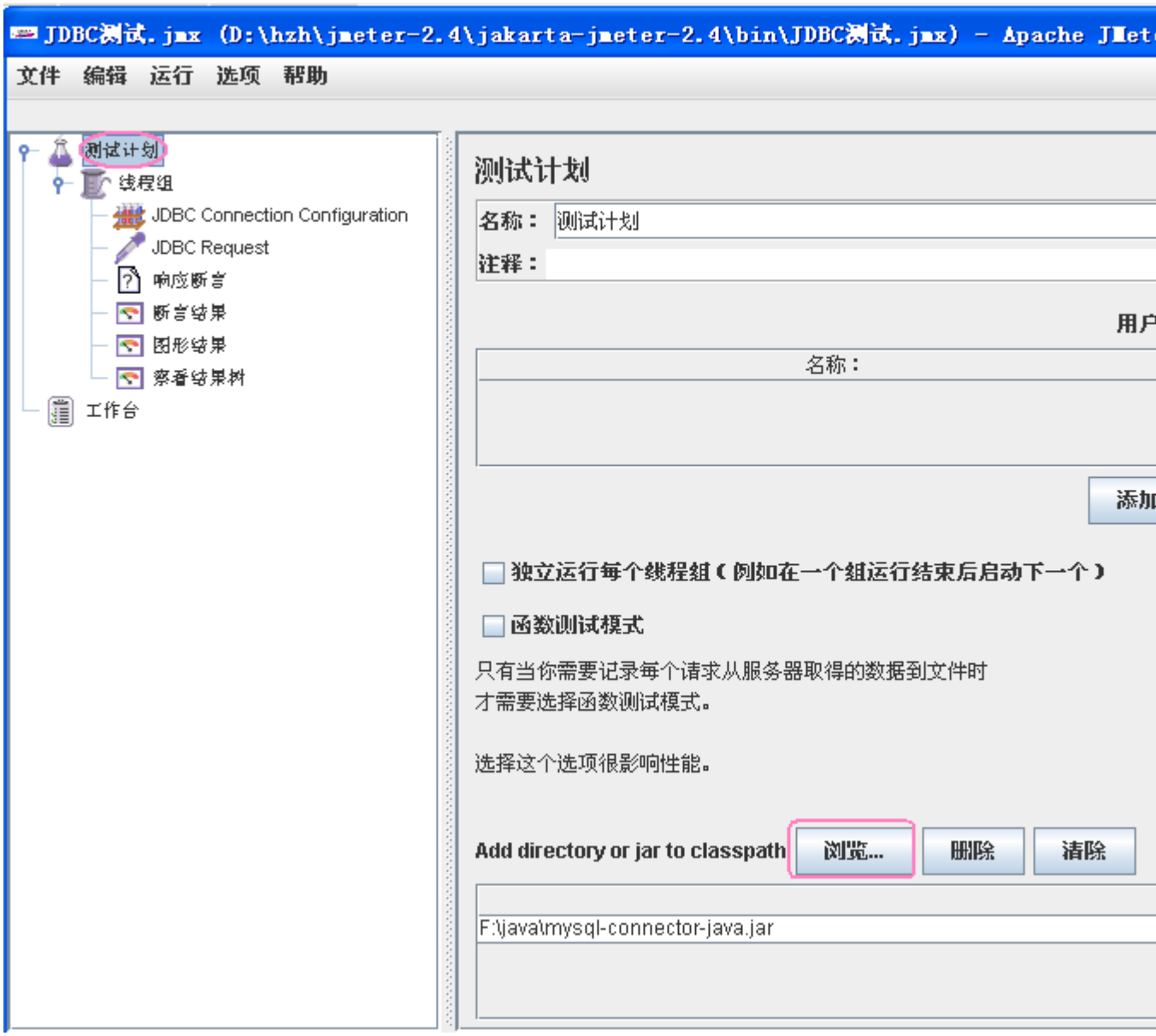
来看一下我的数据：



我可能有点啰嗦 :） 不管你用什么方式，你的数据库里要有点数据。

2.

打开 JMeter,点击测试计划，



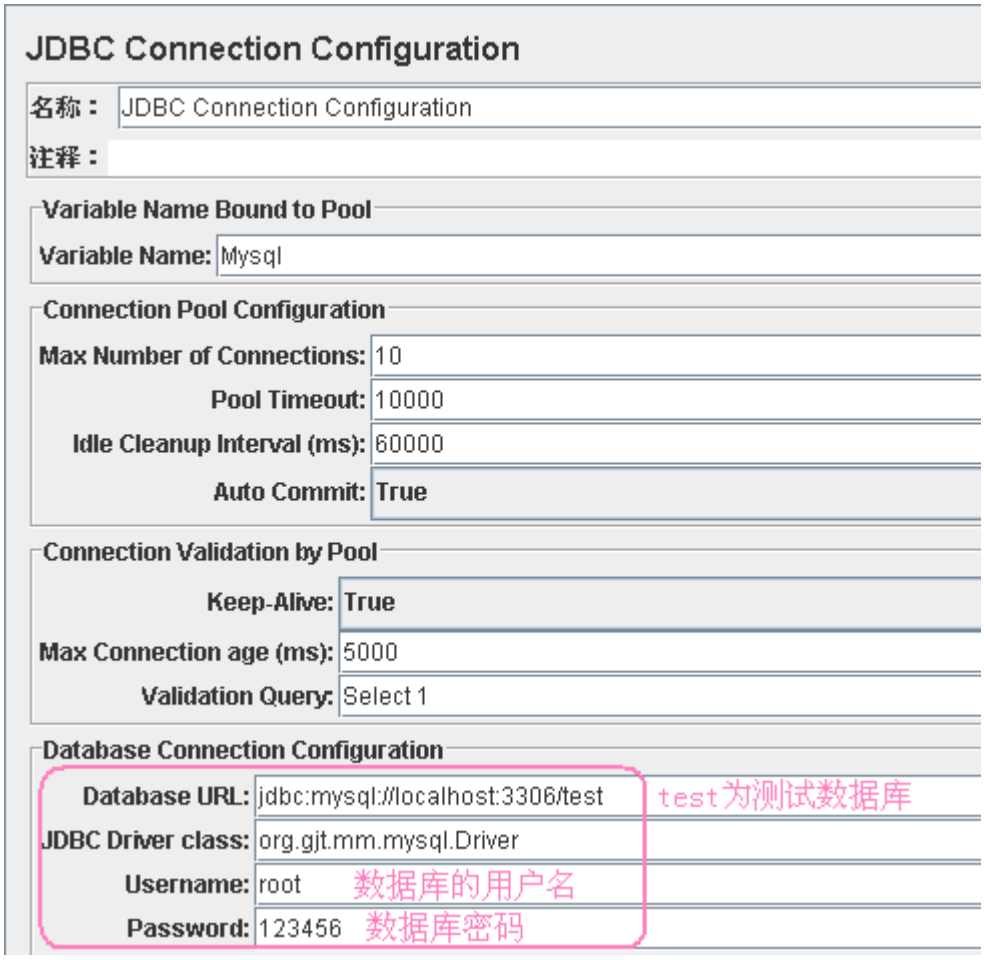
点击“浏览...”按钮，将你的 JDBC 驱动添加进来。

3.

添加一个线程组，

右键点击“线程组”，在下面添加一个“JDBC Connection Configuration”

来配置一下 JDBC Connection Configuration 页面。



4.

右键点击“线程组”，在下面添加一个“JDBC request”

JDBC Request

名称： JDBC Request

注释：

Variable Name Bound to Pool

Variable Name: Mysql

SQL Query

Query Type: Select Statement 要对数据库做查询操作

Query:

```
select *from pet where name like '%chongshi%'
```

select *from pet where name like '%chongshi%';
查询pet表中，name字段包含chongshi 的内容

Parameter values:

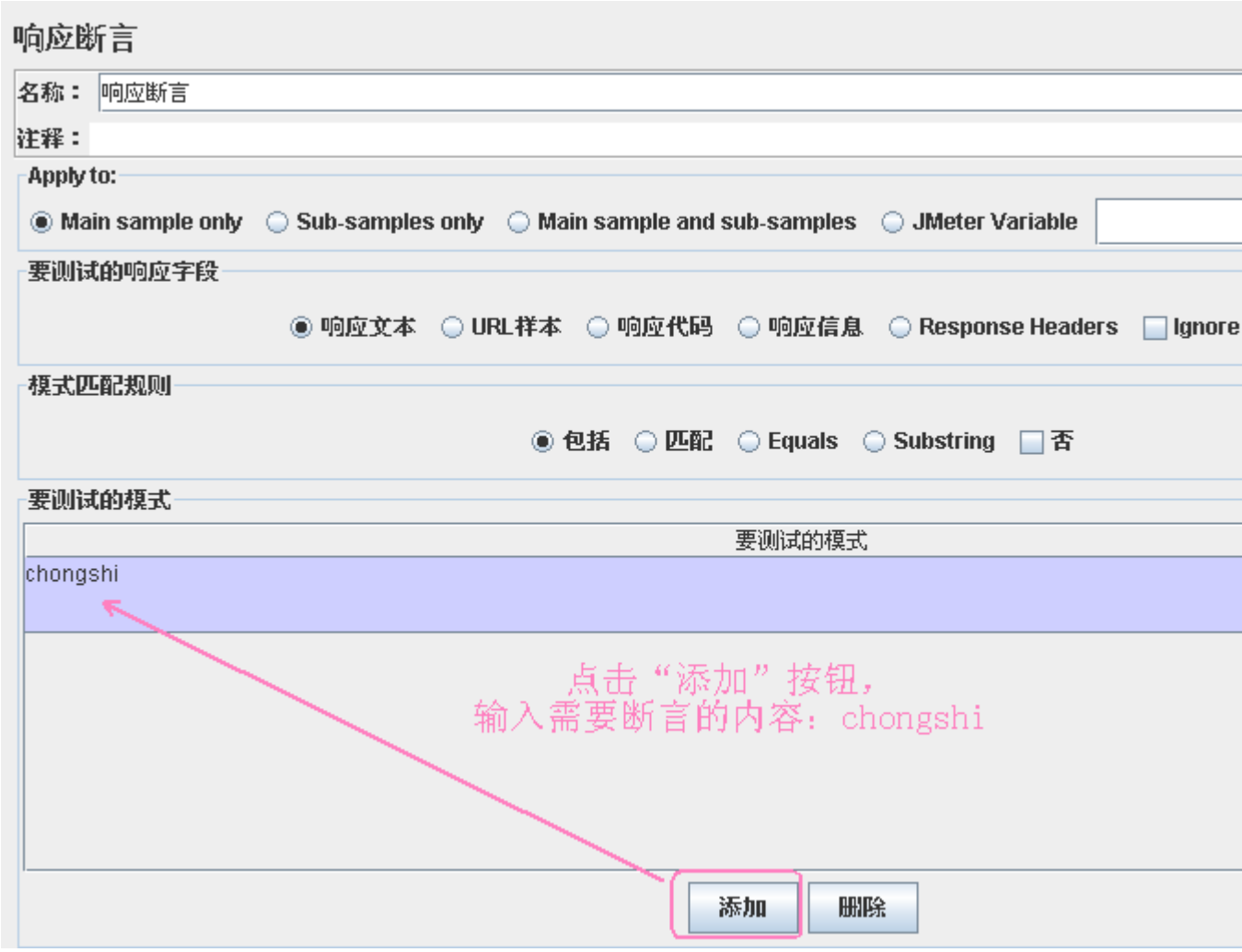
Parameter types:

Variable names:

5.

添加断言。

右键点击线程组---->添加--->断言---->响应断言。



6.

我们来添加一些监听器来行查看

添加一个断言结果：

右键点击线程组---->添加--->监听器---->结果断言。

添加一个图形结果：

右键点击线程组---->添加--->监听器---->图形结果。

添加一个查看结果树：

右键点击线程组---->添加--->监听器---->查看结果树。

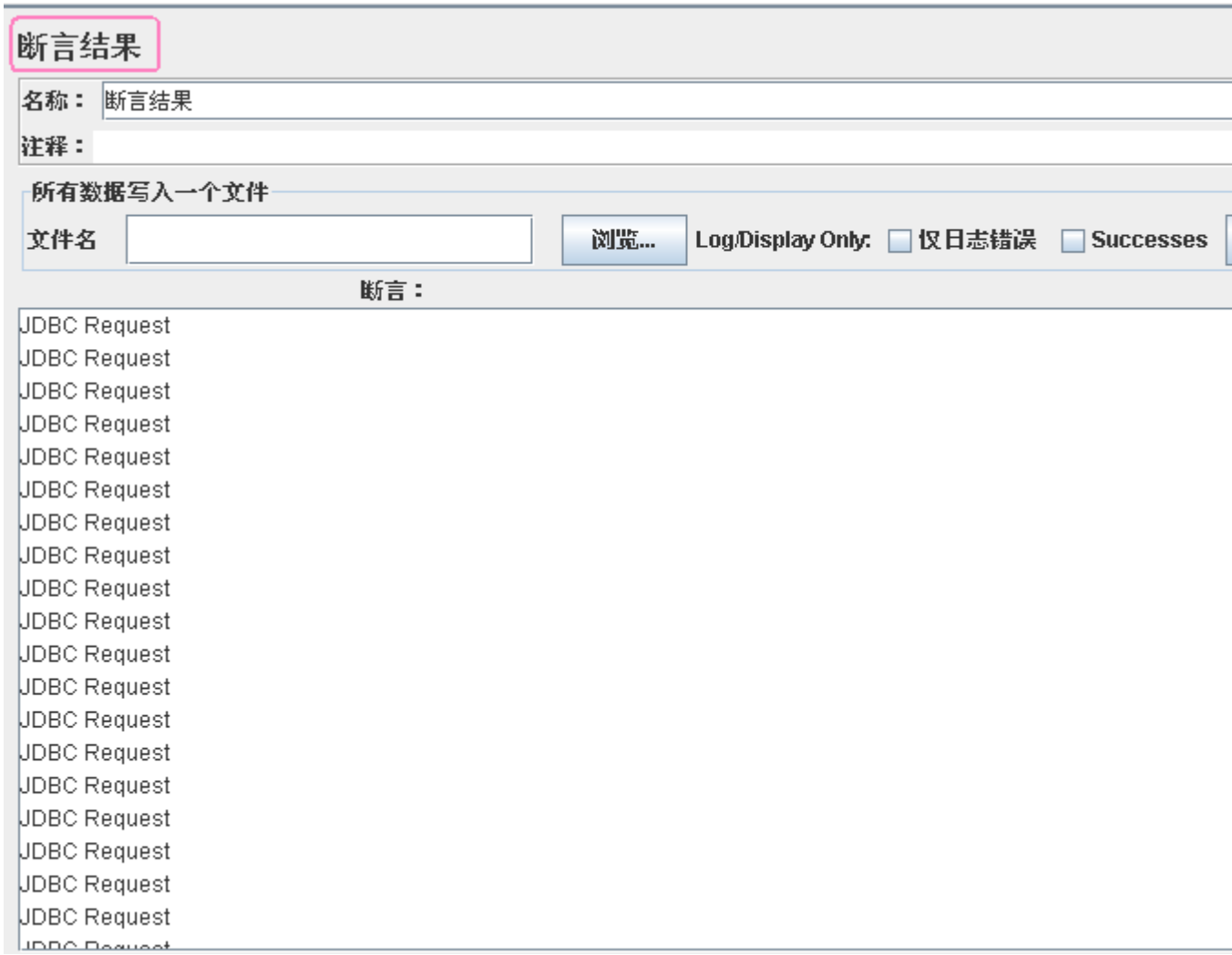
下面是添加所有东东的列表：

7.

在线程组页面设置用户数、启动时间、循环次数

点击菜单栏“运行”----“启动”

下面是结果：



其它的结果图片就不贴了。嘻嘻！！用图片说话，看上去简单，但比较长。。

火龙果 · 整理
uml.org.cn

LR 有的 JMeter 也有之一"参数化"

2011-07-28 19:44 by 虫师, 3393 阅读, 1 评论, 收藏, 编辑

酝酿了几天，一直想写点 JMeter 的东西，算是对学习东西的一个整理。:)

恩，一直觉得自己领悟能力不强，别人写的东西总要看老半天也不懂。好吧！一惯的傻瓜的方式（大量的截图+参数说明）嘻嘻。

参数化：简单的来理解一下，我们录制了一个脚本，这个脚本中有登录操作，需要输入用户名和密码，假如系统不允许相同的用户名和密码同时登录，或者想更好的模拟多个用户来登录系统。

这个时候就需要对用户名和密码进行参数化，使每个虚拟用户都使用不同的用户名和密码进行访问。

前提：

假如，我们录制好了一个脚本（可以用 badboy 工具录制），在 jmeter 中打开，找到有用户名和密码的页面。如下：

HTTP请求

名称：登录　这里，我将网址路径改成了“登录”两个字

注释：

Web服务器

服务器名称或IP：localhost　端口号：8080

Timeout

Connect

HTTP请求

协议：http　方法：POST　Content encoding:

路径：/admin/j_security_check

☑ 自动重定向　☑ 跟随重定向　☑ Use KeepAlive　☐ Use multipart/form-data for HTTP POST

同请求一起发送参数：

名称：	值
j_username	admin
j_password	admin

这里就是用户名和密码

添加　删除

同请求一起发送文件：

文件名称：

添加　浏览...　删除

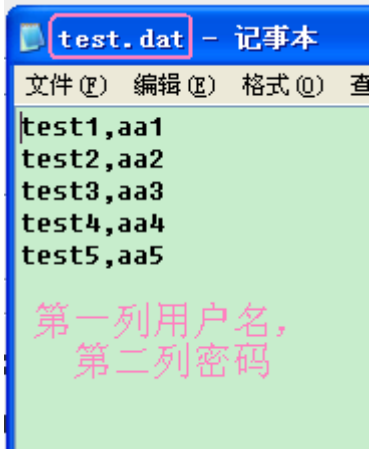
其他任务

☐ 从HTML文件获取所有内含的资源　☐ 用作监视器　☐ Save response as MD5 hash?

Embedded URLs must match:

1.

我们需要“参数化”的数据，这里我用记事本写了五个用户名和密码，保存为.dat 格式的文件。

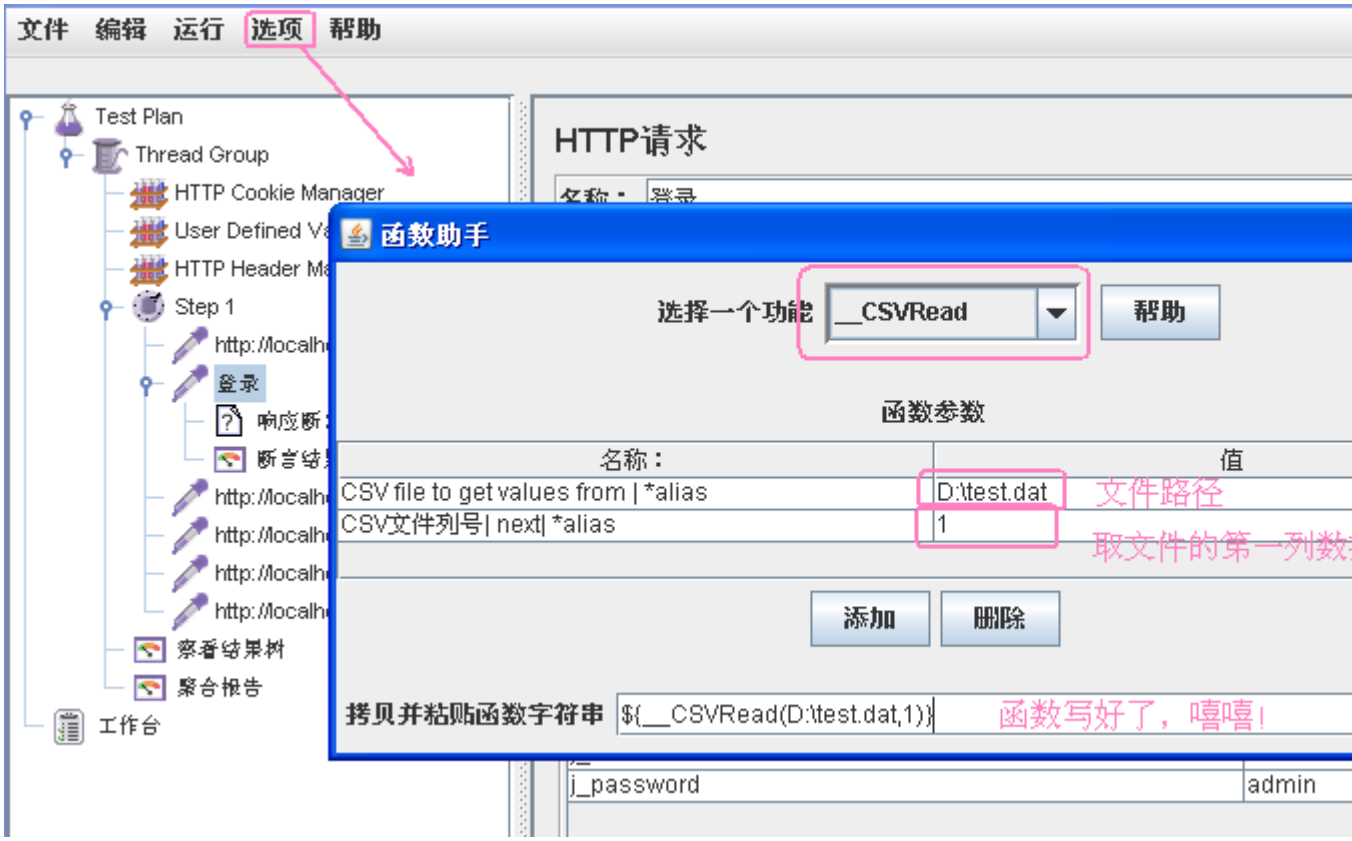


我将这个文件放在了我的（ D:\test.dat ）路径下。关于如何得到成百上千的用户名和密码，首先要在数据库中创建这些数据，将数据导出，整理保存，这里就不深究。

2.

好，我们要编写函数来调用这个 test.dat 文件，怎么弄，

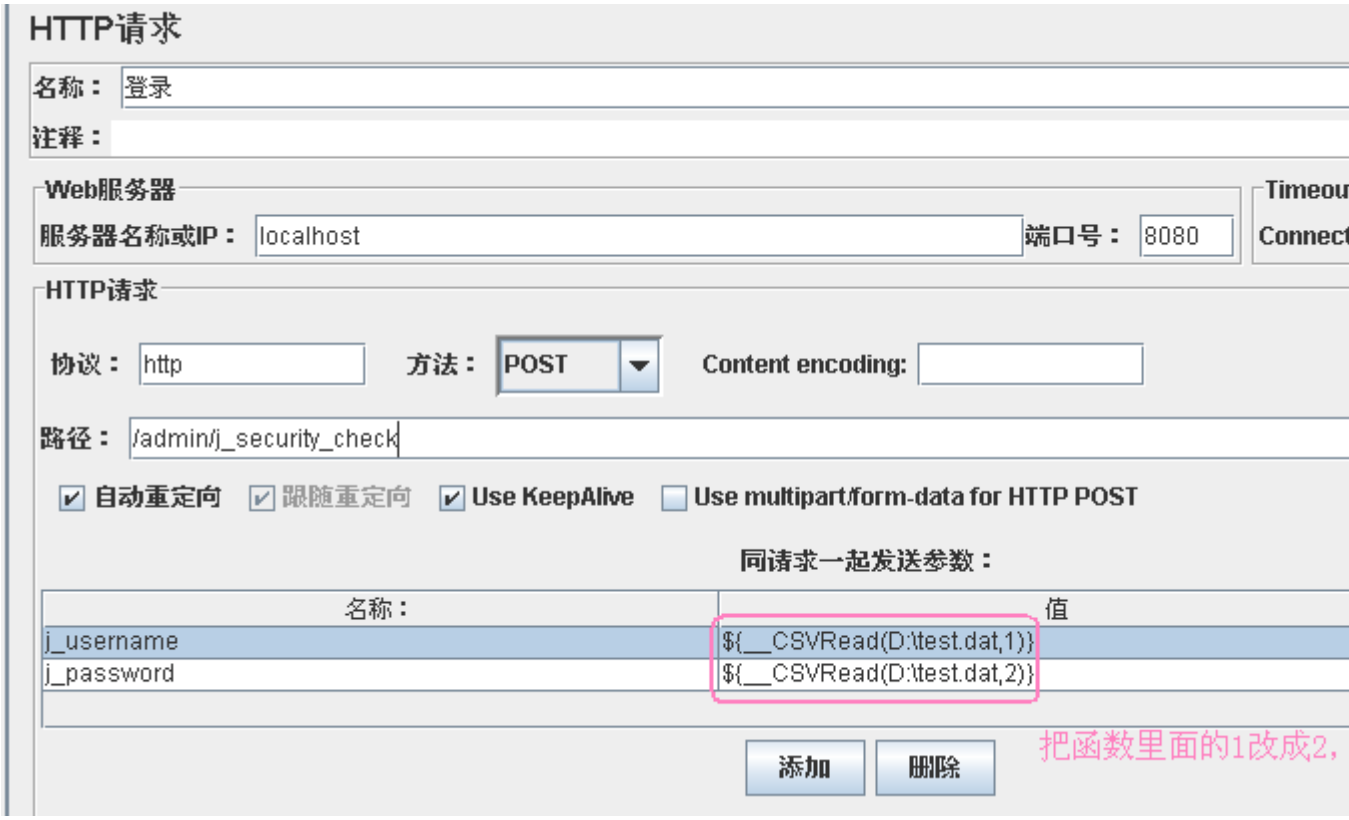
点击菜单栏“选项”---->函数助手对话框，看下图。



更正：.dat 文件第一列从 0 开始取得。

3.

把我们写好的函数复制到“登录”页面用户名和密码胡位置。



好了，现在我们的参数化设置完成，在脚本的时候，会调用我们 D 盘下面的 test.dat 文件，第一列是用户，第二列是密码。

注意用户名和密码是一一对应的，中间用户逗号（，）隔开。

LR 有的 JMeter 也有之二"检查点"

2011-07-28 20:29 by 虫师, 2434 阅读, 1 评论, 收藏, 编辑

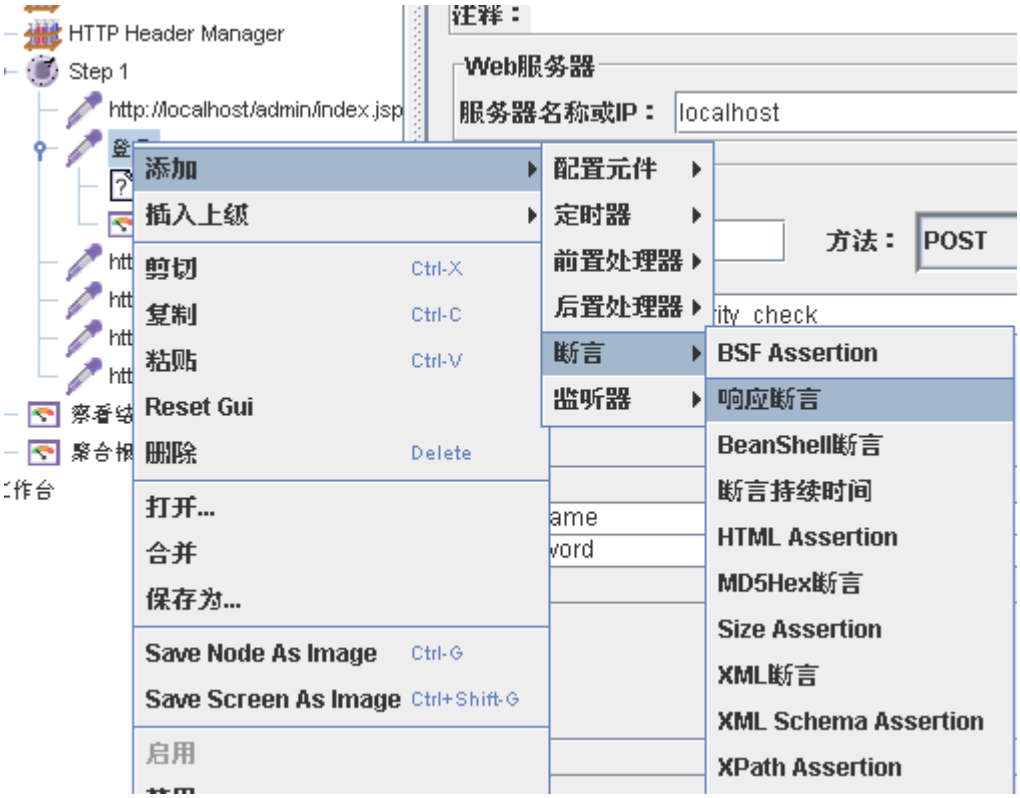
好吧！接着上一篇文章的内容和思路，继续前进。

检查点：简单的来理解一下，上一章讲到，我们对用户名和密码进行了参数化，那么怎样来判断 jmeter 有没有正确调用 test.dat 里面的文件呢。当然，我们可以从结果图表中查看。但我还是想在"登录"这个地方进行一下检查。

jmeter 里面的检查点通过添加断言来完成。

1.

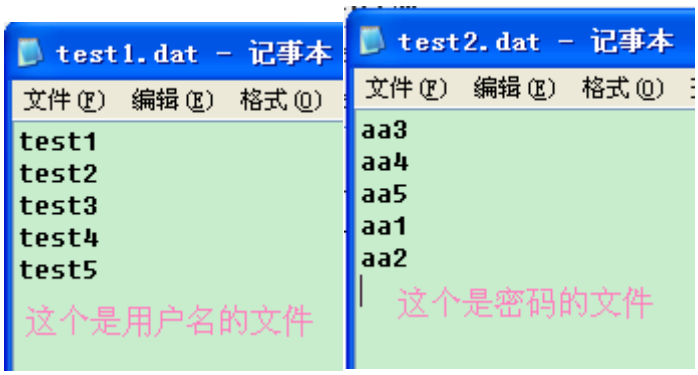
右键点击我们的"登录"页面---->添加---->断言---->响应断言



来设置一下断言页面，我们想断言什么呢？调用的用户名和密码是不是我们 test.dat 文件里的用户名和密码，怎样来进行呢。

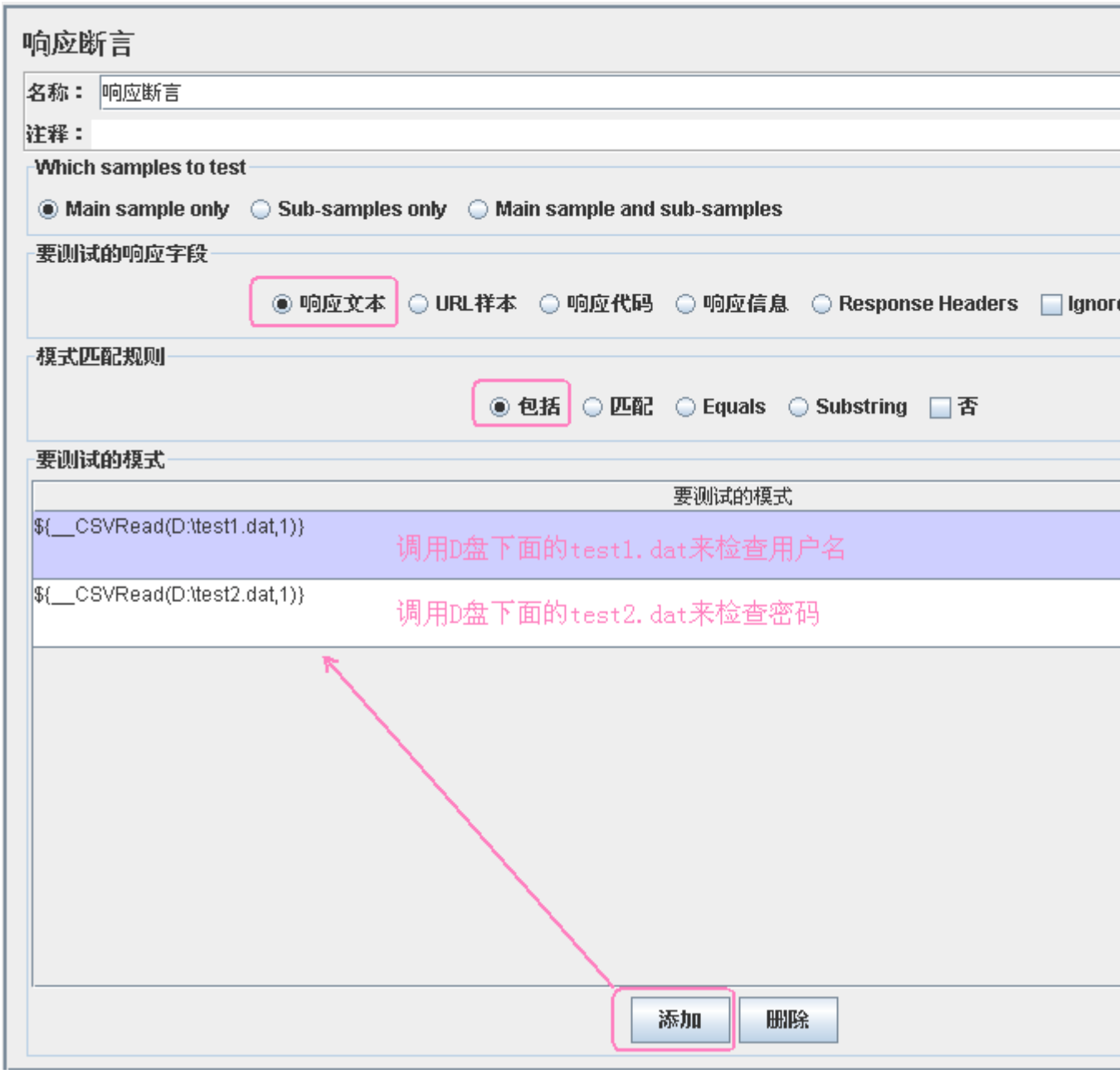
3.

这里我们需要分别准备两个文件，一个放用户名，一个放密码。



下面编写函数来调用这两个文件，如何编写先看我上一篇文章。

在响应断言页面添加两个“模式”



（第三步还有一种方法，直接调用原来 test.dat 的数据进行检查，我试了半天一直失败，不知道原因在哪儿？）

4.

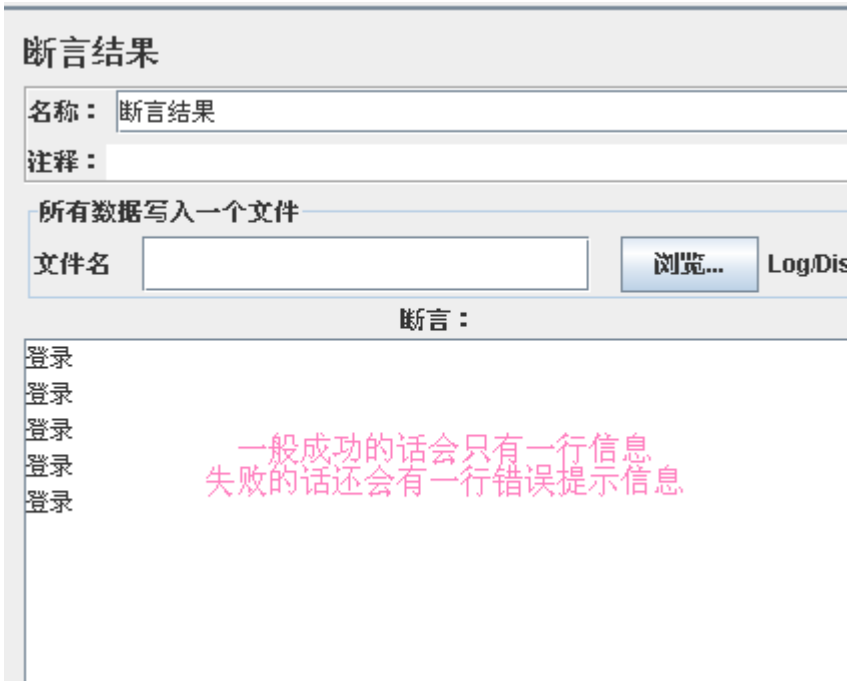
添加一个断言结果

右键点击我们的“登录”页面---->添加---->监听器---->断言结果

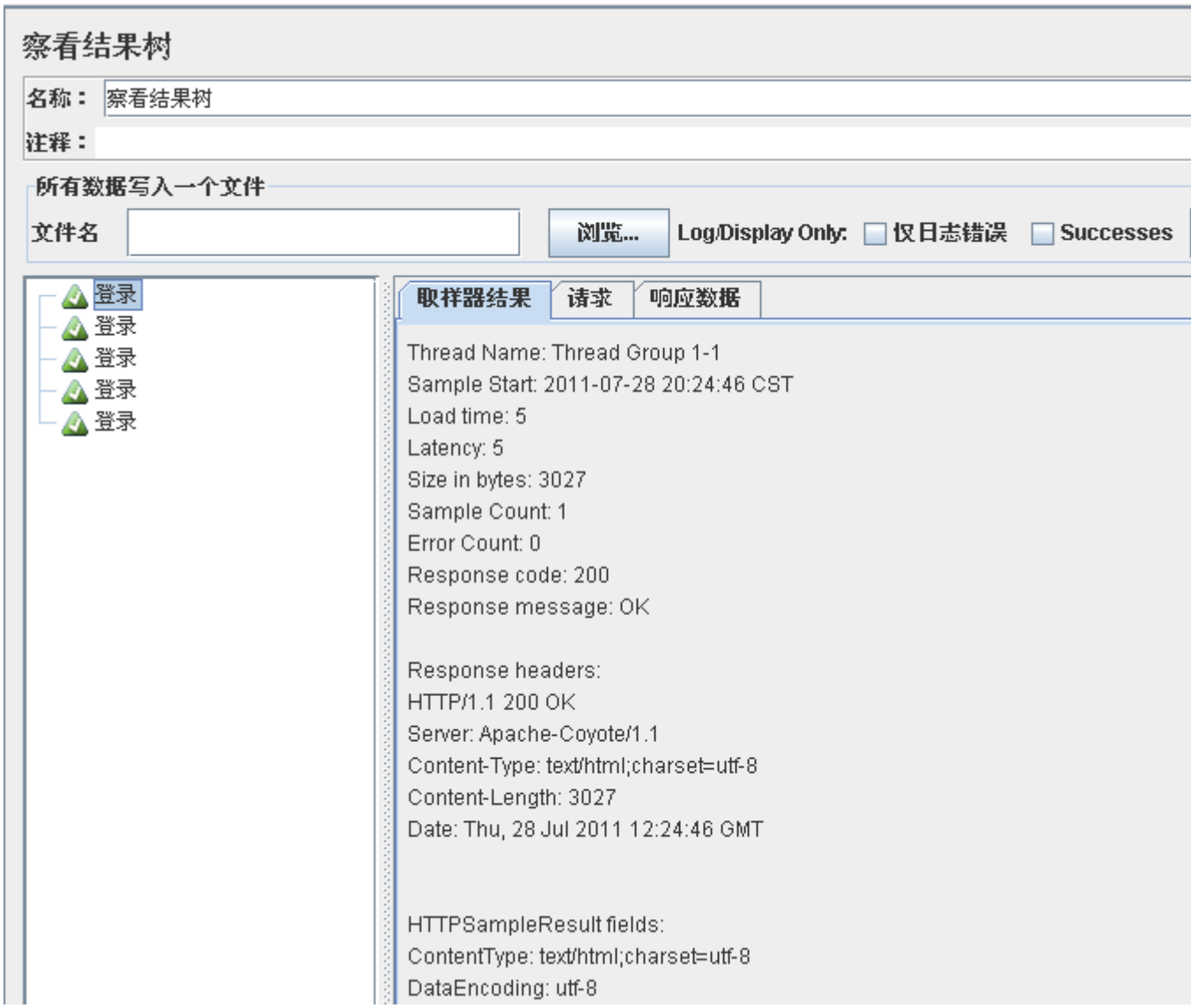
还可以再添加一个“查看结果树” 右键点击我们的“登录”页面---->添加---->监听器---->查看结果树

在线程组中设置 5 个用户，点击运行运行。

查看断言结果：



查看结果做：



好了，检查点添加成功！

LR 有的 JMeter 也有之三"集合点"

2011-07-28 20:56 by 虫师, 2109 阅读, 3 评论, 收藏, 编辑

继续上两篇的文章内容和思路进行。（文思如尿崩，谁与我争锋----韩寒）哈哈！

集合点：简单来理解一下，虽然我们的"性能测试"理解为"多用户并发测试"，但真正的并发是不存在的，为了更真实的实现并发这感念，我们可以在需要压力的地方设置集合点，

还拿那个用户和密码的地方，每到输入用户名和密码登录的地方，所有的虚拟用户都相互之间等一等，然后，一起访问。（红军排长说：等一等！大家一起冲啊！这样给敌人的压力是很大的。嘻嘻！）

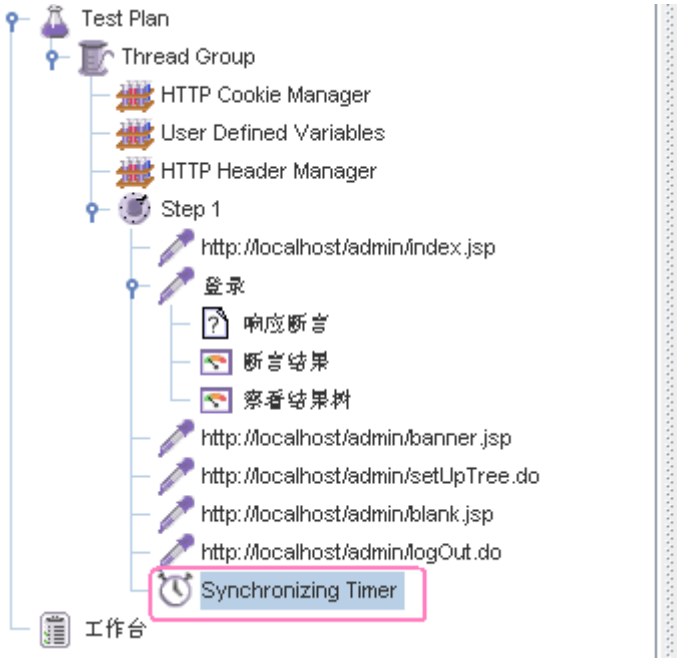
1.

接着之前创建的脚本，右键点击 step1---->定时器---->Synchronizing Timer

这样子就添加了一个“集合点”，下面来设置一下集合点。

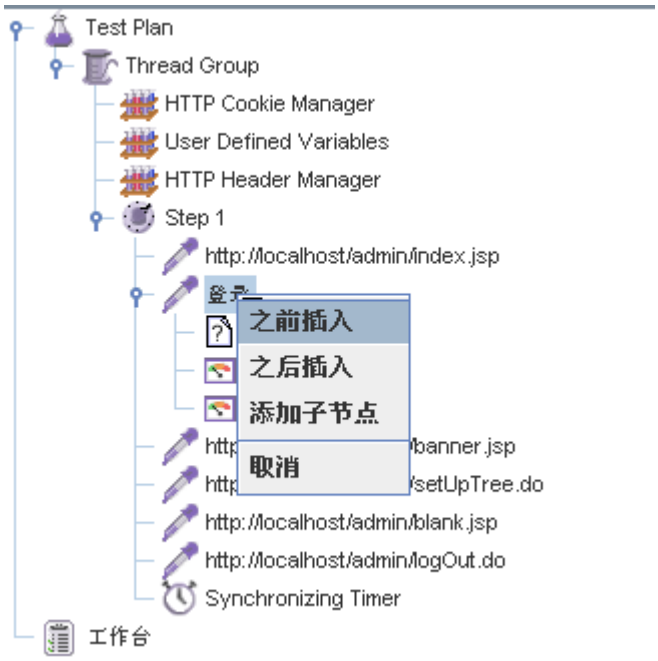
2.

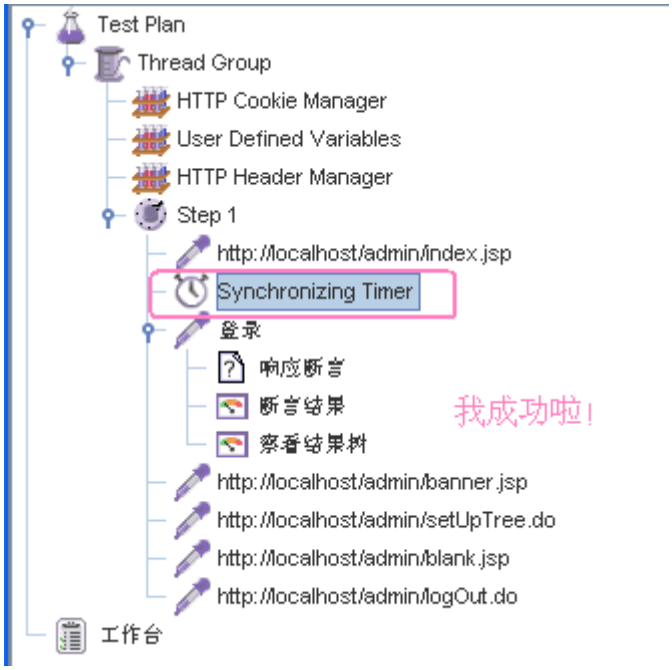
我们添加完之后的列表是这个样子的，



发现了没，我们集合点的位置不对，应该在登录的前面才对。怎么弄呢？

点击“synchronizing Timer”拖动到“登录”面前的位置，不要松鼠标的左键，再点击鼠标的右键，选择在“之前插入”---有点难噢，慢慢来！





OK!!

一种简单的数据库性能测试方法

2011-08-03 23:39 by 虫师, 9862 阅读, 7 评论, 收藏, 编辑

这两天接到一个任务，要测试一个服务器的性能，客户要求向数据库内 1000/s（每插入一千条数据） 的处理能力，当时脑子赌赛了，想的是用 LR 来进行，由于 LR 接触不深，只知道 LR 实现参数化的时候可以调用数据库里面的数据，往里面大批量的插入数据，以前没试过。

翻阅了一下资料，找一到了一篇《一种特殊的数据库性能测试》，大概思路是：通过编写一 java 程序来循环插入数据，编写一个批处理文件来调用 java 程序。再通过 LR 的 system()函数来调用批处理文件来进行压力测试。但是对于我这种菜鸟来说，好多细节不懂。比如那个批处理就让我很为难。呵呵。

其实，通过 jmeter 很简单就可以完成，可以参考我以前的一篇文章《jmeter 创建数据库（MySql）测试》。

前提条件：一个数据库：test 数据库下面有一张表：user 表中有两个字段：username、passworld 。

要求：往数据库内大批量插入数据，1000/s

其实和之前的方法一样，为了简单，我还是把截图贴出来吧。

1.

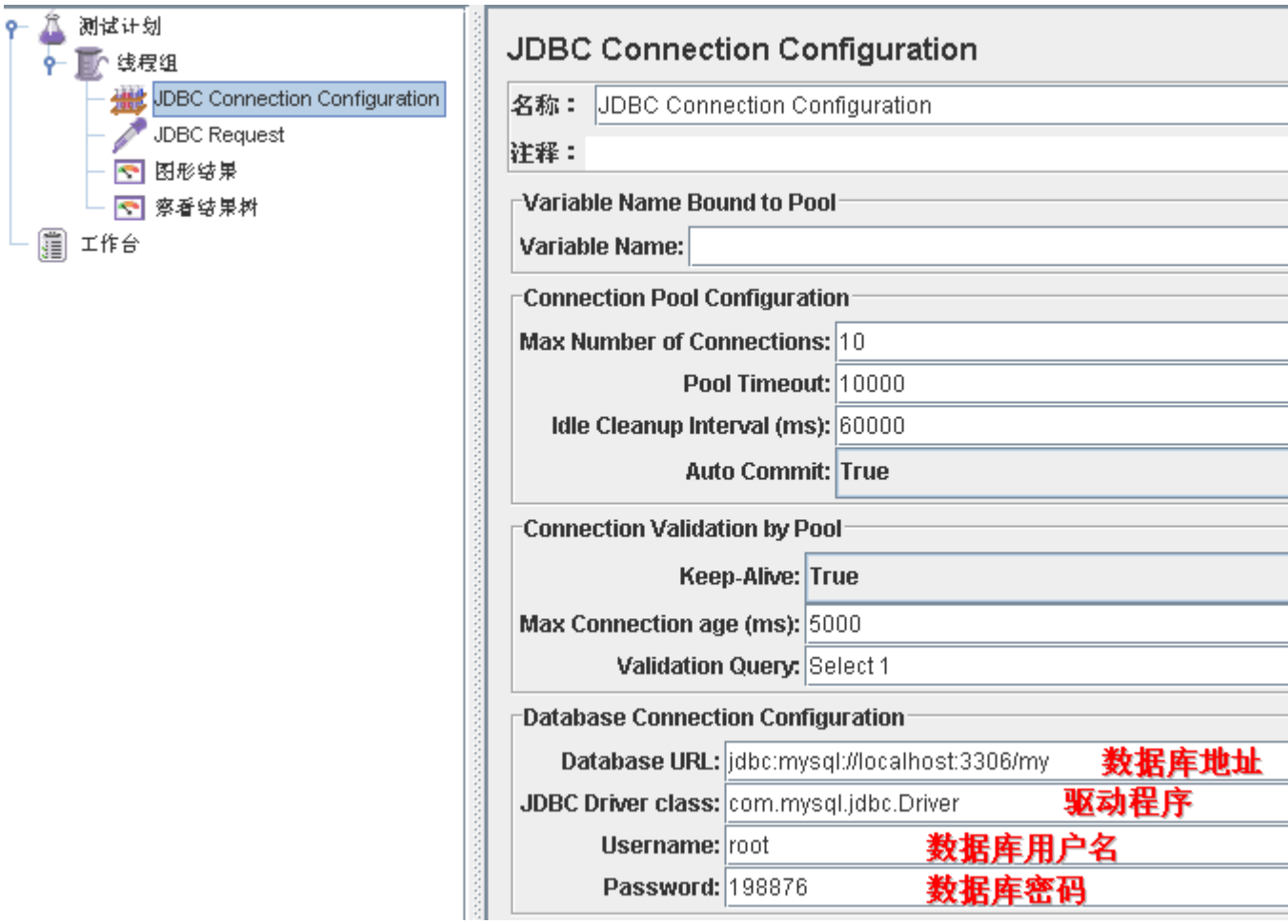
创建一个测试计划，将我们所使用的数据库驱动包导入。

2.

添加一个线程组，并设置我们的虚拟用户数、启动时间、和循环次数

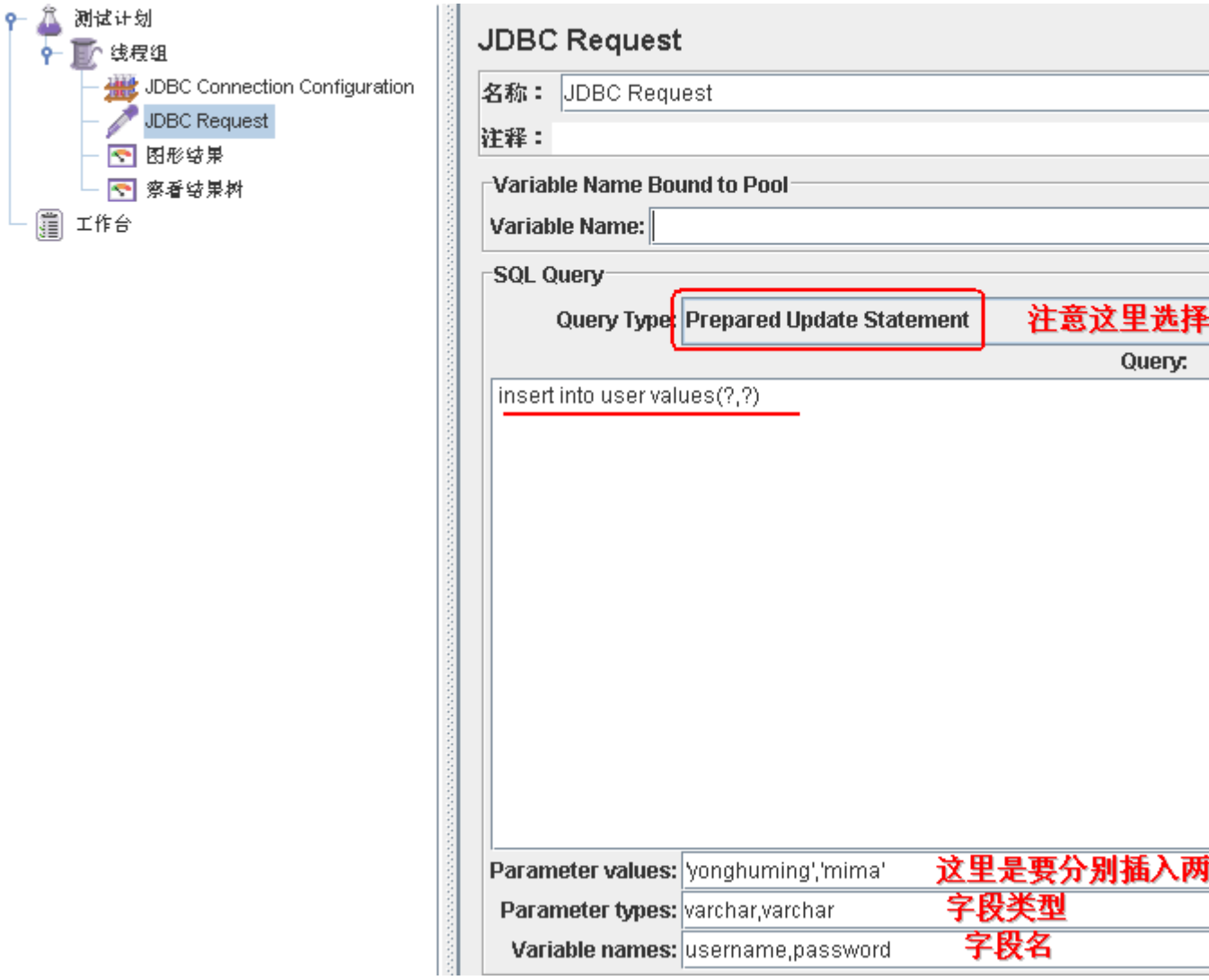
3.

创建一个线程，并在线程下面，创建一个 JDBC Connection Configuration ,设置相关信息。



4.

创建一个 JDBC Request.我们需要对数据库做插入操作。（详细设置，看截图上的说明）

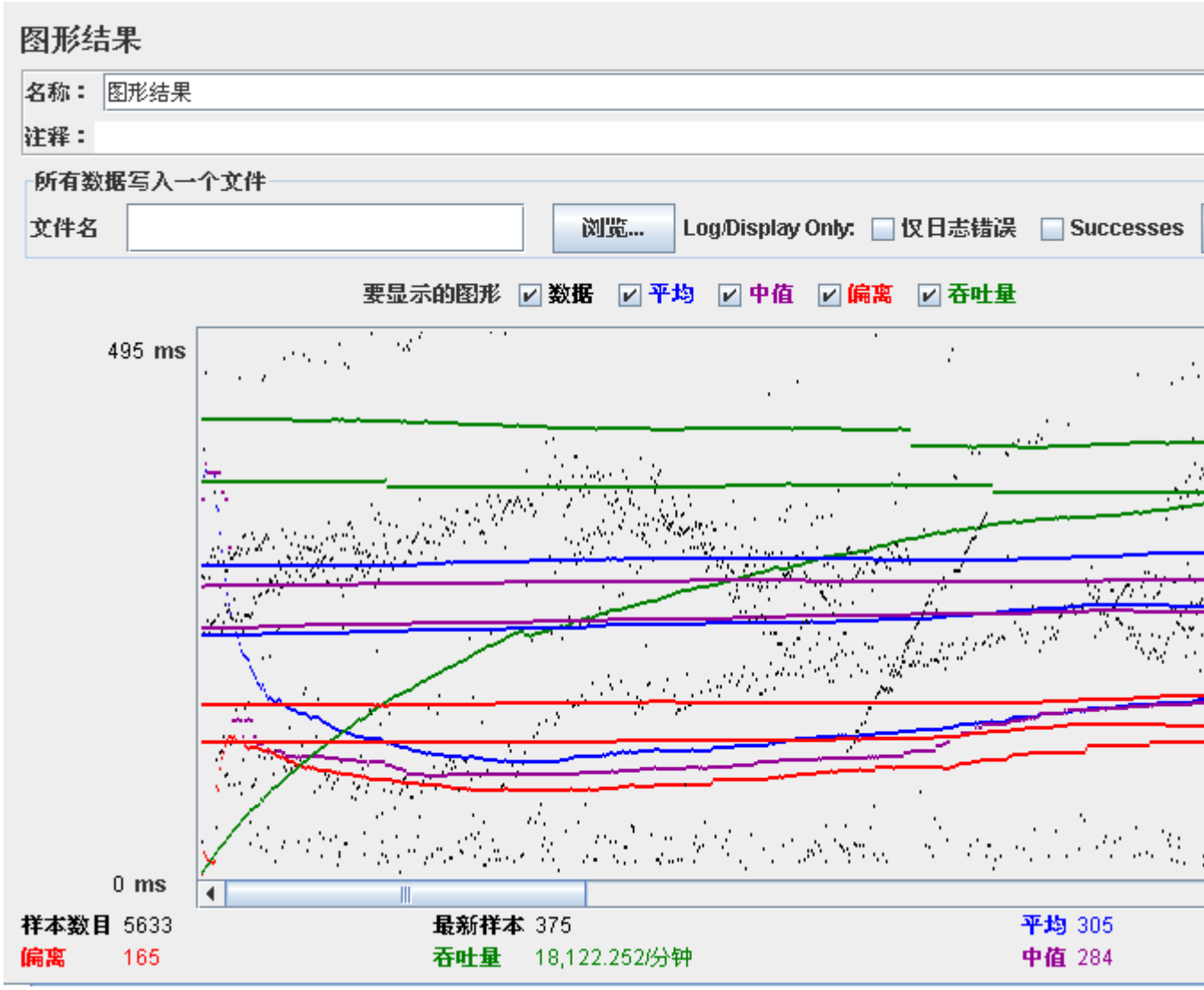


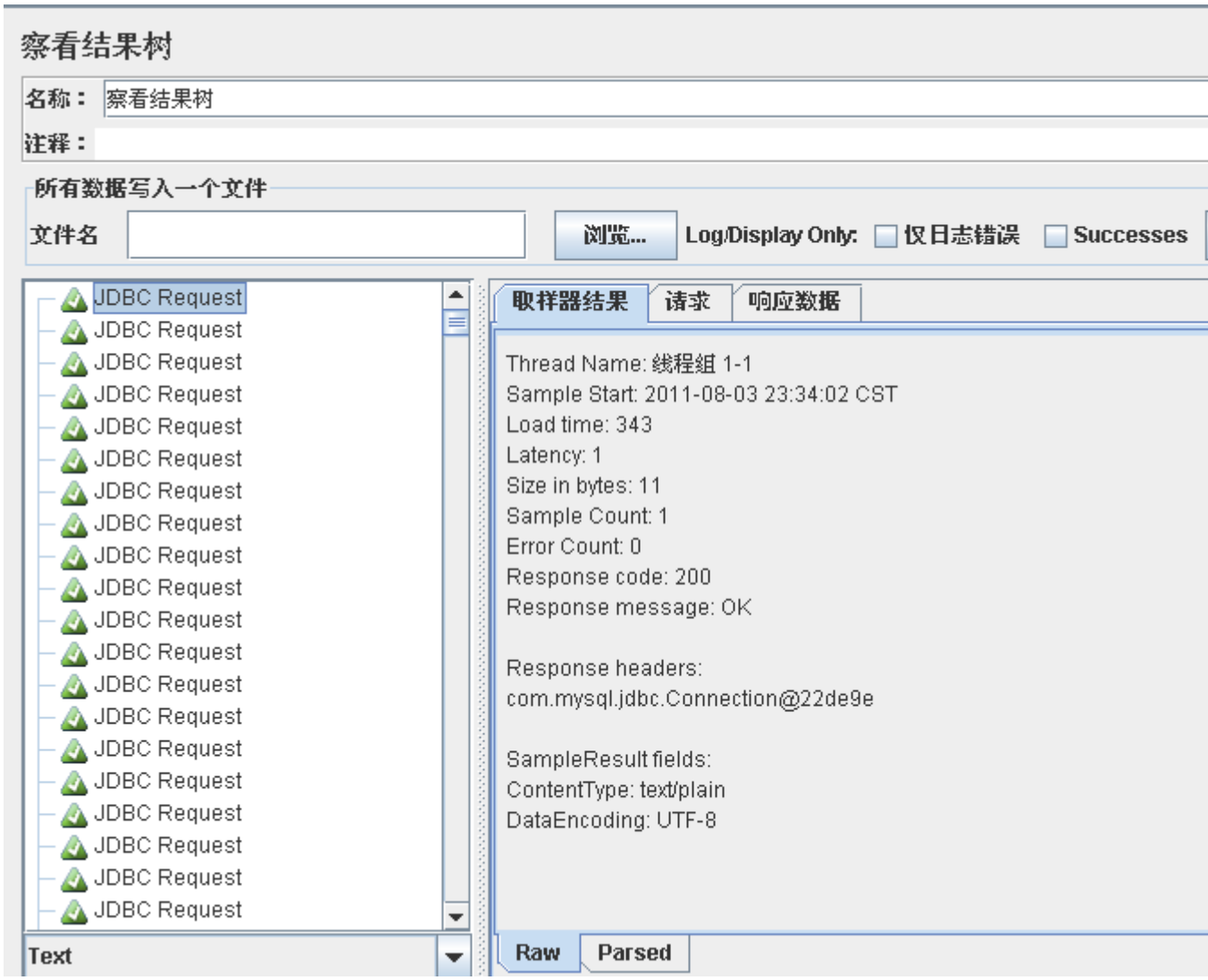
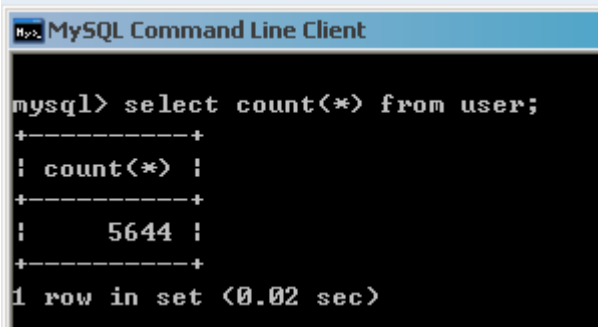
5.

添加监听器，我们这里选择添加“图形结果”和“查看结果树”，点击菜单栏上的“启动”--->运行。

查看我们的运行结果。

火龙果 · 整理
uml.org.cn





在测试的过程中，通过数据库命令，可以查看当前数据库插入了多少数据