

自学与分享平台：www.besttest.cn 微博：<http://weibo.com/besttest>

BestTest 悦分享越快乐

——专业的软件测试自学与分享平台

开源性能测试工具

安大叔

QQ群：49526682

新浪微博：<http://weibo.com/TestAndashu>

腾讯微博：<http://t.qq.com/w223ewe>

besttest 安大叔自学与分享平台：www.besttest.cn 微博：<http://weibo.com/besttest>

常见性能测试工具

Loadrunner优点

- 成熟，学习资料多，功能强大简单可以方便快捷进行性能测试
- 支持协议种类多Web(HTTP/HTML)、Sockets、net 协议、web services、常用数据库协议（ODBC，ORACLE，SQLSERVER 等）、邮件（SMTP、pop3）等等
- 监控报表全面

缺点：

- 商业，收费，破解稍微麻烦

常见性能测试工具

jmeter

- 是Apache组织开发的基于Java的压力测试工具
- 可用于对静态的和动态的资源（文件，Servlet，Perl脚本，java 对象，数据库和查询，FTP服务器等等）的性能进行测试
- jdk1.6以上版本

常见性能测试工具

Jmeter优点

- 开源免费，简单小巧
- 纯java，支持linux跟win

Jmeter缺点

- 无IP欺骗
- 报表类型少
- 上手比lr难
- 场景设置没有lr灵活
- 无进程模式
- 结果数据不是很准确
- 网上资料较少

微博：<http://weibo.com/besttest>

常见性能测试工具

apache ab webbench

优点:小巧便捷, 支持并发多

缺点：不判断返回数据

只支持url形式

功能较弱小

bestres/besttest

jmeter

JMeter，一个100%的纯Java桌面应用，它是Apache组织的开放源代码项目，它是功能和性能测试的工具。JMeter可以用于测试静态或者动态资源（文件、Servlets、Perl脚本、Java对象、数据库和查询、ftp服务器或者其他资源）

- a) 能够对HTTP和FTP服务器进行压力和性能测试，也可以对任何数据库进行同样的测试（通过JDBC）。
- b) 完全的可移植性和100% 纯Java。
- c) 完全 Swing 和轻量组件支持（预编译的JAR使用 javax.swing.*）包。
- d) 完全多线程 框架允许通过多个线程并发取样和 通过单独的线程组对不同的功能同时取样。
- e) 缓存和离线分析/回放测试结果。
- g) 高可扩展性：
- h) 可链接的取样器允许无限制的测试能力。
- i) 各种负载统计表和可链接的计时器可供选择。
- j) 数据分析和可视化插件提供了很好的可扩展性以及 以及个性化。
- k) 具有提供动态输入到测试的功能（包括Javascript）。
- l) 支持脚本变成的取样器。

自学与分享平台：www.besttest.cn 微博：<http://weibo.com/besttest>

jmeter—脚本录制

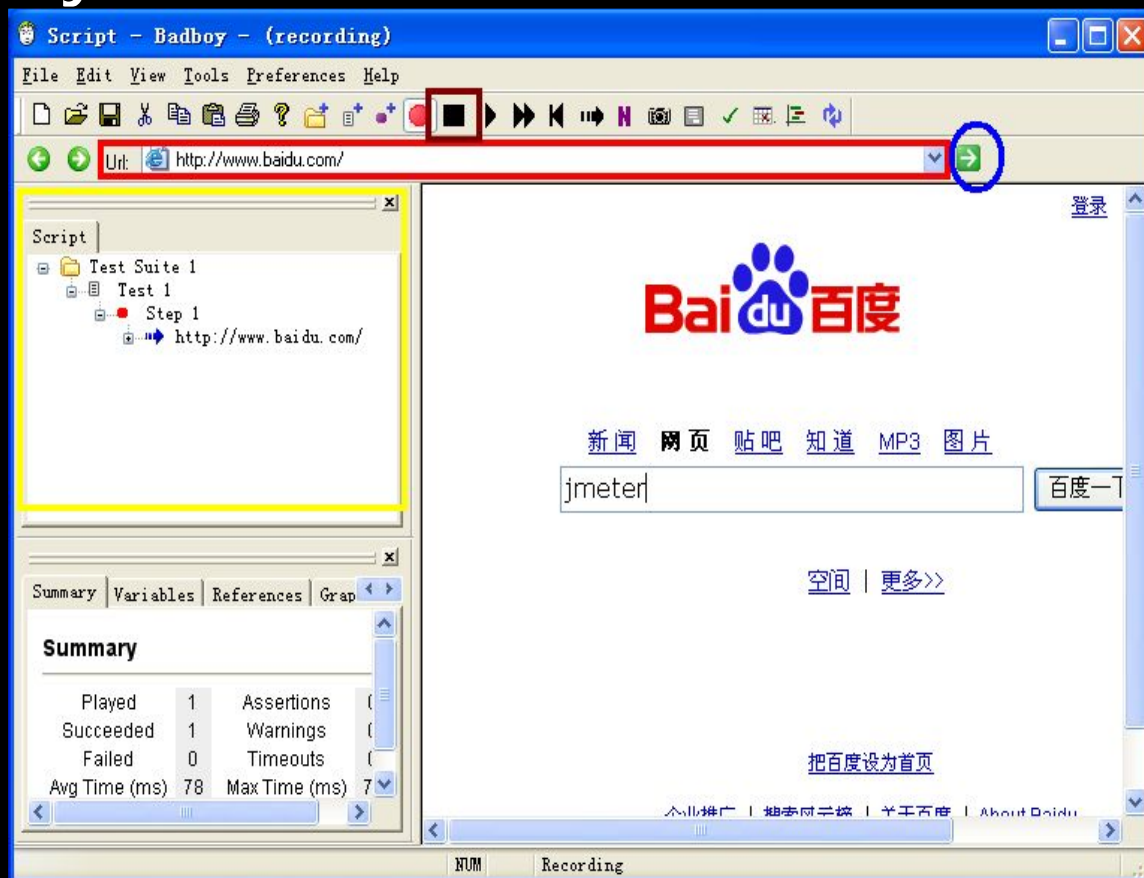
- 1、利用BadBoy进行脚本录制
- 2、利用JMeter的代理服务器功能进行脚本录制

besttest自学与分享平台 www.besttest.cn 微博：<http://weibo.com/besttest>

jmeter—脚本录制

- 通过Badboy的官方网站 (<http://www.badboy.com.au>) 下载Badboy的最新版本；
- 安装Badboy。安装过程同一般的Windows 应用程序没有什么区别，安装完成后你可以在桌面和Windows开始菜单中看到相应的快捷方式——如果找不到，可以找一下Badboy安装目录下的Badboy.exe 文件，直接双击启动Badboy；
- 启动Badboy，你可以看到下面的界面。

jmeter—脚本录制



在地址栏（图中红色方框标注的部分）中输入你需要录制的Web应用的URL——这里我们以

<http://www.baidu.com> 为例。

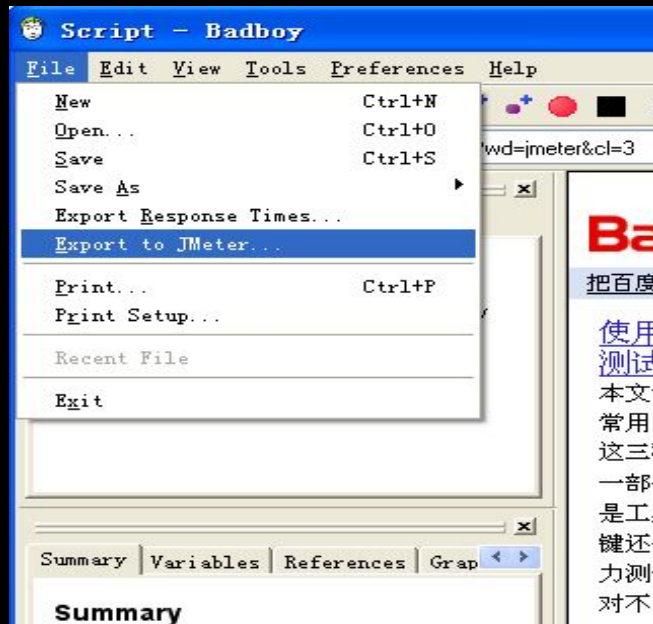
点击 开始录制 按钮（图中蓝色圆圈标注的部分）开始录制。

开始录制后，你可以直接在Badboy内嵌的浏览器（主界面的右侧）中对被测应用进行操作，所有的操作都会被记录在主界面左侧的编辑窗口中（图中黄色方框标注的部分）——在这个试验中，我们在baidu的搜索引擎中输入 [JMeter](#) 进行搜索。不过录制下来的脚本并不是一行行的代码，而是一个个Web对象——这有点像LoadRunner的VuGen中的Tree View视图；

录制完成后，点击工具栏中的“停止录制”按钮（图中紫色方框标注的部分），完成脚本的录制；

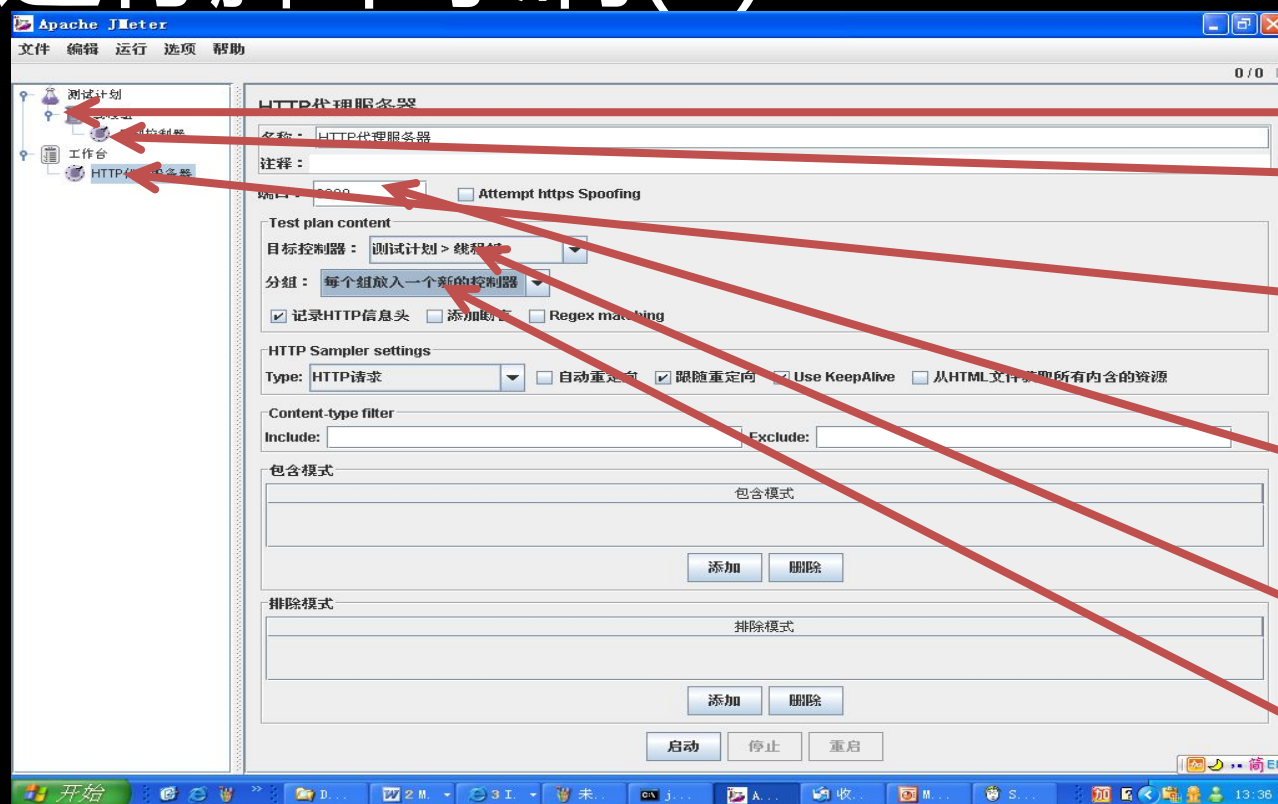
jmeter—脚本录制

- 选择“File -> Export to JMeter”菜单，填写文件名“baidu.jmx”，将录制好脚本导出为JMeter脚本格式。也可以选择“File -> Save”菜单保存为Badboy脚本；
- 启动JMeter并打开刚刚生成的测试脚本，就可以用JMeter进行测试了。



自学与分享平台：www.besttest.cn 微博：<http://weibo.com/besttest>

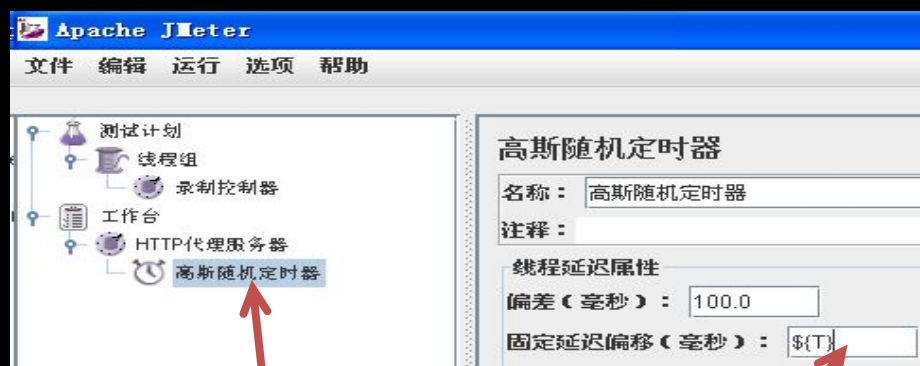
利用JMeter的代理服务器功能进行脚本录制(1)



- 在测试计划中添加线程组，在线程组中添加逻辑控制器 - 录制控制器。
- 在工作台中添加非测试元件 - HTTP代理服务器。
- 端口：即代理服务器的监听端口，我们设为8080。
- 目标控制器选择：测试计划>线程组；
分组选择：每个组放入一个新的

besttest 安大叔自学与分享平台：www.besttest.cn 微博：<http://weibo.com/besttest>

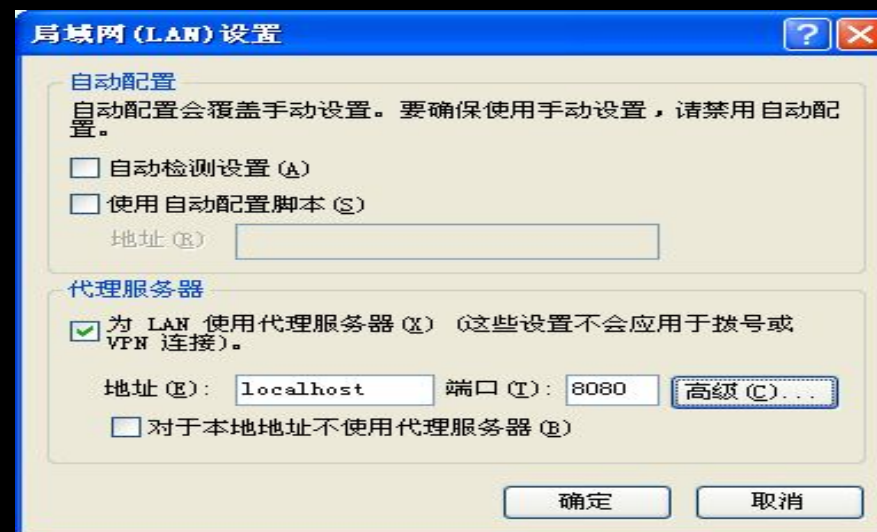
利用JMeter的代理服务器功能进行脚本录制(2)



- 在HTTP代理服务器中添加定时器 - 高斯随机定时器（用于告知Jmeter来在其生成的HTTP请求中自动的增加一个定时器）。定时器将会使相应的的取样器被延迟。延时的规则是，在上一个访问请求被响应并延时了指定的时间后，下一个被定时器影响的取样访问请求才会被发送出去。
- 如果在代理服务器元件里使用了高斯随机定时器，就应该在其中的固定延迟偏移（Constant Delay Offset）设置项里添上`${T}`（用于自动引用纪录的延迟时间）。

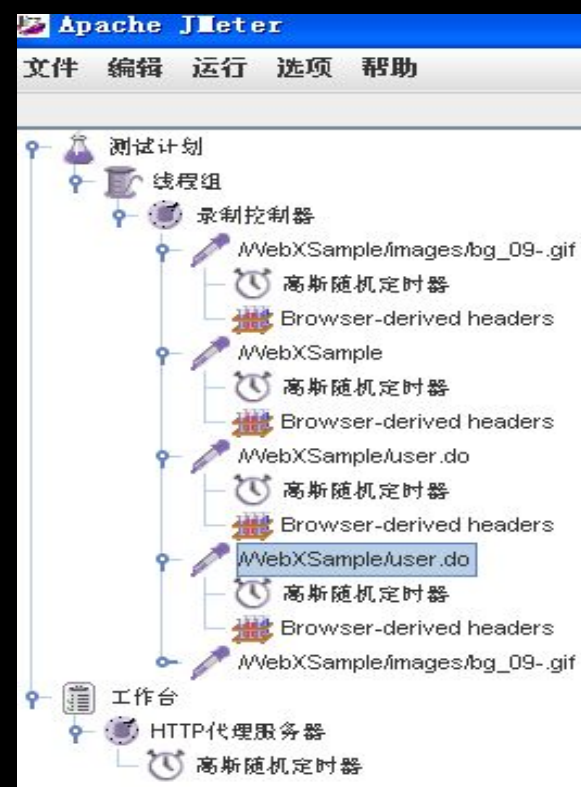
利用JMeter的代理服务器功能进行脚本录制(3)

- 代理服务器配置好以后，点击启动，代理服务器就会开始记录所接受的HTTP 访问请求。
- 打开浏览器，打开Internet选项，将局域网（LAN）设置中的代理服务器设为：localhost，端口为在代理服务器中设的端口：8080



利用JMeter的代理服务器功能 进行脚本录制(4)

- 在浏览器地址栏中输入地址并进行相关的操作进行录制，录制完成后，停止HTTP 代理服务器；在录制控制器元件上单击右键将记录的元件保存为一个文件用于以后重用，另外，不要忘了恢复浏览器的代理服务器设置。

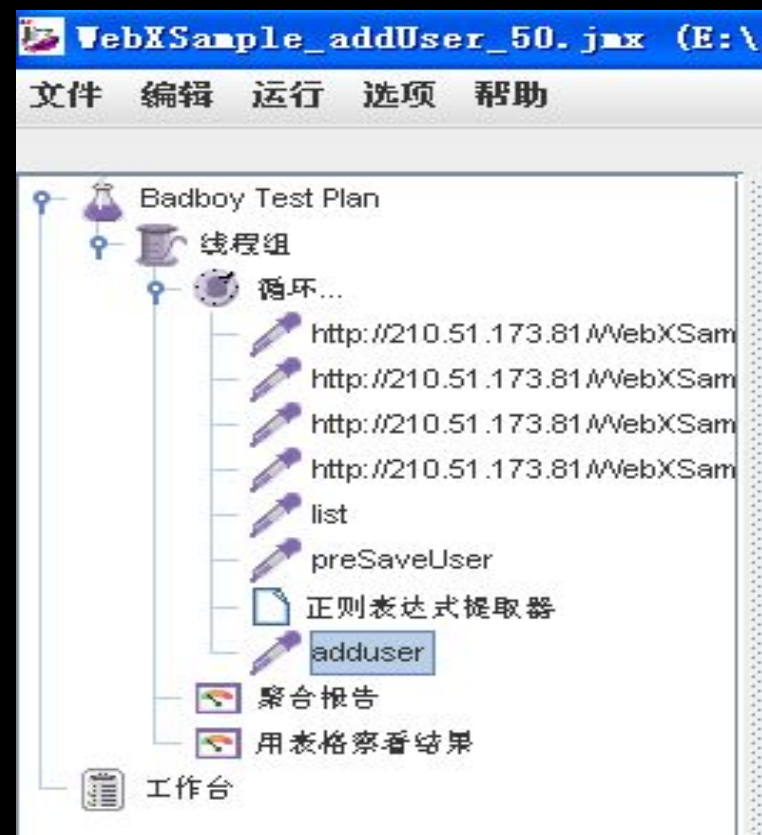


运行Jmeter进行测试(1)

- 脚本录制完毕后，就可以运行JMeter来进行我们的测试了。
- 到JMeter的网站
http://jakarta.apache.org/site/downloads/downloads_jmeter.cgi
- 下载JMeter的最新版本，将zip文件解压缩到D:/JMeter，运行D:/JMeter/bin下的jmeter.bat即可。

运行Jmeter进行测试(2)

- 打开JMeter会有一个默认的测试计划，点击文件 - 打开，选中录制的脚本文件如：WebXSample_addUser.jmx，打开脚本进行测试。
- 在线程组上添加监听器 - 聚合报告（用于分析测试结果）后，点击运行 - 启动，开始测试，测试完毕后在聚合报告中就可以看到测试结果。一个简单的测试计划就完成了。



运行Jmeter进行测试(3)

- 聚合报告显示测试结果

聚合报告

名称：

注释：

所有数据写入一个文件

文件名

☐ 仅日志错误

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Through...	KB/sec
http://21...	50	1152	16	3000	0	9000	0.00%	2.8/sec	.0
http://21...	50	482	16	2422	0	3187	0.00%	5.2/sec	11313.6
http://21...	50	322	16	1469	0	3063	0.00%	5.5/sec	.0
http://21...	50	219	0	844	0	3016	0.00%	5.5/sec	.0
list	50	108	0	250	0	1562	0.00%	5.5/sec	.0
preSave...	50	67	0	297	0	1140	0.00%	5.5/sec	.0
adduser	50	150	0	422	0	2938	0.00%	5.5/sec	.0
总体	350	357	15	1562	0	9000	0.00%	19.3/sec	5943.2

运行Jmeter进行测试(4) - 聚合报告详解

- 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，如果测试计划模拟10个用户，每个用户迭代10次，这里就会显示100
- Average：平均响应时间——默认情况下是单个 Request 的平均响应时间，当使用了事务控制器时，也可以以事务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%：错误率，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）
- KB/Sec：每秒从服务器端接收到的数据量

JMeter主要组件介绍

- 测试计划(Test Plan)是使用 JMeter 进行测试的起点，它是其它 JMeter 测试元件的容器。
- 线程组(Thread Group)代表一定数量的并发用户，它可以用来模拟并发用户发送请求。
- 取样器(sampler)定义实际的请求内容，被线程组包含，我们主要用 HTTP 请求。
- 监听器(Listener)
- 逻辑控制器(Logic Controller)
- 断言(Assertions)
- 配置元件(Config Element)
- 前置处理器(Pre Processors)和后置处理器(Post Processors)
- 定时器(Timer)

JMeter主要组件介绍 - 测试计划

测试计划

名称：

注释：

用户定义的变量

名称：	值
url	http://www.baidu.com

☐ 独立运行每个线程组（例如在一个组运行结束后启动下一个）

☐ 函数测试模式

只有当你需要记录每个请求从服务器取得的数据到文件时才需要选择函数测试模式。

选择这个选项很影响性能。

Add directory or jar to classpath

Library

- 测试计划(Test Plan)是使用 JMeter 进行测试的起点，它是其它 JMeter 测试元件的容器。
- 名称：你可以为你的测试计划取一个有意义的名字。
- 注释：对测试计划的注释。
- 用户定义的变量：用户可以自己定义变量，在用到此变量的时候直接用\${变量名}引用即可。例：变量名 = url，值 = http://www.baidu.com，在需要<http://www.baidu.com>时直接用\${url}即可。
- Add directory or jar to classpath：向类路径即%JMMETER-HOME%\lib中添加目录及jar包。

JMeter主要组件介绍 - 线程组

线程组

名称：

注释：

在取样器错误后要执行的动作

☒ 继续 ☐ 停止线程 ☐ 停止测试

线程属性

线程数：

Ramp-Up Period (in seconds):

循环次数 ☐ 永远

☒ 调度器

调度器配置

启动时间

结束时间

持续时间 (秒)

启动延迟 (秒)

- 名称：为线程组起名字。
- 线程属性 - 线程数：设置发送请求的用户数目，即并发数。
- 线程属性 - Ramp-Up Period(in second)：线程间的时间间隔，单位是秒。即所有线程在多少时间内启动。
- 线程属性 - 循环次数：请求的重复次数，如果选择后面的forever（默认），那么请求将一直继续，如果不选择forever，而在输入框中输入数字，那么请求将重复指定的次数，如果输入1，那么请求将执行一次，如果是0，会出现问题。
- 调度器配置 - 启动时间：测试计划什么时候启动，启动延迟会覆盖它。
- 调度器配置 - 结束时间：测试计划什么时候结束，持续时间会覆盖它。
- 调度器配置 - 持续时间：测试计划持续多长时间，会覆盖结束时间。
- 调度器配置 - 启动延迟：测试计划延迟多长时间启动，会覆盖启动时间。

JMeter主要组件介绍 - 取样器(HTTP请求)

HTTP请求

名称：

注释：

Web服务器

服务器名称或IP： 端口号：

HTTP请求

协议： 方法： Content encoding：

路径：

☐ 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	编码？	包含等于？
method	doSave	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
bookName	111\${__Random(100,200,random...}	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
bookConcern	111\${__Random(100,200,random...}	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

同请求一起发送文件：

文件名称：

参数名称：

MIME类型：

其他任务

☐ 从HTML文件获取所有内含的资源 ☐ 用作监视器

Embedded URLs must match:

- 名称：HTTP请求的名字。
- Web服务器 - 服务器名称或IP、端口号，脚本录制时自动添加，也可以使用“用户自定义变量”。
- HTTP请求：用默认值即可。
- 同请求一起发送参数：请求中的参数值可以在此设置，需要用到参数化及动态数据关联，我们在后面专门介绍。
- 同请求一起发送文件：我可以制定同请求一起发送哪个文件。
- 其他任务包括：从HTML文件中获取所有内含的资源、用作监视器、Embedded URLs must match。

JMeter主要组件介绍 - 监听器

- 监听器(Listener)负责收集测试结果，同时也被告知了结果显示的方式。我们常用的包括：[聚合报告](#)、[查看结果树](#)、[用表格查看结果](#)，都支持将结果数据写入文件。其他的添加上去看看就行。[聚合报告](#)前面我们介绍过，后面是[查看结果树](#)和[用表格查看结果](#)的截图。



JMeter主要组件介绍 - 监听器(用表格查看结果)

用表格察看结果

名称：

注释：

所有数据写入一个文件

文件名

浏览...

☐ 仅日志错误

Configure

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes
1	13:47:16.031	Thread Group 1-1	addBook	625		42637
2	13:47:16.031	Thread Group 1-1	事务控制器	625		42637
3	13:47:16.031	Thread Group 1-3	addBook	672		42637
4	13:47:16.031	Thread Group 1-3	事务控制器	672		42637
5	13:47:16.046	Thread Group 1-5	addBook	844		42637
6	13:47:16.046	Thread Group 1-5	事务控制器	860		42637
7	13:47:16.031	Thread Group 1-2	addBook	890		42573
8	13:47:16.031	Thread Group 1-4	addBook	906		42637
9	13:47:16.031	Thread Group 1-4	事务控制器	906		42637
10	13:47:16.031	Thread Group 1-2	事务控制器	906		42573

样本数目 10

最新样本 906

平均 790

偏离 118

JMeter主要组件介绍 - 监听器(查看结果树)


察看结果树


名称：


注释：

所有数据写入一个文件


文件名 ☐ 仅日志错误


 addBook


 事务控制器


 addBook


 事务控制器


 addBook

 事务控制器

 addBook

 事务控制器

 addBook

 事务控制器

取样器结果

请求

响应数据

图书搜索

t>

图书名称:

图书分类:

图书作者:

图书价格:

出版日期:

(yyyy-mm-dd)

搜索

JMeter主要组件介绍 - 逻辑控制器

- 逻辑控制器(Logic Controller)可以自定义JMeter发送请求的行为逻辑，它与Sampler结合使用可以模拟复杂的请求序列。



JMeter主要组件介绍 - 逻辑控制器

The image shows two configuration panels from the JMeter GUI. The top panel is for a '循环控制器' (Loop Controller). It has a '名称' (Name) field set to '循环控制器', an empty '注释' (Comment) field, and a '循环次数' (Loop count) section with an unchecked '永远' (Forever) checkbox and a text box containing '1'. The bottom panel is for a '事务控制器' (Transaction Controller). It has a '名称' (Name) field set to '事务控制器', an empty '注释' (Comment) field, and a 'Generate parent sample' checkbox which is unchecked.

循环控制器

名称：

注释：

循环次数 ☐ 永远

事务控制器

名称：

注释：

☐ Generate parent sample

- 循环控制器可以设置请求的循环次数或永远循环（如果选中永远的话）。
- 事务控制器可以将多个请求放在同一个事务中。如果选中Generate parent sample，则聚合报告中只显示事务控制器的数据，而不会显示其中的各个请求的数据，反之则全部显示。

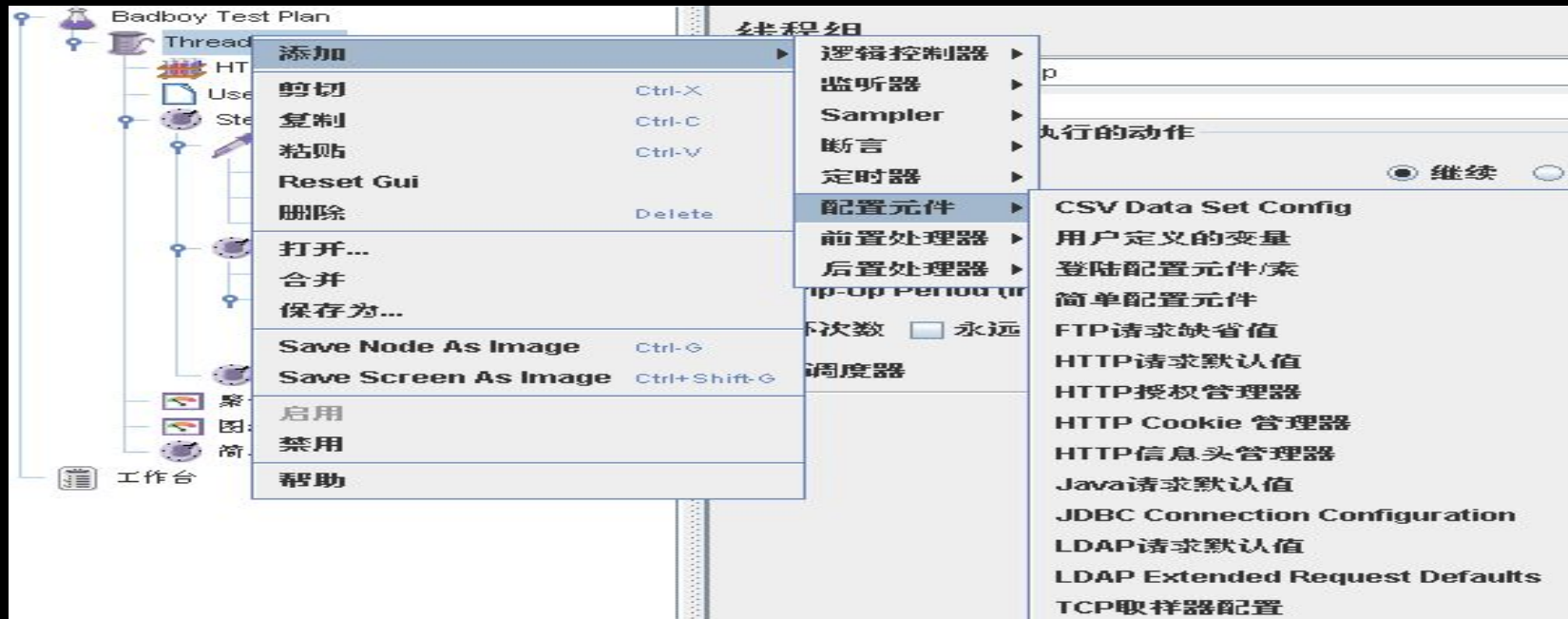
JMeter主要组件介绍 - 断言

- 断言(Assertions)可以用来判断请求响应的结果是否如用户所期望的。它可以用来隔离问题域，即在确保功能正确的前提下执行压力测试。这个限制对于有效的测试是非常有用的。



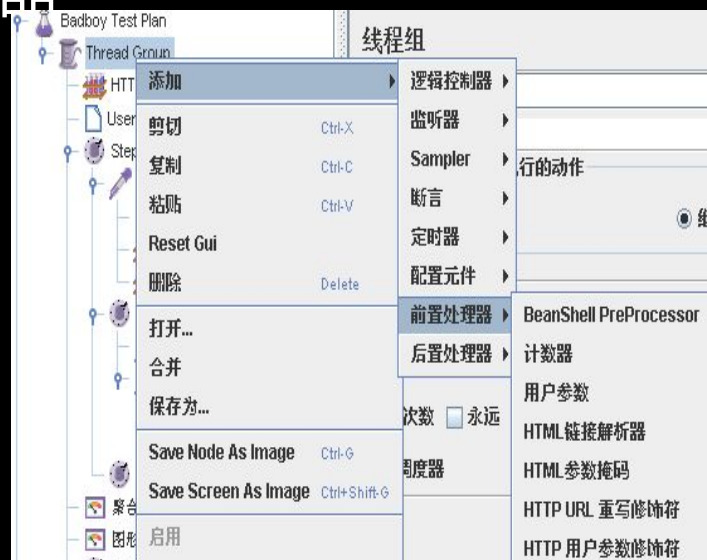
JMeter主要组件介绍 - 配置元件

- 配置元件(Config Element)维护Sampler需要的配置信息，并根据实际的需要会修改请求的内容。我们主要在参数化中用到CSV Data Set Config



自学与分享平台：www.besttest.cn 微博：<http://weibo.com/besttest>

JMeter主要组件介绍 - 前置处理器和后置处理器



- 前置处理器(Pre Processors)和后置处理器(Post Processors)负责在生成请求之前和之后完成工作。前置处理器常常用来修改请求的设置，后置处理器则常常用来处理响应的数据。我们主要在动态关联中用到后置处理器的正则表达式提取器。

besttest 安大叔自学与分享平台：www.besttest.cn 微博：<http://weibo.com/besttest>

JMeter主要组件介绍 - 定时器

- 定时器(Timer)负责定义请求之间的延迟间隔。



参数化设置

- 在我们的测试计划中我们使用的都是HTTP请求，在HTTP请求中常常会提交一些参数，我们为了不想在每一次请求中都使用相同的参数值，就要想办法让每一次请求的参数的值都是不一样的，我们就需要进行参数化设置，这里我们介绍两种参数化设置的方法。
一种是利用函数助手中的_Random函数进行参数化设置；
一种是利用配置元件中的CSV Data Set Config进行参数化设置。

参数化设置 - 函数助手(1)

- JMeter中有一个函数助手的功能，里面内置了多个函数，我们可以利用其中的_Random函数来进行请求中的参数化设置。

参数化设置 - 函数助手(2)

- 在JMeter中选择“选项 - 函数助手对话框”打开函数助手对话框。



参数化设置 - 函数助手(3)



- 选择一个功能_Random。
- 设定最小值为1。
- 设定最大值为100。
- 函数名称设为func1。
- 点击生成将生成一个引用字符串\${_Random(1,100,func1)}，在需要的请求参数中拷贝该字符串即可。
- 其中函数名称用来在引用中作区别，即我们可以设定不同的名称在不同的地方引用同一个函数的功能。

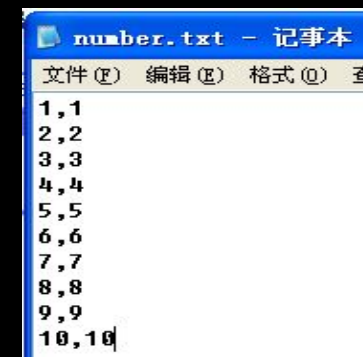
参数化设置 - 函数助手(4)

- 例如：我们生成了两个串`${_Random(1,100,func1)}`、`${_Random(1,100,func2)}`，在添加图书的请求中，我们在bookName的参数值中引用`${_Random(1,100,func1)}`，在bookConcern的参数值中应用`${_Random(1,100,func2)}`，见截图，这样测试运行时不同的线程将产生不同的bookName和bookConcern。

同请求一起发送参数：				
名称：	值	编码？	包含等于？	
method	doSave	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	▲
bookName	bookName\${_Random(1,100,func1)}	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	≡
bookConcern	bookConcern\${_Random(1,100,func2)}	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	▼

参数化设置 - CSV Data Set Config(1)

- 在线程组上点击右键选择添加 - 配置元件 - CSV Data Set Config，打开CSV Data Set Config。
- Filename文件名：即参数化要引用的文件名。这里是d:\number.txt
- File encoding文件编码：可以不填。
- Variable Names(comma-delimited)变量名(用逗号分割)：多个变量可以引用同一个文件，用逗号分割。这里是num1, num2
- Delimiter(use '\t' for tab)：参数文件中多个变量值的分隔符，\t表示用tab键分割，默认是逗号。
- Recycle on EOF?：结束后是否循环？默认是True。
- Number.txt文件内容见截图。



参数化设置 - CSV Data Set Config(2)

- 在添加图书的请求中，我们在bookName的参数值中引用\${num1}，在bookConcern的参数值中应用\${num2}，见截图，这样测试运行时不同的线程将产生不同的bookName和bookConcern。

同请求一起发送参数：				
名称：	值	编码？	包含等于？	
method	doSave	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	▲
bookName	bookName\${num1}	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	☰
bookConcern	bookConcern\${num2}	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	▼

动态数据关联

- 在HTTP请求的参数中经常会遇到一些参数的值是从服务器相应的动态数据，这些数据需要进行关联才能使得每一次请求都能成功得被服务器接受，在JMeter中采用正则表达式提取器来获取这些动态的数据。
- 正则表达式允许用户从服务器响应中获取数据，作为一个后置处理器，正则表达式提取器会在每一个请求执行后再执行，正则表达式提取请求的参数值，产生模板字符串，并将结果保存到给出的变量中

动态数据关联(1) - 正则表达式提取器

- 在HTTP请求的参数中经常会遇到一些参数的值是从服务器相应的动态数据，这些数据需要进行关联才能使得每一次请求都能成功得被服务器接受，在JMeter中采用正则表达式提取器来获取这些动态的数据。比如我们想要获得从服务器相应的TOKEN，就可以使用下面的正则表达式提取器。

正则表达式提取器	
名称：	正则表达式提取器
注释：	
要检查的响应字段	
<input checked="" type="radio"/> 主体 <input type="radio"/> 信息头 <input type="radio"/> URL	
引用名称：	token
正则表达式：	name="org.apache.struts.taglib.html.TOKEN" value="(.)">
模板：	\$1\$
匹配数字（0代表随机）：	0
缺省值：	0

动态数据关联(2)

- 使用时就可以使用`${token_g1}`来表示获取到的TOKEN值。
如下图：

同请求一起发送参数：		
名称：	值	编辑
method	save	
id		
corpid	1	
rolesString	:1:3:2	
org.apache.struts.taglib.html.TOKEN	<code>\${token_g1}</code>	
username	<code>userName\${__Random(0001,9999,nu9999)}</code>	



使用命令行运行JMeter脚本

- JMeter *-n -t my_test.jmx -l log.jtl*
- -n告知JMeter采用非图形化运行JMeter脚本
- -t指定要运行的脚本文件
- -l指定保存执行结果的文件名
- 执行命令前要检查当前目录是否是%JMeter_Home%\bin目录；
- 如果 JMeter 脚本不在当前目录，需要指定完整的路径；如果要把执行的结果保存在其他地方也要指定完整的路径。

利用XSLT分析JMeter结果文件(1)

- 将结果文件的扩展名改为xml，例：将log.jtl改为log.xml。

- 在log.xml的文件头

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<testResults version="1.2">
```

中添加

```
<?xml-stylesheet type="text/xsl" href="jmeter-results-report.xsl" ?>
```

使之变成

```
<?xml version="1.0" encoding="UTF-8"?>
```

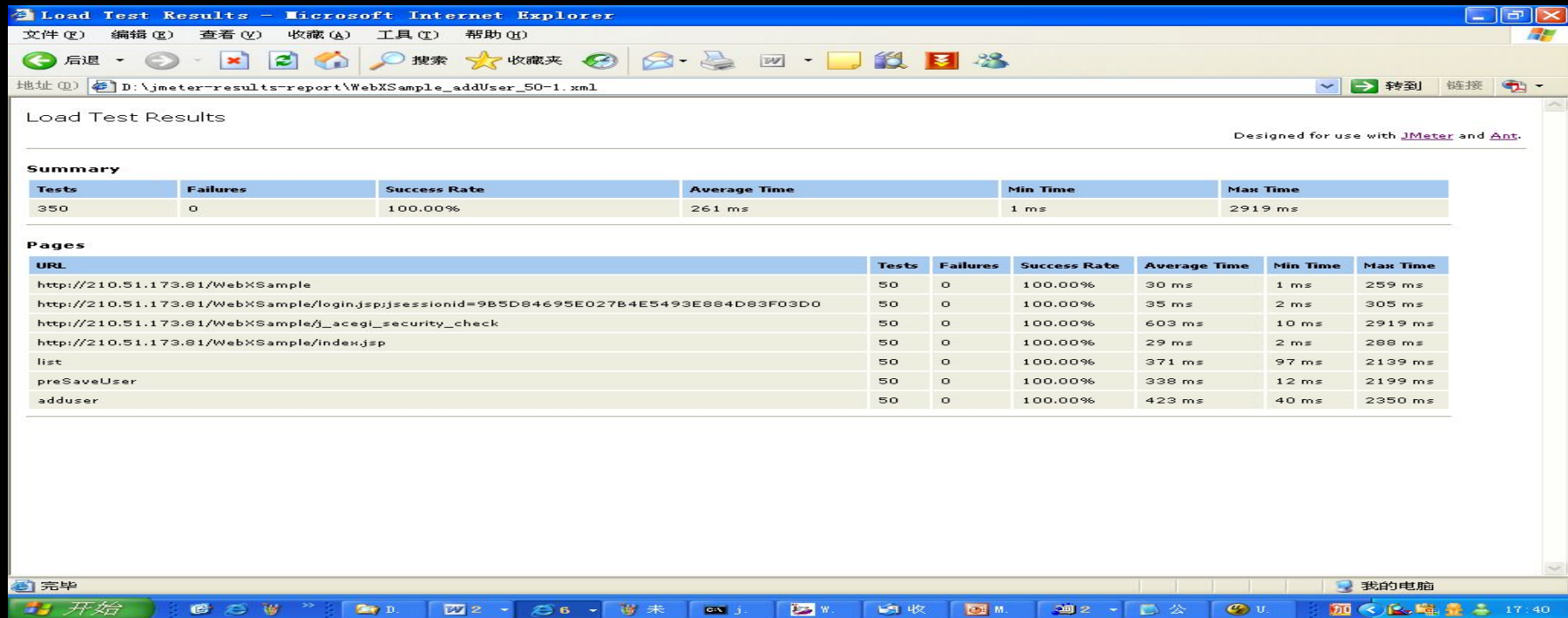
```
<?xml-stylesheet type="text/xsl" href="jmeter-results-report.xsl" ?>
```

```
<testResults version="1.2">
```

其中href指定xsl文件的路径，如果与xml文件在同一目录，则只写文件名即可，否则要写明详细路径。

利用XSLT分析JMeter结果文件(2)

- 将修改后的xml文件用浏览器打开即可得到统计结果



Load Test Results - Microsoft Internet Explorer

文件(F) 编辑(E) 查看(V) 收藏(A) 工具(T) 帮助(H)

后退 搜索 收藏夹

地址(D) D:\jmeter-results-report\WebXSample_addUser_50-1.xml 转到 链接

Load Test Results

Designed for use with JMeter and Ant.

Summary

Tests	Failures	Success Rate	Average Time	Min Time	Max Time
350	0	100.00%	261 ms	1 ms	2919 ms

Pages

URL	Tests	Failures	Success Rate	Average Time	Min Time	Max Time
http://210.51.173.81/WebXSample	50	0	100.00%	30 ms	1 ms	259 ms
http://210.51.173.81/WebXSample/login.jsp;jsessionid=9B5D84695E027B4E5493E884D83F03D0	50	0	100.00%	35 ms	2 ms	305 ms
http://210.51.173.81/WebXSample/j_acegi_security_check	50	0	100.00%	603 ms	10 ms	2919 ms
http://210.51.173.81/WebXSample/index.jsp	50	0	100.00%	29 ms	2 ms	288 ms
list	50	0	100.00%	371 ms	97 ms	2139 ms
preSaveUser	50	0	100.00%	338 ms	12 ms	2199 ms
adduser	50	0	100.00%	423 ms	40 ms	2350 ms

完毕 我的电脑

开始 2 6 未 C:\ J. W. 收 M 2 公 U. 17:40

- 作为一个纯 JAVA 的GUI应用，JMeter对于CPU和内存的消耗还是很惊人的，所以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，甚至还会引起JAVA内存溢出的错误。不过，JMeter也可以像 LoadRunner 一样通过使用多台机器运行所谓的代理来分担负载产生器自身的压力，并借此来获取更大的并发用户数，我们只需手动配置一下即可。
- 1、在所有期望运行JMeter作为 负载产生器的机器上安装JMeter，并确定其中一台机器作为 控制器，其他的机器作为 代理。然后运行所有 代理 机器上的JMeter-server.bat文件——假定我们使用两台机器172.20.80.47和172.20.80.68作为 代理；
- 2、在Controller 机器的JMeter安装目录下找到 bin 目录，再找到 jmeter.properties 这个文件，使用记事本或者其他文字编辑工具打开它；
- 3、在打开的文件中查找 “remote hosts=” 这个字符串，你可以找到这样一行 “remote hosts=127.0.0.1”。其中的 127.0.0.1 表示运行JMeter代理的机器，这里需要修改为 “remote_hosts=172.20.80.47, 172.20.80.68”；
- 4、保存文件，并重新启动控制器机器上的JMeter，并进入 启动 -> 远程启动 菜单项。就会看到我们刚才添加的两个代理的地址，选中即可运行，如果想同时启动所有代理，选择 远程全部启动 即可。
- 要进行分布式测试代理机器上需要添加环境变量，即添加用户变量JMETER_HOME = d:\jmeter，系统变量中的path中添加d:\jmeter\bin（假设jmeter放在d盘根目录下）。

自学与分享平台：www.besttest.cn 微博：<http://weibo.com/besttest>

- 分布式测试，jmeter运行步骤：
- 1.客户端设置环境变量：
- window下配置：
新增JMETER_HOME=D:\soft\test\apache-jmeter-2.9
CLASSPATH=;%JMETER_HOME%\lib\ext\ApacheJMeter_core.jar;%JMETER_H
OME%\lib\jorphan.jar;%JMETER_HOME%\lib\logkit-2.0.jar;
PATH=;%JMETER_HOME%\bin
- linux 下配置：
export JMETER_HOME=/usr/local/apache-jmeter-2.9
export
PATH=\$JMETER_HOME/lib/ext/ApacheJMeter_core.jar:\$JMETER_HOME/lib/j
orphan.jar:\$JMETER_HOME/lib/logkit-2.0.jar:\$JMETER_HOME/bin:\$PATH

besttest 安大叔自学与分享平台：www.besttest.cn 微博：<http://weibo.com/besttest>

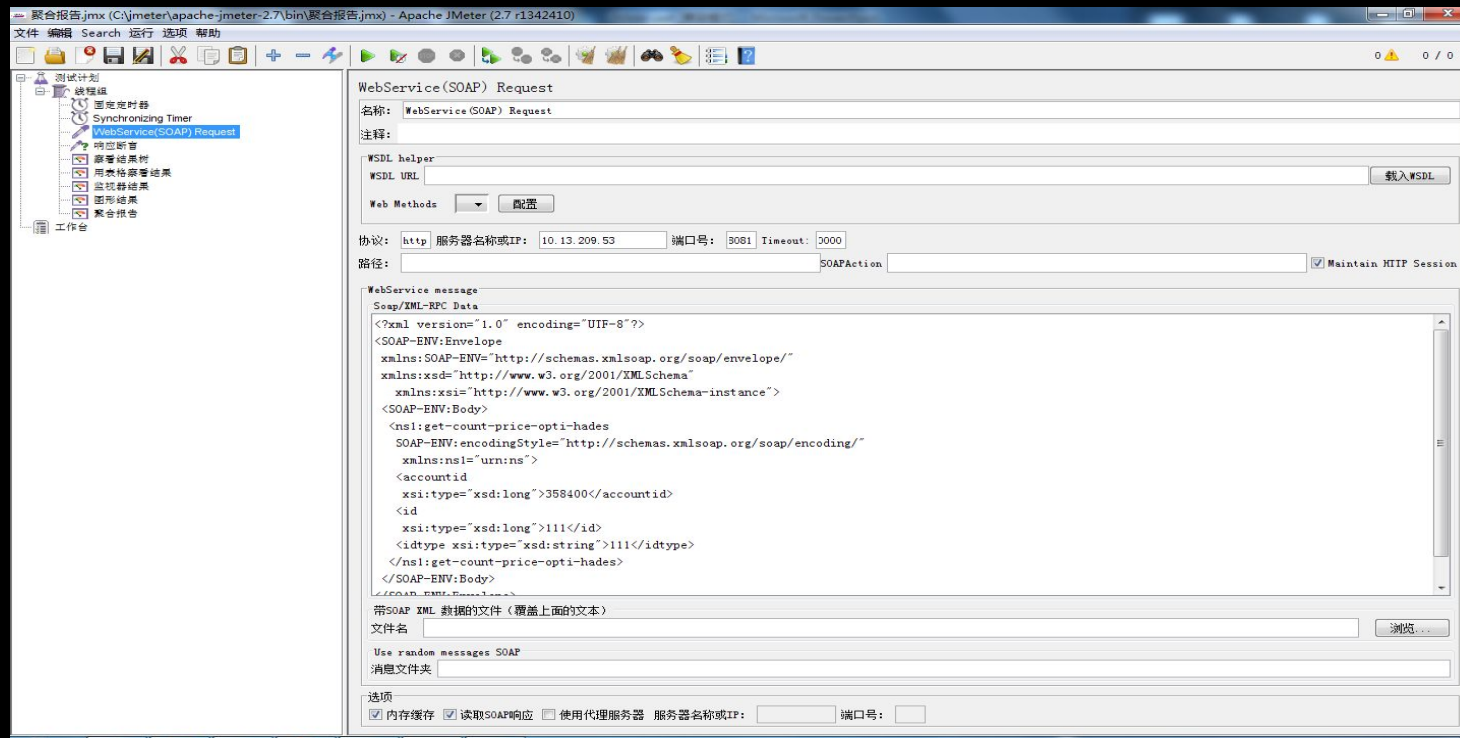
自学与分享平台：www.besttest.cn 微博：<http://weibo.com/besttest>

- 分布式测试，jmeter运行步骤：
- 2.修改客户端的配置文件jmeter.properties：
remote_hosts=10.1.164.159:1099,10.14.134.225:1099,127.0.0.1:1099
- 3.客户端和代理服务器启动JMETER-SERVER（默认端口号1099）
如果想更换端口号，windows运行如下命令：
SET SERVER_PORT=1099
JMETER-SERVER
如果想更换端口号，linux运行如下命令：
SERVER_PORT=1099 jmeter-server
- 4.运行客户端的jmeter.bat，运行-》远程启动具体IP即可
- 备注：Jmeter分布式控制过程中，各个Agent启动的线程数等于线程组中的配置，不是均分线程组中的配置。

besttest 安大叔自学与分享平台：www.besttest.cn 微博：<http://weibo.com/besttest>

自学与分享平台：www.besttest.cn 微博：<http://weibo.com/besttest>

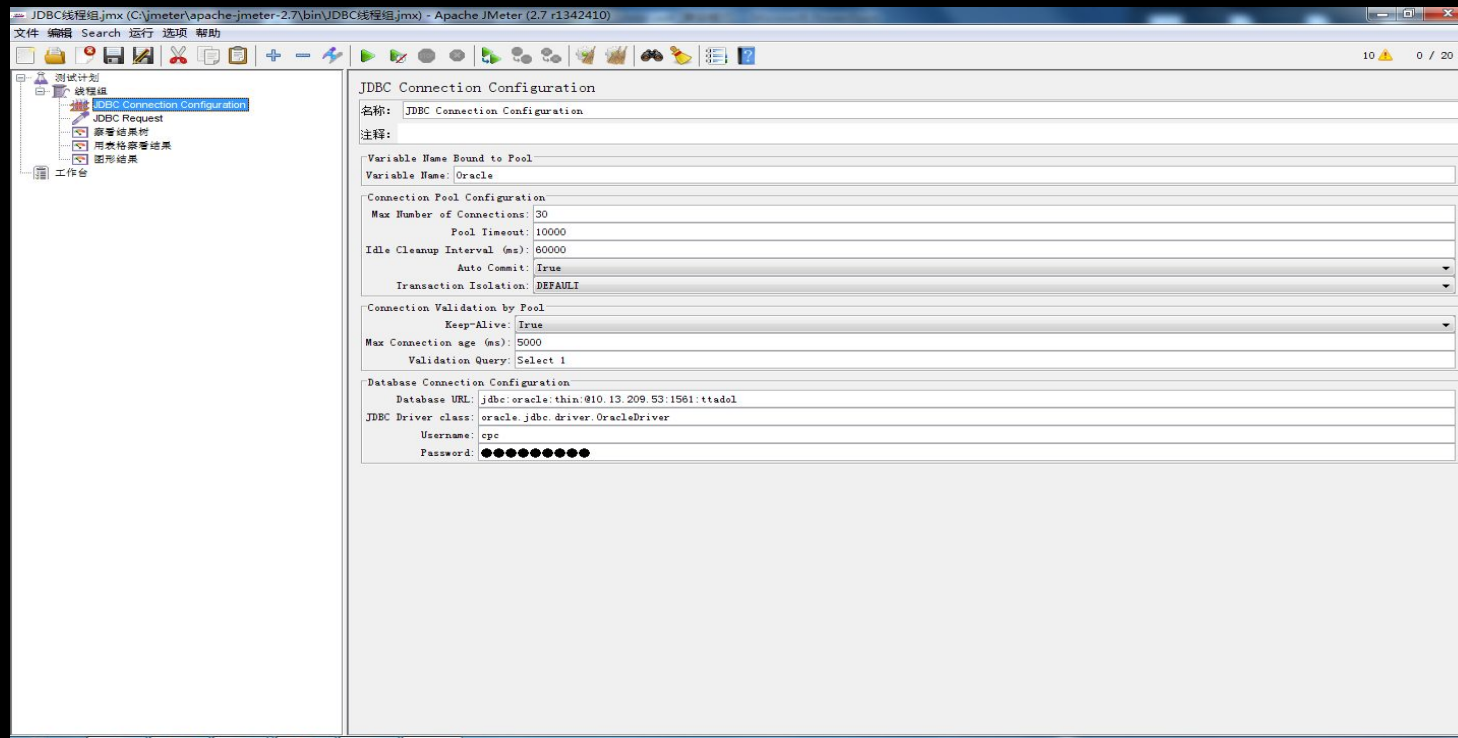
JMeter实战—WS接口、性能测试



besttest 安大叔自学与分享平台：www.besttest.cn 微博：<http://weibo.com/besttest>

自学与分享平台：www.besttest.cn 微博：<http://weibo.com/besttest>



JMeter实战—JDBC性能测试



besttest 安大叔自学与分享平台：www.besttest.cn 微博：<http://weibo.com/besttest>

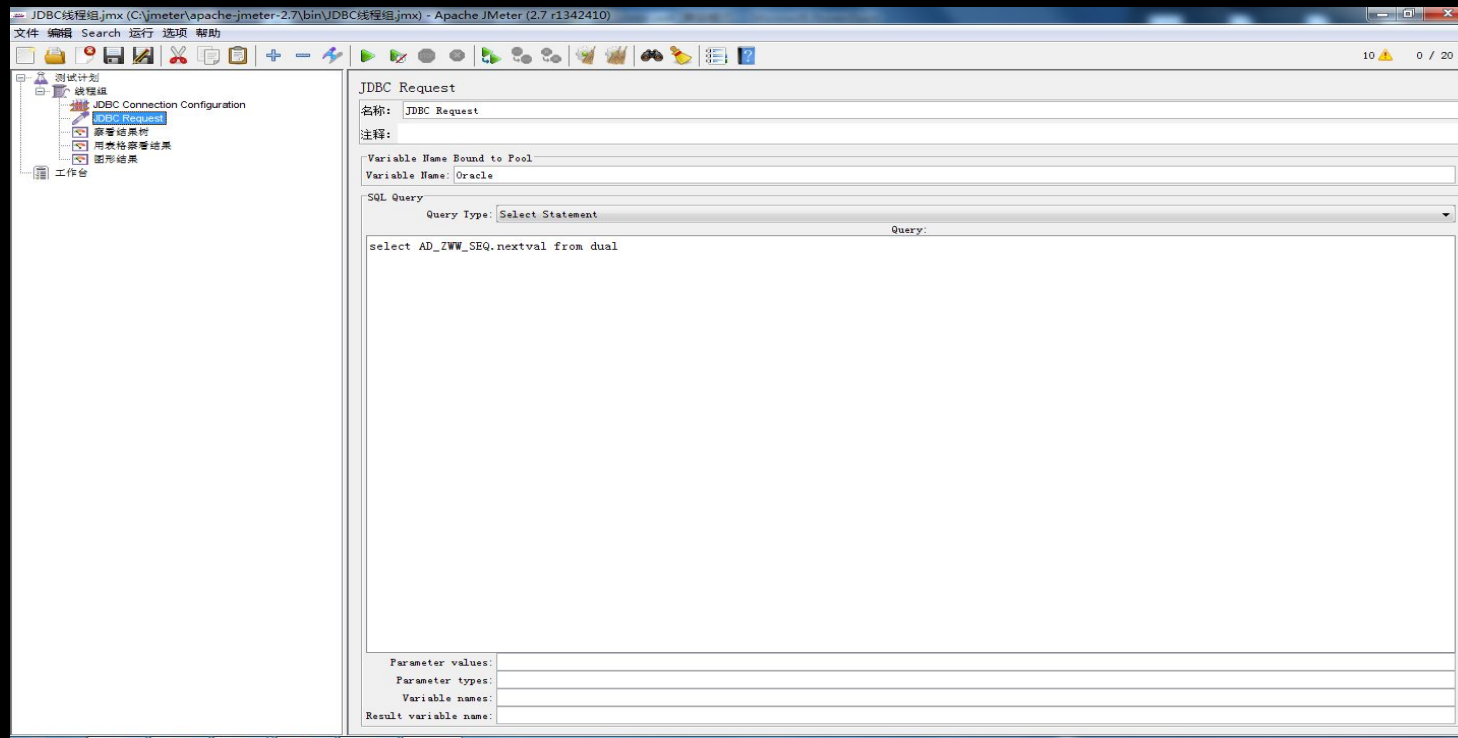
JMeter实战—JDBC性能测试

1. 下载JDBC驱动。将JDBC解压，
将jdbc驱动包，copy到jmeter/lib下
2. 打开JMeter，设置JDBC请求。
 - 1) JDBC URL: jdbc:mysql://localhost:3306/loaddb?autocollection=true .
其中 localhost:3306，这个是你测试的server地址
loaddb，这个是你测试的数据库名字。
这个url是jdbc的连接
 - 2) Driver Class: org.gjt.mm.mysql.Driver 或者 com.mysql.jdbc.Driver
 - 3) Username: 这个数据库的登录名
 - 4) password: 这个是数据库的登录密码
 - 5) 在Sql Query String中写sql语句进行测试。

 mysql-connector-java.jar	2013/5/15 18:20	Executable Jar File	529 KB
 mysql-connector-java-3.0.17-ga-bin.jar	2013/5/15 18:11	Executable Jar File	241 KB

自学与分享平台：www.besttest.cn 微博：<http://weibo.com/besttest>

JMeter实战—JDBC性能测试



besttest 安大叔自学与分享平台：www.besttest.cn 微博：<http://weibo.com/besttest>

apache ab

Apache自带，只判断200状态，不接受返回值

`./ab -c 1000 -n 10000 url`（-n表示执行次数，-c表示并发数）

Concurrency Level:	10
Time taken for tests:	0.496 seconds
Complete requests:	1000
Failed requests:	0
Write errors:	0
Non-2xx responses:	1000
Total transferred:	400000 bytes
HTML transferred:	211000 bytes
Requests per second:	2015.93 [#/sec] (mean)
Time per request:	4.960 [ms] (mean)
Time per request:	0.496 [ms] (mean, across all concurrent requests)
Transfer rate:	787.47 [Kbytes/sec] received

Connection Times (ms)					
	min	mean[+/-sd]	median	max	
Connect:	0	0	0.4	0	1
Processing:	2	5	1.1	4	12
Waiting:	2	4	1.1	4	12
Total:	2	5	1.1	5	12

apache ab

Concurrency Level 表示并发用户数，这是我们设置的参数之一

Time taken for tests 表示所有这些请求被处理完成所花费的总时间

Complete requests 表示总请求数量，这是我们设置的参数之一

Failed requests 表示失败的请求数量，这里的失败是指请求在连接服务器、发送数据等环节发生异常，以及无响应后超时的情况。如果接收到的HTTP响应数据的头信息中含有2XX以外的状态码，则会在测试结果中显示另一个名为“Non-2xx responses”的统计项，用于统计这部分请求数，这些请求并不算在失败的请求中。

Total transferred 表示所有请求的响应数据长度总和，包括每个HTTP响应数据的头信息和正文数据的长度。注意这里不包括HTTP请求数据的长度，仅仅为web服务器流向用户PC的应用层数据总长度。

HTML transferred 表示所有请求的响应数据中正文数据的总和，也就是减去了Total transferred中HTTP响应数据中的头信息的长度。

Requests per second 吞吐率，计算公式： $\text{Complete requests} / \text{Time taken for tests}$

Time per request 用户平均请求等待时间，计算公式： $\text{Time taken for tests} / (\text{Complete requests} / \text{Concurrency Level})$

Time per request(across all concurrent request) 服务器平均请求等待时间，计算公式： $\text{Time taken for tests} / \text{Complete requests}$ ，正好是吞吐率的倒数。也可以这么统计： $\text{Time per request} / \text{Concurrency Level}$

Transfer rate 表示这些请求在单位时间内从服务器获取的数据长度，计算公式： $\text{Total transferred} / \text{Time taken for tests}$ ，这个统计很好的说明服务器的处理能力达到极限时，其出口宽带的需求量。

自学与分享平台：www.besttest.cn 微博：<http://weibo.com/besttest>

webbench

安装：

官网：<http://home.tiscali.cz/~cz210552/webbench.html>

```
tar zxvf webbench-1.5.tar.gz
```

```
cd webbench-1.5
```

```
make && make install
```

使用：

```
webbench -c 500 -t 30 http://127.0.0.1/test.jpg
```

参数说明：-c表示并发数，-t表示时间(秒)

besttest 安大叔自学与分享平台：www.besttest.cn 微博：<http://weibo.com/besttest>

自学与分享平台 : www.besttest.cn 微博 : <http://weibo.com/besttest>

webbench

测试结果示例:

```
webbench -c 100 -t 1
```

```
"http://10.11.204.140/sugg?yyid=abvdd&assn=1&sp=%E6%90%BA%E7%A8%8B&key=xiecheng&can=%E6%90%BA%E7%A8%8B;%E5%86%99%E6%88%90;%E9%9E%8B%E5%9F%8E;%E6%96%9C%E6%92%91;%E8%B0%A2%E6%88%90"
```

Webbench - Simple Web Benchmark 1.5

Copyright (c) Radim Kolar 1997-2004, GPL Open Source Software.

Benchmarking: GET

```
http://10.11.204.140/sugg?yyid=abvdd&assn=1&sp=%E6%90%BA%E7%A8%8B&key=xiecheng&can=%E6%90%BA%E7%A8%8B;%E5%86%99%E6%88%90;%E9%9E%8B%E5%9F%8E;%E6%96%9C%E6%92%91;%E8%B0%A2%E6%88%90
```

100 clients, running 1 sec.

Speed=228839 pages/min, 19304320 bytes/sec.

Requests: 3814 succeed, 0 failed.

分析: 每分钟228839pages, 每秒钟传输数据量19304320 bytes/sec.

besttest 安大叔自学与分享平台 : www.besttest.cn 微博 : <http://weibo.com/besttest>

自学与分享平台：www.besttest.cn 微博：<http://weibo.com/besttest>

感谢您的支持



安大叔

QQ群：49526682

新浪微博：<http://weibo.com/TestAndashu>

腾讯微博：<http://t.qq.com/w223ewe>

besttest 安大叔自学与分享平台：www.besttest.cn 微博：<http://weibo.com/besttest>