今天的年会已过，仍然是空手而归，不过俺坚信能让生活稳定永远都是努力。由于隔壁组负责年会的发红包项目，因而趁此机会把通过工具模拟高并发的知识补了补，通过和身边大师的交流，总算是对压力测试有了个简要的了解，尤其是熟悉JMeter的使用（之前还想过自己写个多线程客户端，被大师说重复造轮子不可取）。

Apache JMeter是Apache组织开发的基于Java的压力测试工具，是非常简洁有效的选择，页面使用swing构建。支持很多类型的测试，包括最常见的Web(Http/Https)，FTP等，尽管是JAVA语言写成的， 仍然可以很好地在.NET平台使用，只需要在对应的压测机上安装相应版本的JRE即可。对于Web开发者而言，主要用到的就是页面的调用和Web服务的调用，这也是介绍的重点。

Jmeter下载页面: <http://jmeter.apache.org/download_jmeter.cgi>

所需环境Jdk下载页面: http://www.oracle.com/technetwork/java/javase/downloads/index-jsp-138363.html

程序执行

jmeter.bat – 默认的运行JMeter

jmeterw.cmd – 在不包含控制台的情况下运行JMeter的GUI页面

jmeter-n.cmd – 通过一个JMX file来运行，省略掉冗长的配置

jmeter-server.bat – 运行JMeter在服务器模式下

服务模式

针对分布式测试，JMeter需要通过服务器模式运行在远程节点上，之后通过GUI界面控制服务器，当然也可以使用没有图形界面的模式运行远程测试，可以通过运行jmeter-server.bat在服务器主机上运行Jmeter服务。这儿注意的是，需要指定防火墙和代理服务器的相关信息。

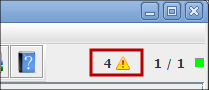
-H 代理服务器主机名或者IP地址

-P 代理服务器端口

例子: jmeter-server -H my.proxy.server -P 8000， 如果需要在单个测试运行完毕后关闭服务，可以设置参数， server.exitaftertest=true。

日志和错误信息

JMeter的GUI不会提供错误信息的弹出框，而是将错误信息写入到日志文件，该文件的名称定义在jmeter.properties配置文件，可以通过菜单栏中的 Options > Log Viewer 来现在日志信息，此外错误信息的个数会显示在工具的右上角，一个简短的日志信息如下所示。

[](http://jmeter.apache.org/images/screenshots/log_errors_counter.png)

|  |
| --- |
| 10/17/2003 12:19:20 PM INFO - jmeter.JMeter: Version 1.9.20031002  10/17/2003 12:19:45 PM INFO - jmeter.gui.action.Load: Loading file: D:\jmeter\customConfig\BSH.jmx  10/17/2003 12:19:52 PM INFO - jmeter.engine.StandardJMeterEngine: Running the test!  10/17/2003 12:19:52 PM INFO - jmeter.engine.StandardJMeterEngine: Starting 1 threads for group BSH. Ramp up = 1.  10/17/2003 12:19:52 PM INFO - jmeter.engine.StandardJMeterEngine: Continue on error  10/17/2003 12:19:52 PM INFO - jmeter.threads.JMeterThread: Thread BSH1-1 started  10/17/2003 12:19:52 PM INFO - jmeter.threads.JMeterThread: Thread BSH1-1 is done  10/17/2003 12:19:52 PM INFO - jmeter.engine.StandardJMeterEngine: Test has ended  The log file can be helpful in determining the cause of an error, as JMeter does not interrupt a test to display an error dialogue. |

* 配置JMeter

Jmeter配置文件的修改比较合理的方式是通过user.properties来进行个性化配置，包括设置ssl.provider、xml.parser、**remote\_hosts**等，简单应用的话默认值即可。

此外，jMeter还提供**测试计划模板**，当对基础知识有了一定理解的情况下，使用模板可以非常高效的构建测试计划。

JMeter官方文档地址：<http://jmeter.apache.org/usermanual/index.html>

页面功能测试说到底就是模拟用户浏览点击页面的全过程，很多的测试工具都可以对该过程进行录制后模拟用户操作，而压力测试就是将这个过程在单位时间内重复成千上万次，看检测应用的高可用，接下来就来一步步的构建我们的页面测试计划吧。比如说我们现在有10个用户，每人请求2个页面并且请求2次，总请求次数为10 \* 2 \* 2 = 40。

第一步，创建模拟用户，实际上就是线程。（ Test Plan（右键） –> 添加 -> Threads -> 线程组 ），在该页面中可以修改线程组名，在该场景下，将线程数设置为10、循环次数设置为2、ramp-up period设置为1秒（10个线程/1秒 = 10线程每秒，即每秒创建10个线程，设置为0则直接创建）。

（之后相类似截图省略）

第二步，添加默认的http请求属性，在之前创建的线程组User（右键） -> 添加 -> 配置元件 -> Http请求默认值，这个默认值会在所有的http请求中使用，最常见的设置是请求的目标服务器名称或IP，如本机localhost。

第三步，添加cookie的支持，线程组User（右键） -> 添加 -> 配置元件 -> cookie管理器，我们知道Cookie中主要用于存放用户的相关信息，尤其是登录信息，如果不能正确的设置这部分内容，我们请求页面可能会一直把请求跳转到登录页面。

第四步，添加Http请求，这是整个操作的核心，线程组User（右键） -> 添加 -> Sampler –> HTTP请求。比如请求的页面为Home页，路径Path设置为”/”表示根路径，接着再另外添加一个页面及其路径，比如”/Product”。 比如针对.NET的Form认证模式，可以通过POST用户名密码到login页面来完成相应的操作。

第五步，最后为请求对应添加监听器用于获取测试结果，线程组User（右键） -> 添加 -> 监听器 –> 图形化结果，只需要指定好文件路径和文件名即可。

补充:

处理通过URL重写的用户会话：如果web应用使用URL重写来代替Cookie保存Session信息的场景下，需要一些额外的操作，使用适当的HTTP URL重写器去完成这部分工作。简单的输入sessionID参数到重写器，它会将该内容添加到每一个请求，如果请求已经有该值，则替换，如果检测是否缓存了Session ID，那么最近的sessionID将被保存并被之前不包含sessionID的请求使用。这而为了控制这个重写器的工作范围，将在该用户（线程）组下建立一个简单控制器，线程组User（右键） -> 添加 -> 逻辑控制器 -> 简单控制器， 之后在其中添加一个URL重写器，这个想想也能知道，这应该属于一个前置的拦截器，因而菜单项为, 简单控制器（右键）-> 前置处理器 -> HTTP URL重写操作符，之后设置会话参数名称即可。

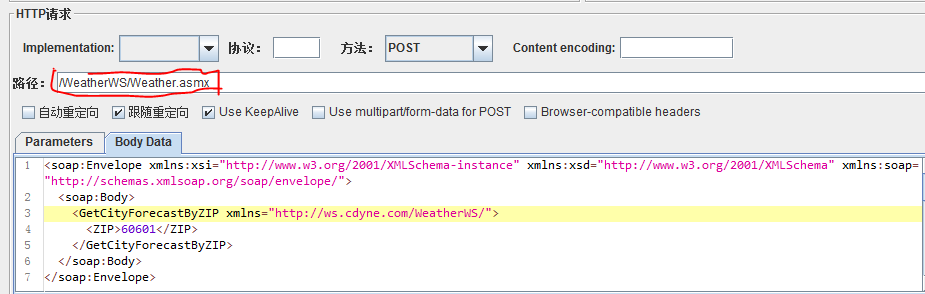


Http头管理器(Header Manager): 这部分内容在Web服务时会使用的比较多，应用服务的验签一般是通过请求头中存放token来处理的。

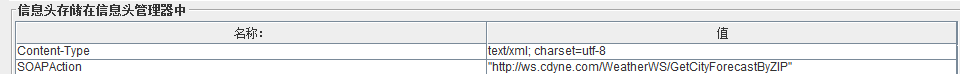
虽然SOAP协议现在已经不再是搭建公共服务的主流，但毕竟大部分的SOA项目还是使用该协议，因此还是对它进行简要的介绍，这儿的例子用到之前提到的及其方便的模板， 文件-> Template模板， 选择”Building a SOAP Webservice Test Plan”，创建后的步骤如下所示。

Step1: 在HTTP请求默认参数页或用户定义变量页（不同版本有细微差异）设置指定服务器名或IP。

Step2: 在用户组的Soap请求页，修改Path



Step3: 在该请求下的HTTP请求头管理器中指定SOAPAction（仅用于.NET WebService，其他情况下删除，而JAVA的相关服务包括Weblogic、Axis、JWSDP等）



Step4: 修改之前请求中的Body data即可，一个简单的服务压测就完成了，是不是很棒，哈哈。

**Restful Service**

这是现在最常见的服务形式，整个操作和以前基本一致，只需要做部分细微修改即可。在HTTP请求页面中，将httpMethod修改为所需方法，Body data设置为指定的数据格式，如Json。最后在Http头管理中将Content-Type选择为指定的数据格式即可，比如application/json。

之前介绍Jmeter的demo应用，现在来介绍其在实际应用中的一些重要知识，首先通过几个问题引入当前的话题。单位时间内平均的用户数量(并发数)是多少?用户的峰值数量是多少？什么时候是比较合适的压力测试的时机（在没有相应替补生产环境的情况下，考虑如何达到接近真实测试的目的）？我们的web应用程序是有状态么，如果有，应用程序是如何管理诸如cookies、session-rewriting等状态对象的？我们的测试目标是什么？

为了达到我们的测试目标，首先需要考虑的搭建测试环境时，申请到相关的资源，比如防火墙、代理服务器等网络问题，CPU的核数、内存的大小等硬件资源问题。此外，考虑使用什么样的操作系统，比如像windows系统其自身就会暂用很多的资源，可能无法模拟出你所需的并发数。这些东西在实际的项目中显得非常重要，因为这些可能不是一个人技术好就可以决定的，需要充分的沟通和合作。如果你需要很多的机器来测试网络延迟，也可以考虑使用云的方式，Jmeter支持商业云PAAS来运行基准测试和压力测试，需要注意的是，在压力测试时，需要运行Jmeter的非GUI版本，GUI版本只是用于创建测试计划和Debug。

这儿只是对Jmeter进行了一个非常简要的介绍，若有不足之处，请见谅！

此外也推荐大家去看看小坦克大师的博文：<http://www.cnblogs.com/TankXiao/p/4045439.html>，内容非常详实