1、下载Jmeter  
　　下载地址：http://jmeter.apache.org/download\_jmeter.cgi  
　　目前最新版为2.9，其余文件如源代码等也可从如下官网下载：  
　　http://jmeter.apache.org/download\_jmeter.cgi  
　　2、安装Jmeter之前  
　　安装Jmeter之前需要先配置Java[**环境**](javascript:;)，我们下载的是jmeter2.9，所以java版本最好是选用java6以后的版本。  
　　安装JDk1.6的步骤如下：  
　　点击下载的JDK文件-dk-6u18-windows-i586.exe，点击下一步直至安装完成，然后开始配置环境：  
　　点击我的电脑----属性----高级----环境变量----在系统变量中----点击新建，  
　　在变量名中输入：JAVA\_HOME  
　　变量值中输入：C:Program Files (x86)Javajdk1.6.0\_18  
　　再次点击新建：  
　　在变量名中输入：CLASSPATH，  
　　变量值中输入：.;%JAVA\_HOME%/lib/dt.jar;%JAVA\_HOME%/lib/tools.jar;  
　　然后再系统变量中找到Path，点击编辑，在变量值中加上：  
　　%JAVA\_HOME%/bin;%JAVA\_HOME%/jre/bin;（如果前面没有“;”需要加上）  
　　配置完成之后，点击确定保存，然后cmd打开命令窗口输入：java或者javac，出现大串字符串表示配置成功。  
　　也可以输入：java -version 查看java版本来查看是否配置成功。  
　　OK，JDK安装成功之后，就可以下一步安装Jmeter了。  
　　3、安装Jmeter  
　　解压apache-jmeter-2.9.zip文件至目录，我的是D:Program Files目录。  
　　点击我的电脑----属性----高级----环境变量----在系统变量中----点击新建，  
　　变量名输入：JMETER\_HOME  
　　变量值输入：D:Program Filesapache-jmeter-2.9  
　　然后编辑CLASSPATH变量，加上%JMETER\_HOME%libextApacheJMeter\_core.jar;%JMETER\_HOME%libjorphan.jar;%JMETER\_HOME%liblogkit-2.0.jar;然后确定即可  
　　操作完上面的步骤，我们就可以点击Jmeter中bin目录下面的jmeter.bat文件即可打开Jmeter了。  
　　注意：打开的时候会有两个窗口，Jmeter的命令窗口和Jmeter的图形操作界面，不可以关闭命令窗口。  
　　二、录制Jmeter脚本  
　　录制Jmeter脚本有两种方法，一种是设置代理；一种则是利用badboy软件，badboy软件支持导出jmx脚本。  
　　这里我们介绍第二种方法，利用badboy录制脚本，然后导出Jmeter需要的jmx文件。  
　　首先，在地址栏中输入要录制脚本的地址，然后点击绿色按钮即可开始录制（工具栏中的红色按钮也可以点击进行录制），点击红色圆形按钮旁边的正方形黑色按钮结束脚本录制（如果使用过loadrunner，那么步骤是一样的），在内嵌的浏览器上面操作你需要测试的动作即可，它会自己记录你的相关行为动作。  
　　然后点击黑色按钮录制完成之后，我们可以导出jmx文件：  
　　保存为“登录开发者.jmx”。

三、使用Jmeter开始测试  
　　1、打开脚本  
　　用Jmeter打开我们上面保存的文件。  
　　打开之后如图所示：  
　　2、修改脚本  
　　a、线程组  
　　在测试计划里面选择添加既可以看到线程组。  
　　如上图：  
　　名称：可以随意取，我们保持创建是自动分配的名称  
　　注释：随意  
　　在取样器错误后要执行的动作：我们选择继续，错误之后依然继续执行  
　　线程数：一般我们用来表示多少个用户，即我们测试时的用户数量  
　　Ramp-up Period(in Seconds)：表示每个用户启动的延迟时间，上述我设为1秒，表示系统将在1秒结束前启动我设置的1000个用户，如果设置为1000秒，那么系统将会在1000秒结束前启动这1000个用户，开始用户的延迟为1秒， 如果我设置为0秒，则表示立即启动所有用户。  
　　循环次数：如果你要限定循环次数为10次的话，可以取消永远的那个勾，然后在后面的文本框里面填写10；在这里我们勾上永远，表示如果不停止或者限定时间将会一直执行下去， 是为了方便调度器的调用。  
　　调度器的配置：我们勾选调度器时，将会出现这个面板  
　　启动时间：表示我们脚本开始启动的时间，当你不想立即启动脚本测试，但是启动脚本的时间不会再电脑旁的时候，你可以设定一个启动的时间，然后再运行那里点击启动，系统将不会立即运行，而是会等到你填写的时间才开始运行。  
　　结束时间：与启动时间对应，表示脚本结束运行的时间。  
　　持续时间：表示脚本持续运行的时间，以秒为单位，比如如果你要让用户持续不断登录1个小时，你可以在文本框中填写3600。如果在1小时以内，结束时间已经到达，它将会覆盖结束时间，继续执行。  
　　启动延迟：表示脚本延迟启动的时间，在点击启动后，如果启动时间已经到达，但是还没有到启动延迟的时间，那么，启动延迟将会覆盖启动时间，等到启动延迟的时间到达后，再运行系统。  
　　注意：如果我们需要用到调度器来设定持续时间，如果线程数不够多到持续时间结束，我们就必须将循环次数勾选为永远，特别地，如果线程组里面有其他的循环，我们也需将该循环次数勾选为永远（如我上面录制的脚本中的Step1也是一个循环，需要将永远勾选），否则，按我如上配置，将永远去掉勾选，文本里填1，那么无论你将持续时间启动时间结束时间等设置多少，系统运行1000次后，将会停止不再运行。  
　　b、关于HTTP请求  
　　关于HTTP请求，我们录制的脚本里面就有，这里就介绍一些常用的属性。  
　　如图所示：  
　　点击HTTP请求，将会出现如下界面：  
　　其中：  
　　名称：可以随意取，我们可以根据录制脚本的路径来命名  
　　注释：可以对该界面做一个简单介绍  
　　服务器名称或IP：即我们的服务器或者IP的地址，我因为是在本机上的项目，故填写localhost，可以填写127.0.0.1。  
　　端口号：我用的是tomcat，自己设置的端口号为8086（tomcat默认为8080）  
　　后面的超时定义可以不用填写。  
　　Implementation：这里我们录制时默认填写为JAVA  
　　协议填写为：http  
　　方法为：GET  
　　Content encoding：编码可以不用填写  
　　路径：即页面的路径；下面的重定向等选择可以根据需要勾选，这里我们保持录制不修改。  
　　Parameters参数：即跟着路径一起发送的参数及文件  
　　下面的代理服务器等可以忽略，按需要填写。  
　　c、参数化  
　　如下，我们为请求发送的参数来参数化。  
　　上述的例子为登录的用例，登录有两个参数：EMAIL和DEVELOPER\_PASSWORD，在实际测试中，当我们需要用到不同的用户来登录的时候，我们就可以用到参数化。  
　　1)、我们新建一个文件，这里我新建的是一个txt文件，命名为c.txt，里面的内容为  
　　web@qq.com，111111  
　　col@qq.com，111111  
　　mon@qq.com，111111  
　　2)、在jmeter中的【选项】中选择【函数助手对话框】，将会弹出如下对话框：  
　　其中：  
　　CSV file to get values from | \*alias：要读取的文件路径，为绝对路径  
　　CSV文件列号| next| \*alias：从第几列开始读取，注意第一列是0  
　　如上图所示，我们读取的是c.txt里面的第一列用户名(如果要读取第二列的密码，只需将0改成1即可，往后类推)，点击【生成】按钮即可生成函数，我们使用时即拷贝生成的函数字符串：${\_\_CSVRead(E:c.txt，0)}。  
　　其他函数后面介绍。  
　　d、断言验证  
　　在web测试中，有的时候，即使我们测试返回的response code为200，也不能保证该测试是正确的，这时候，我们可以用到响应断言，通过对比响应的内容来判断返回的页面是否是我们确定要返回的页面。  
　　选择我们要判断的页面，点击添加断言----响应断言，将会出现如下图：  
　　名称：随意，这里我们不做改动  
　　注释：随意  
　　Apply to：这里我们选择默认值  
　　要测试的响应字段：这里我们选择响应的文本，即返回的页面信息  
　　模式匹配规则：这里我们选择包括，即响应的文本是否包括我们验证的信息  
　　要测试的模式：开发者名称，即我们需要验证，开发者名称是否在响应文本中存在，也可以添加更多的字段来验证。  
　　其他还有文件大小和响应时间等的断言，后面介绍。  
　　e、添加监听器  
　　我们做性能测试的时候，经常需要各种数据来验证我们的测试结果，Jmeter里面也提供了相关的功能，这里主要说一下Aggregate Report聚合报告。  
　　点击添加---监听器，选择：查看结果树，和聚合报告，如果需要看到断言信息也可以选择断言结果，将设置的一切保存之后，然后点击启动，系统将会运行，将结果在监听器中表现出来。

　　四、分析结果  
　　1、查看结果树  
　　如图所示：成功的为绿色，失败则显示为红色。如果测试的结果太多，你只需要看到错误的页面，则勾选【仅日志错误】  
　　2、聚合报告(Aggregate Report)  
　　其中：  
　　Label：标签，即我们上面的请求名称  
　　#Samples：本次场景中一共发出了多少个请求  
　　Average：平均响应时间  
　　Median：中位数，也就是50%的用户的响应时间  
　　90%Line：表示90%的用户的响应时间，如果最小值和最大值相差很大的话，我们一般选择这个作为最终测试结果  
　　Min：最小响应时间  
　　Max：最大响应时间  
　　Error%：出错率，本次测试中出现错误的请求的数量/请求的总数  
　　Throughput：吞吐量  
　　KB/sec：每秒从服务器端接受到的数据量  
　　五、监控内存及CPU等  
　　很多时候，我们测试性能都需要查看内存和CPU等信息来判断系统瓶颈，关于CPU和内存的监控，Jmeter并没有很好的支持，很多时候，我们都只能通过系统的资源监控器来观察，一闪而过不会记录下来，很不方便。幸好google开发了一款专门监控的Jmeter插件，弥补了Jmeter这方面的不足，下面就来介绍一些这款插件——JMeterPlugins，目前我用的是0.5.6版本。  
　　1、JMeterPlugins的下载地址  
　　地址：http://code.google.com/p/jmeter-plugins/downloads/list，选择需要的版本，下载  
　　2、JMeterPlugins的配置使用  
　　将下载的文件解压，得到如下目录：  
　　将JMeterPlugins.jar包复制到Jmeter的lib目录下面的ext目录下面，重新启动Jmeter，我们点击添加就可以看到出现了很多的jp@gc-开头的文件.  
　　这里监控内存我们使用的是：jp@gc - PerfMon Metrics Collectot  
　　在使用之前，我们需要运行/serverAgent/startAgent.bat这个文件，我们需要将serverAgent目录及下面的文件复制到我们测试的服务器上，然后点击打开(我这里是本机，直接在本机上面打开这个应用系统即可)，它的默认端口为4444。  
　　一切准备好后，点击启动，即可得到如下图：  
　　你就可以得到系统运行时，你所需要的常用的性能值了。  
　　六、监控内存及CPU等（jconsole）  
　　最近逛论坛的时候，发现了一个比较好的监控内存CPU等的小工具，本着开源小工具多多益善的原则，记录一下。  
　　打开这个小工具的步骤很简单，如果你已经配置好了Jmeter运行的环境，那么你也就不用去做其他的配置，直接 点击：开始——》运行——》输入cmd——》然后在出现的命令行界面输入“jconsole”即可弹出一个【java监视和管理控制台】，  
　　将会弹出如下界面：  
　　这里我们选择本地进程，并点击一下sun.tools.jconsole.JConsole这一行，然后点击连接，就可以查看我们的内存和CPU的使用情况了。  
　　当然，我们要得到服务器的内存使用等信息，也可以选择 本地进程下面的远程进程，在文本框中输入我们需要测试的服务器的IP地址：端口，然后在下面输入用户名和密码，点击连接，看看会出现什么情况。