　　随着市场上推出越来越多的测试工具，大家使用起来也比较盲目，就我个人而言，使用过LoadRunner和Jmeter  
　　很多人倾向于LoadRunner，因为LR的功能更强大，其实工具的选择，不在于工具本身有多强大，取决于我们的实际需求  
　　我接触的项目基本都是web应用，从性能测试需求的角度来说，性能测试工具需要具有以下一些特点  
　　1、支持HTTP/HTTPS协议，能够发送HTTP/HTTPS请求  
　　2、可以实现不同的用户场景，能够根据实际业务模拟场景，对用户的比例进行合理分配  
　　3、支持事务；  
　　4、支持参数化和关联；  
　　5、可以生成报告；  
　　既然Jmeter满足以上特性，使用特别灵活，又是免费的，所以我很明显的偏向了JMeter  
　　以下是整理的相关jmeter的基础介绍和一些术语  
**1.基本介绍**  
　　Jmeter是Apache组织的开放源代码项目，是功能和性能测试的工具，它是100%的Java应用。Jmeter提供的操作界面支持多种配置和结果显示。Jmeter能够模拟对**[服务器](javascript:;" \t "_self)**或者网络系统逐步加负载（压测包括HTTP/FTP/JDBC/JAVA等）。Jmeter支持多线程并发取样和单线程组对不同的功能同时取样。Jmeter也可以作为Web服务器与浏览器之间的代理网关，以便捕获浏览器的请求和Web服务器的响应。  
**Jmeter包含4部分：**  
　　负载发生器：用于产生负载，通常以多线程或多进程的方式模拟用户行为。  
　　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。  
　　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。  
　　报表生成器：根据测试分析结果生成报表，提供可视化的数据展现方式  
**2.基础元件介绍**  
　　测试计划是测试计划是性能测试的起点，其它元件都需要依赖测试计划，是所有元件的容器  
**2.1 线程Threads(Users)**  
　　2.1.1线程组  
　　(1) 取样错误后要执行的动作  
　　a.继续：默认选项，出错后后续的线程继续运行  
　　b.Start Next Thread Loop：出错后启动下一组线程运行  
　　c.停止线程：后续的线程停止运行，但是会返回出错前的结果  
　　d.停止测试：与停止线程的差异在于不返回出错前的结果  
　　(2)线程数：虚拟用户数，代表发送请求的用户数，一个线程数占用一个进程或线程。  
　　(3)Ramp-Up Period(in seconds)-准备时长  
　　-所有线程在多少时间内启动，单位是秒，为0时代表一次请求即并发。如果线程数为5，Ramp-Up Period为10  
　　(4)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 请求数是2000  
　　(5)Delay Thread Creation until needed：直到需要时延迟线程的创建  
　　(6)调度器：设定测试运行时间  
**2.2配置元件**  
　　2.2.1 CSV Data Set Config  
　　参数化配置，一般用在数据驱动，通过文件来存储测试数据。调用时也支持参数化一部分数据。Jmeter参数化方式是一次读入一行，分割后存入若干变量中交给一个线程处理。  
　　2.2.2 HTTP信息头管理器  
　　2.2.3 HTTP Cookie管理器  
　　2.2.4 HTTP COOKIE Manager管理cookie  
　　每个Jmeter线程都有自己的session  
**2.3前置处理器**  
　　全局的参数设置，设置后每次请求前读取的值，比如地址、端口、token  
**2.4后置处理器(Post Processors)**  
　　正则表达式提取器可以从请求的响应结果中提取需要的内容，从而实现关联  
　　(1)引用名称：外部引用时使用  
　　(2)正则表达式：提取的内容需要用()括起来。  
　　(3)模板：使用$$引用起来 ，值从1开始  
　　(4)匹配数据（0代表随机）：0代表随机，-1代表所有，其余正整数代表将在已提取的内容中匹配第 几个内容  
　　(5)缺省值：正则匹配失败时，取的值。  
**2.5Sampler（取样器）**  
　　sampler是与服务器进行交互的单元  
　　(1)HTTP请求  
　　(2)Web服务器  
　　(3)Http请求  
　　(4)Parameters参数  
**2.6监听器**  
　　2.6.1 聚合报告  
　　聚合报告的相关指标：  
　　(1)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。  
　　(2) #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。  
　　(3) Average：平均响应时间，默认情况下是单个 Request 的平均响应时间，使用了事务控制器后表示为整个事务的平均响应时间。  
　　(4) Median： 50 ％用户的响应时间。  
　　(5) 90% Line： 90 ％用户的响应时间。  
　　6) Min：最小响应时间。  
　　(7) Max：最大响应时间。  
　　(8) Error%：错误率，本次测试中出现错误的请求的数量 / 请求的总数。  
　　(9) 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。  
　　(10) KB/Sec：每秒从服务器端接收到的数据量。  
　　2.6.2 查看结果树  
　　(1)Thread Name: 线程组名称  
　　(2)Sample Start: 启动开始时间  
　　(3)Load time: 加载时长  
　　(4)Latency: 等待时长  
　　(5)Size in bytes: 发送的数据总大小  
　　(6)Headers size in bytes: 发送头大小  
　　(7)Body size in bytes: 发送数据的其余部分大小  
　　(8)Sample Count: 发送统计  
　　(9)Error Count: 交互错误统计  
　　(10)Response code: 返回码  
　　(11)Response message: 返回信息  
　　(12)Response headers: 返回的头部信息  
**2.7断言**  
　　确定Jmeter有没有按照预期工作，应当考虑使Jmeter提供的断言，加到Sampler 里面可以对返回的结果进行判断，例如判断HTTP返回结果里面是否含有某个字符串  
　　附上一张jmeter的界面截图

