Jmeter接口自动化测试方法简介

一、 思路简洁

1. 了解待测接口参数规范，具体参考wiki，明确get参数和post参数，是否需要验证cookie、ua等
2. Jmeter参数化方式配置请求host、url、header消息头等
3. 配置csv文件，编写测试用例参数和预期结果格式校验
4. 根据需要编写beanshell脚本或导入辅助性jar包，用于解析接口返回结果，比如解密数据
5. 在Jmeter中添加必要的断言或监听器，用于收集用例执行的结果
6. 执行测试，查看用例结果，重点分析Fail的用例，和开发沟通，上报bug

二、 一个简单的性能测试

**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。

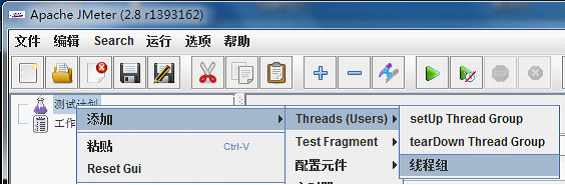
**建立测试计划**

 　　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。

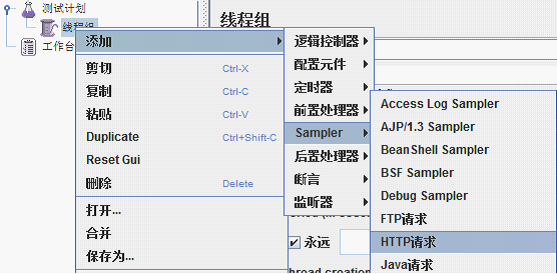
**添加HTTP请求**

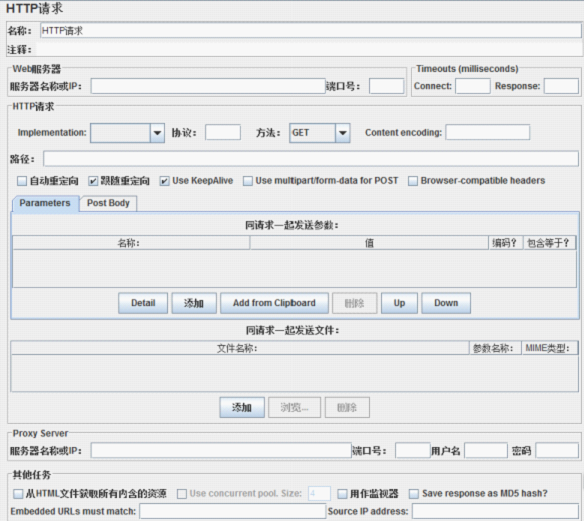
　　添加完成线程组后，在线程组上右键菜单（添加--->Sampler--->HTTP请求）选择HTTP请求。对于jmeter来说，取样器（Sampler）是与服务器进行交互的单元。一个取样器通常进行三部分的工作：

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**

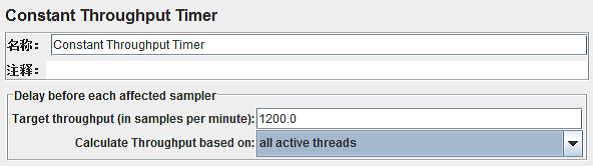
　　本次性能测试的需求中提到测试的目的是“了解博客的首页在负载达到20 QPS时的响应时间”，因此需要控制向博客首页发送请求的负载为20QPS。

　　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：

