

jmeter+dubbo压测说明

Jmeter+dubbo压测工具使用说明

[jmeter-plugins-dubbo-1.2.7-SNAPSHOT-jar-with-dependencies.jar](#)

说明书

版本 0.1

项目名称：

修订历史

版本号	作者	修订章节	修订原因	修订日期
0.1	甘建新	初稿	初稿	2018-06-05

目 录

Jmeter+dubbo压测工具使用说明

1 概述

1.1 什么是jemter?

2 压测要求

3 使用说明

3.1 下载jmeter

3.2 设置JAVA_HOME和堆内存

3.3 下载插件

3.4 插件安装

3.5 接口包依赖添加方式

3.5.1 1.3.x版本

3.5.2 1.2.x版本

3.6 脚本准备

3.6.1 线程组

3.6.2 添加Sampler

3.6.3 参数化处理

3.7 服务器资源情况查看

3.8 报表数据查看

3.8.1 查看结果树

3.8.2 聚合报告解析

3.9 注意事项

4 参考资料

概述

什么是jemter?

Apache JMeter™应用程序是开源软件，一种100%纯Java应用程序，用于加载测试功能行为并测量性能。它最初设计用于测试Web应用程序，但后来扩展到其他测试功能。

压测要求

对于接口压测，我们主要关注以下三点：

- 脚本准备
- 服务器资源监测
- 报表数据查看

使用说明

下载jmeter

支持Jmeter版本：3.0+；

下载地址：https://jmeter.apache.org/download_jmeter.cgi

设置JAVA_HOME和堆内存

```
文件jmeter.bat
set JAVA_HOME=D:\Program Files\Java\jdk1.7.0_79
set PATH=D:\Program Files\Java\jdk1.7.0_79\bin;%PATH%
echo %JAVA_HOME%
echo %Path%
set HEAP=-Xms512m -Xmx1024m
set NEW=-XX:NewSize=128m -XX:MaxNewSize=512m
```

下载插件

下载地址：<https://github.com/dubbo/jmeter-plugins-dubbo/tree/master/dist>

1.2.X下载地址，

<https://github.com/dubbo/jmeter-plugins-dubbo/blob/master/dist/jmeter-plugins-dubbo-1.2.7-SNAPSHOT-jar-with-dependencies.jar>

jmeter-plugins-dubbo-1.2.7-SNAPSHOT-jar-with-dependencies.jar包含所有依赖，推荐使用这个包。

对于插件版本分为1.2.X和1.3.X版本，区别在于1.3.X版本不用添加接口包，而1.2.X需要添加接口及其依赖的包。我们仍然推荐使用1.2.X版本，因为1.3.X在某种情况下会无法使用。

问题一：1.3.X在什么情况下无法使用？

比如：当一个类没有默认构造函数时，无法反序列化，比如我们的Money中引用了java.util.Currency。

问题二：1.3.X为什么不用引用接口依赖包？

在dubbo中有一个功能叫泛化引用，有兴趣的同学可以去了解下。参见：

<http://dubbo.apache.org/books/dubbo-user-book/demos/generic-reference.html>

插件安装

将插件包放入Jmeter的lib的ext下。

\$(Path)\apache-jmeter-3.0\lib\ext

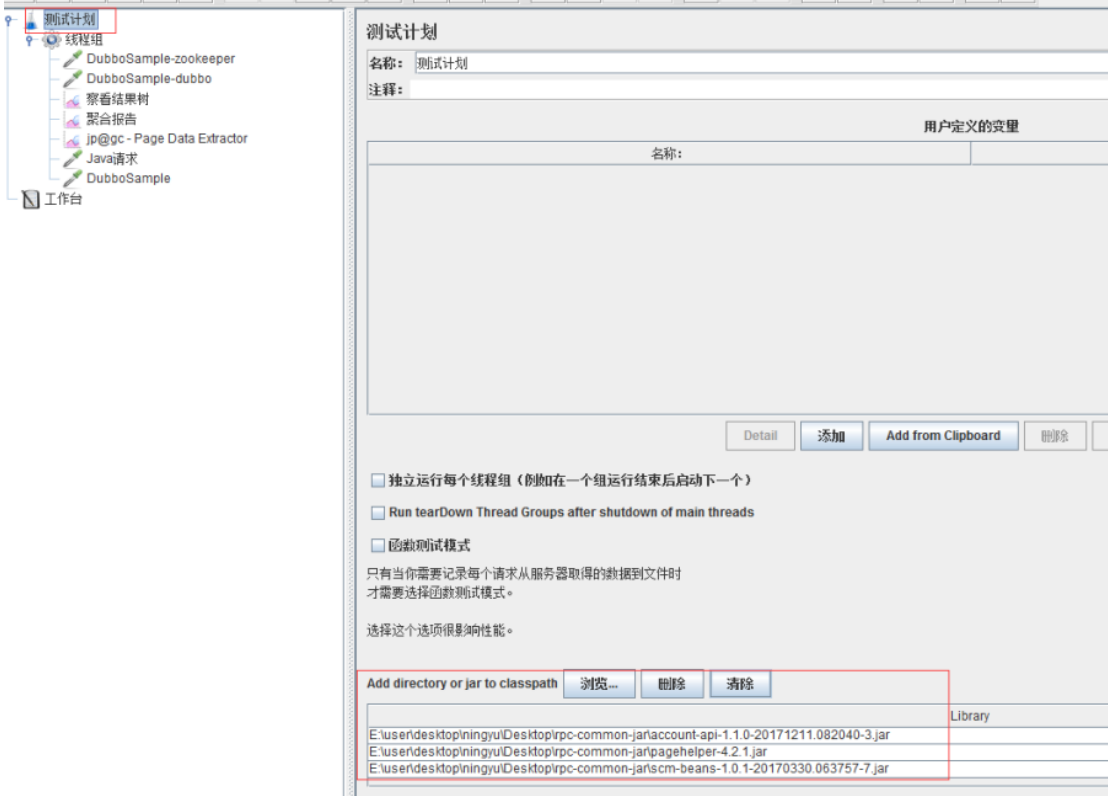
接口包依赖添加方式

1.3.x版本

不需要再添加接口包（api-jar）到classpath下。

1.2.x版本

接口包及接口包依赖的其他jar包请添加到classpath下或放在apache-jmeter-3.0\lib\ext下，也可以通过下图方式添加：



脚本准备

线程组

线程组有很多种，以常用的一种为例



线程组

名称:	模拟并发压力
注释:	
在取样器错误后要执行的动作	
<input checked="" type="radio"/> 继续 <input type="radio"/> Start Next Thread Loop <input type="radio"/> 停止线程 <input type="radio"/> 停止测试 <input type="radio"/> Stop Test Now	
线程属性	
线程数:	6000
Ramp-Up Period (in seconds):	60
循环次数	<input type="checkbox"/> 永远 1
<input type="checkbox"/> Delay Thread creation until needed	
<input type="checkbox"/> 调度器	

线程组主要包含三个参数：

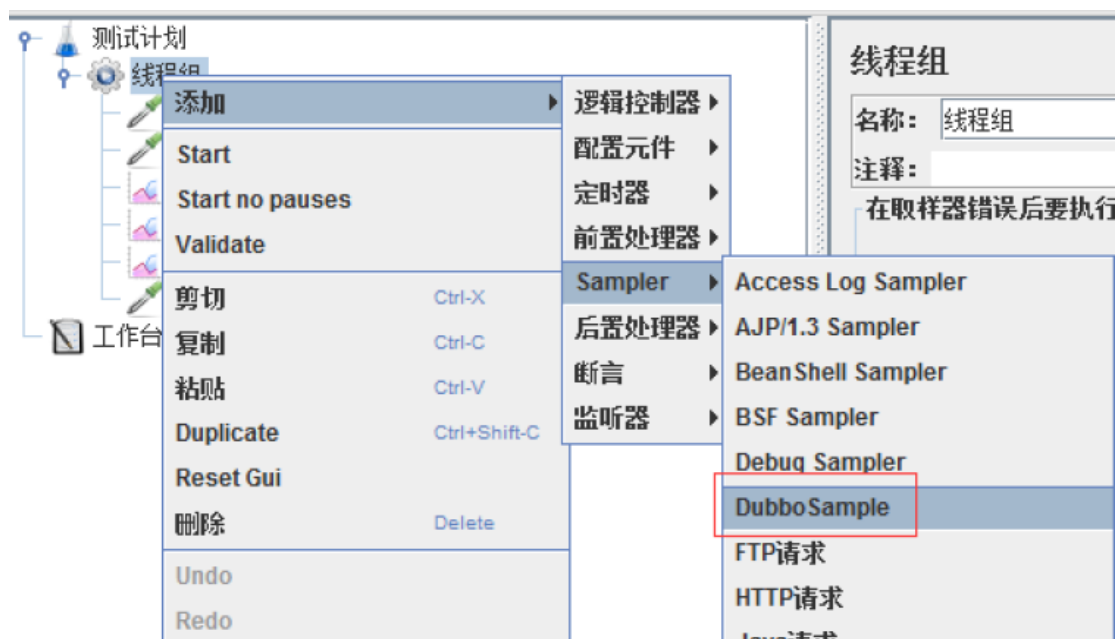
线程数：虚拟用户的数量；

Ramp—Up Period(in seconds)：准备时长。设置的线程数需要多久全部启动，比如上图，线程数为6000，启动时间为60，那么需要60S内启动6000个线程；

循环次数：每个线程发送请求的次数。

还可以通过勾选调度器，设置持续时间或开始和结束时间来控制。

添加Sampler



DubboSample

名称: DubboSample

注释:

Server Settings

Protocol: zookeeper ← 选择协议: zookeeper或dubbo

Address: P: PORT ← 使用zookeeper协议, 填写zk地址, 集群地址用"/"分隔

Timeout: 1200000 Version: 1.0.0 Retries: 0 Cluster: failfast

Interface Settings

Interface: 接口完全名称, 例如: com.xxx.api.UserService ← 接口类完全名称, 含包名

Method: 方法名称: getUserById ← 方法名称

paramType	paramValue
参数类型: java.lang.Long ← 参数类型完全类名, 含包名	1
参数类型: long, 小long类型 ← 基础类型直接填写: int、float、double、short、long、byte、boolean、char	
com.xxx.vo.UserVo	{"startCreateTime":"2017-04-07 13:29:00","endCreateTime":"2017-04-11 18:11:57"}

Args: 复杂参数值使用json格式

增加 删除

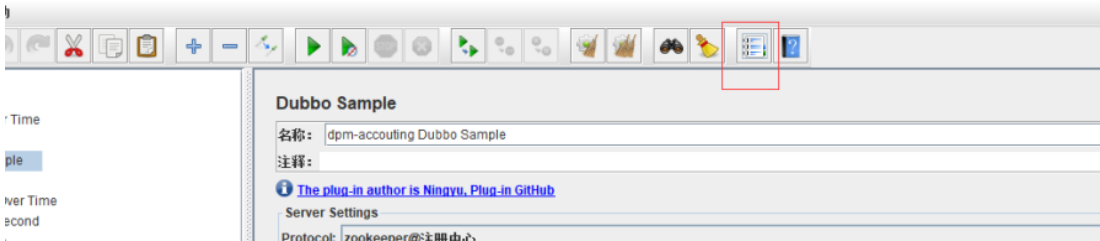
根据上图提示传入值即可。

参数化处理

通常在接口中会有些数据是动态的, 有两种形式:
一种函数生成的随机数据或固定数据;
另外一种是一些固定的业务数据。

第一种, 函数生成的随机数据或固定数据

打开函数助手, 选择要使用的函数



时间函数

函数助手

选择一个功能: time 帮助

函数参数

名称:	值
Format string for SimpleDateFormat (optional)	yyyyMMddHHmmssSSS
Name of variable in which to store the result (optional)	

Detail 添加 Add from Clipboard 删除

拷贝并粘贴函数字符串: `$[_time(yyyyMMddHHmmssSSS,)]` 生成

将生成的函数字符串拷贝到脚本中直接使用即可。

随机字符串函数

函数助手

选择一个功能

RandomString

帮助

函数参数

名称:	值
Random string length	10
Chars to use for random string generation	1234567890
Name of variable in which to store the result (optio...	

Detail

添加

Add from Clipboard

删除

拷贝并粘贴函数字符串


`$_RandomString(10,1234567890,)`

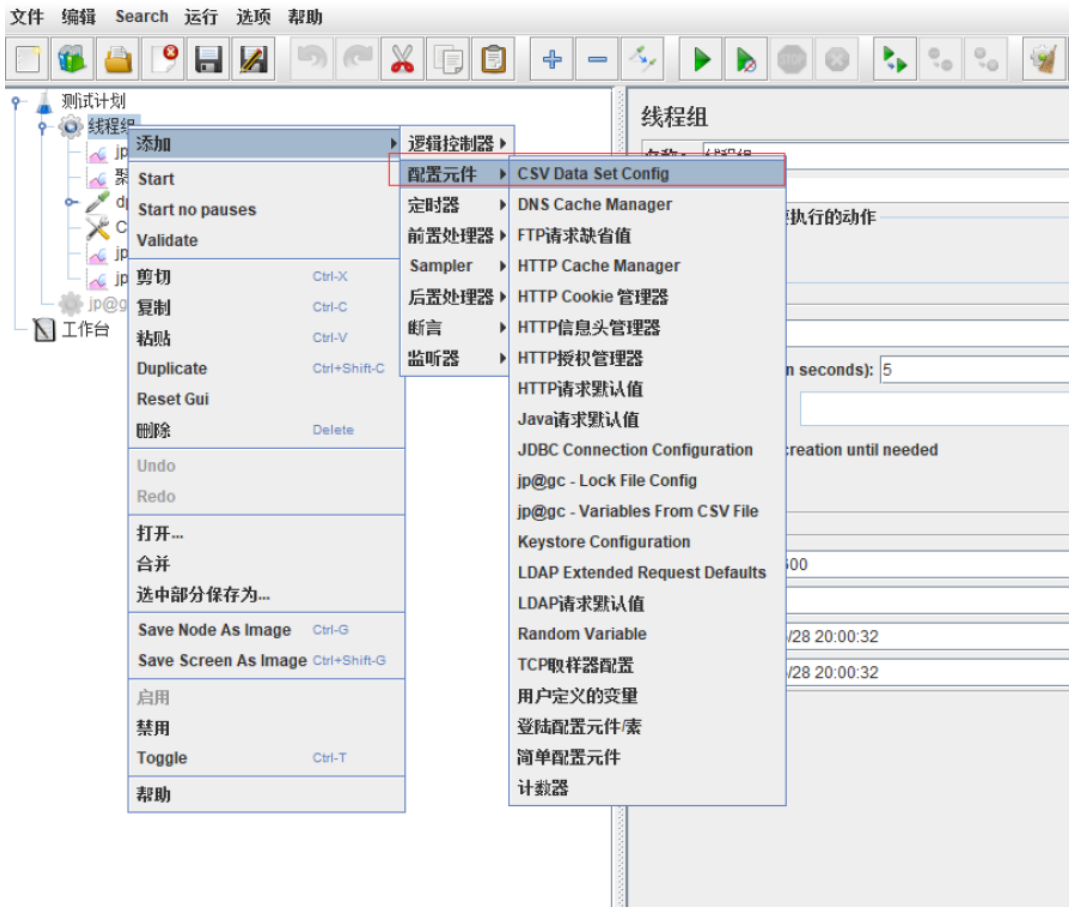
生成

将生成的函数字符串拷贝到脚本中直接使用即可。

另外一种是一些固定的业务数据

对于此类数据我们通常预存在文本或csv中，在jmeter中右键选择"配置元件"，再选择"CSV Data Set Config"；

 Dubbo Sample.jmx (D:\5-技术研究\jmeter\apache-jmeter-3.1\bin\Dubbo Sample.jmx) - Ap



CSV Data Set Config

名称: CSV Data Set Config

注释:

Configure the CSV Data Source

Filename: D:\1-工作\72-账务系统优化\accountParam.txt

File encoding: UTF-8

Variable Names (comma-delimited): accoutNo1,memberId1,accoutNo2,memberId2

Delimiter (use '\t' for tab):

Allow quoted data?: False

Recycle on EOF?: True

Stop thread on EOF?: False

Sharing mode: All threads

Filename：参数文件名，可以写绝对路径，个人强烈建议采用相对路径，避免脚本迁移时需要修改路径。

File encoding：参数文件的编码格式。推荐选择 UTF-8。

Variable Names：对对应参数文件每列的变量名。类似于 Excel 文件的文件头，起到标示的作用，同时也是后续引用的标识符，建议采用有意义的英文标示。

Delimiter：参数文件分隔符。与参数文件中的分隔符保持一致即可。

Allow quoted data？：是否允许引用数据。默认设置为 false。例如数据样式为："101-005-98536","29357","1","1993575","477948510289","android","45" 时，此处需设置为 true，一般默认为 false 即可。

Recycle on EOF?：是否循环读取参数文件内容。默认设置为 true。设置为 true 时，当已经读取完参数文件内的测试用例数据，还需要继续获取用例数据时，此时会循环读取参数文件数据；设置为 false 时，若已至文件末尾，则不再继续读取测试数据。通常在 线程组的线程数 * 线程组的循环次数 > 参数文件行数时，才需要将此项设置为 true。

Sotp thread on EOF?：当读取到参数文件末尾时，是否停止读取线程。默认为 false。当Recycle on EOF? 设置为 true 时，此项不起任何作用。当且仅当 Recycle on EOF? 为 false 时，此项配置才生效。

若为 true，则在读取到参数文件行末尾时，终止参数文件读取线程。例如：线程组的线程数 * 线程组的循环次数 = 10，参数文件行数 = 7，那么将在第 8 次开始停止线程。

若为 false，此时线程会继续读取，但是会请求错误，因此时读取的数据为 EOF。以上同例，自第 8 次开始，线程的请求数据为 EOF。

Sharing mode：共享模式，即参数文件变量作用域。主要有以下几种方式：

All threads：当前测试计划中的所有线程组中的所有的线程均有效。默认。

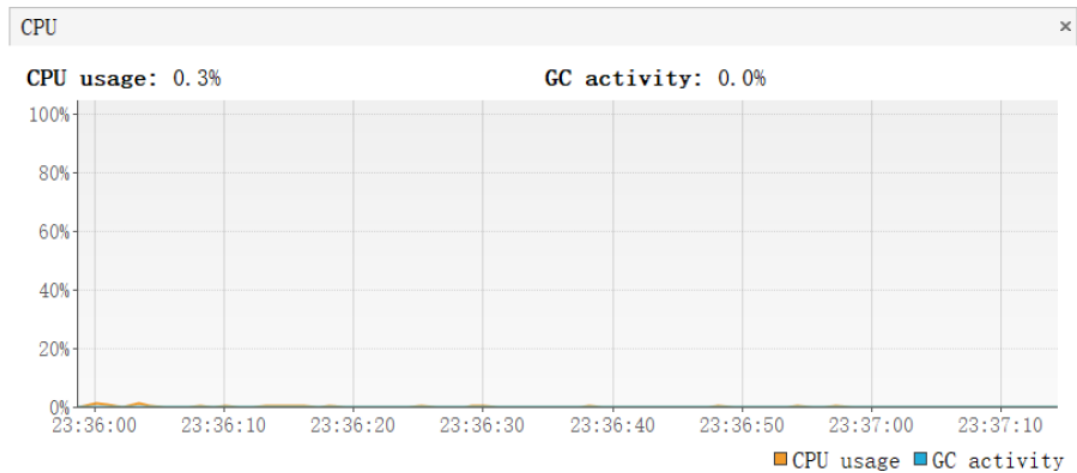
Current thread group：当前的线程组中的线程有效。

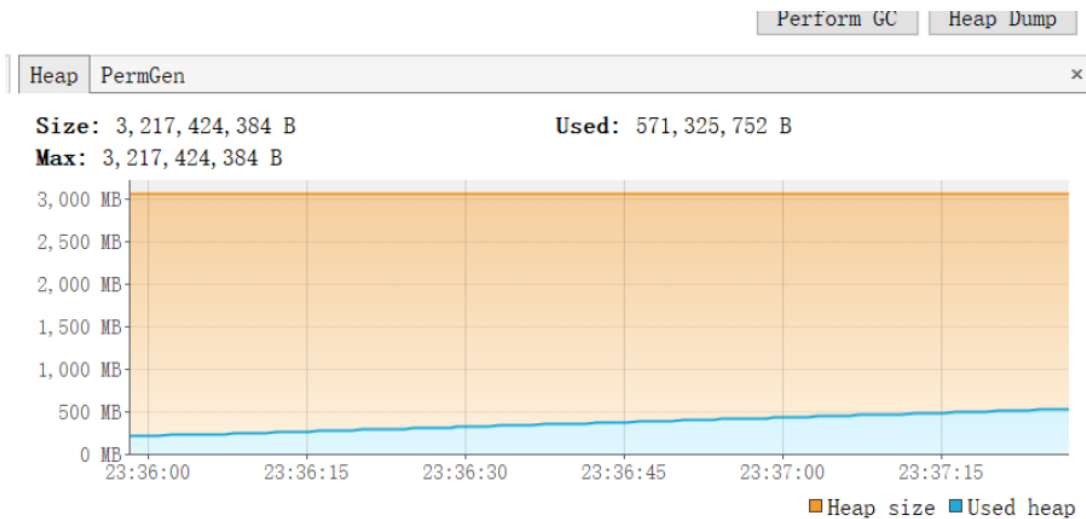
Current thread：当前线程有效。

服务器资源情况查看

可通过jmx或命令监控服务器的CPU，内存，IO等指标。
以下示例：

Uptime: 126 hrs 49 min 35 sec

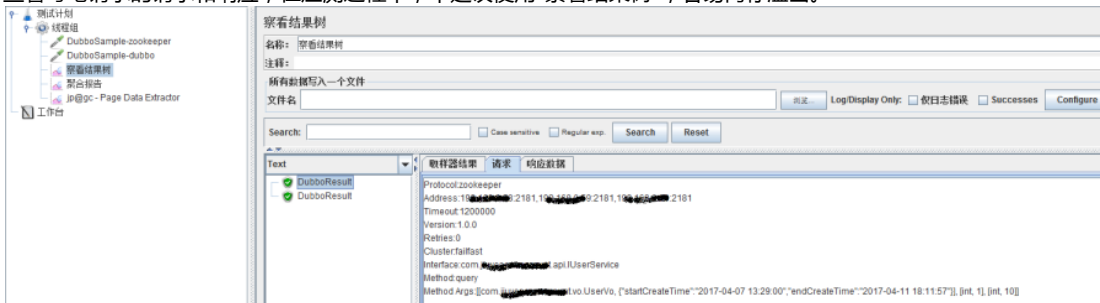




报表数据查看

查看结果树

查看每笔请求的请求和响应，在压测过程中，不建议使用"察看结果树"，容易内存溢出。



聚合报告解析

聚合报告

名称: 聚合报告

注释:

所有数据写入一个文件

文件名: [] [浏览] Log/Display Only: ☐ 仅日志错误 ☐ Successes [Configure]

Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Max	Error %	Throughput	KB/sec
总体	0	0	0	0	0	0	92233720...	-92233720...	0.00%	.0/hour	.0

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，如果模拟10个用户，每个用户迭代10次，那么这里显示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 Transaction Controller 时，也可以以 Transaction 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%: 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)，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

注意事项

- 当使用 zk，address 填入 zk 地址 (集群地址使用“,”分隔)，使用 dubbo 直连，address 填写直连地址和服务端口
- timeout: 服务方法调用超时时间(毫秒)

- version：服务版本，与服务提供者的版本一致
- retries：远程服务调用重试次数，不包括第一次调用，不需要重试请设为0
- cluster：集群方式，可选：failover/failfast/failsafe/failback/forking
- 接口需要填写类型完全名称，含包名参数支持任何类型，包装类直接使用java.lang下的包装类，小类型使用：int、float、short、double、long、byte、boolean、char，自定义类使用类完全名称。
- 参数值，基础包装类和基础小类型直接使用值，例如：int为1，boolean为true等，自定义类与List或者Map等使用json格式数据。

参考资料

<https://github.com/dubbo/jmeter-plugins-dubbo>
<https://jmeter.apache.org/>