

# 性能测试培训

北京臻云科技有限责任公司

# 性能测试初印象：公众事件



## 美国大选川普赢了 加拿大移民部官网直接崩溃

加拿大家园 iAsk.ca 2016-11-08 21:05 来源: 综合新闻 作者: 加拿大移民局官网已经崩溃。 . . . . .

↑ 收起 | Q 查看大图 | C 向左旋转 | C 向右旋转

We couldn't find that Web page  
Error 404

Sorry, we couldn't find the page you're looking for. Sometimes an old link gets moved or deleted, but hopefully we can help you find what you're looking for. What next?

- Return to the homepage.
- Find out if you can come to Canada.
- Visit our Help Centre to find what you need.

Nous ne pouvons trouver cette page Web  
Erreur 404

Hélas, nous n'avons pas pu trouver la page que vous recherchez. Il arrive parfois qu'une page soit déplacée ou supprimée. Heureusement, nous pouvons vous aider à trouver ce que vous cherchez. Que faire?

- Retourner à la page d'accueil.
- Découvrir si vous pouvez venir au Canada.
- Visiter notre Centre d'aide pour trouver ce dont vous avez besoin.

Search

加拿大移民局  
iAsk.ca

# 什么是性能测试

- 评估和保障系统性能相关的NFR(非功能性需求)  
例如：响应，吞吐率，支持负载，资源消耗
- 优化系统实现：找到并发错误，系统瓶颈，提供调优建议等
- 基本方法：工具模拟实际用户，对目标系统加压
- 测试形态：基线，负载，压力，长时间可靠性测试等

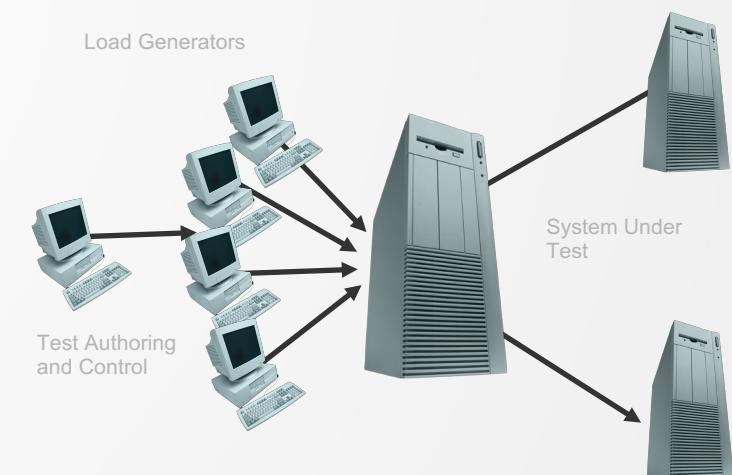
## 谁负责性能测试

- ✧ Nobody (somebody?)
- ✧ 开发兼顾
- ✧ 项目中专门的测试组
- ✧ 验收部门, 项目后期或部署之前介入

人人测试，尽早测试

## 性能目标举例

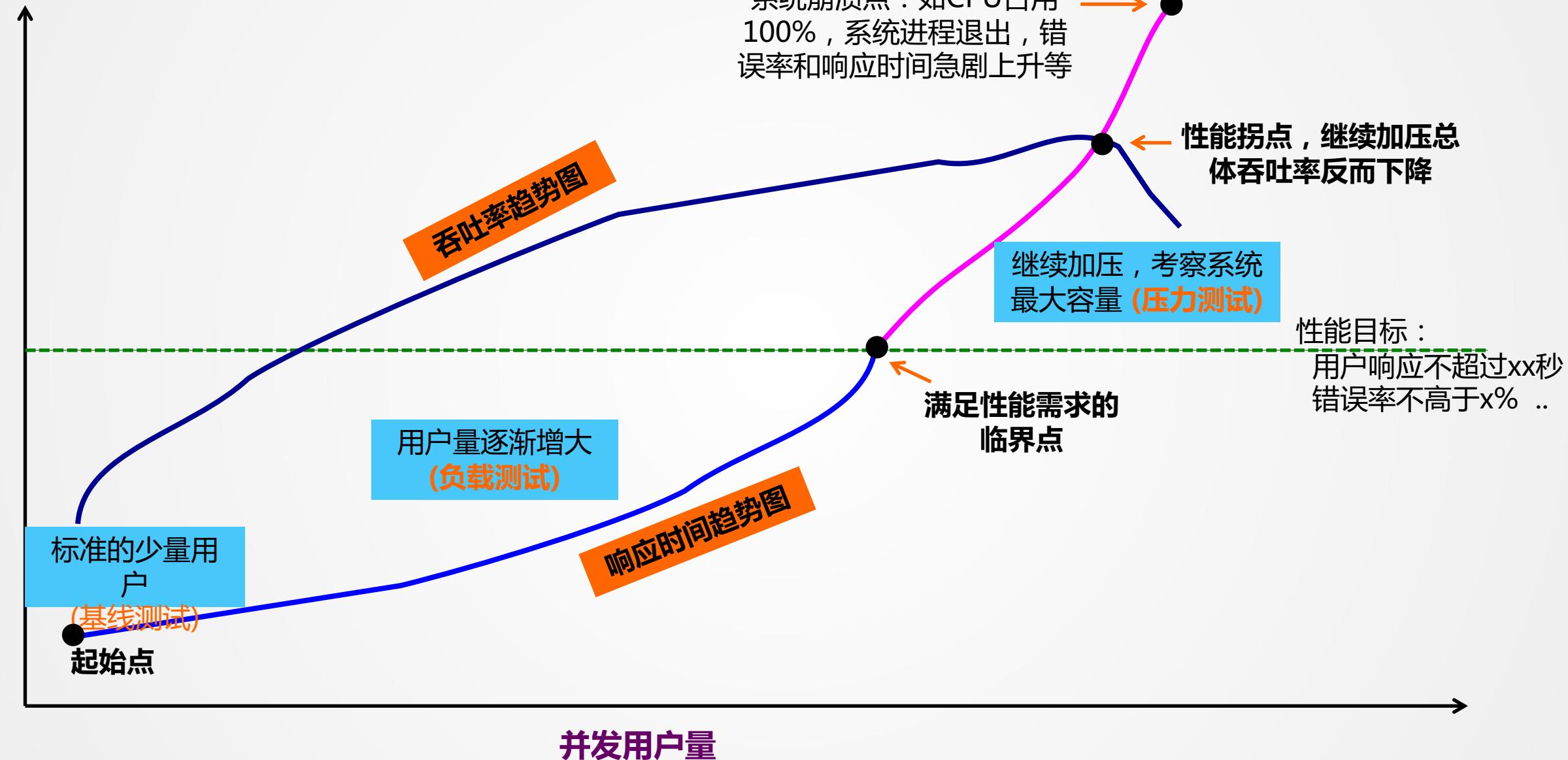
- ✓ Web操作的成功率>95%
- ✓ 平均响应时间 (除初始加载)<20秒
- ✓ 应用服务器集群负载分配合理
- ✓ 测试过程没有出现显著内存泄漏迹象
- ✓ 没有服务器崩溃或重启
- ✓ .....



# 性能测试的类型

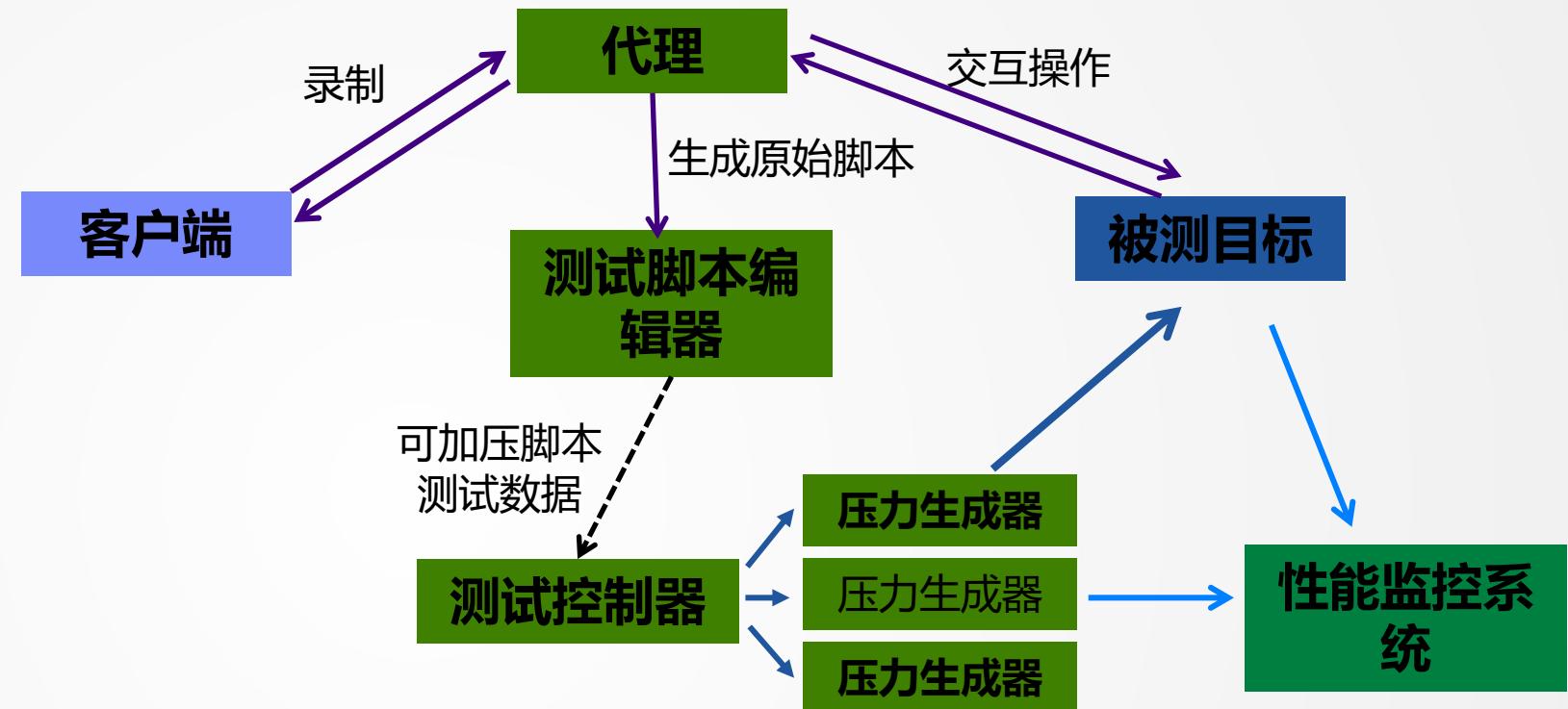
- 客户端性能 v.s. 服务器端性能
- API和组件级性能 : 基线 , 回归
- 系统级性能
  - 基线(Baseline): 单个或少量用户
  - 基准测试(Benchmark):  
严格定义的系统环境和工作负载, 结果具有可比性, 发现性能衰退
  - 负载和压力测试(Load/Stress):  
验证可扩展性, 制定容量计划, 找出最优配置, 可支持的最大并发数 ..
  - 可靠性测试(Reliability/Long Run):  
模拟系统连续运行, 发现诸如内存泄漏等问题

# 典型的系统压力模型

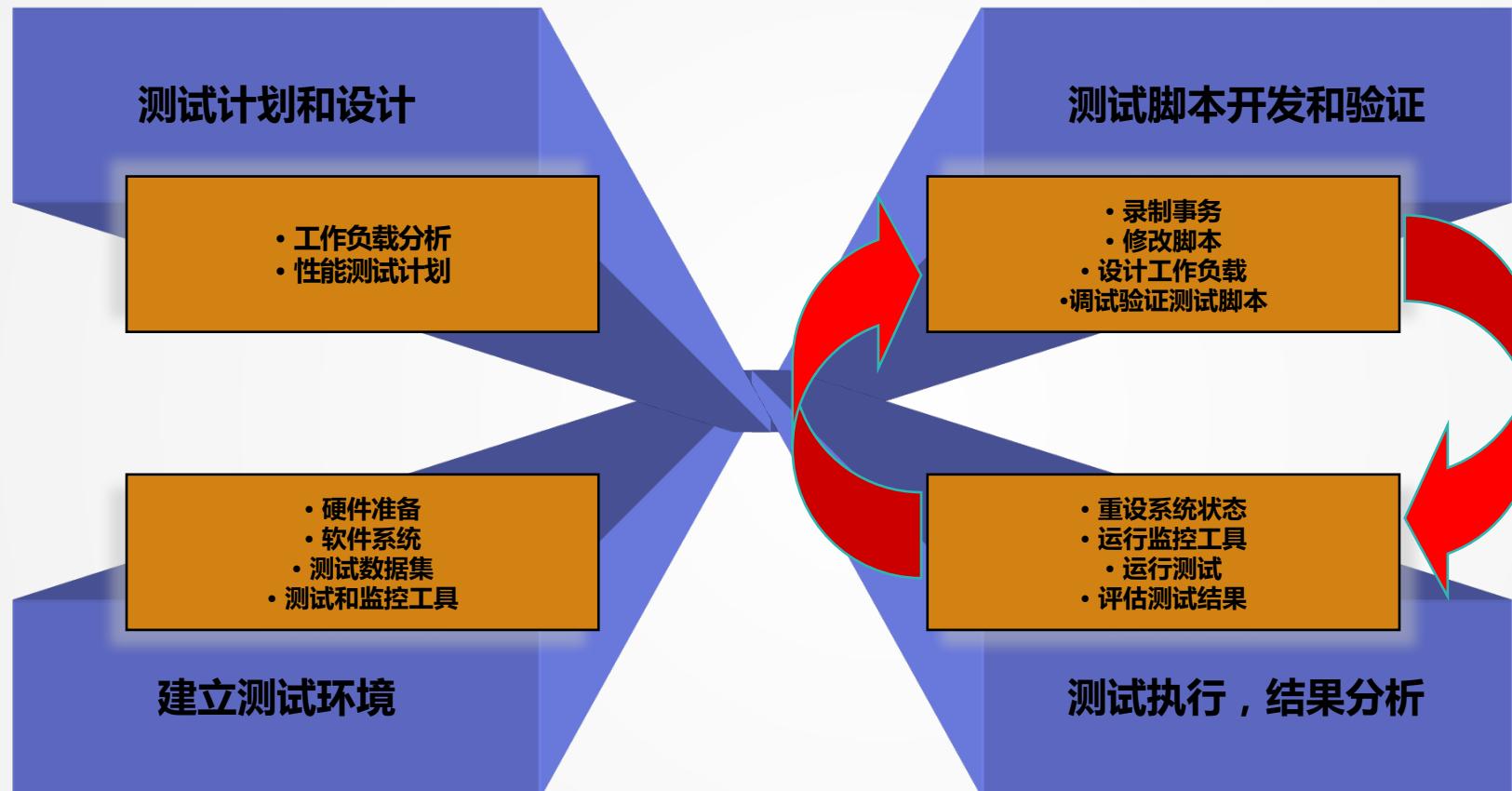


# 性能测试的基本工作原理

- 还原并模拟并发的虚拟用户请求，对被测系统加压
- 监控，收集数据，发现瓶颈和系统最大处理能力，分析改进
- 善用工具（录制，编辑，施压，结果展现，分析）



# 性能测试基本流程



# 测试准备

- 测试管理准备：
  - 工具，人员，流程，文档模板
  - 如何与需求及开发方沟通
- 熟悉被测系统，理解需求，编写测试计划
- 测试环境准备
  - 压力机系统
  - 搭建被测系统监控
- 测试数据准备

# 测试开发和验证

- 录制，编写测试脚本
- 设计工作负载
- 调试运行
  - 单用户及小量并发，排错
- 与开发同步进行
  - 及时沟通，及时反馈 (例如：缺陷汇报 )

# 正式的测试执行

- 时机：系统开发及功能测试已接近尾声
- 逐渐增加并发用户数和系统负载，衡量性能指标，理解变化的趋势
- 与开发、运维协作，测试系统的伸缩性（如：服务器扩容）及稳定性（长时间运行）
- 找到系统最大的承载量
- 结合监控分析测试结果
- 编写测试报告

## 提示

- 确保每次执行是有效的，结果真实可信
- 每次执行前的状态恢复，如测试机清理，数据库重新初始化

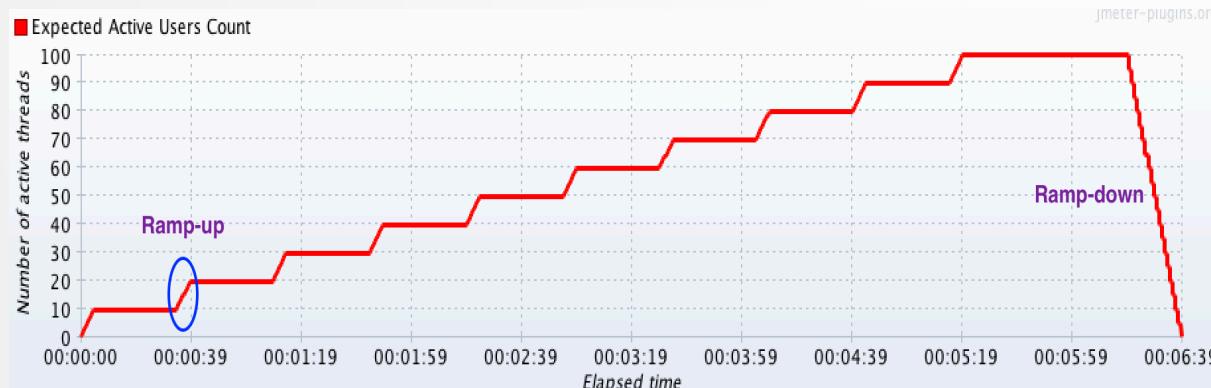
# 基本概念阐释

学习中结合实例加深概念理解

- 用户组 (线程组)
- 虚拟用户数
- Ramp-up, Ramp-down
- 页面，事务 (混合负载)
- Workload, Userload
  - 模拟的哪些操作，以及实施的频度(rate)



一个线程通常代表了一个“虚拟用户”，同被测系统交互



**线程组**

名称:

注释:

继续  Start Next Thread

**线程属性**

线程数:

Ramp-Up Period (in seconds):

循环次数  永远

▼ 测试计划

- ▶ 一般查询线程组
- ▶ 02.02 志愿服务
- ▶ 02.03 安全提示
- ▶ 02.04 服务帮助
- ▶ 02.05 12306查询车票信息
- ▶ 02.06 火车乘车 信息
- ▶ 02.07 公交信息查询
- ▶ 02.08 地铁信息查询
- ▶ 02.09 出租车信息查询
- ▶ 监听报告

▼ 测试计划

- ▶ 一般查询线程组
- ▶ 用户登录线程组
- ▶ 10\_用户登录案例
- ▶ 打开登录页面
- ▶ 提交登录请求
- ▶ 查看个人主页

事务操作

↑

页面操作

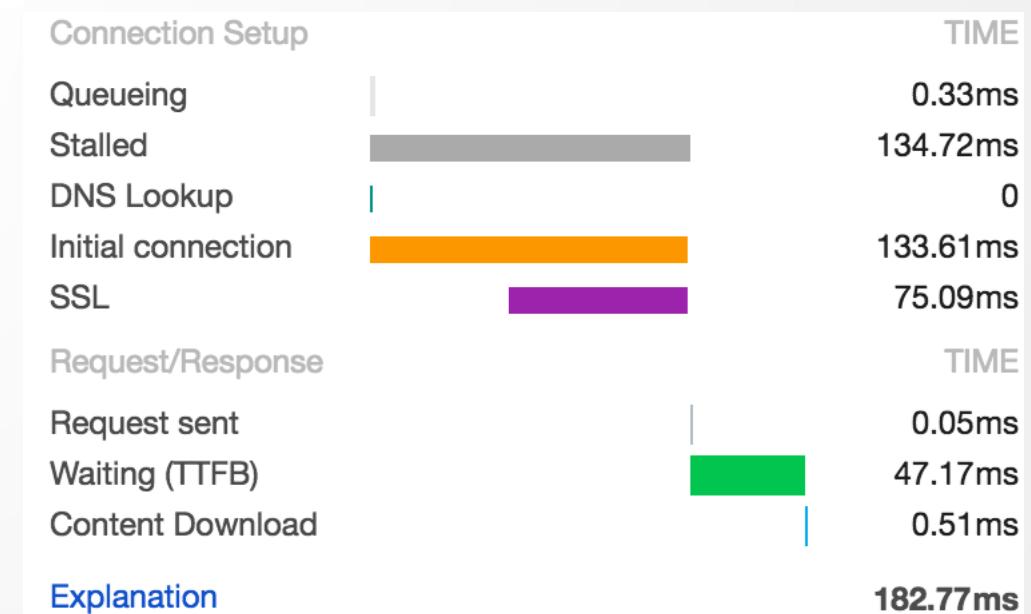
↑

# 基本概念阐释

- 响应时间
- 吞吐率 (throughput)
- 思考时间, 延时
  - 与响应时间关系
  - Latency, Sample time
- 同步 , 集合点
  - ▶ 随便逛逛
  - ▶ ...跟我一起数,3,2,1 ...
  - ▶ 秒杀开始!
- 活跃用户 , 并发用户(狭义)

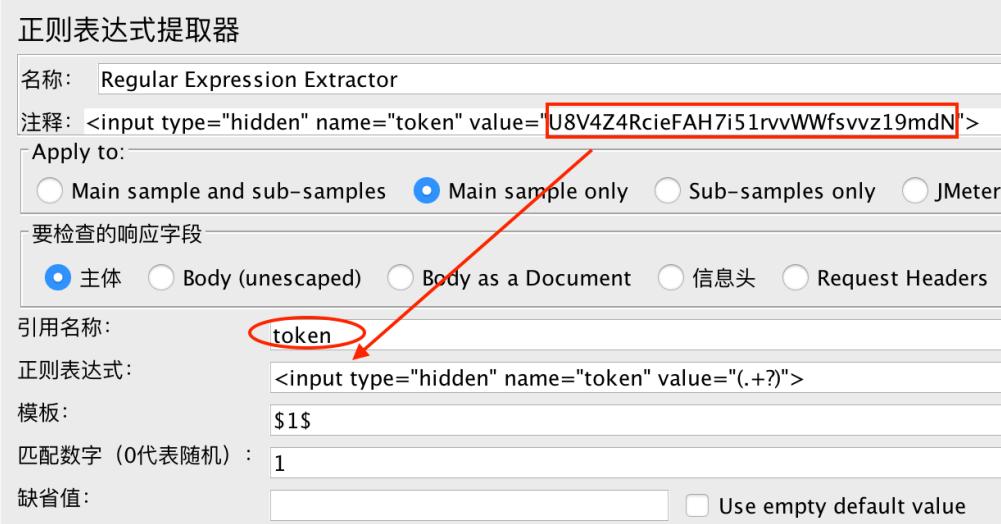
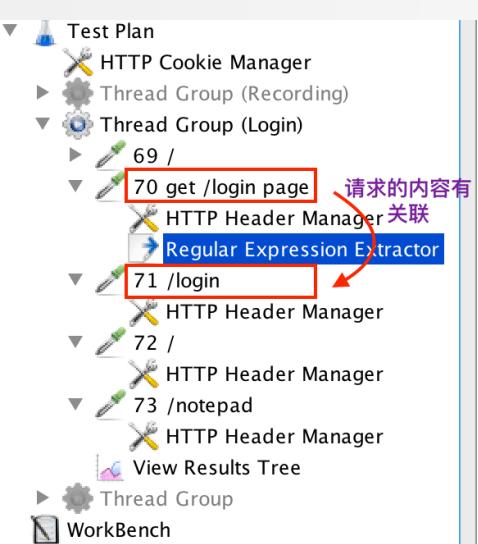
Start Time	Bytes	Status	Sample Time(ms)	Latency	Connect Time(ms)
10:04:54.613	49773	✓	332	250	75
10:04:54.950	49773	✓	393	257	70
10:04:55.281	49773	✓	335	251	62

- ▶ 🛍 检阅购物车
- ▶ ⏳ ...考虑一下, 冲动是魔鬼
- ▶ 💸 豁出去了, 买!



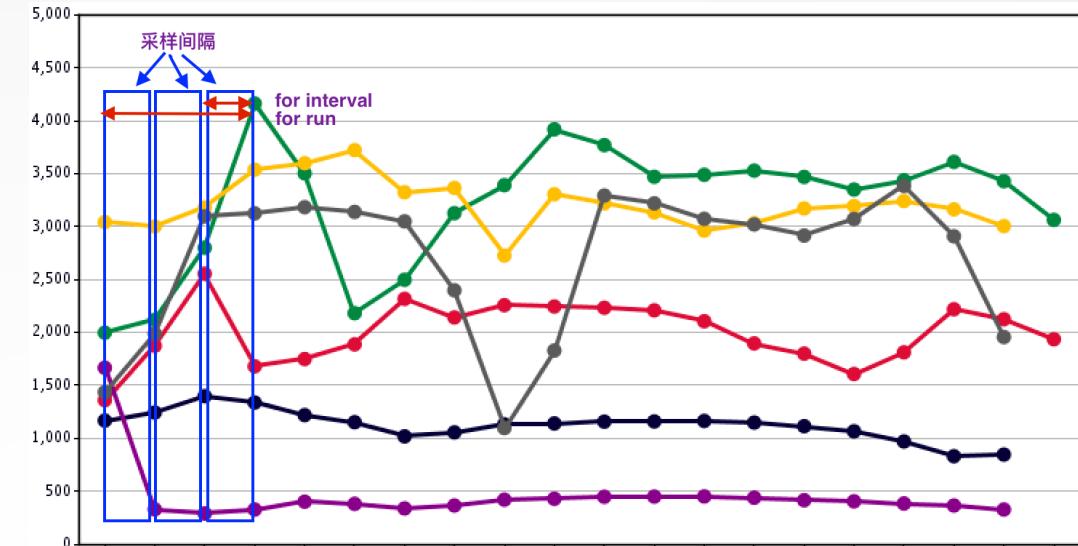
# 基本概念阐释

- HTTP请求，响应，方法，header, body
- 参数化，动态数据关联（以及随机化）
- 执行控制逻辑



# 基本概念阐释

- 采样：
  - For run v.s. **for interval**
- Attempt v.s. Hit
- 平均，最大，最小, Percentile
  - 去除少量误差大的坏样本



Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Max	Error %
jianshu	50	451	362	498	785	2282	254	2282	0.00%
baidu	50	246	138	511	541	1116	84	1116	0.00%
总体	100	348	328	511	785	1853	84	2282	0.00%

90% line 511ms : 90%的样本的响应时间低于511ms

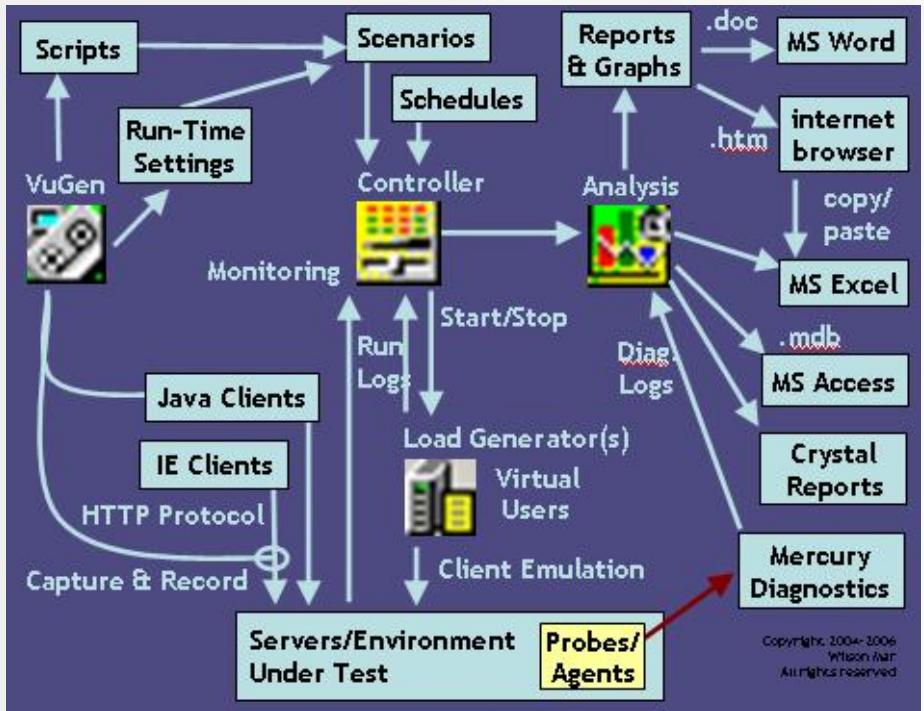
- 方差
  - 个体样本之间的差异程度（值越大差异越大，理想值=0）

## 常用的性能度量指标

- 系统响应时间：页面，事务级别
- 吞吐率 (throughput): TPS, QPS
- 出错率：超过阈值则为无效测试执行
- 被测系统(OS)的监控指标：CPU，内存，磁盘IO，网络IO (带宽)
- 应用相关指标：比如，应用服务器线程连接池，JDBC连接池的用量

分析问题：“端到端”的性能度量与后台收集的数据相结合，例如  
Java相关的gc log, thread dump, heap dump  
消息服务器，数据库的日志

# 主要性能测试工具介绍



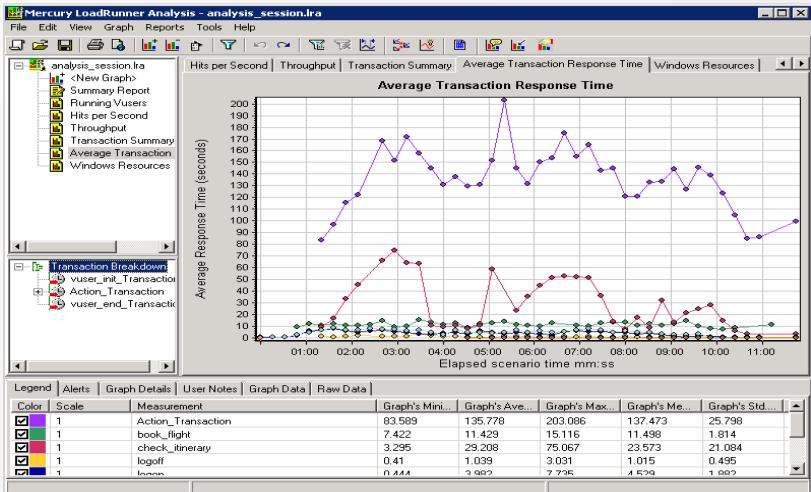
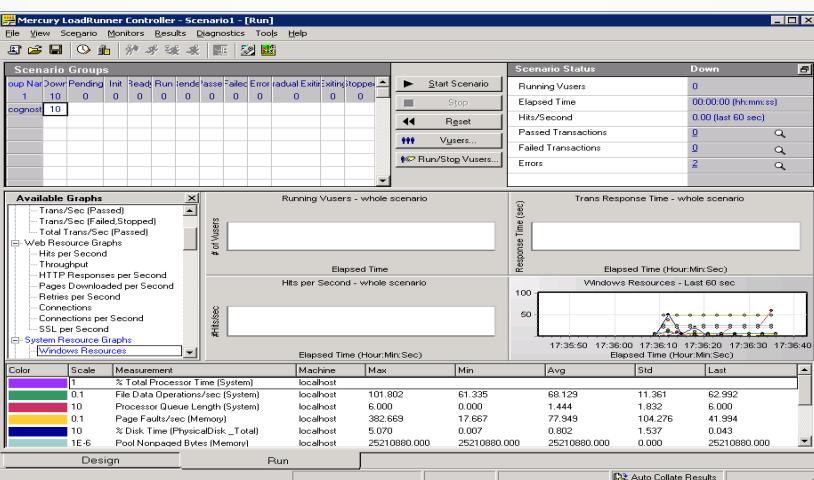
- 老牌的性能测试工具，功能强大，使用广泛
- 主要组件：VuGen, Controller, LoadGen, Analysis
- 测试脚本以C语言为主，可录制，编辑，扩展
- 灵活的测试场景，支持协议多，监控，报表
- license费用高昂
- 近年来发展趋于平缓

```
#include "as_web.h"

Action1()
{
    web_url("duwamish7",
            "URL=http://192.168.6.204/duwamish7/",
            "Resource=",
            "ReContent-Type=text/html",
            "Referer=",
            "Snapshot=t1.inf",
            "Mode=HTML",
            EXTRARES,
            "Url../Duwamish7/images/banner/bannerslice.gif", ENDITEM,
            LAST);

    lr_think_time( 2 );

    web_submit_form("Default.aspx",
                    "Snapshot=t2.inf",
                    ITEMDATA,
                    "Name=ModuleSearch:SearchDropDownList", "Value=Title", ENDITEM,
                    "Name=ModuleSearch:SearchTextBox", "Value=", ENDITEM,
                    "Name=ModuleDailyPick:DailyPickList:_ctl10:AddToCartButton", "Value=Add To
                    EXTRARES,
```



# 主要性能测试工具介绍



- 性能测试工具的后起之秀，基于Java的开源软件，商用免费
- 社区活跃，关注度和使用量追平甚至超过LoadRunner
- 小巧轻便，快速上手，众多可选插件
- 测试脚本可录制，图形化的脚本编辑器，可调试运行
- 可自行扩展支持协议，二次开发
  
- 易用性稍弱，不够美观
- 报表，结果分析能力不足

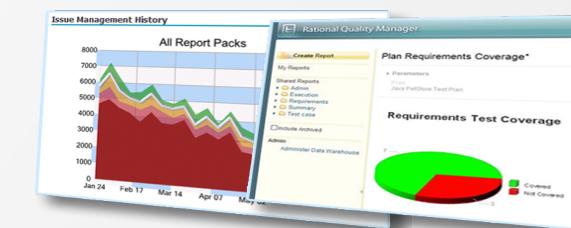
The screenshot shows the Apache JMeter graphical user interface. At the top is a toolbar with various icons for file operations like Open, Save, and Print, and other tools like Run and Help. Below the toolbar is the main workspace.

The left side features a tree view of the "Test Plan". Under "Test Plan", there is a "Thread Group" node which contains several items: "MQTT\_Publisher" (selected), "MQTT\_Subscriber", "Constant Timer", "View Results Tree", "View Results in Tab", and "While Controller". Below the "Thread Group" is another node "WorkBench" which contains "HTTP(S) Test Script Re".

To the right of the tree view is a configuration panel titled "MQTT Publisher". It includes fields for "Name" (set to "MQTT\_Publisher"), "Comments", "Connection Info" (with a "Provider URL" field containing "tcp://192.168.100.12:1883"), and "List topic" (with a value of "topic"). There are also checkboxes for "One connection per topic" and "Use Authorization? User".

# 性能测试报告：简明清晰、有内涵

- 先明确报告的对象，再组织和定制报告的内容
- 尽早，及时地提供（非正式）报告，获得反馈
- 基于模板的统一的报告格式
- 正确理解测试数据，再做汇总统计和分析
- 图表直观，视觉效果好
- 辅以简明的文字解读图表，给出评价和建议
- 留存各类原始数据，随时可访问



# 测试报告文档样例

## 综述

测试项目的目标，人员时间安排等  
通过测试观察到的主要结果  
结论，包括系统评价，调优建议

## 测试环境

被测系统描述：拓扑结构，系统配置等  
测试工具  
测试机软硬件配置  
测试数据  
用户负载，工作负载说明

## 测试用例

用例场景描述，负责人  
测试的负载，配置参数等  
需要收集的指标和其它数据

## 测试结果

针对测试用例执行结果，汇集相应的性能度量指标  
图表展现  
多角度分析对比，给出结论、推理预测及缺陷报告

## 原始数据归档

包括导出的性能测试数据，服务器日志等

# 性能调优若干基本原则

- 维护系统配置的基线(baseline)，在此基础上记录每次的改变
- 每次只调整一个因素，重新一轮测试，衡量对结果的影响
- 特定的调优措施作用于组件，但要从“端到端”来检验系统的性能变化
- 组件调优后，要重新建立整个系统的基准(benchmark)
- 记录每次测试的结果，分析问题，以及推荐解决办法
  
- 可调优组件的一些例子
  - ▶ 操作系统参数 (VM setting, TCP/IP timeout)
  - ▶ 应用程序日志级别
  - ▶ JVM (Heap size, thread pool, connection pool)
  - ▶ 数据库 (建立索引, 优化SQL)

# 性能测试的几点趋势

- 微服务架构需要更多接口和组件级的性能测试
- DevOps让性能测试融入持续集成
- 大数据时代的高负载及海量处理分析
- 云服务
  - 提升测试资源配置效率
  - 减轻人工执行的负担
  - 更加高效的性能分析汇报

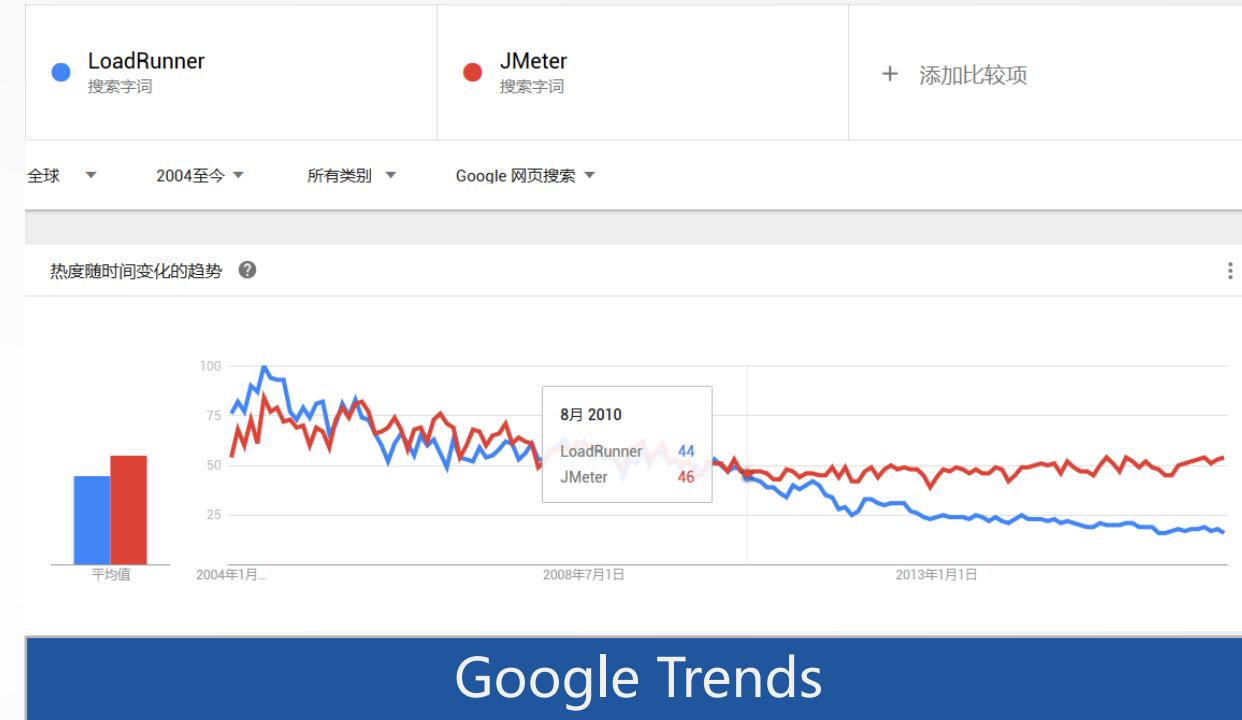
# JMeter基础

# 什么是JMeter ?

Apache JMeter is a 100% pure Java application designed to load test client/server software (such as a web application). It may be used to test performance both on static and dynamic resources such as static files, Java Servlets, ASP.NET, PHP, CGI scripts, Java objects, databases, FTP servers, and more. JMeter can be used to simulate a heavy load on a server, network or object to test its strength or to analyze overall performance under different load types.

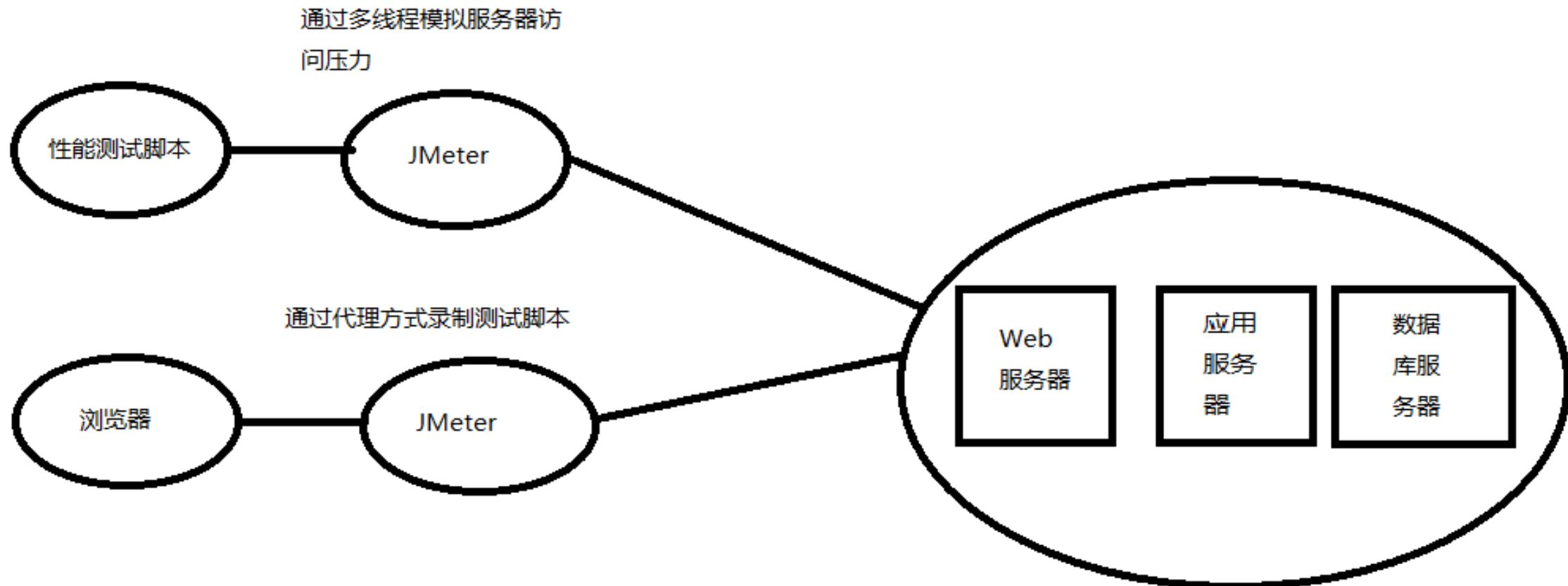
- 开源的性能测试工具
- 100%的Java应用，跨平台
- 可用于测试基于多种协议的静态或动态资源
- 通过模拟大量负载来测试并分析系统在不同情况下的性能

# JMeter的生态



- 与传统的性能测试工具的代表LoadRunner相比，JMeter显得更有生命力；
- 百度指数，JMeter从2014年开始稳步上升，近期已接近或超过LoadRunner；
- Google Trends，2011 年全球对JMeter的兴趣已经超过了LoadRunner。

# JMeter工作基本原理



- JMeter根据性能测试脚本，通过线程组（虚拟用户）来模拟真实用户的访问压力
- JMeter也可以作为web服务器与浏览器之间的代理网关以捕获浏览器请求和web服务器的响应生成性能测试脚本

## JMeter系统要求

- JVM 7或者更高版本。
- 操作系统： JMeter是100%Java应用，支持Java官方支持的所有平台。

# JMeter安装步骤

- <https://www.jianshu.com/p/0e4daecc8122>



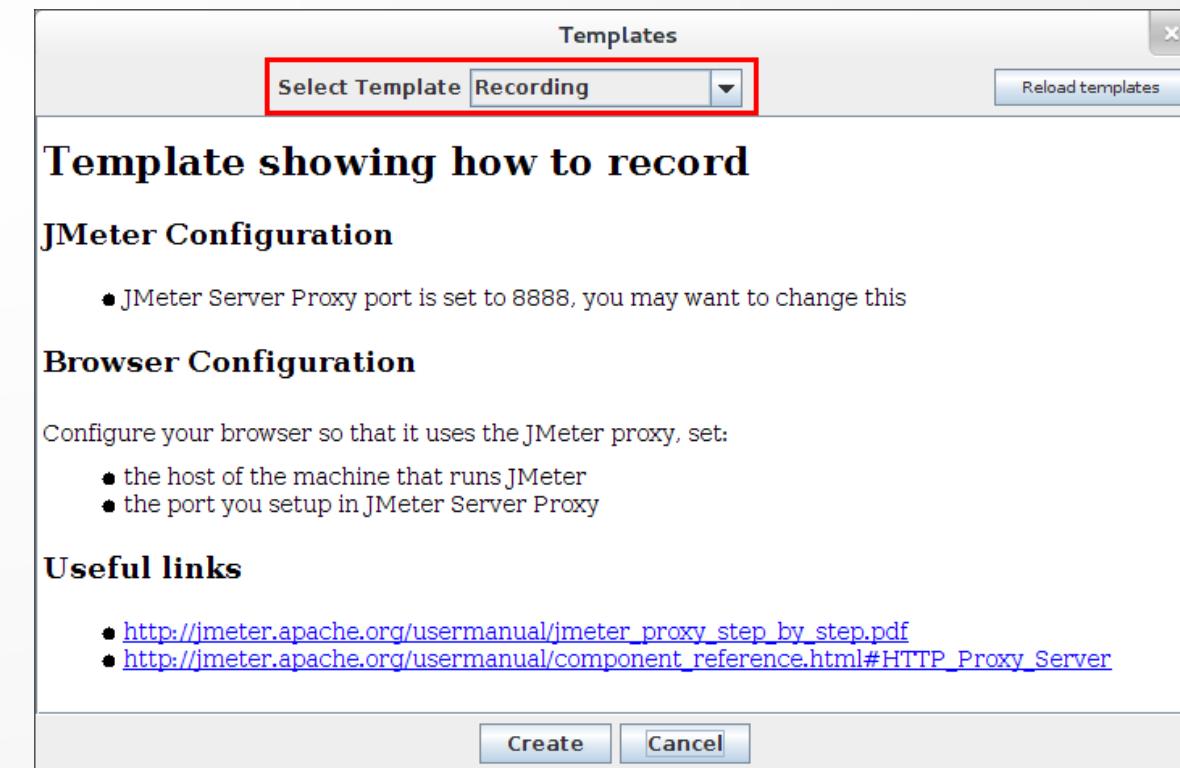
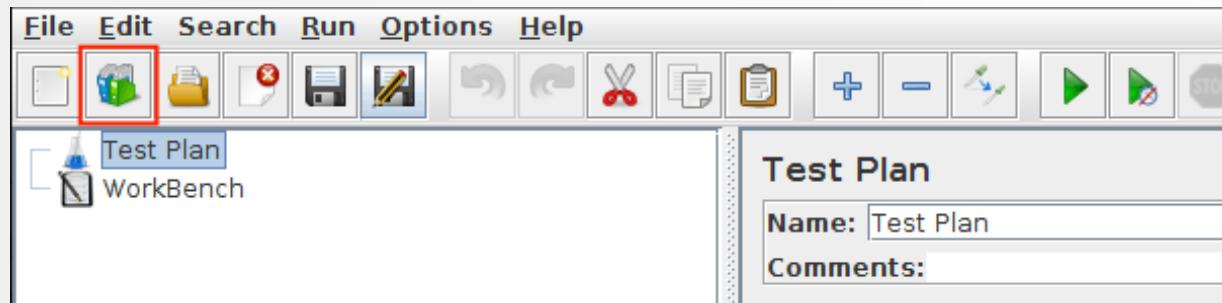
- 安装Java 8 → 下载JMeter → 运行JMeter

## 练习1: 手动创建一个HTTP测试用例 – 15分钟

- 事先准备好JMeter运行环境
- 创建第一个测试：百度首页测试
  - 创建线程组
  - 创建HTTP取样器、并设定目标地址
  - 加入结果Listener（“查看结果树”和“聚合报告”），查看测试结果
  - 增加虚拟用户数目

# 通过模版创建一个测试用例

- JMeter提供了多种测试模版（可在<JMeter\_HOME>/bin/templates下找到模版文件）
- 可以通过点击“文件”->“templates”或者通过UI来创建测试用例。



# JMeter的测试元素

- 测试计划 ( Test Plan )
- 线程 ( 用户 ) Thread ( User )
- 采样器
- 逻辑控制器
- 配置元件
- 定时器
- 前置处理器
- 后置处理器
- 断言
- 监听器

# 测试计划 ( Test Plan )

- 测试计划使整个测试脚本的基础，所有的测试元素都会被包含在测试计划中
- 测试计划中的配置项：
  - 用户定义变量：可以被整个测试计划中的测试元素或脚本调用  
调用方式 \${变量名}
  - 独立运行每个线程组（在一个线程组结束后启动下一个）
  - 在关闭(shutdown)主线程后运行tearDown线程组
  - 函数测试模式：只有当你需要记录每个请求从服务器取得的数据文件时才需要选择。非常影响性能。一般用于功能测试模式。
  - 添加目录或jar到JMeter classpath

# 线程组 ( Thread Group )

- 线程组提供了一个虚拟用户池，用来执行测试脚本中的指定测试。
- JMeter提供了三中线程组：

- 线程组 ( Thread group )

通俗的讲一个线程组，可以看做一个虚拟用户组，线程组中的每个线程都可以理解为一个虚拟用户。线程组中包含的线程数量在测试执行过程中是不会发生改变的。

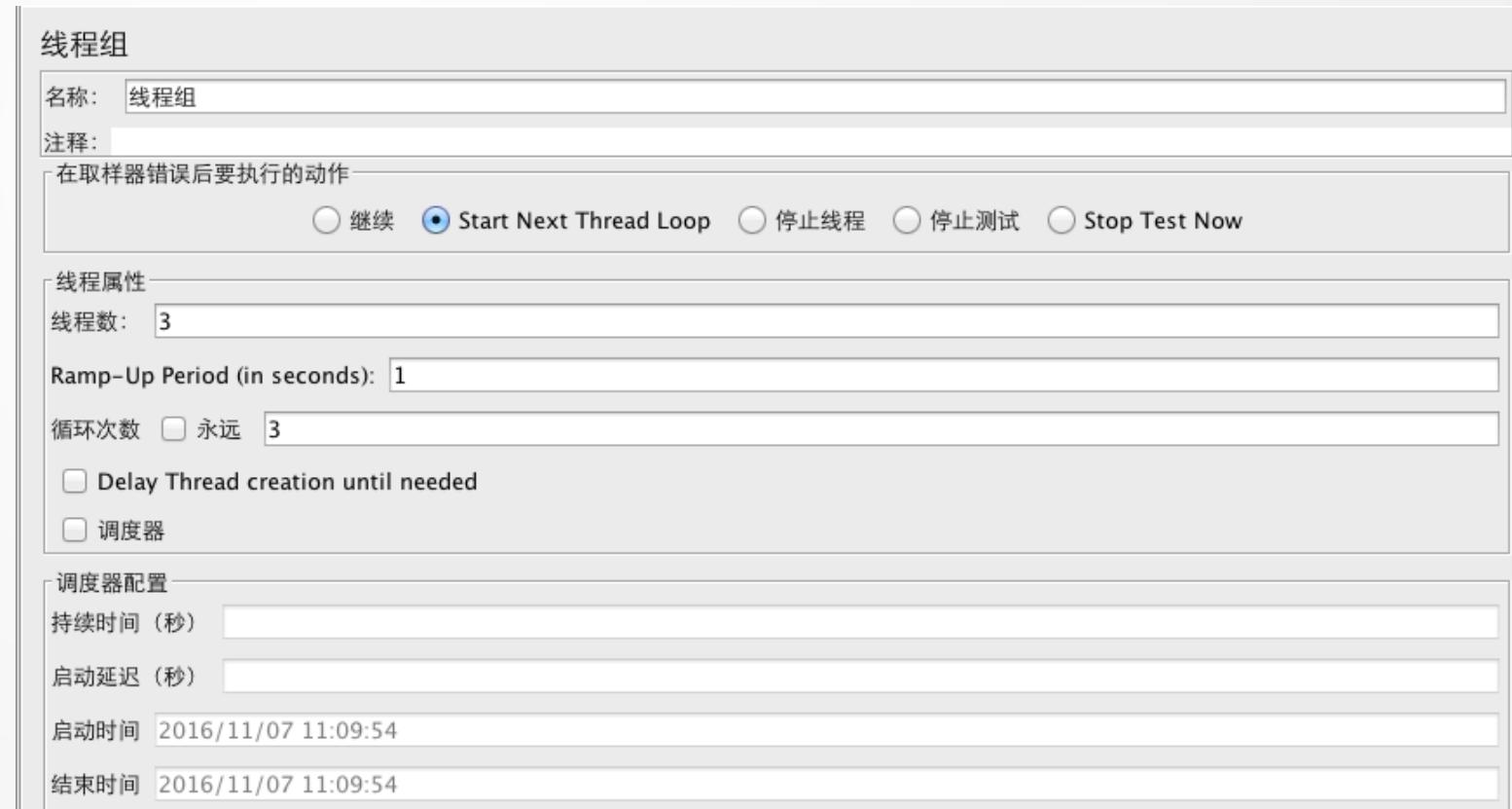
- setup thread group

一种特殊类型的ThreadGroup，可用于执行预测试操作。

- Teardown thread group

一种特殊类型的ThreadGroup，可用于执行测试后动作。

- 线程组常用配置参数：



# 采样器 ( Sampler )

- 采样器(Sample)是性能测试中向服务器发送请求，记录响应信息，响应时间的最小单元，JMeter 原生支持多种不同的sampler，每一种不同类型的 sampler 可以根据设置的参数向服务器发出不同类型的请求。
- 采样器只能包含在线程组中
- JMeter自带的采样器：其中Java Request Sampler 和 Beanshell Request Sampler 是两种特殊的可定制的采样器

Access Log Sampler  
AJP/1.3 Sampler  
BeanShell Sampler  
BSF Sampler  
Debug Sampler  
FTP请求  
HTTP请求  
Java请求  
JDBC Request  
JMS Point-to-Point  
JMS Publisher  
JMS Subscriber  
JSR223 Sampler  
JUnit Request  
LDAP Extended Request  
LDAP请求  
Mail Reader Sampler  
OS Process Sampler  
SMTP Sampler  
SOAP/XML-RPC Request  
TCP取样器  
Test Action

# HTTP请求采样器

## ■ 添加HTTP请求配置参数：

**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URL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。

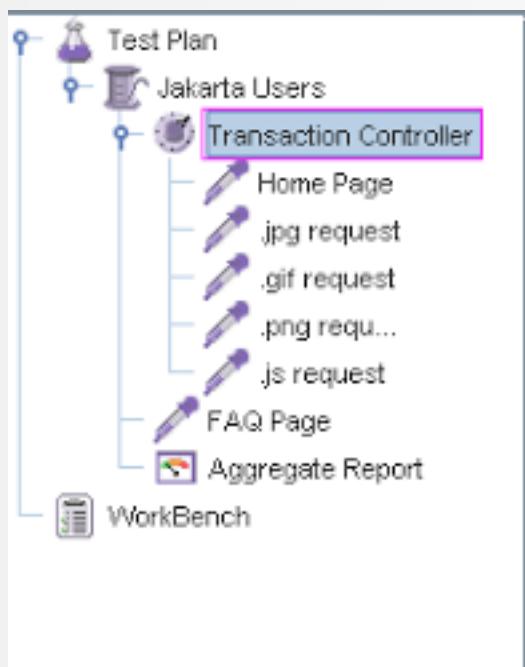
**从HTML文件获取所有有内含的资源**： 当该选项被选中时， JMeter在发出HTTP请求并获得响应的HTML文件内容后， 还对该HTML进行Parse 并获取HTML中包含的所有资源（图片、 flash等）， 默认不选中， 如果用户只希望获取页面中的特定资源， 可以在下方的Embedded URLs must match 文本框中填入需要下载的特定资源表达式， 这样， 只有能匹配指定正则表达式的URL指向资源会被下载。

**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**： 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。 在数据量非常大的测试时， 建议选中该项以减少sampler记录响应数据的开销。

# 事务控制器

- 我们在做某一项操作时通常会包含多个请求，当我们需要把多个请求看成是一个整体事务，并在聚合报告中看到一个统一的条目时我们就需要使用事务控制器。



The screenshot shows the JMeter Test Plan tree on the left. A Transaction Controller node is selected and highlighted with a pink border. It has several child requests listed under it: Home Page, jpg request, gif request, png request, js request, FAQ Page, and Aggregate Report.

**Transaction Controller**

Name: Transaction

Comments:

Generate parent

Include timer during iteration

WorkBench

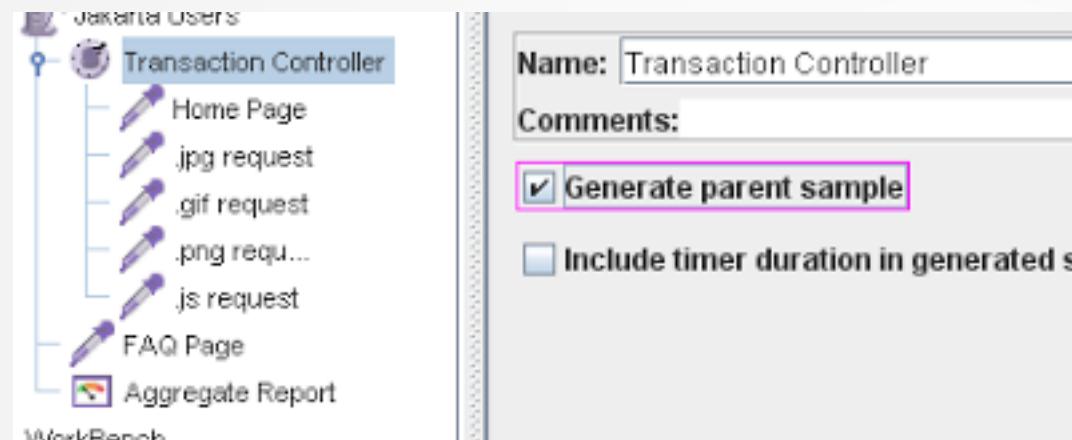
- Home Page
- jpg request
- gif request
- png request
- js request
- FAQ Page
- Aggregate Report

Comments:  
Write results to file / Read from file  
Filename:   
Browse... Log/Display Only:  Err

Label	# Samples	Average	Median	90% L...	Min	Max	Error %	Through...
Home Page	2	3505	3250	3754	3250	3754	0.00%	32.0/min
jpg request	2	498	309	687	309	687	100.00%	2.5/sec
gif request	2	3554	3433	3676	3433	3676	0.00%	31.6/min
png request	2	3631	3628	3634	3628	3634	0.00%	30.1/min
js request	2	3673	3671	3675	3671	3675	0.00%	29.8/min
Transaction Controller	2	14064	14600	15041	14600	15041	100.00%	0.0/min
FAQ Page	2	437	357	518	357	518	100.00%	2.8/sec
TOTAL	14	4309	3628	14600	309	15041	42.06%	54.5/min

## 事务控制器 (Cont.)

- 我们可以通过设置“Generate parent sample”来仅现实整体事务报告。



Write results to file / Read from file								
Label	# Samples	Average	Median	90% Li.	Min	Max	Error %	Throughput
Transaction Controller	2	15079	15277	16482	15277	16482	100.00%	7.3/min
FAQ Page	2	933	649	1217	649	1217	100.00%	1.1/sec
TOTAL	4	8406	1217	16482	649	16482	100.00%	14.0/min

# 配置元件 ( Config element )

- 配置元件用于提供对静态数据配置的支持。

比如CSV Data Set Config 可以将本地数据文件形成数据池；而对应于HTTP Request Sampler和TCP Request Sampler等类型的配制元件则可以修改采样器的默认数据。例如HTTP Cookie Manager 可以用于对 HTTP Request Sampler 的cookie 进行管理。

- 配置元件可定义在测试计划层级下，  
也可以定义在线程组或采样器层级下。  
定义在不同层级的作用域不同。

- JMeter支持的配置元件：

- CSV Data Set Config
- DNS Cache Manager
- FTP请求缺省值
- HTTP Cache Manager
- HTTP Cookie 管理器
- HTTP信息头管理器
- HTTP授权管理器
- HTTP请求默认值
- Java请求默认值
- JDBC Connection Configuration
- Keystore Configuration
- LDAP Extended Request Defaults
- LDAP请求默认值
- Random Variable
- TCP取样器配置
- 用户定义的变量
- 登陆配置元件/素
- 简单配置元件
- 计数器

# HTTP Cookie Manager

## ■ 自动管理cookie：

像浏览器一样的存储和发送Cookie，如果发送一个http请求他的响应中包含Cookie，那么Cookie Manager就会自动地保存这些Cookie并在所有后来发送到该站点的请求中使用这些Cookie的值。每个线程都有自己存储cookie的区域。在cookie manager中看不到自动保存的cookie，我们可以在View Results Tree的Request界面看到被发送的Cookie Data。

接受到的Cookie的值可以被存储到JMeter 线程变量中。要把Cookies保存到线程变量中，要在 jemeter.properties中定义属性 “CookieManager.save.cookies=true” 变量名为COOKIE\_ + Cookie名。属性CookieManager.name.prefix= 可以用来修改默认的COOKIE\_的值。

- 手动管理Cookie：手动添加Cookie到Cookie Manager，这些Cookie的值会被所有线程共享。
- 在一个测试计划中最好只有一个cookie manager

## HTTP Header Manager

- 该属性管理器用于定制Sampler发出的HTTP请求的请求头的内容。

## HTTP Cache Manager

- 该属性管理器用于模拟浏览器的Cache行为。为Test Plan增加该属性管理器后，Test Plan运行过程中会使用Last-Modified、ETag和Expired等决定是否从Cache中获取相应的元素。

## HTTP Request Defaults

- 该属性管理器用于设置其作用范围内的所有HTTP的默认值，可被设置的内容包括HTTP请求的host、端口、协议等。一个Test Plan中可以有多个HTTP Request Defaults，处于多个HTTP Request Defaults作用域内的Sampler使用HTTP Request Defaults中设置值的叠加值。

# CSV数据集配置 ( CSV Data Set Config )

- 在性能测试过程中我们往往需要一些参数化的输入参数，比如登录操作里面的用户名密码。当并发量比较大的时候我们就可以借助 CSV Data Set Config 来作为参数来源
- 以下是CSV Data Set Config各个参数的简要说明：
  - FileName:即同目录下csv文件的名称
  - File Encoding: 默认为ANSI
  - Variable Names: 定义文本文件中的参数名,参数之间逗号分隔.定义后可在脚本在以Shell变量同样的方式引用
  - Allow Quoted data: 是否双引号
  - Delimiter: 数据分隔符
  - Recycle on EOF: 设置为True后,允许循环取值
  - Stop Thread on EOF: 当Recycle on EOF为false并且Stop Thread on EOF为true,则读完csv文件中的记录后,停止运行
  - Sharing Mode: 设置是否线程共享

# 定时器 ( Timer )

- 定时器(Timer)用于操作之间设置等待时间，等待时间是性能测试中常用的控制客户端请求频率的手端。类似于LoadRunner里面的“思考时间”。
- 配置元件可定义在测试计划层级下，也可以定义在线程组或采样器层级下。定义在不同层级的作用域不同。
- JMeter支持的定时器类型：

BeanShell Timer  
BSF Timer  
Constant Throughput Timer  
JSR223 Timer  
Poisson Random Timer  
Synchronizing Timer  
Uniform Random Timer  
固定定时器  
高斯随机定时器

# 后置处理器(Post Processor )

- 用于对Sampler发出请求后得到的服务器响应进行处理。一般用来提取响应中的特定数据。例如，XPath Extractor则可以用于提取响应数据中通过给定XPath值获得的数据。
- 后置处理器定义在测试计划层级下，也可以定义在线程组或采样器层级下。定义在不同层级的作用域不同。
- JMeter支持的后置处理器类型：

BeanShell PostProcessor  
BSF PostProcessor  
CSS/JQuery Extractor  
Debug PostProcessor  
JDBC PostProcessor  
JSON Path PostProcessor  
JSR223 PostProcessor  
Result Status Action Handler  
XPath Extractor  
正则表达式提取器

# 正则表达式提取器

- 正则表达式提取器是一个非常常用的后置处理器
- 通过正则表达式，我们可以很灵活地从响应中查找，提取我们想要的内容并把它作为参数传递给后面的采样器。
- 例子：
  - 我们希望从login页面请求响应的html body中找到token
  - 找到后把token赋给一个参数并传到下一个请求中

The screenshot shows the JMeter interface with the 'RegExp Extractor' configuration dialog open. On the left, the tree view shows a 'Load login page' test plan element containing a 'RegExp Extractor' node, which is highlighted in blue. The main window displays the 'RegExp Extractor' settings:

- 名称:** 正则表达式提取器
- 注释:** (empty)
- Apply to:**  Main sample only
- 要检查的响应字段:**  主体
- 引用名称:** token
- 正则表达式:** <input type="hidden" name="token" value="(.\*)">
- 模板:** \$1\$
- 匹配数字 (0代表随机):** 1
- 缺省值:** (empty input field)
- Use empty default value:**

# 断言 (Assertions )

- 断言用于检查测试中得到的相应数据等是否符合预期，断言一般用来设置检查点，用以保证性能测试过程中的数据交互是否与预期一致。
- JMeter支持的断言类型：

BeanShell断言  
BSF Assertion  
Compare Assertion  
HTML Assertion  
JSR223 Assertion  
MD5Hex断言  
Size Assertion  
SMIME Assertion  
XML Schema Assertion  
XML断言  
XPath Assertion  
响应断言  
断言持续时间

# 监听器(Listener )

- 监听器是用来对测试结果数据进行处理和可视化展示的一系列元件。 图形结果、 查看结果树、 聚合报告。 都是我们经常用到的元件。
- JMeter支持的监听器类型：

- Aggregate Graph
- Backend Listener
- BeanShell Listener
- BSF Listener
- Comparison Assertion Visualizer
- JSR223 Listener
- Response Time Graph
- Simple Data Writer
- Summary Report
- 保存响应到文件
- 图形结果
- 查看结果树
- 断言结果
- 生成概要结果
- 用表格察看结果
- 监视器结果
- 聚合报告
- 邮件观察仪

# 聚合报告 ( aggregate report )

■ 聚合报告是监听器里我们常用的一种类型。在报告中有一些名词我们在这里解释一下

- Samples -- 本次场景中一共完成了多少个Transaction
- Average -- 平均响应时间
- Median -- 统计意义上响应时间的中值
- 90% Line -- 所有transaction中90%的transaction的响应时间都小于xx
- Min -- 最小响应时间
- Max -- 最大响应时间
- PS: 以上时间的单位均为ms
- Error -- 出错率
- Throughput -- 吞吐量，单位：transaction/sec
- KB/sec -- 以流量做衡量的吞吐量

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	<input type="text"/>										
<input type="button" value="浏览..."/> Log/Display Only: <input type="checkbox"/> 仅日志错误 <input type="checkbox"/> Successes <input type="button" value="Configure"/>											
Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Max	Error %	Through...	KB/sec
Load login page	1	19	19	19	19	19	19	19	0.00%	52.6/sec	255.7
login post	1	111	111	111	111	111	111	111	0.00%	9.0/sec	57.0
总体	2	65	19	111	111	111	19	111	0.00%	15.2/sec	84.7

# 练习2: 创建一个比较复杂的HTTP测试

- 聚合报告是监听器里我们常用的一种类型。在报告中有一些名词我们在这里解释一下
  - Samples -- 本次场景中一共完成了多少个Transaction
  - Average -- 平均响应时间
  - Median -- 统计意义上响应时间的中值
  - 90% Line -- 所有transaction中90%的transaction的响应时间都小于xx
  - Min -- 最小响应时间
  - Max -- 最大响应时间
  - PS: 以上时间的单位均为ms
  - Error -- 出错率
  - Throughput -- 吞吐量 , 单位 : transaction/sec
  - KB/sec -- 以流量做衡量的吞吐量

# JMeter 支持的其他协议测试

- JMeter除了支持HTTP协议的测试以外还支持多种协议, 我们可以在采样器列表中找到相应的采样器
  - JDBC
  - TCP
  - FTP
  - LDAP
  - JMS
  - SMTP
  - WebSocket
  - MQTT
  - 等等。 . . .
- 除了开箱即用的协议支持外 , JMeter还支持用户自己开发新协议的plugin来扩展

# JMeter 测试脚本录制

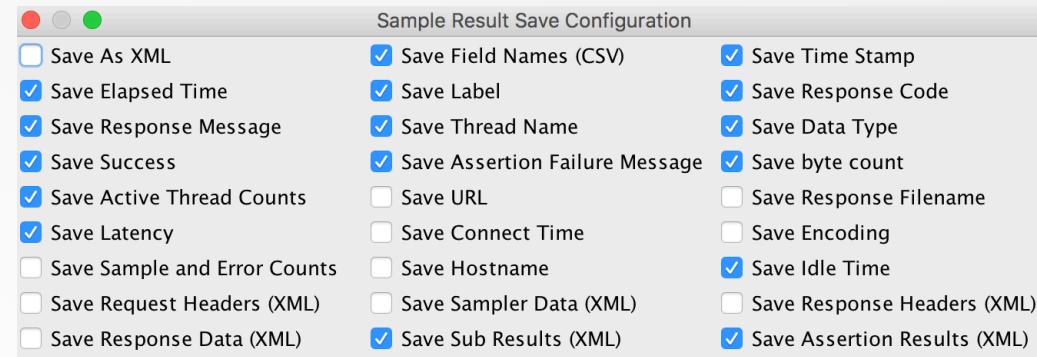
- JMeter除了手动创建脚本以外还可以通过录制的方法创建脚本
- 录制脚本可以帮助我们快速的创建一个测试用例的基本骨架。
- 录制后的脚本往往不能直接使用，需要对其进行一定的修改
- 常用的录制方式包括
  - 使用JMeter自带的录制器录制脚本
  - 使用第三方工具，比如badboy

# 使用JMeter自带的录制器

- JMeter是通过自带的HTTP 代理服务器来录制Http的脚本
- 录制方法：
  - 在测试计划中创建一个线程组
  - 在工作台中创建HTTP代理服务器
  - 在浏览器中配置http代理，将代理服务器地址设置为 localhost 或者 127.0.0.1, 端口设置成在 JMeter中的端口号
  - 点击JMeter中HTTP代理服务器下面的录制按钮
  - 到浏览器中进入被测网站执行操作
  - 执行完毕，回到JMeter，停止录制
  - 禁用浏览器中的http代理

# JMeter 测试数据收集

- JMeter 通过监听器记录测试数据
  - 格式: csv, xml
- 监听器类型繁多，还可通过插件扩展
- 每种监听器可以独立地指定测试数据记录文件，以及记录哪些字段
- NonGUI 执行 Jmeter 也可指定数据记录文件
  - `./jmeter.sh -n -t test.jmx -l data.jtl`
- “Simple Data Writer” 监听器只记录数据，执行开销比其它监听器小很多（正式的大并发长时间性能测试不要开其他监听器）
- 可在任意监听器中加载保存过的 jtl 文件，查看不同的结果展现



Summary Report

名称: Summary Report

注释:

所有数据写入一个文件

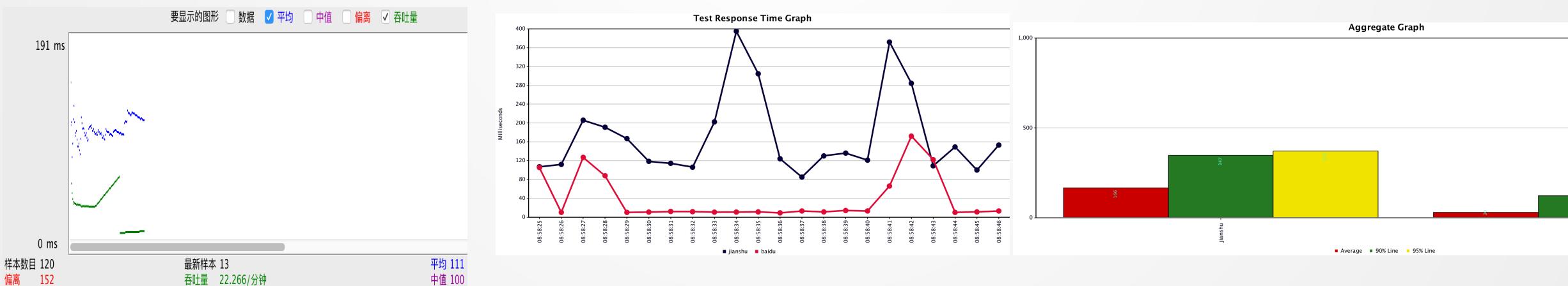
文件名 /tmp/sum.jtl 浏览... Log/Display Only:  仅日志错误  Successes Configure

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	KB/sec	Avg. Bytes
jianshu	20	166	85	443	98.32	0.00%	57.8/min	46.66	49616.3
baidu	20	30	9	172	46.90	0.00%	58.1/min	2.63	2781.0
总体	40	98	9	443	102.75	0.00%	1.8/sec	46.99	26198.7



# 报告：记录的数据在监听器中的展现

- 实时展现：运行时实时刷新
- 事后展现：在相应的监听器中加载jtl文件
- 常见的监听器介绍
  - View Result Tree (详细的请求响应信息，方便调试脚本)
  - 断言结果（查看断言错误，用于调试脚本）
  - 用表格察看结果（列表一目了然，可查看请求发出的顺序，但不能灵活排序）
  - Summary Report，聚合报告（包含统计percentile, 方差等统计信息）
  - Simple Data Writer（开销小，先记录后分析）
  - 图形结果（表现形式及其丑陋，慎入）
  - Response Time Graph（可设置采样间隔，报表格式，可导出png图片，但只处理响应时间）
  - Aggregate Graph（也较拙劣，只处理响应时间）
- 总结：**JMeter的内置报告实在泛善可陈**（考虑到其免费使用和体量的瘦小，我们暂且忍了。。。）



# JMeter Report Dashboard

- HTML的汇总报告，数据来源依然是jtl结果数据文件
- 生成报告

`./jmeter.sh -g [log-file] -o [path to output folder]`

`./jmeter.sh -n -t test.jmx -l result.jtl -e -o /path/to/dashboard/folder`

(测试并记录jtl, 执行结束时生成报告)

- 查看报告

访问output文件夹里的index.html

- 修改报告的配置参数 (参考JMeter[文档](#))

jmeter.reportgenerator.xxx

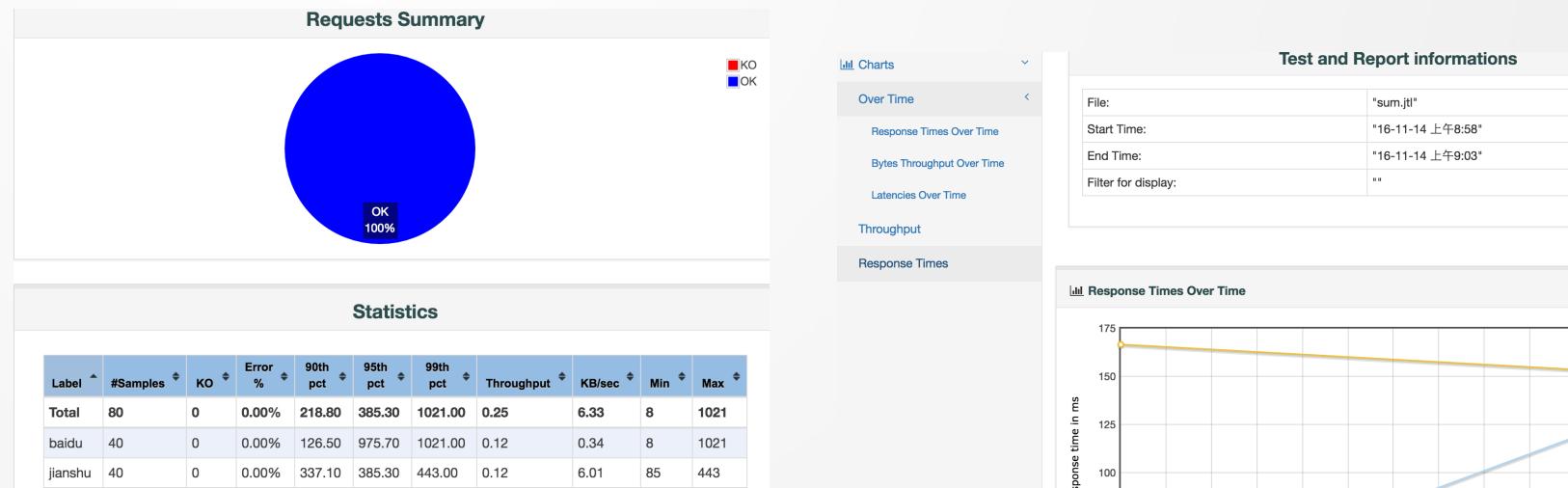
例如：jmeter.reportgenerator.overall\_granularity=60000 (缺省1分钟)

jmeter.reportgenerator.graph.responseTimeDistribution.property.set\_granularity=500

- 总结：比内置报告改善很多，但仍然不够灵活，不方便定制，扩展

The screenshot shows the Apache JMeter Dashboard interface. In the top left corner, there's a sidebar with 'Apache JMeter Dashboard' and a 'Dashboard' button. Below it is a 'Jtl Charts' button with a dropdown arrow. The main area has a title 'Test and Report informations'. It contains a table with the following data:

File:	"sum.jtl"
Start Time:	"16-11-14 上午8:58"
End Time:	"16-11-14 上午9:03"
Filter for display:	" "



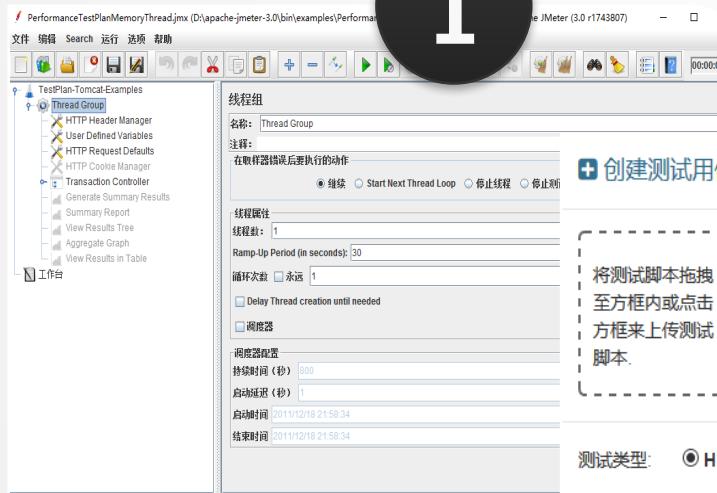
# XMeter — 性能测试云服务



**基于SaaS分布式压力测试，取代传统性能测试；**

**无需安装软件、配置环境，只需开通云端服务；**

**即需即测，三步操作，简便易行。**



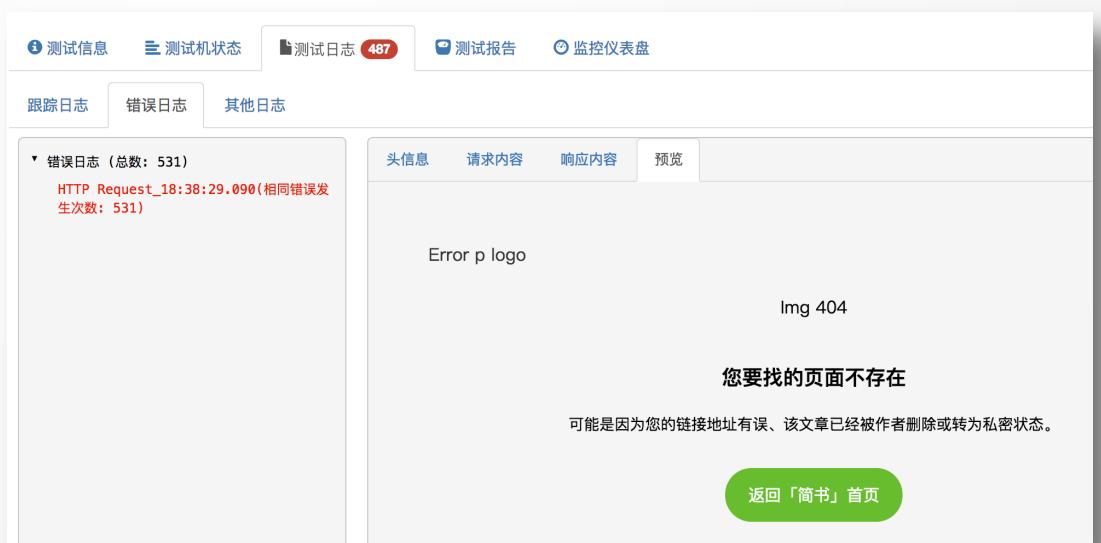
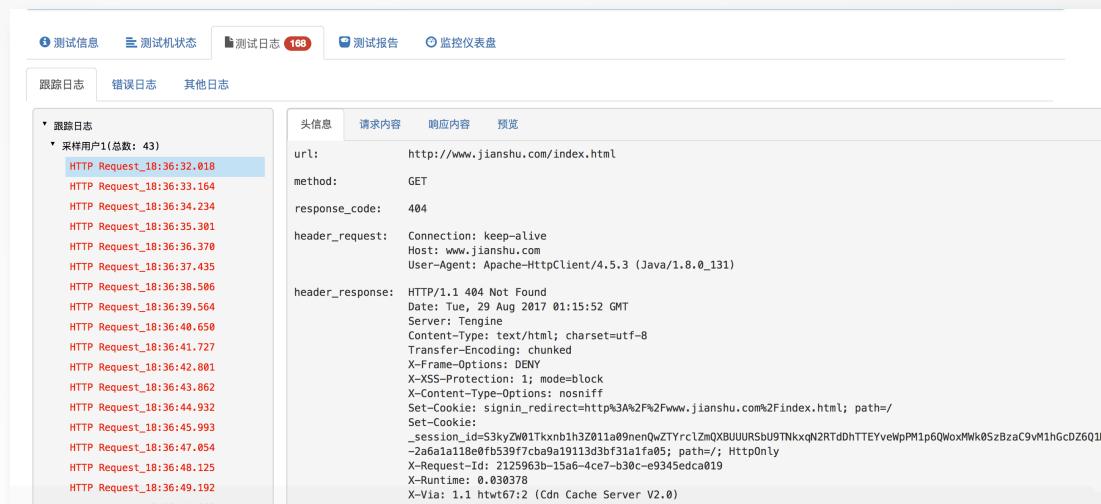
测试类型:  HTTP/HTTPS 测试  MQTT 测试

当新文件和已存在的测试用例重名时是否允许覆盖已有测试用例

关闭

上传测试用例

# XMeter性能测试平台截图



臻云科技成立于2016年，是一家专注于软件性能测试领域的云端性能测试服务提供商。成员由性能测试领域的高端人才组成，具有丰富的实践经验和技术创新，拥有自主研发的云端SaaS性能测试产品XMeter。不同于传统性能测试软件，XMeter让性能测试变得简单易行！

臻云科技致力于在软件性能测试领域为客户提供高效便捷的性能测试生产工具及技术领先的低成本性能测试解决方案。





# 谢谢!

手机 : 13718453305(金瑞妹)  
邮箱 : jinruimei@xmeter.net  
Q Q : 3448363935

