

性能测试基础知识培训

——性能测试系列培训课程 2019/03/27 共享技术部-谢佳——

性能测试理论知识篇

性能测试工具基础知识篇

JMeter安装、配置、界面及主要功能说明

JMeter HTTP协议模板&脚本录制和过滤

性能测试基础知识-需要学习哪些知识？

QPS
RT
用户量
数据量
业务场景

用户眼中的性能是？
开发眼中的性能是？
测试眼中的性能是？

哇塞这么多啊，简直让我a蒙逼b蒙逼c蒙逼。。。

CPU
负载
内存
网络io
磁盘io

Java
Python
Shell

性能测试
基准测试
并发测试
负载测试
压力测试
浪涌测试
可靠性测试
稳定性测试
抗疲劳测试
容量测试
扩展性测试
.....

操作系统
网络
数据库
中间件
开发语言
测试工具

注册用户数
在线用户数
并发用户数

性能测试基础知识-性能测试所处阶段

单元测试（白盒）

接口测试（灰盒）

功能测试（黑盒）

性能测试

安全测试

先看看性能测试处于分层测试的哪个节点。当系统无阻断性功能bug时，性能测试可以和功能测试同时开展，尽早发现系统性能问题，避免后续时间不足发现问题来不及修改

性能测试基础知识-性能测试测什么？

性能测试 ≠ JMeter 或 LR
性能测试 ≠ 代码和脚本

性能测试是要站在全系统的
角度做分析和评估，工具只
是其实施压力的手段

问题诊断

能力验证

容量规划

性能调优

基准测试

基础

扩展

回归

性能测试基础知识-不同角色对性能理解

用户视角

还要让我等多久? ——响应时间

为什么总是失败? ——稳定性

开发视角

架构设计是否合理? ——架构设计

数据库设计是否合理? ——数据库设计

代码是否存在性能问题? ——代码

是否有不合理的内存使用? ——代码

是否有不合理的线程同步操作? ——代码

是否有不合理的资源竞争? ——代码

代码算法是否还能有进一步提升? ——代码

管理员视角

服务器资源使用合理吗? ——资源利用率

数据库使用合理吗? ——资源利用率

系统能否实现扩展? ——可扩展性

最多支撑多少用户访问? ——系统容量

最大业务处理量? ——系统容量

系统有哪些潜在的瓶颈? ——可扩展性

更换哪些设备, 添加哪些机器可以提高系统性能? ——可扩展性

7 X 24 小时连续不间断业务访问? ——稳定性

性能测试基础知识-总结概述

性能测试是指借助自动化的测试工具或代码，通过多线程的方式模拟高并发下的多种正常、峰值以及异常用户量负载情况，观察系统的各项性能指标和监控状况是否达标。

性能测试是对系统的处理能力和存在风险的综合评估，包括硬件、网络、软件、技术架构、业务逻辑等多方面内容，对知识的储备量、知识宽度、人的分析和思考能力等要求较高。

很多人觉得测试入门简单，各分支想要做的精深和全面，也是需要不断积累和实践才行的。

性能测试基础知识-基本概念

负载测试：检验系统在给定负载下的性能表现、资源利用情况、是否达到预期性能指标等，通常并不强调系统的瓶颈。通常需要测试现有负载和预期负载下的性能表现，并进行对比分析。

容量测试：在已确定的环境下（软件环境、硬件设备、网络条件、基础数据等）、且满足一定条件（比如交易响应时间）的情况下，获取应用系统的最大处理能力。

稳定性测试：通过给系统加载一定的业务压力的情况下，让应用持续运行一段时间，测试系统在这种条件下是否能够稳定运行。一般让系统处于峰值压力下，进行8-10小时的测试。

疲劳强度测试：在稳定性测试的基础上增加一定比例的压力（一般在可接受最大用户数的基础上增加20%的压力）运行24-72小时。

压力测试：对特定的软硬件环境不断施加“压力”，测试系统在压力情况下的性能表现，通过确定一个系统的瓶颈或者不能接受的性能点（系统崩溃点），来获得系统能提供的最大服务级别的测试。

大数据量测试：通常是针对某些系统存储、传输、统计、处理等业务进行大数据量的综合测试，如银行业务系统中的批处理业务（日终批、联机批），主要关注处理时间窗口是否满足要求。

失效恢复测试：针对有冗余备份和负载均衡的待测系统，检验如果系统发生局部故障，用户是否能够继续使用系统；以及如果这种情况发生，用户将受到多大程度的影响（如交易响应时间是否在可接受的范围内）。

配置对比测试：通过调整系统软/硬件配置，了解不同环境对系统性能的影响，找到系统各项资源的最优分配原则，一般用于性能调优和系统规划。

性能测试基础知识-常见指标释义

VU

- 全称为 **Virtual user** , 虚拟用户数, 也叫并发用户数
- 通常情况下, 用一个线程或者一个进程执行脚本函数来模拟一个真实用户访问系统的行为
- 通过多线程来模拟不同用户对系统持续性访问造成的压力

TPS

- 全称为 **Transactions per second** , 每秒事物数
- TPS是衡量系统处理能力的重要指标, 一个页面、一个接口、一个业务流都可以当做一个事物
- 和QPS (query per second) 含义类似

ART

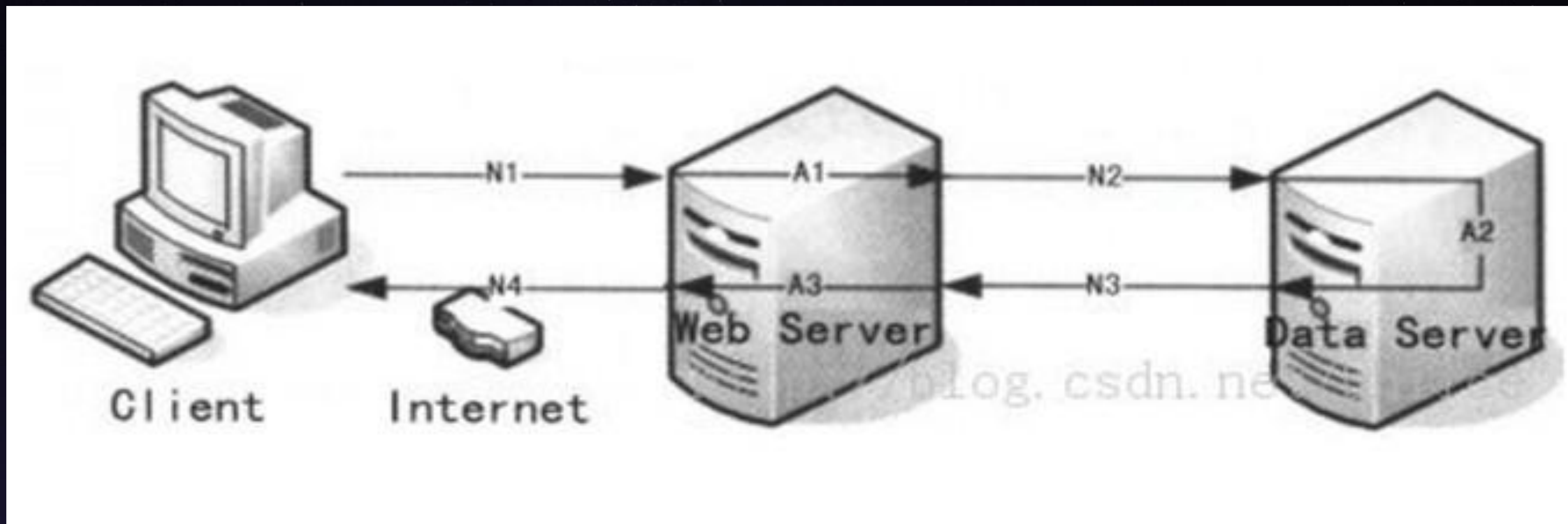
- 全称为 **Average response time** , 平均响应时间
- 响应时间是衡量系统处理快慢的重要参考指标, 关系到用户的直观使用体验, 一般统计的是页面、接口、交易或者一个完整流程的耗时

SR

- 全称为 **Success Rate** , 成功率
- 成功率是衡量系统稳定性的重要参考指标, 一般要求在99%以上

阐述这些指标时, 是要基于一定的**前置条件**才有意义, 比如: 系统的某个接口或者功能, 在**什么业务场景**、基于**多少数据量**、**多少并发用户量**情况下, 去观察统计其系统吞吐量和响应时间是多少, 事物成功率能达到多少, CPU、内存、磁盘IO、网络IO使用情况如何

性能测试基础知识-响应时间基本构成



**Response
Time**

=

$(N1+N2+N3+N4)$
+
 $(A1+A2+A3)$

网络延时 + 系统处理时间

性能测试基础知识-响应时间 “2-5-8原则”

所谓“2-5-8原则”,是指用户面对系统的不同响应速度时的直观感受和行为。它是我们开展性能测试评估系统速度的一个行业通用参考值

响应时间

< 2秒

- 用户能够在2秒以内得到响应时，会感觉系统的响应很快

2-5秒

- 会感觉系统的响应速度一般

5-8秒

- 会感觉系统的响应速度很慢，但是还可以接受

> 8秒

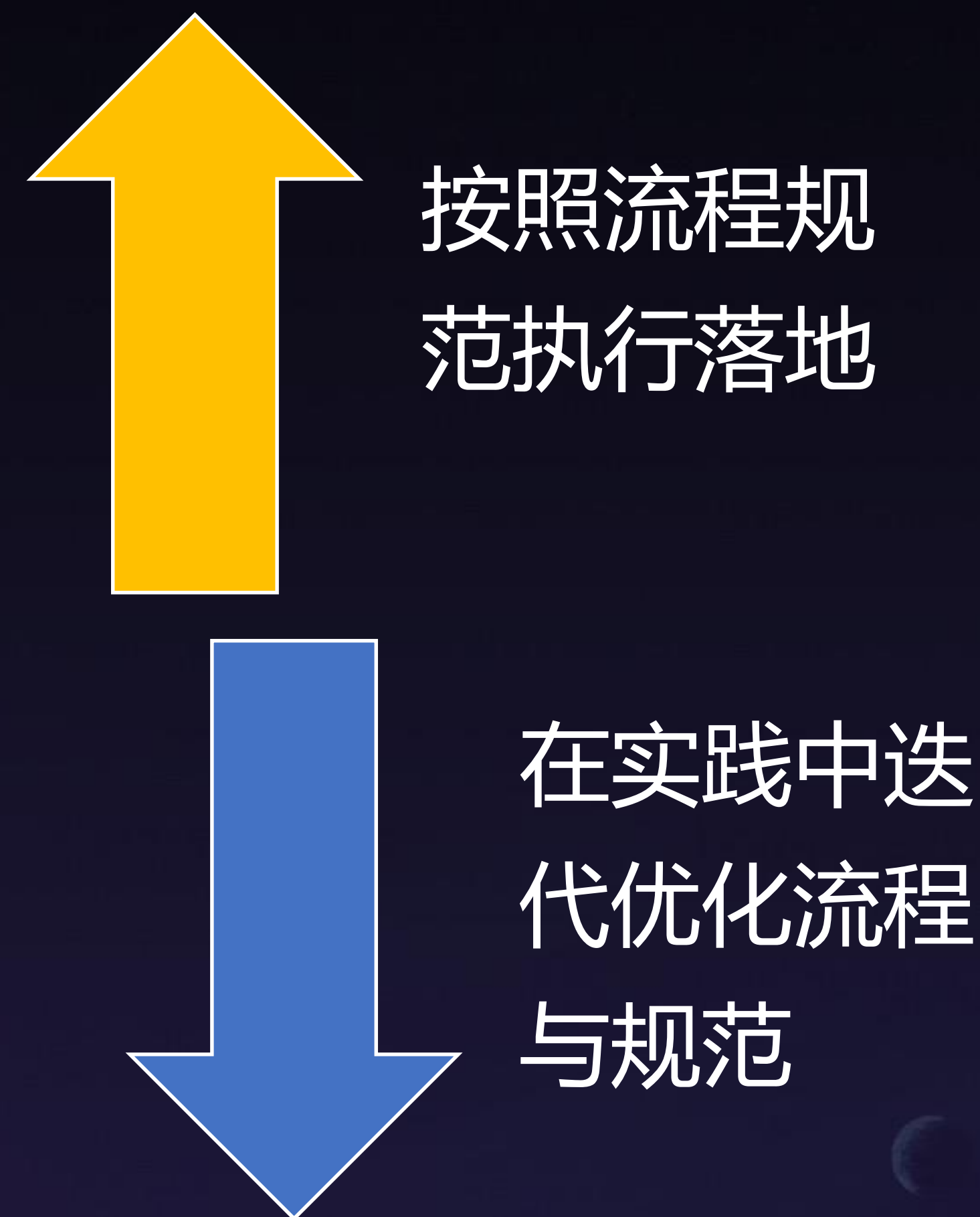
- 会感觉系统糟透了，或者认为系统已经失去响应，而选择离开这个应用，或者发起第二次请求

性能测试基础知识-性能测试开展时间踩对了？

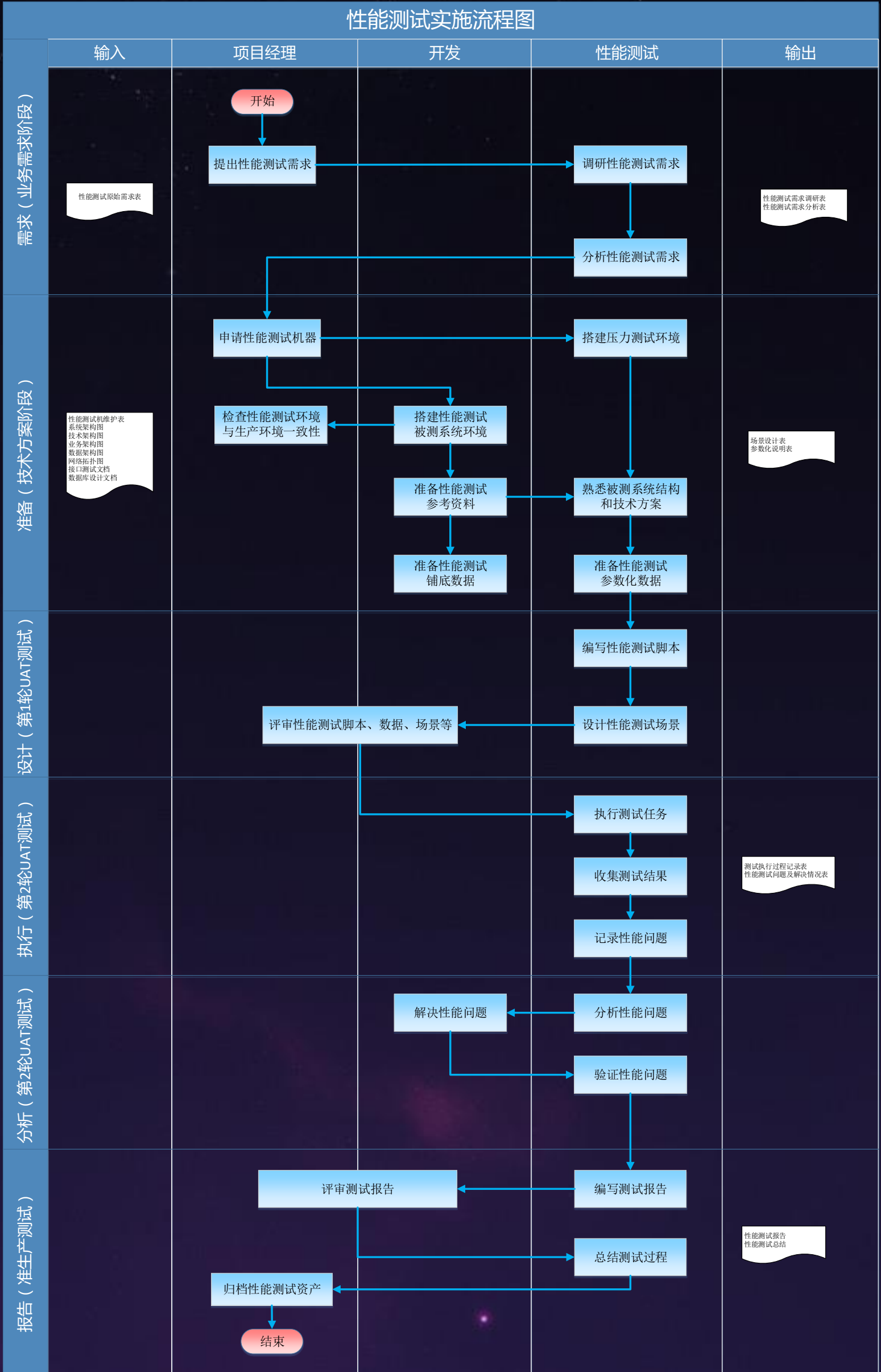


红色：被动滞后型
蓝色：主动规划型

性能测试的开展，最好是开发设计方案完成之后，此时需求、设计方案、接口文档、数据库设计文档等都非常明确了，基于这些开展性能测试的准备和设计工作；等开发版本出来时候，性能测试的各种脚本场景已准备完毕，就可以快速开展性能测试实施和执行工作。性能测试的问题发现的更早更及时，而不是等到功能测试结束后、或者临上线前去开展，不仅准备不充分，即使发现了问题，短期内也很难解决。



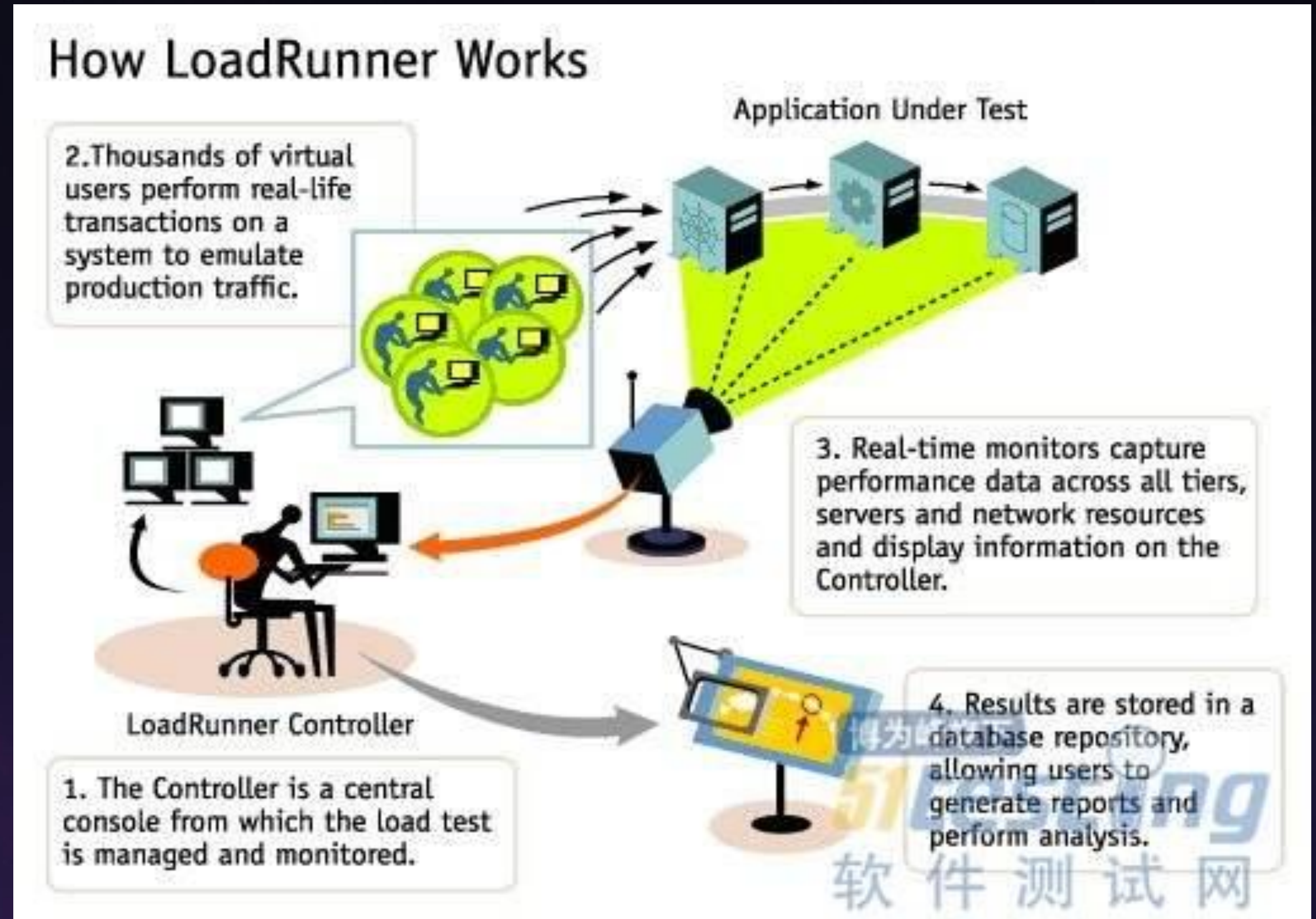
性能测试基础知识-性能测试实施流程图



性能测试基础知识-性能测试实施参考模板

序号	性能测试阶段	模板名称	备注
1	系统调研	1. 性能测试环境配置调研表.xlsx	网络拓扑、系统架构图、生产环境和压测环境硬件配置
		2. 性能指标.xlsx	系统处理能力、系统资源利用率、系统稳定性、其他应用或中间件性能指标
		3. 系统技术特征调研表.xlsx	报文格式、协议类型、同步\异步、
		4. 业务交易调研表.xlsx	典型交易占比、用户流量分布信息，以及未来业务拓展
2	需求分析	需求分析报告.docx	测试目的、测试范围、测试环境、业务交易、测试压力模型、性能指标、性能测试数据规模、用户流量分布信息
3	计划方案	性能测试方案.docx	测试简介、性能测试指标、测试约束、测试方案、测试资源、进度计划、测试风险分析
4	环境搭建	测试环境搭建.xlsx	被测系统环境、压力机环境
5	数据准备	数据表.xlsx	1. 业务系统铺底数据表 2. 脚本参数化数据表
6	脚本编写	脚本协议模板.jmx	通用组件模板
7	场景设计	1. 测试任务列表.xlsx	任务类型及优先级划分
		2. 业务模型分析.xlsx	正常业务模型、推算业务模型
		3. 性能测试场景设计执行.xlsx	1. 业务类型：单交易负载、综合交易负载 2. 测试类型：基准测试、压力测试、负载测试、并发测试、稳定性测试、批量业务测试、异常案例测试
8	测试执行	1. 执行过程记录表.xlsx	测试类型、测试场景、数据使用情况、并发用户数、并发用户加载时间、脚本循环次数、执行状态、执行指令、执行机器ip、请求地址
		2. 执行结果记录表.xlsx	测试类型、测试场景、执行结果
		3. 出错信息统计表.xlsx	测试类型、出错交易名称、错误信息描述、错误类型
9	测试分析	性能测试场景执行结果分析.xlsx	场景描述、场景执行情况、交易执行结果、TPS-VU曲线图、ART-VU曲线图、主机资源分析图、CPU使用率、内存使用情况、磁盘I/O读写情况
10	测试报告	性能测试报告.docx	测试简介、测试方案、测试实施情况、测试结果、测试结论
11	测试总结	测试总结模板.docx	测试过程描述、经验技巧积累、已解决问题的方案、遗留问题的应对策略、工具和方法的改进

性能测试工具基础知识篇-子系统构成



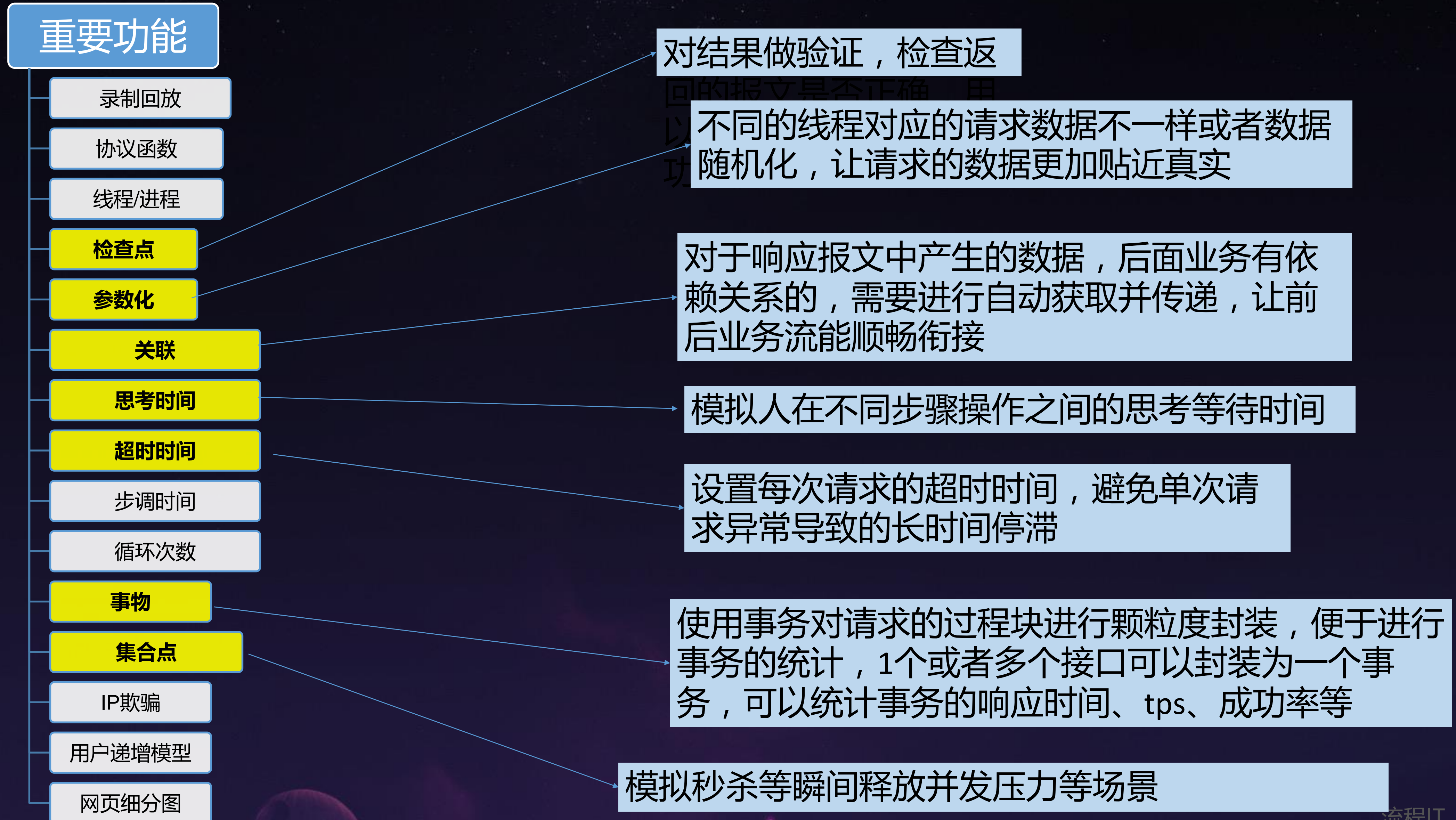
性能测试工具基础知识篇-子系统作用

从性能工具的原理划分：

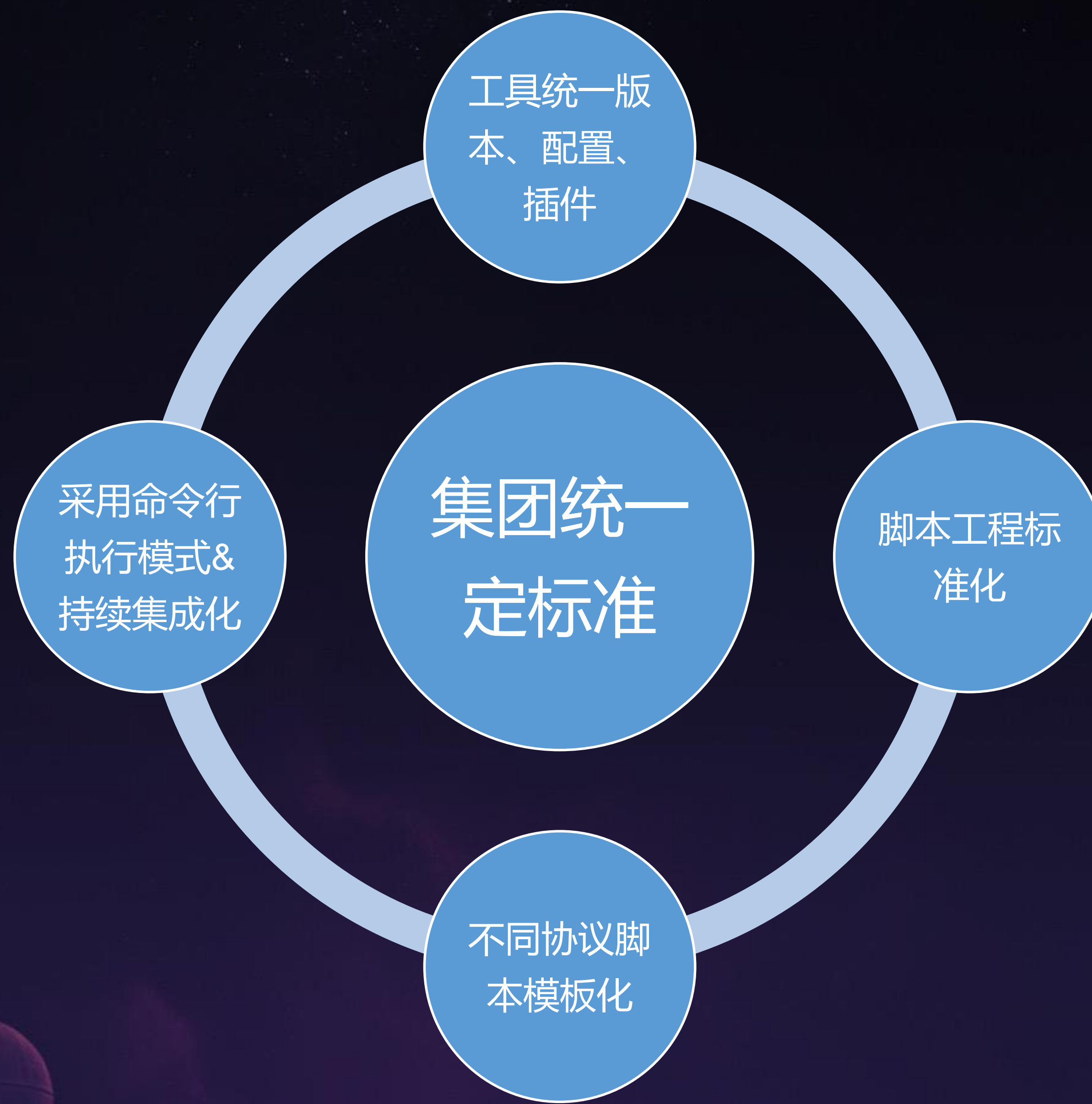
JMeter工具和其他性能工具在原理上完全一致，包含5个部分：

- （1）**脚本生成器**：通常是一个脚本运行引擎，脚本生成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。
- （2）**场景设计器**：它提供了对场景执行流程、并发用户数、用户加载和退出方式、执行时间或循环次数、集合点策略、脚本等待时间、各种超时时间、执行压力机、资源阈值等进行设置。
- （3）**任务调度器**：任务调度与分发、结果收集等。
- （4）**负载生成器**：用于产生负载，通常以多线程或是多进程的方式模拟用户行为。
- （5）**报告生成器**：根据测试统计结果数据生成报表，提供可视化的数据显示方式。
- （6）**执行代理器**：提供监听本地端口的网络通信行为，能把用户的操作行为录制为对应的脚本函数。
- （7）**性能计数器**：能够按指定的时间间隔，周期性的收集服务器的各种性能指标

性能测试工具基础知识篇-常见概念释义



JMeter-初学者如何减少基础问题概率



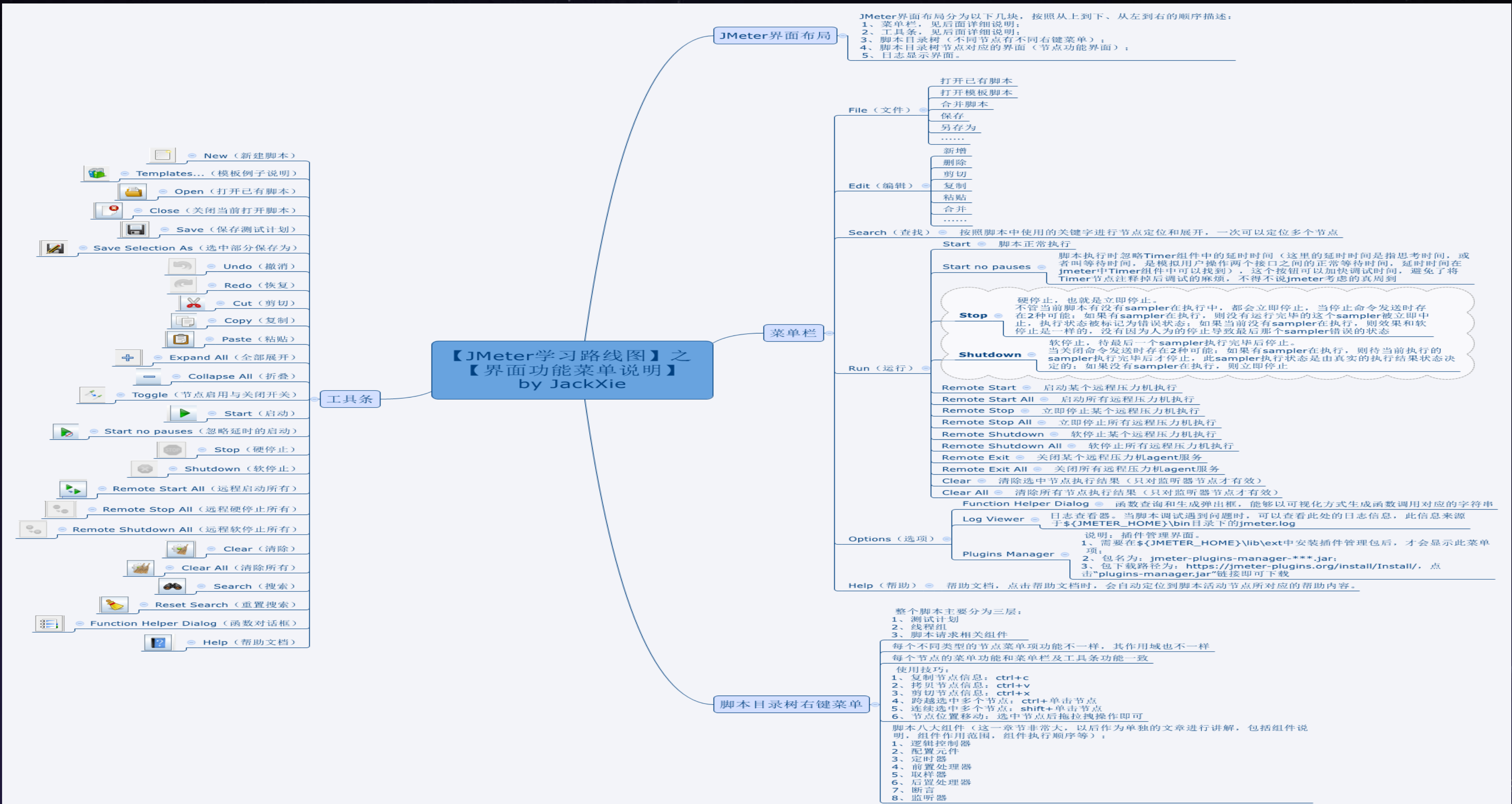
JMeter-安装与配置



【JMeter学习路线图】之 【安装与配置】 by JackXie



JMeter-界面介绍



JMeter-主要元件构成及说明

JMeter主要元件

线程组

Sampler

逻辑控制器

前置
处理
器

后置
处理
器

断言

定时器

配置元件

监听器

Test
Fragment

非测试元件
(HTTP代理服务
器)

JMeter-脚本目录树右键菜单主要元件

HTTP协议接口测试计划

用户定义的变量

CSV数据源-键值对集合

CSV数据源-多行数据表

计数器-16LengthData

HTTP请求默认值

HTTP信息头管理器

HTTP Cookie 管理器

线程组类型1-新用户-按迭代次数循环

线程组类型2-计划名称-按...

察看结果树

聚合报告

jp@gc - Active Threads (Current)

jp@gc - Response Time (Average)

jp@gc - Transactions per Second

jp@gc - Transaction Throughput

jp@gc - Response Time (Maximum)

工作台

HTTP代理服务器

添加

粘贴 Ctrl-V

Reset Gui

Undo

Redo

打开...

合并

选中部分保存为...

Save Node As Image Ctrl-G

Save Screen As Image Ctrl+Shift-G

启用

禁用

Toggle Ctrl-T

帮助

Threads (Users) ▸

Test Fragment ▸

线程组

名称: 线程组类型1-新用户-按迭代次数循环

注释:

在取样器错误后要执行的动作

☒ 继续 ☐ Start Next

添加

Start

Start no pauses

Validate

剪切 Ctrl-X

复制 Ctrl-C

粘贴 Ctrl-V

Duplicate Ctrl+Shift-C

Reset Gui

删除 Delete

Undo

Redo

逻辑控制器 ▸

配置元件 ▸

定时器 ▸

前置处理器 ▸

Sampler ▸

后置处理器 ▸

断言 ▸

监听器 ▸

元件设计顺序

测试计划

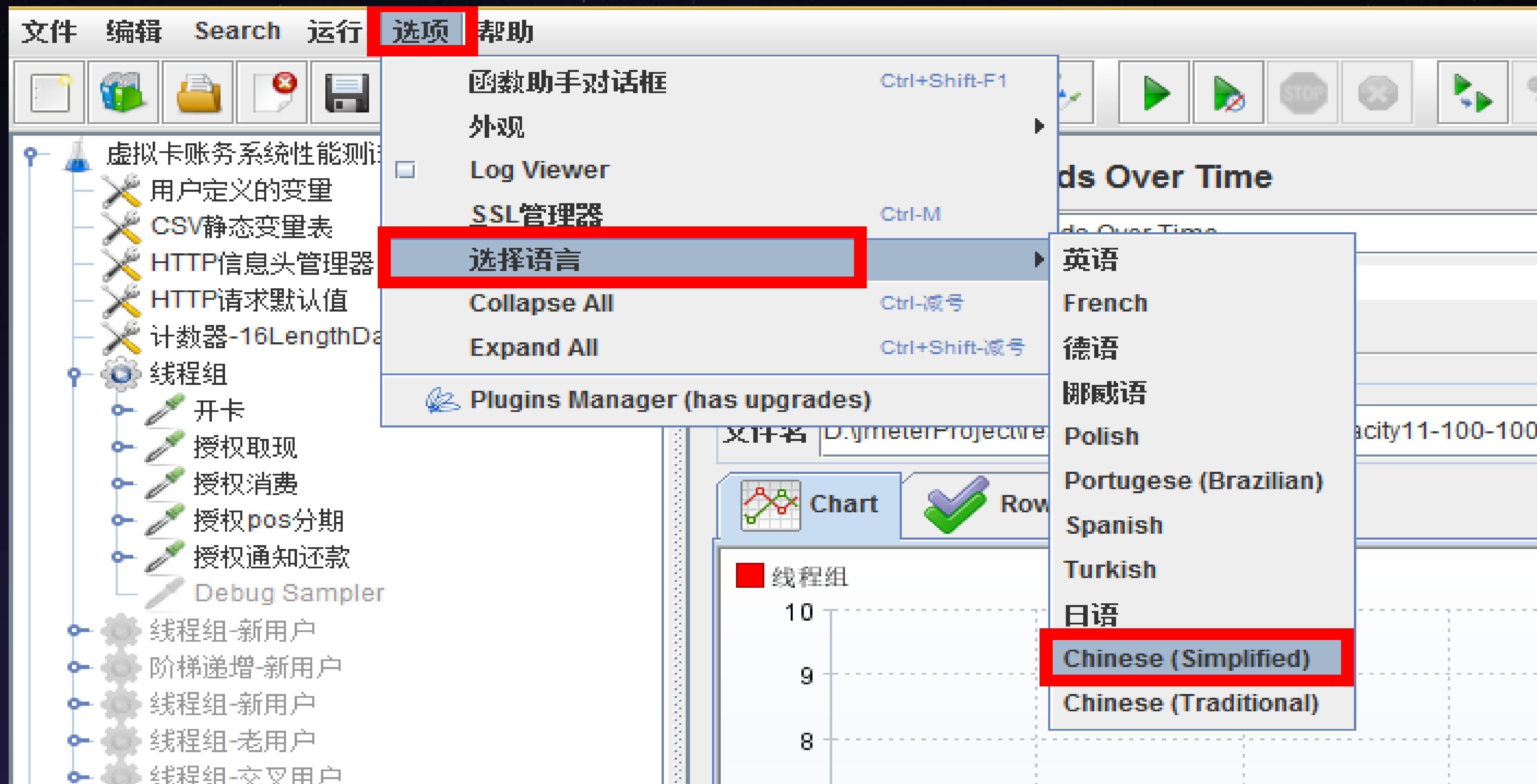
- Config (计数器、用户参数表、数据文件设置、默认请求设置等)
- 逻辑控制器 (if、switch、while、for、foreach等逻辑控制)
- Sampler
- Timer (定时器 , 包含执行延时、超时时间设置等)
- 前置处理器 (复杂逻辑处理、参数设置、请求超时设置等)
- 后置处理器 (复杂逻辑处理、参数提取)
- 断言 (检查点 , xml、json、正则)
- 监听器 (图形报表 , 所有执行完成后显示)

JMeter-元件执行顺序

元件执行顺序



JMeter-界面语言设置



JMeter-脚本执行错误处理方式

The screenshot displays the JMeter configuration interface. On the left, a tree view shows the test plan structure, including 'HTTP协议接口测试计划', '用户定义的变量', 'CSV数据源-键值对集合', 'CSV数据源-多行数据表', '计数器-16LengthData', 'HTTP请求默认值', 'HTTP信息头管理器', 'HTTP Cookie 管理器', '线程组类型1-新用户-按迭代次数循环' (selected), '线程组类型2-计划名称-按执行时间循环', '察看结果树', '聚合报告', and several 'jp@gc -' monitoring plugins. The main panel is titled '线程组' (Thread Group) and shows the configuration for '线程组类型1-新用户-按迭代次数循环'. The '名称' (Name) field is filled with '线程组类型1-新用户-按迭代次数循环'. The '注释' (Comment) field is empty. Under '在取样器错误后要执行的动作' (Action to perform after sampler error), five radio buttons are shown: '继续' (Continue), 'Start Next Thread Loop', '停止线程' (Stop Thread), '停止测试' (Stop Test), and 'Stop Test Now'. The '继续' option is selected and highlighted with a red box. The '线程属性' (Thread Properties) section shows '线程数' (Number of threads) as 10, 'Ramp-Up Period (in seconds)' as 100, and '循环次数' (Number of iterations) as 10. There are checkboxes for '永远' (Forever), 'Delay Thread creation until needed', and '调度器' (Scheduler). The '调度器配置' (Scheduler Configuration) section includes fields for '持续时间 (秒)' (Duration in seconds), '启动延迟 (秒)' (Startup delay in seconds), '启动时间' (Start time) set to '2016/12/27 18:17:50', and '结束时间' (End time) set to '2016/12/27 18:17:50'.

线程组

名称: 线程组类型1-新用户-按迭代次数循环

注释:

在取样器错误后要执行的动作

☒ 继续 ☐ Start Next Thread Loop ☐ 停止线程 ☐ 停止测试 ☐ Stop Test Now

线程属性

线程数: 10

Ramp-Up Period (in seconds): 100

循环次数 ☐ 永远 10

☐ Delay Thread creation until needed

☐ 调度器

调度器配置

持续时间 (秒)

启动延迟 (秒)

启动时间 2016/12/27 18:17:50

结束时间 2016/12/27 18:17:50

JMeter-HTTP协议元件构成

The image displays the JMeter configuration interface for HTTP components, showing four overlapping windows:

- HTTP请求默认值 (HTTP Request Default):** Name: HTTP请求默认值. Implementation: HttpClient.
- HTTP信息头管理器 (HTTP Information Manager):** Name: HTTP信息头管理器. Information stored in the information manager: User-Agent, Content-Type.
- HTTP Cookie 管理器 (HTTP Cookie Manager):** Name: HTTP Cookie 管理器. Implementation: HC4CookieHandler. Options: ☐ 每次反复清除Cookies?.
- HTTP请求 (HTTP Request):** Name: 用户第一次初始化. Implementation: HttpClient4. Method: POST. Path: /api/flow/apply. Body Data:

```
{
  "userType": "0",
  "applyNo": "${16LengthData}",
  "userId": "${16LengthData}",
  "productCode": "1101",
  "uniqueId": "${16LengthData}",
  "form": {
    "applyNo": "${16LengthData}",
    "applySerialNo": "${16LengthData}",
    "productCode": "1101",
    "subProductCode": "1101",
    "appRemark": "123",
    "subTerminalType": "IOS"
  }
}
```

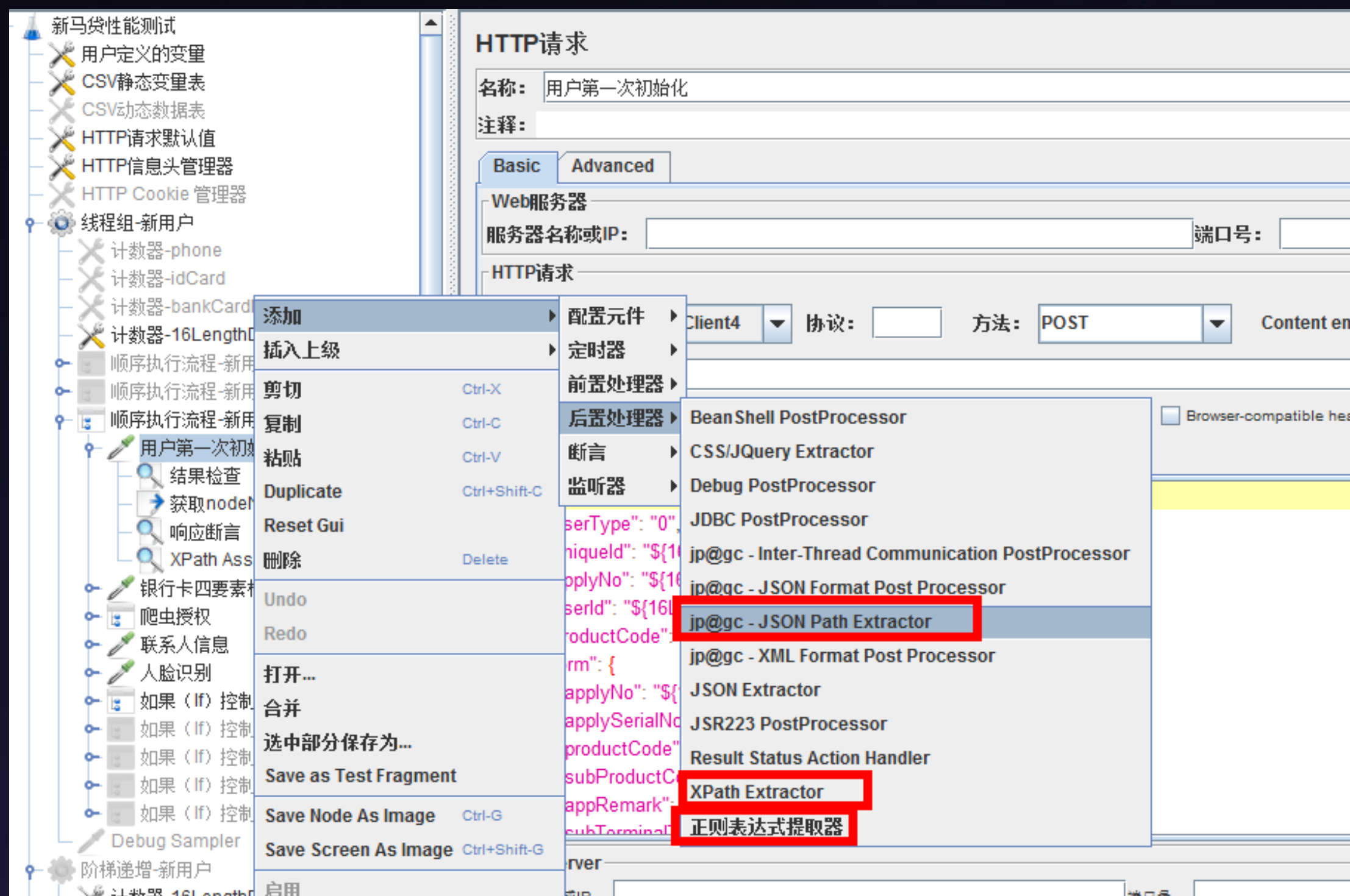

JMeter-检查点

The image displays three overlapping screenshots of the Apache JMeter configuration interface, specifically focusing on assertion plugins.

- Top Screenshot: JSON Path Assertion**
 - 名称:** 结果检查
 - JSON Path:** \$.pageCode
 - ☒ Validate against expected value
 - ☐ Match as regular expression
 - Expected Value:** 300
- Middle Screenshot: XPath Assertion**
 - 名称:** XPath Assertion
 - Apply to:** Main sample only (selected)
 - XML Parsing Options:**
 - ☐ Use Tidy (tolerant parser)
 - ☒ Quiet
 - ☐ Report errors
 - ☐ Show warnings
 - ☐ Use Namespaces
 - ☐ Validate XML
 - ☐ Ignore Whitespace
 - ☐ Fetch external DTDs

- Bottom Screenshot: 响应断言 (Response Assertion)**
- 名称:** 响应断言
- Apply to:** Main sample only (selected)
- 要测试的响应字段:** 响应代码 (selected)
- 模式匹配规则:** Equals (selected)
- 要测试的模式:** 200

JMeter-结果提取



XPath Extractor

名称: XPath Extractor

注释:

Apply to:

☐ Main sample and sub-samples ☒ Main sample only ☐ Sub-samples only ☐ JMeter Variable

XML Parsing Options

☐ Use Tidy (tolerant parser) ☒ Quiet ☐ Report errors ☐ Show warnings

☐ Use Namespaces ☐ Validate XML ☐ Ignore Whitespace ☐ Fetch external DTDs

☐ Return entire XPath fragment instead of text content?

引用名称: jsessionid

XPath query: /jsessionid

缺省值:

正则表达式提取器

名称: token提取器

注释:

Apply to:

☒ Main sample and sub-samples ☐ Main sample only ☐ Sub-samples only ☐ JMeter Variable

要检查的响应字段

☒ 主体 ☐ Body (unescaped) ☐ Body as a Document ☐ 信息头 ☐ Request Headers ☐ URL ☐ 响应代码 ☐ 响应信息

引用名称: token

正则表达式: "xAuthToken": "(.+?)"

模板: \$1\$

匹配数字 (0代表随机): 1

缺省值: false ☐ Use empty default value

jp@gc - JSON Path Extractor

名称: 获取nodeNo1

注释:

[Help on this plugin](#)

Apply to:

☒ Response Text ☐ JMeter Variable:

Destination Variable Name: nodeNo1

JSONPath Expression: \$.nodeNo

Default Value:

JMeter-报告构成

The screenshot displays the JMeter GUI. On the left, the 'Tree' view shows the test plan structure. A red box highlights the 'Report' (聚合报告) option under the 'Debug Sampler' node. On the right, the 'Report Configuration' (聚合报告配置) panel is visible, showing fields for 'Name' (名称: 聚合报告), 'Comment' (注释), and 'All data written to one file' (所有数据写入一个文件). Below these fields is a table with the following data:

Label	# Samples	Average	Median	90% Line	95% Line	99% Line
总体	0	0	0	0	0	0

Below the table, there is a list of report types that can be generated, including 'View Results Tree' (察看结果树), 'Aggregate Report' (聚合报告), 'View Results in Table' (用表格察看结果), 'Generate Summary Report' (生成概要结果), and several 'jp@gc' (JMeter Plugins) reports for monitoring user count, response time, TPS, and throughput over time.

JMeter-聚合报告指标详解



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	Received KB...	Sent KB/sec
开卡	20	252	283	403	491	614	111	614	50.00%	8.9/min	0.10	0.77
授权取现	20	363	326	468	485	602	309	602	100.00%	8.9/min	0.14	0.31
授权消费	20	170	166	224	228	270	126	270	50.00%	9.0/min	0.16	0.31
授权pos分期	20	165	136	197	255	265	124	265	50.00%	9.0/min	0.16	0.31
授权通知还款	20	160	161	199	220	264	122	264	50.00%	9.0/min	0.16	0.31
总体	100	222	181	337	415	602	111	614	60.00%	44.3/min	0.71	2.00

JMeter-脚本录制及过滤

1、配置JMeter

添加HTTP代理服务器

配置代理端口

选择脚本存放路径：使用录制控制器

设置脚本过滤条件：(?!).*\.(bmp|css|js|gif|ico|jpe?g|png|swf|woff|woff2|html|aspx|php|cgi|gif)

生成证书：启动HTTP代理服务器证书会自动生成，生成路径：
//JMeter_HOME/bin/ApacheJMeterTemporaryRootCA.crt

安装证书：ApacheJMeterTemporaryRootCA.crt

设置浏览器代理端口号

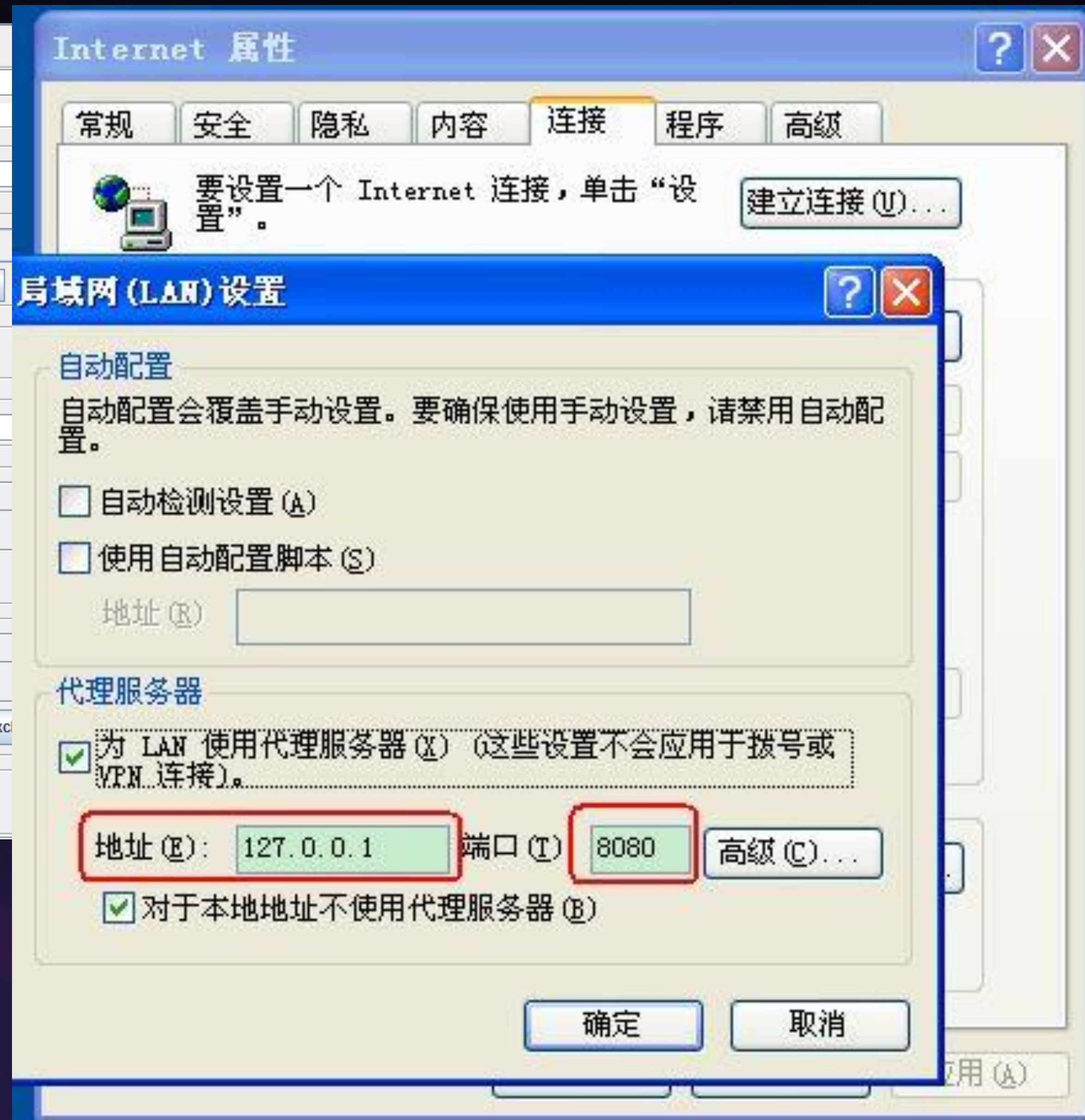
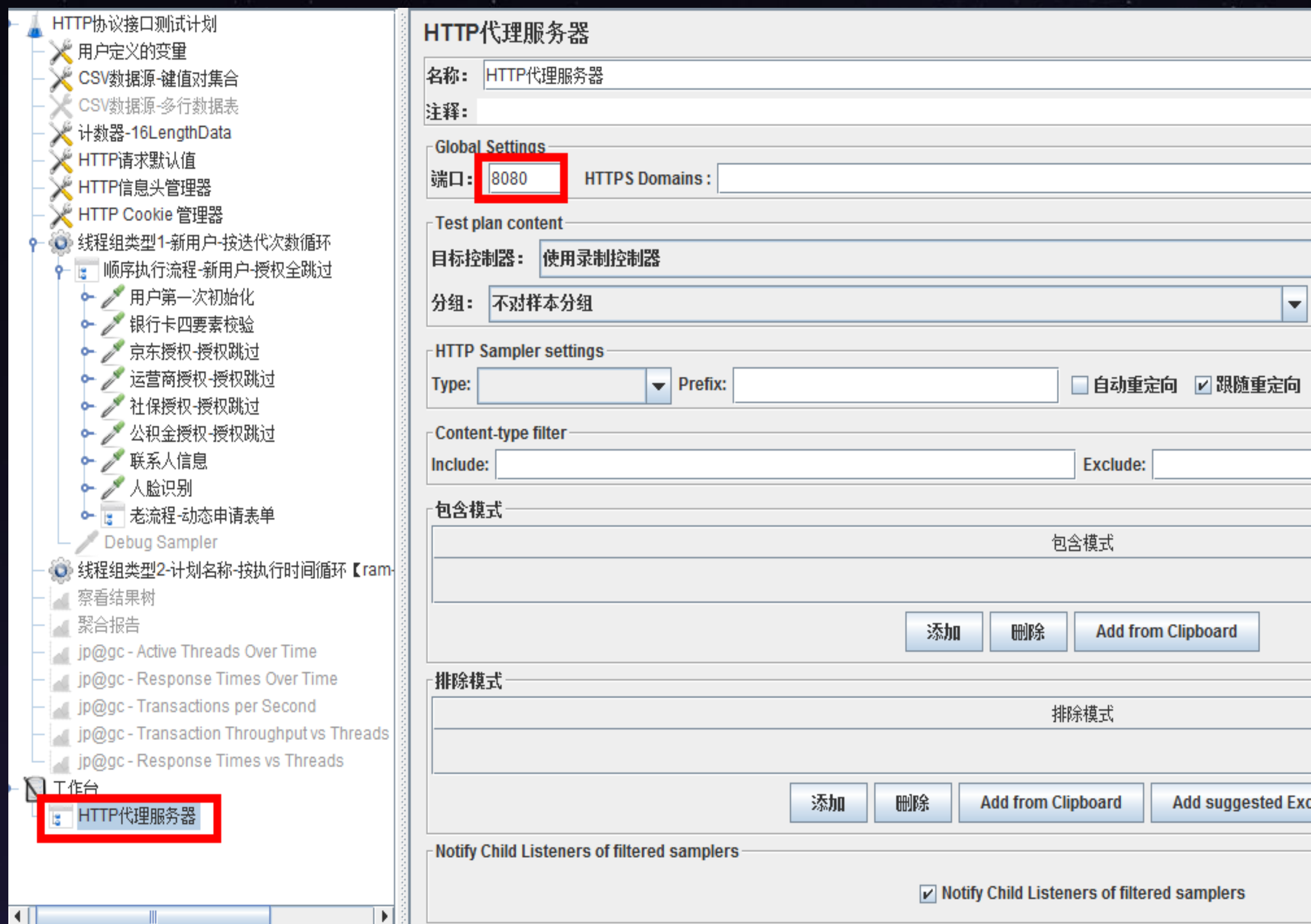
使用浏览器访问系统，JMeter会将脚本录制下来

录制步骤

2、安装证书

3、配置Browser

JMeter-脚本录制及过滤



学习参考地址-百度云盘学习视频分享

百度云盘JMeter学习视频分享

链接：https://pan.baidu.com/s/1_9lvQ5jXOGM4dSfux30_bQ

提取码：4qv5



学习参考地址-公司wiki学习地址

公司confluence学习地址

<http://confluence.midea.com/pages/viewpage.action?pageId=21466592>

<http://confluence.midea.com/pages/viewpage.action?pageId=17865945>

性能测试

- 性能测试体系
- 性能测试工具
 - JMeter
 - JMeter使用规范标准**
 - JMeter学习路线图
 - 【JMeter学习路线图】之【JM...
 - 【JMeter学习路线图】之【JM...
 - 【JMeter学习路线图】之【JM...
 - 【JMeter学习路线图】之【JM...
 - 【JMeter学习路线图】之【JM...
 - 【JMeter学习路线图】之【JM...
 - 【JMeter学习路线图】之【JM...
 - 【JMeter学习路线图】之【JM...
 - 脚本模板
- 接口测试
- 流程管理规范
- 测试周报
- 线上问题跟踪列表

空间管理

页面 / ... / JMeter

JMeter使用规范标准

创建：谢佳，最新修改：陈国涛 于一月 15, 2019

本文档主要以附件形式体现，包含了以下5个附件：

- ✓ JMeter工程与脚本命名规范_V1.0_R1_201805191800.xlsx
- ✓ JMeter脚本执行过程记录表_V1.0_R1_201805191800.xlsx
- ✓ JMeter命令行参数详解及范例_V1.0_R1_201805191800.xls
- ✓ 数据准备-被测系统数据库铺底数据说明.xlsx
- ✓ 数据准备-测试脚本参数化数据说明.xlsx

远程站点: /home/jmeter/jmeterProject/systemName

- jmeter
 - apache-jmeter-3.1
 - 系统名称
 - jmeterProject
 - 子系统名称
 - systemName
 - 数据及配置文件夹
 - subSystemName
 - data
 - result
 - 测试报告
 - html
 - 测试日志：执行日志和统计日志
 - ttl
 - 测试脚本
 - script
 - zjcgpt

性能测试工程命名规范说明：

1. 压力机的性能测试根路径“/home/jmeter/”，此路径下有2个子目录：

页面 / ... / 培训

HTTP协议详解（20180718培训ppt材料）

创建：谢佳，最新修改：匿名用户 于 七月 23, 2018

MIP问题整改计划

安全管理工作

性能测试

接口测试

培训

- HTTP协议详解（20180718培训ppt材料）
- JMeter-http协议接口测试（20180721）
- JMeter-使用命令行生成html报告（20180801）
- JMeter-命令行执行详解（20180830）
- JMeter常用功能介绍（20180801培训）
- JMeter录脚本&过滤脚本（20180919培训）
- JMeter-聚合报告详解（20180823培训）
- JMeter脚本工程目录结构及命名规范
- 接口测试前为什么要学http协议？

PDF

5人赞了它



THANKS