



接口压力测试入门

—— 大卫



**霍格沃兹测试学院致力于培养专业的测试人才，推动测试行业的技术更新和发展，
本着此宗旨，为您提供最新的测试技术培训和实战，让您真正的可以在测试能力上获得提升。**



前言

希望大家通过学习接口压力测试课程，了解压力测试的几点用途。

- 通过压力测试验证系统的性能指标是否满足需求；**
- 及早发现系统潜在的性能风险，提前防范；**



第一章 压力测试的用途和场景

- 压力测试定义
- 压力测试用途
- 压力测试场景





什么是压力测试

压力测试是对系统不断施压的测试，是通过确定一个系统的瓶颈或不能接收用户请求的性能点，来获得系统能提供的最大服务级别的测试。

例如，模拟实际软硬件环境，在超出用户常规负荷下，长时间运行测试工具来测试被测系统的可靠性，和测试被测系统的响应时间，目的是在极限负载下识别程序的弱点。



什么是接口测试

接口测试的原理是，通过测试程序或工具，模拟客户端向服务器发送请求报文，服务器接收请求报文后对相应的报文做出处理，然后再把应答报文发送给客户端，客户端接收应答报文这一个过程。





服务器设备

CPU主频1GHz以上，内存1GB以上，硬盘自由空间1GB以上。

支持软件

操作系统：Windows2003 Server或Windows XP

数据库服务器：MySQL-5.1.28

应用服务器：Tomcat6.0

Java：JDK1.6.0_07

应用软件：Liferay Portal 5.1.1

浏览器：IE6+sp2

Word：office 2000或office XP或office 2003



并发/在线用户数

并发用户数是指在某一给定时间内，某个特定点上进行会话操作的用户数，即与服务器进行交互的在线用户数量。并发用户数并不是在线用户数，比如用户在浏览一个网站时，虽然在线，但是并没有与服务器相互交互，所以该用户并不属于并发用户，属于在线用户。



响应时间

响应时间指的是客户端发出请求到得到响应的整个过程所经历的时间

吞吐量/系统处理能力

单位时间内能处理多少事务/请求/单位数据等



测试需求	测试过程说明	过程标引
登录系统 (login)	单用户执行登录系统	1.1
	10用户并发执行登录系统	1.2
	20用户并发执行登录系统	1.3
查询稿件 (search_menuscript)	单用户执行查询稿件	2.1
	10用户并发执行查询稿件	2.2
	20用户并发执行查询稿件	2.3
新增稿件 (add_menuscript)	单用户执行新增稿件	3.1
	10用户并发执行新增稿件	3.2
	20用户并发执行新增稿件	3.3



压力测试指标需求通常涵盖以下：

本次性能测试需要测试的性能指标包括：

- 1、交易响应时间：核心系统处理交易的平均响应时间
- 2、交易吞吐量：后台主机每秒能够处理的交易笔数（TPS）
- 3、并发交易成功率
- 4、批处理效率
- 5、资源使用指标：前置和核心系统各服务器CPU占用率、内存占用率、I/O占用率；LoadRunner压力产生器CPU占用率、内存占用率



第二章 JMeter的使用

- JMeter基本介绍
- JMeter使用方法





JMeter介绍

Apache JMeter 是Apache组织的开放源代码项目，是一个100%纯Java桌面应用，用于压力测试和性能测量。

安装包: <http://jakarta.apache.org>



安装要求：

JVM：Java版本1.8以上。需要配置JAVA_HOME、PATH和CLASSPATH。需要保障JAVA版本满足待安装JMeter4.0的需求。

操作系统：JMeter是一个100%的Java应用，它可以在当前任何一个已经部署了Java的操作系统上运行。

内存、CPU和存储：根据压力要求。



安装方法：

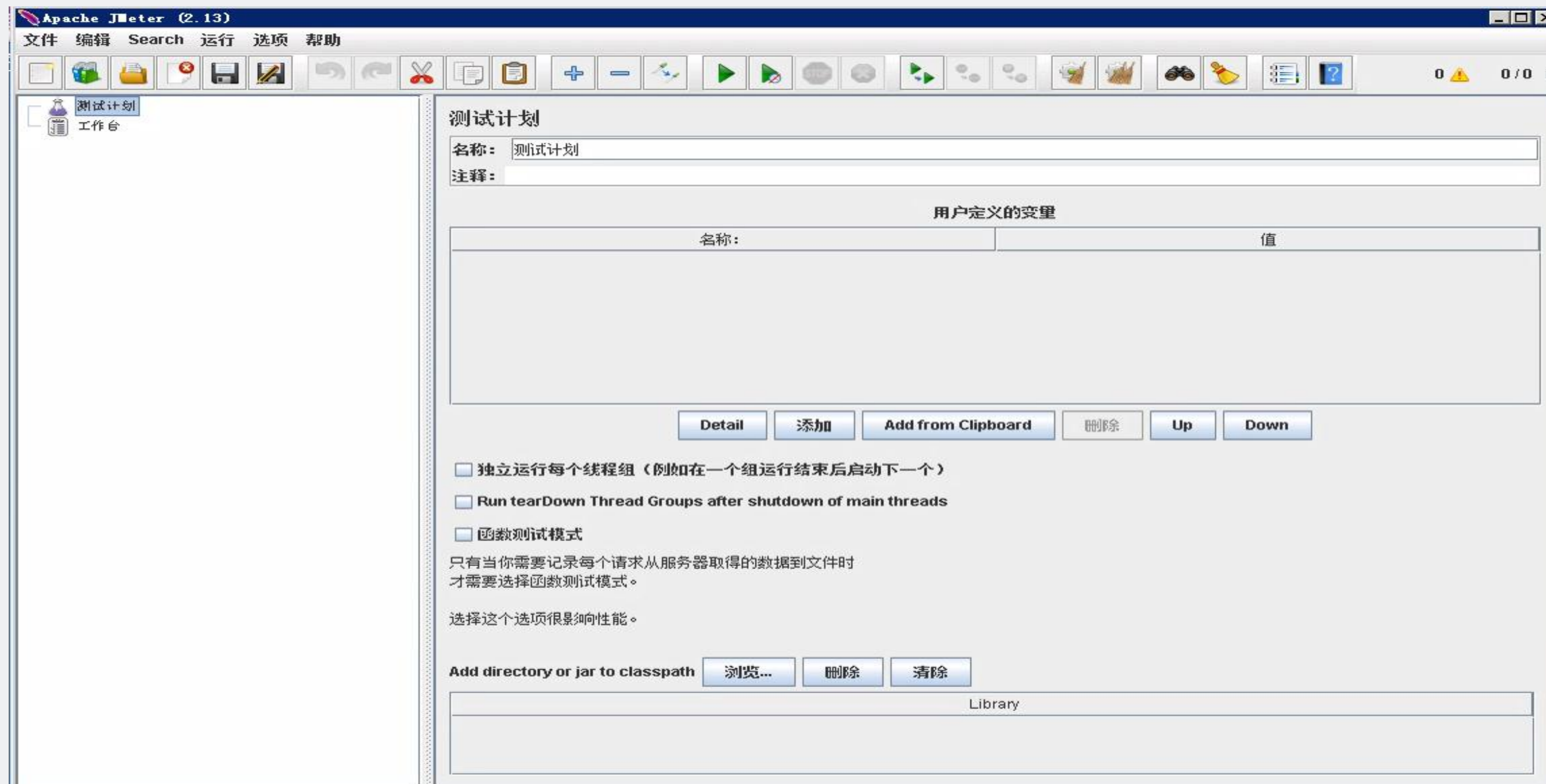
安装JMeter，只需解压zip或tar文件到你想要安装的目录里。

Windows版本执行机需要配置JMETER_HOME。

若需要JMeter自身外的功能，需要安装插件。



Windows下运行bin目录下的 jmeter.bat





Apache Jmeter模拟测试功能包括：

1. 能够加载和性能测试许多不同的应用程序/服务器/协议类型：Web - HTTP , HTTPS (Java , NodeJS , PHP , ASP.NET , ...)
2. SOAP / REST Web服务
3. FTP
4. 数据库通过JDBC
5. LDAP
6. 面向消息的中间件 (MOM) 通过JMS
7. 邮件 - SMTP (S) , POP3 (S) 和IMAP (S)
8. 本机命令或shell脚本
9. TCP
10. Java对象



Http报文

连接: Connection

一个传输层的实际环流，它是建立在两个相互通讯的应用程序之间。

在http1.1, request和reponse头中都有可能出现一个connection的头，此header的含义是当client和server通信时对于长链接如何处理。

消息: Message

HTTP通讯的基本单位，包括一个结构化的八元组序列并通过连接传输。

请求: Request

一个从客户端到服务器的请求信息包括应用于资源的方法、资源的标识符和协议的版本号。

响应: Response

一个从服务器返回的信息包括HTTP协议的版本号、请求的状态(例如“成功”或“没找到”)和文档的MIME类型。



线程组

线程组元件是任何测试计划的起点。一个测试计划的所有元件必须在一个线程组下。线程组元件控制JMeter运行测试时使用的线程数。线程组元件包括下面3个参数：

- ❑ 设置线程数（这里的线程数相当于LR的虚拟用户的概念）
- ❑ 设置ramp-up period(in seconds)
- ❑ 设置循环次数





测试计划



工作

添加

粘贴 Ctrl-V

Reset Gui

打开...

合并

保存为...

Save Node As Image Ctrl-G

Save Screen As Image Ctrl+Shift-G

启用

禁用

帮助

Threads (Users)

配置元件

定时器

前置处理器

后置处理器

断言

监听器

线程组

名称:



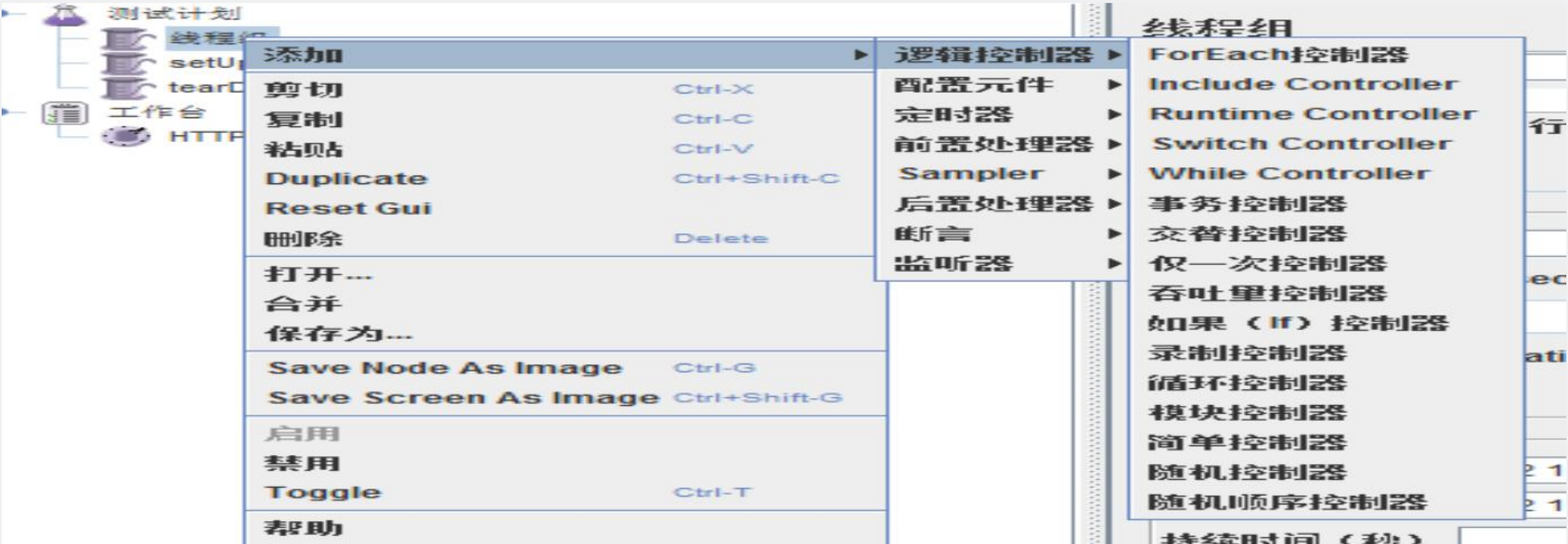
采样器

采样器（ sampler ，又叫取样器 ）告诉JMeter发送一个请求到指定服务器，并等待服务器的请求。采样器会按照其在测试树中的顺序去执行，还可以用逻辑控制器来改变采样器运行的重复次数。



逻辑控制器（Controller）

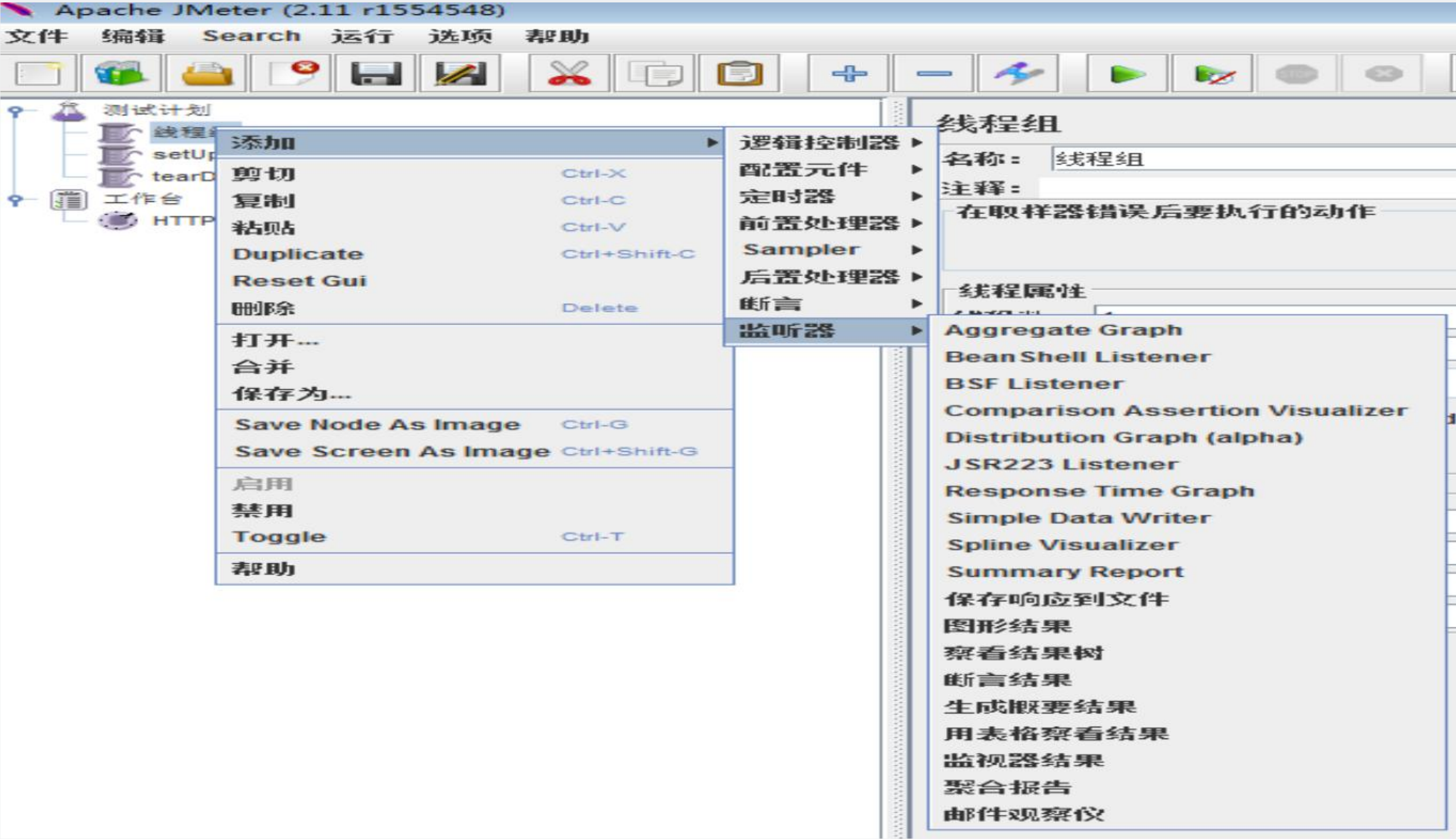
可以帮助用户控制JMeter的测试逻辑，特别是何时发送请求。逻辑控制器可以改变其子测试元件的请求执行顺序。





监听器 (Listener)

提供了对JMeter在测试期间收集到的信息的访问方法。"图形结果"监听器会将系统响应时长绘制在一张图片之中。"查看结果树"监听器会展示采样器请求和响应的细节，还可以将测试数据导入到文件之中，以供后续分析。





定时器

定时器（Timer）会让作用域内的每一个采样器都在执行前等待一个固定时长，如果不设定这种延迟，JMeter可能会在短时间内产生大量访问请求，导致服务器被大量请求所淹没。如果为线程组添加了多个定时器，那么JMeter会将这些定时器的时长叠加起来，共同影响作用域范围内的采样器。定时器可以作为采样器或者逻辑控制器的子项，目的是只影响作用域内的采样器。



测试计划

线程组

添加

Ctrl-X

Ctrl-C

Ctrl-V

Ctrl+Shift-C

Delete

Ctrl-G

Ctrl+Shift-G

Ctrl-T

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

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

☒ 调度器

调度器配置

启动时间2015/06/02 13:51

结束时间2015/06/02 13:51

持续时间（秒）

启动延迟（秒）



断言

用户可以使用断言（ Assertions ）来检查从服务器获得的响应内容。通过断言可以测试服务器返回的响应与测试人员的期望是否相符。

在JMeter中，断言就类似于Loadrunner中的检查点，但比Loadrunner中的检查点功能强大。它对上一个请求返回的信息，可以做字符串、数据包大小、HTML、XML、图片等做判断，确保返回信息的准确性。



建立测试计划

创建一个基础的测试计划来测试网站。

5个用户向<http://47.100.105.183/login>网站网页发送请求。

让每个用户发送两次。这样，总的请求数为：（5个用户）
 \times （2个请求） \times （重复2次）=20个HTTP请求。



作业1

创建一个测试计划

5个用户向http://47.100.105.183/login网站网页发送请求。

让每个用户发送两次。这样，总的请求数为： $(20\text{个用户}) \times (15\text{个请求}) \times (\text{重复}5\text{次}) = 1500\text{个HTTP请求}$ 。

并设置1秒的定时。

作业文件保存为 0908_01_姓名+.jmx 文件提交。



第三章 并发控制与场景定义

- 用户并发控制
- 压测场景定义





试验阶段	任务	交付
评估阶段	收集需求，测试范围，测试目标	性能检查表
计划阶段	最后确定测试范围、里程碑（阶段）、目标、工具	性能计划
设计阶段	设计详细的测试用例，转换为测试脚本	性能测试用例
执行阶段	创建场景；执行场景	性能测试结果
性能调整阶段	分析问题和重新测试	再测试结果
后生产阶段	回顾最后的性能测试结果	性能测试报告



手工场景（ Manual Scenario ）：

创建虚拟用户组，设置虚拟用户数目以及其它运动信息。

面向目标场景（ Goal-Oriented Scenario ）：

先定义测试要达到的目标，然后自动基于这些目标创建场景，运行过程中，会不断地把结果和目标相比较，以决定下一步怎么。



场景设计要素

制定计划： 测试点，测试环境，测试数据，测试方法，etc

开发脚本： 业务操作，参数化，etc

创建运行场景：测试模型，并发数，运行时间，etc

执行测试： 自动执行

监视场景： TPS，响应时间，吞吐量，CPU占用，硬盘占用,etc

分析测试结果：图表，日志。



根据业务量（例子）：

网站稿件管理核心系统当前生产环境高峰日交易总量为7500笔。根据二八原则（80%的交易量发生在20%的时间段内当前生产环境对主机的交易吞吐量指标要求为：），

$$\text{TPS}_1 \geq 7500(\text{交易}) * 80\%(\text{交易量}) / (24(\text{小时}) * 20\% * 3600(1\text{小时} \\ 60\text{分钟} * 1\text{分钟} 60\text{秒})) = 0.34 \text{ 笔/秒} * 3$$

根据规划，网站稿件管理系统未来1年内核心系统的处理能力应达到高峰日交易总量10000笔，则3年后对主机的交易吞吐量指标要求为：

$$\text{TPS}_2 \geq 10000 * 80\% / (24 * 20\% * 3600) = 0.46 \text{ 笔/秒}$$



常用的监控指标：

Average：事务的平均响应时间

Min：事务最少响应时间

Max：事务最大响应时间

90%Line：90%的响应时间

Std. Deviation：标准差

Fail：错误的事务

Pass：通过=**100用户，事务成功率100%，1000用户，**

事务成功率95%，

Error：错误

Throughput/sec：吞吐量



基准测试

在测试环境经过确认，脚本预验证之后对本次测试涉及的全部联机交易做基准测试。目的是验证测试脚本及后台环境、初步检查交易本身是否存在性能缺陷。

目的：是获取单用户执行时的各项性能指标，为多用户并发和混合场景的性能测试分析提供参考依据；



并发测试 ≥ 2

并发测试是指并发不同数目的虚拟用户执行检查点操作，目的是对检查点进行压力加载测试。预测系统投入使用后在一定用户压力情况下的系统响应时间,根据此响应时间分析、确定系统存在的性能瓶颈，为系统的优化和调整提供依据。



综合场景测试

通过对系统体系机构和功能模块的分析以及对系统用户的分布和使用频率的分析，来构造系统综合场景的测试模型，模拟不同用户执行不同操作，如10%的用户执行登录操作，50%的用户执行查询操作，40%的用户执行上传文档操作，最大限度地模拟系统的真实场景，使用户预知系统投入使用后的真实性能水平。从而，对系统做出相应的优化及调整，避免实际情况中出现系统长时间不响应及崩溃的情况。



测试数据

数据对象	数据	数据文档	位置
登录用户	用户名 密码	user.dat	D:\LR_Project\ Data
查询稿件	稿件名称 稿件版本号 稿件标题 稿件内容	search_manuscript.dat	
增加稿件	稿件标题 稿件内容	add_manuscript.dat	
显示稿件	稿件编号	show_manuscript.dat	
查询文档	文档名称	search_document.dat	
上传文档	文档名称 文档重命名 文档说明	upload_document.dat	



场景一								
序号	模块名称	功能点	执行用户数	用户加载方式	Duration	Pacing, 限定	Think time +时间	备注
1	稿件	查询稿件 (20VU)	50	每2秒加载1个	1h	Random:6s-9s	Random50%-150%	
2		显示稿件 (15VU)						
3		新增稿件 (15VU)						



场景二								
序号	模块名称	功能点	执行用户数	用户加载方式	Durati on	Pacin g	Think time	备注
1	文档+稿件	查询文档 (20VU)	50	每2秒加载1个	1h	Rand om:6s-9s	Rand om50%-150%	



第四章 常用性能指标及关系分析

- 响应时间与并发关系
- 系统处理能力





常用性能指标的名称与含义

并发：所有用户在同一时刻对系统执行操作，一般指做同一件事情或操作。

在线：所有用户在一段时间内对系统执行操作。

请求响应时间 从client端发出请求到得到响应的整个时间；包括：

client端响应时间<-->网络响应时间(传输时间)<-->服务器响应时间<--> Server端响应间<-->数据库相应时间事务请求响应时间完成相应事务所用的时间；这个是性能测试中重点关注的指标。



TPS (Transaction Per Second-**事务-业务意义**)

每秒钟系统能够处理的交易或事务的数量。它是衡量系统处理能力的重要指标。**TPS**是中重要的性能参数指标。

点击率 (**Hit Per Second-请求。**)

每秒发送的HTTP请求的数量；点击率越大对Server的压力越大
资源利用率

对不同资源的使用程度，如CPU，I/O,内存，.....



对象	计数器名称
mysql	Available Mbytes 可用内存
Memory	Page/sec Page Faults/sec Pages Input/sec Page Reads/sec Transition Faults/sec
Process	Working Set
System	Processor Queue Length
Processor	%Processor Time
PhysicalDisk	%Disk Time



性能常用分析工具

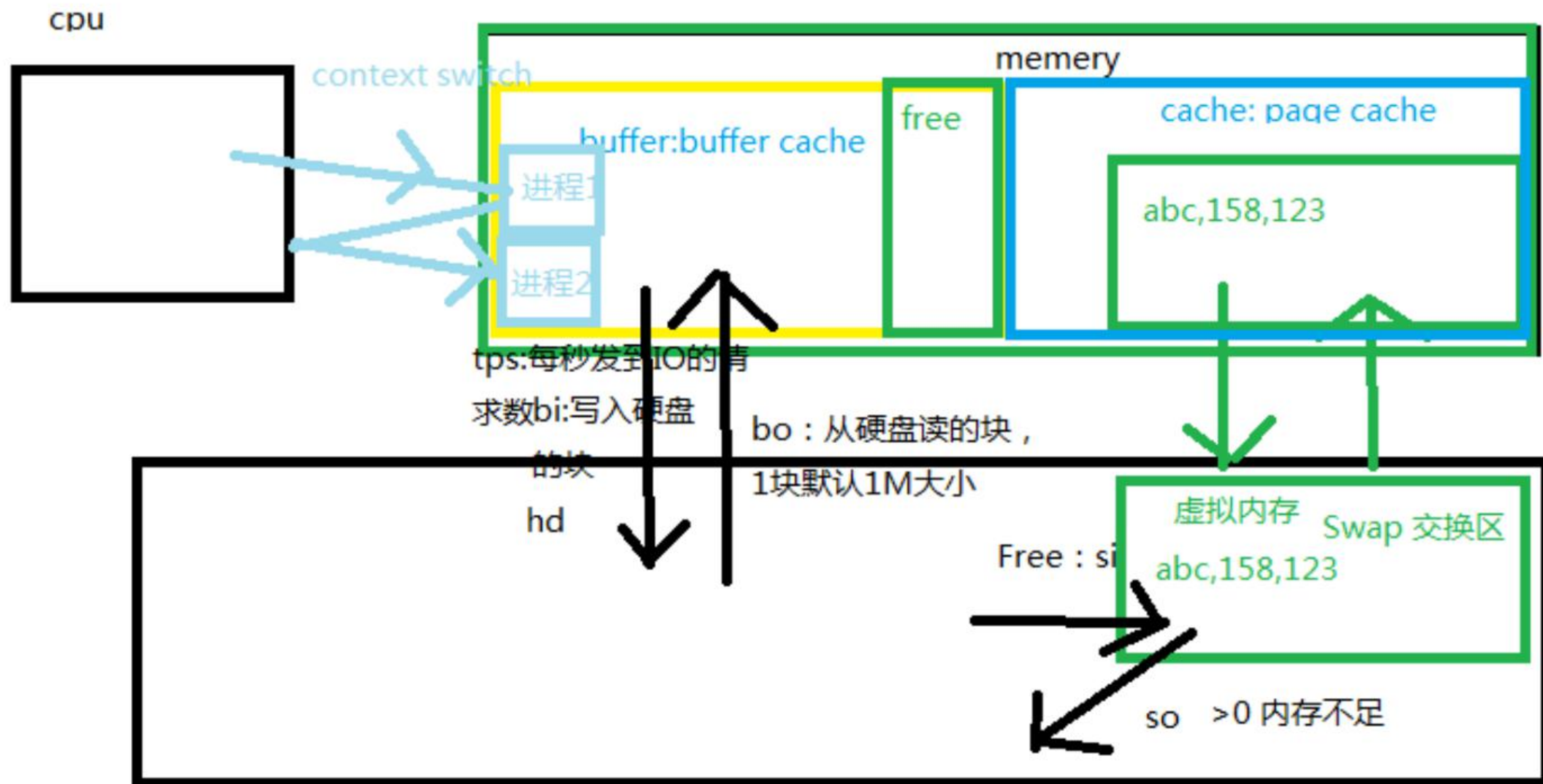
Top命令 iostat命令分析各个磁盘的传输闲忙状况
iostat命令分析各个磁盘的传输闲忙状况

vmstat 命令 报告关于内核线程、虚拟内存、磁盘、陷阱和CPU活动的统计信息

netstat 是用来检测网络信息的工具

sysctl 是一个可用于改变正在运行中的Linux系统的接口。用sysctl可以读取设置超过

第四章 常用性能指标及关系





事务摘要描述 (transaction summary)

显示事物的通过数量和失败数量

成功率 = 成功数 / (成功数 + 失败数)

失败率 = 失败数 / (成功数 + 失败数)



平均事务响应时间 (Average Transaction Response Time)

显示所有场景中出现的事务在执行时的响应时间的情况
随着测试时间的加长，系统处理事务的能力就会开始逐渐下降，总体的事务时间情况应该是缓步进行变更的，如果出现大起大落的现象，则为缺陷，说明该事务不稳定



初步判定

1.当服务器处理能力远大于服务器受压情况时，这种性能测试不是特别必要

2.当服务器处理能力小于服务器受压情况时，能容忍一定的事务数报错

3.随着压力的增大，事务报错数逐渐增加

4.在压力一定的情况下，查看失败率和需求做对比确认，以验证此次性能测试通过与否

一般性能测试的结束基本的通过以下内容进行判定

1.成功率

2.事务平均时间

3.并发用户数



举例

用户数	100	200	300
平均响应时间	10	10	5
吞吐量	1000	500	300
总计请求数	10000	1000	1000



用户数与服务器处理请求的平均时间关系图

- 1.本图表示服务器处理请求的平均响应时间
- 2.最佳性能是随着并发用户数的增加，平均事务响应时间比较平稳
- 3.本图可以清晰看到，对着并发用户的增加事务的响应不平稳

用户数与服务器处理请求关系图

- 1.表示服务器每秒处理请求个数
- 2.最佳性能服务器处理请求数是随着用户的增加而增加



作业2

组号	1	2	3
用户数	5	10	15
平均响应时间	1	20	25
吞吐量	1000	50	40

请大家说明哪一组测试结果最佳



谢谢 !

<http://weibo.com/teliss>