

性能测试前如何做好需求调研

——性能测试系列培训课程 2019/04/09 共享技术部-谢佳——



性能测试基础知识

性能测试需求调研

注意事项及问答

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，想精深和全面掌握，也是需要不断积累和实践才行的。

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

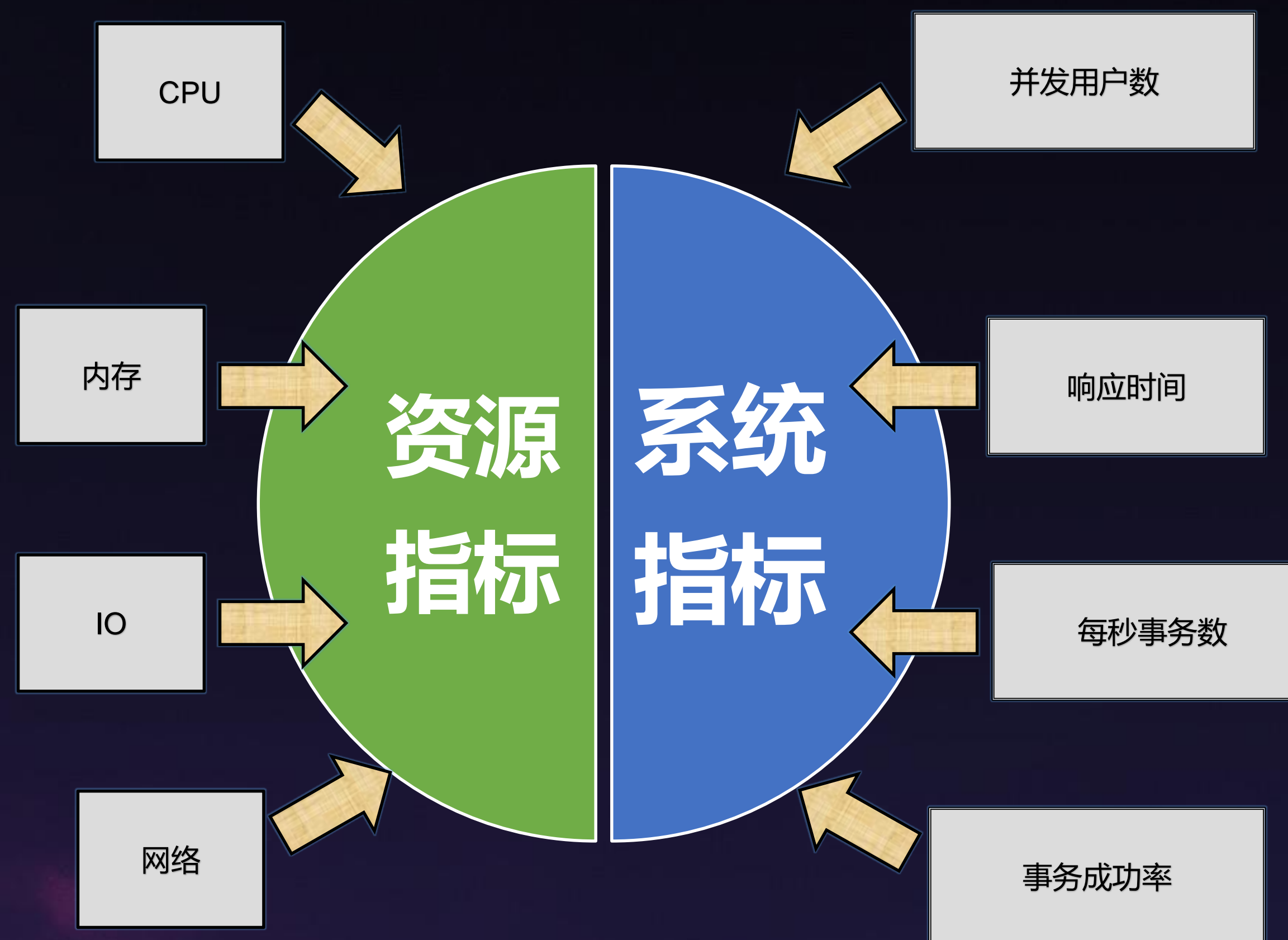
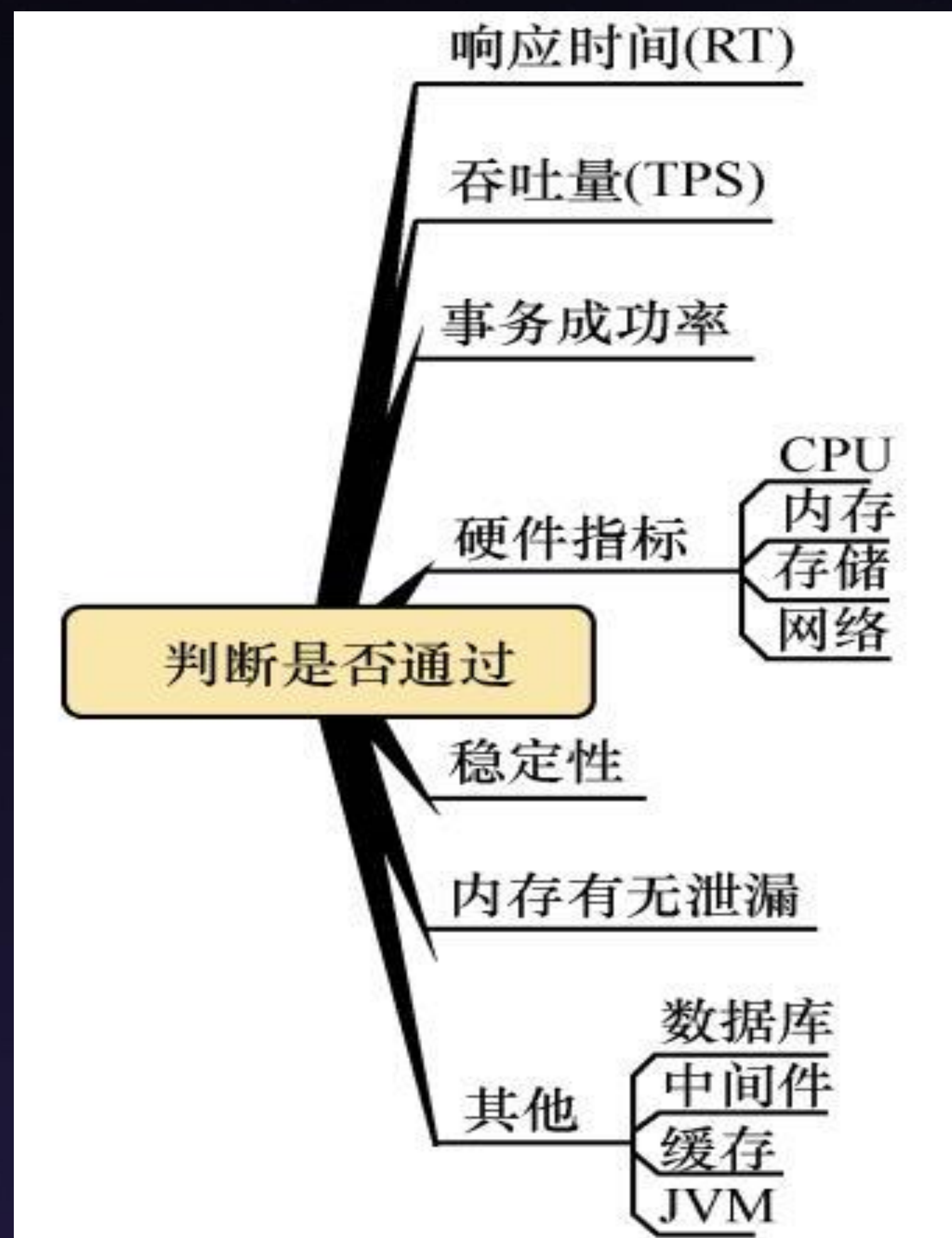
最大业务处理量? ——系统容量

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

性能测试关注主要指标



4个最常用系统指标说明

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

RT

- 全称为 **Response time** , 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

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

性能测试测什么

性能测试 \neq JMeter 或 LR
性能测试 \neq 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

性能测试类型-日常要求

- 1、**基准测试**（实现对一类测试对象的某项性能指标进行定量的和可对比的测试）
- 2、**负载测试**（在期望性能指标内，系统承受的最大用户数）
- 3、**压力测试**（系统处于失效前，承受的最大用户数，如错误率超过上限、超时率或响应时间超过上限，同时分析曲线拐点来分析系统瓶颈）
- 4、**并发测试**（压力通过积攒到指定的线程数后瞬间释放，通过集合点策略实现）
- 5、**稳定性测试**（大压力比如最大用户数、长时间测试比如8小时或24小时，检测系统处理能力响应时间成功率等稳定性、是否有内存泄露等）
- 6、**数据量测试**（大数据量测试，一般指定时程序的触发，程序的并发不是由工具决定的，而是由程序本身的机制决定的，所以这种情况使用单线程测试即可）

性能测试类型-全面要求

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。

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。

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。

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。

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。

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。

性能测试开展时机



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。

性能测试实施流程



按照流程规范
执行落地

在实践中迭
代优化流程
与规范

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等
		2. 性能指标.xlsx	系统处理能力、事物响应时间、系统稳定性、系统资源利用率、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、是否加密、是否挡板
		4. 业务交易调研表.xlsx	用户占比、典型交易占比、用户流量分布、业务数据量、以及未来业务拓展趋势
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、压力模型、性能指标、用户流量分布、业务数据量
3	方案编写	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（服务器环境）、压力机环境（客户端压力模拟器）
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	JMeter脚本协议模板.jmx	直接复用项目工程模板和脚本模板
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(用例).xlsx	1. 业务类型：单交易负载、混合交易负载 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，有请求响应的jtl文件，以及summary摘要的log文件
		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	结果分析	性能测试结果分析.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、风险说明
11	测试总结	性能测试总结.xlsx	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

性能测试基础知识

性能测试需求调研

注意事项及问答

需求调研关注什么？



找谁调研？

产品经理

- 当前及1-3年业务指标
- 业务架构
- 历史运营数据
- 业务发展趋势

技术经理

- 技术架构
- 技术特征
- 中间件及服务组件

系统管理员

- 系统环境配置
- 网络拓扑
- 服务器资源监控历史数据

| 如何调研？

沟通询问

专业引导

二次分析

性能需求统计维度

并发用户量



- A、日常访问量（当前）
- B、峰值访问量（当前）
- C、极限承受量（当前）
- 6个月后的指标
- 3年后的指标



注册用户数
在线用户数
交易用户数
日活量
月活量

业务数据量



- 当前线上系统数据量
- 6个月后的需求量
- 3年后的需求量

参考文档



- 系统架构图
- 业务架构图
- 技术架构图
- 网络拓扑图
- 接口文档
- 数据库表结构

技术特征



- 协议类型
- 报文格式
- 同步异步
- 是否mock

性能测试准备工作

性能环境



- 线上和测试服务器配置对比表
- 性能测试服务器准备及环境搭建
- 性能测试压力机准备及工具安装

铺底数据来源



- 线上脱敏（运维）
- 通过Sql造假数据（开发）
- 通过接口（性能测试）

业务场景



- 业务构成及比例
- 业务交易操作路径
- 用户角色构成及比重
- 性能测试覆盖类型
- 业务及系统指标



- 1、线上和测试环境尽量保持配置等同或者等比例缩小；
- 2、服务器搭建后开发要调整常用的参数配置，和线上环境保持一致，比如jvm参数、线程池、连接池、日志级别、数据库SGV等参数设置

准备数据的几种方式对比

实现方式 比较类型	接口 灌数据	JDBC协议 灌数据	sql语句 灌数据	备注
实现方式	比如jmeter的http协议	比如jmeter的jdbc协议； jdbc不要操作普通的sql语句， 执行效率低，要把sql封装为数据 库的存储过程，然后在jmeter中 调用，效率会很高	编写sql语句； 最好是存储过程，因为存储过 程速度快	
是否已实现	是	是	否	
实现难易程度	易	难	难	
执行速度（条/小时）	慢（需要统计具体值）	快	最快（需要统计具体值）	
数据一致性	好	比较好	比较好	程序升级后，接口的报文有可能改变，如果通过 调用接口的方式去执行，很容易发现错误，性能 测试人员很容易即时观察到并做出调整；如果通 过sql语句的方式去插入，就需要开发和性能测试 紧密沟通，否则由于sql语句的调整导致数据插入 的不正确
数据库表结构升级带来的风险	高 由于数据生成慢，如果通 过新建数据库重新准备数 据的方式，会花费很长时 间准备到指定的数据量， 大大延迟了性能测试的进 度	低（风险等同第三种）	低（由于能够快速生成数据， 可以新建一套表结构，快速生 成大量的数据）	当前线上表结构变化也很频繁 而且都是有专人去做增量升级 过程比较琐碎和复杂耗费时间比较多 测试环境很可能没有专人去做这块处理，导致版 本差异太大，很难升级表结构而且还保留已有数 据，而且现在也不会像线上生产环境那样及时同 步到性能测试环境中，即使同步了，之前循序渐 进准备的数据如果表结构落差太大也全没有用 了，对数据的维护带来了比较大的工作量
是否能跨库操作	是	是	否	由于业务逻辑的复杂性，以及分库分表等情况的 存在，很多时候准备数据，表与表、库与库之间 是有关联性的，所以在准备数据时，需要同时考 虑不同数据库中数据的一致性和关联性，所以在 准备数据时，最好能支持多库同时操作方式
是否能覆盖程序压测	是	否	否	
是否能覆盖mysql压测	是（对数据库冲击力不一定	是（能模拟对数据库直接并发压力	否（并发只能靠外部程序的调 用）	

性能测试环境要求

独立性

- 应该有独立的性能测试环境，不得有其它应用干扰，以免影响性能测试结果的正确性

硬件一致

- 测试所用硬件环境（硬件类型、cpu数量、内存大小等）要尽量与生产环境一致。若不一致，最好按统一的比例进行缩放。

软件一致

- 测试所用软件环境（操作系统、数据库、中间件等）要与生产环境相一致（包括版本号、补丁号等）。

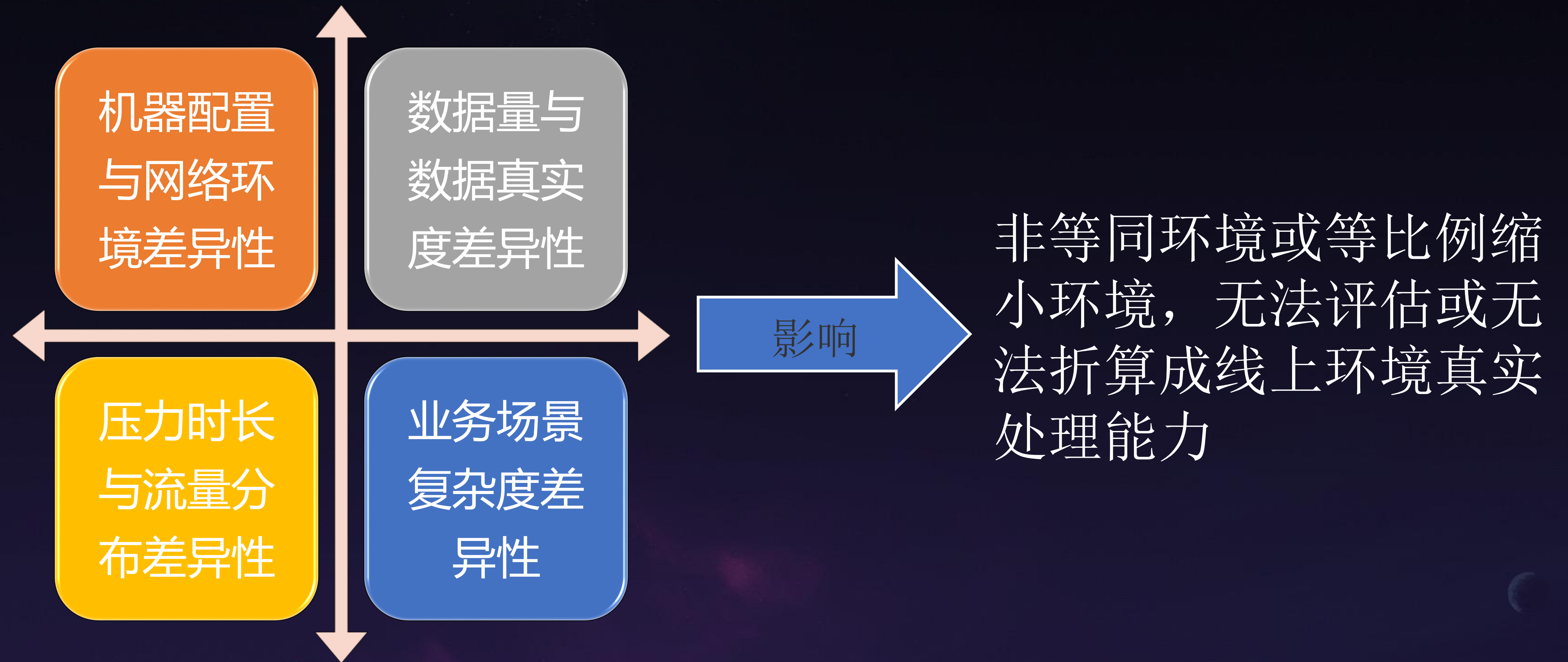
部署一致

- 测试环境中的应用部署模式（如web服务器、应用服务器、数据库服务器是否分离部署）和网络环境（包括网络拓扑结构、网络参数等）要与生产环境相一致。

稳定可重复

- 测试环境应该是稳定的、可重复的，以保证能够得到正确的、可重复的以及可理解的测试结果。

线上线下差异性因素及影响



性能测试基础知识

性能测试需求调研

注意事项及问答

性能测试开展思路

应更早介入

- 开发引入的技术框架和技术原型，需要对其性能、负载、容量做验证
- 不要等产品要临发布前或发布后才开展，遇到大的性能问题再修改就会牵一发而动全身，甚至必须重构才能解决性能瓶颈，代价太大。所以压力测试在大的发版前一两周才启动，很难进行充分准备，而且执行过程中遇到问题解决难度和耗时都很大。

先单点后全链路

- 性能测试需要采用隔离法，对不同的服务瓶颈进行阶段性性能评估
- 独立的服务需要进行性能评估，集成的整个系统也需要做性能评估

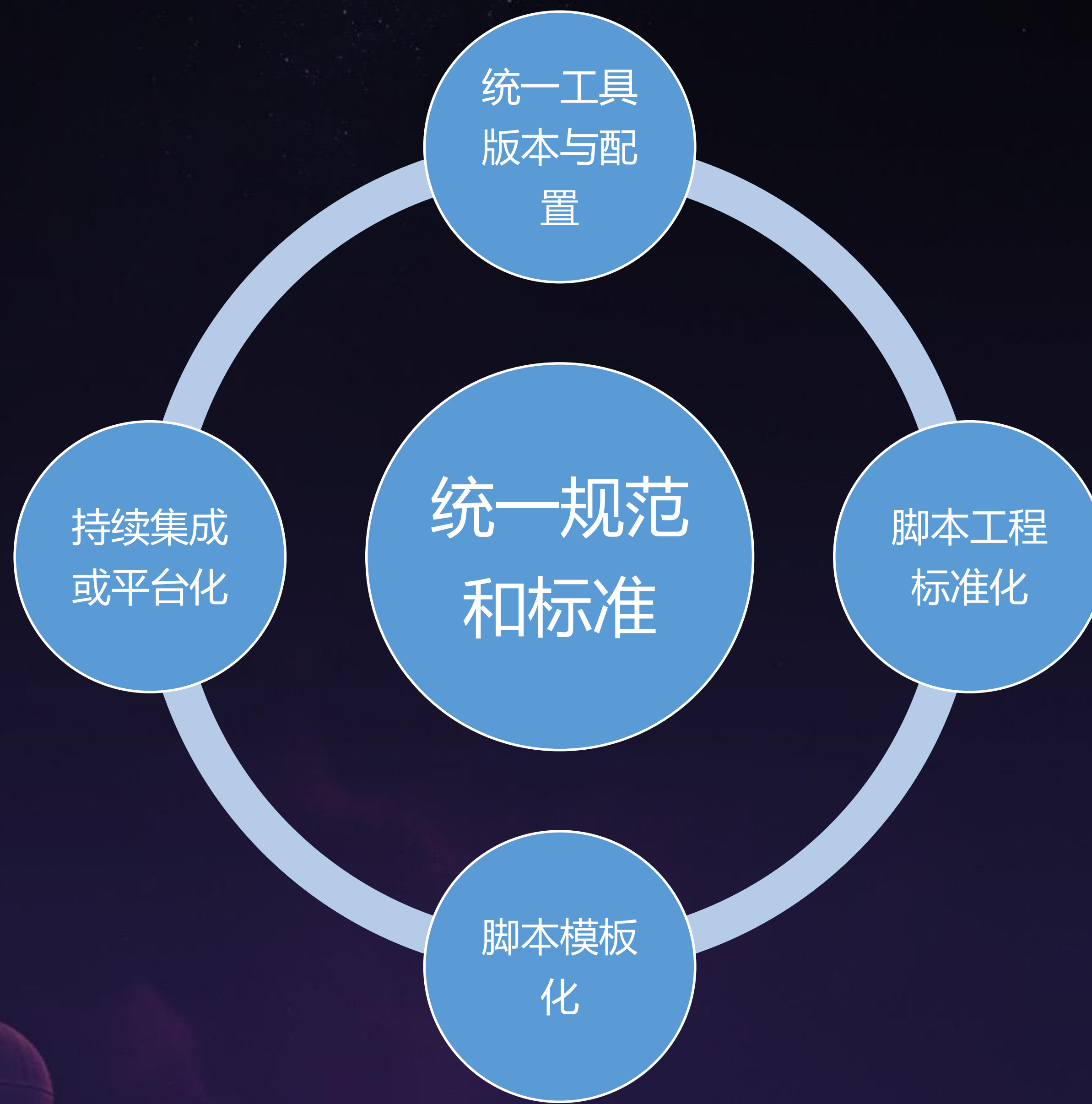
体系化标准化

- 领导层重视，统筹把控，对性能测试重视，成立专门小组
- 站在全行角度去引入测试工具和测试框架，规范化标准化，性能测试资产能复用

自动化常态化

- 性能测试应该和功能回归测试一样常态化，持续性、自动化开展性能基准测试，一旦有问题，立即发现、立即解决，避免问题积累后成为大的毒瘤难以清除

统一规范减少非技术问题



脚本编写要求

- 1、必须遵照真实的业务流程及操作路径进行脚本录制或脚本编写（确保脚本正确性）
- 2、测试数据的丰富性和随机性（让场景数据更贴近真实性）
- 3、添加必要的检查点（返回报文结果检查，只需要检查最核心的内容）
- 4、并发测试必须添加集合点（模拟瞬时压力，比如秒杀场景）
- 5、增强脚本并提升脚本的执行健壮性（健壮的脚本能确保容错性，特别是长时间执行时）
- 6、确保性能测试脚本的简洁性（复杂的配置会耗费资源，特别是大并发时尤其明显）

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

性能常见问题梳理



序号	问题分类	问题描述
1	数据库	部分表缺失索引
2		部分接口传入字段非索引字段
3		连接池配置存在问题导致连接数上不去
4		当一个商户下有多用户并发时行锁导致任务线性排队
5	应用	唯一id生成函数在并发下会有重复值
6		单接口逻辑处理响应时间过长导致吞吐量上不去
7		使用了线程不安全的集合类导致代码抛异常错误
8		内存泄露问题，线程池在执行处理时，处理不够及时，导致Task任务阻塞，产生了大量的FutureTask对象，占用了过多的内存无法释放。
9	日志	日志级别设置问题（ALL、TRACE、DEBUG、INFO、WARN、ERROR、FATAL、OFF）
10		日志打印方式设置问题（stdout、file）
11		日志文件没有切分导致太大后写入缓慢
12	网络	数据库连接没有释放导致端口几分钟内被用完
13		网络内核参数设置问题导致端口释放不及时

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
 - jmeterProject
 - systemName
 - subSystemName
 - data
 - result
 - html
 - testReport
 - jtl
 - testLog
 - script
 - testScript

系统名称

子系统名称

数据及配置文件夹

测试报告

测试日志：执行日志和统计日志

测试脚本

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（20180721）
- 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

问答时刻



THANKS