性能测试基础知识

Agenda

一．软件性能、性能测试概念

二．性能测试的目的

三．性能测试工程师职责描述

四．性能测试基本过程 五．性能测试类型 六.性能测试工具

七．术语

教学目标

一．了解性能测试概念

二．了解性能测试的目的

三．掌握性能测试工程师职责描述

四．掌握性能测试流程

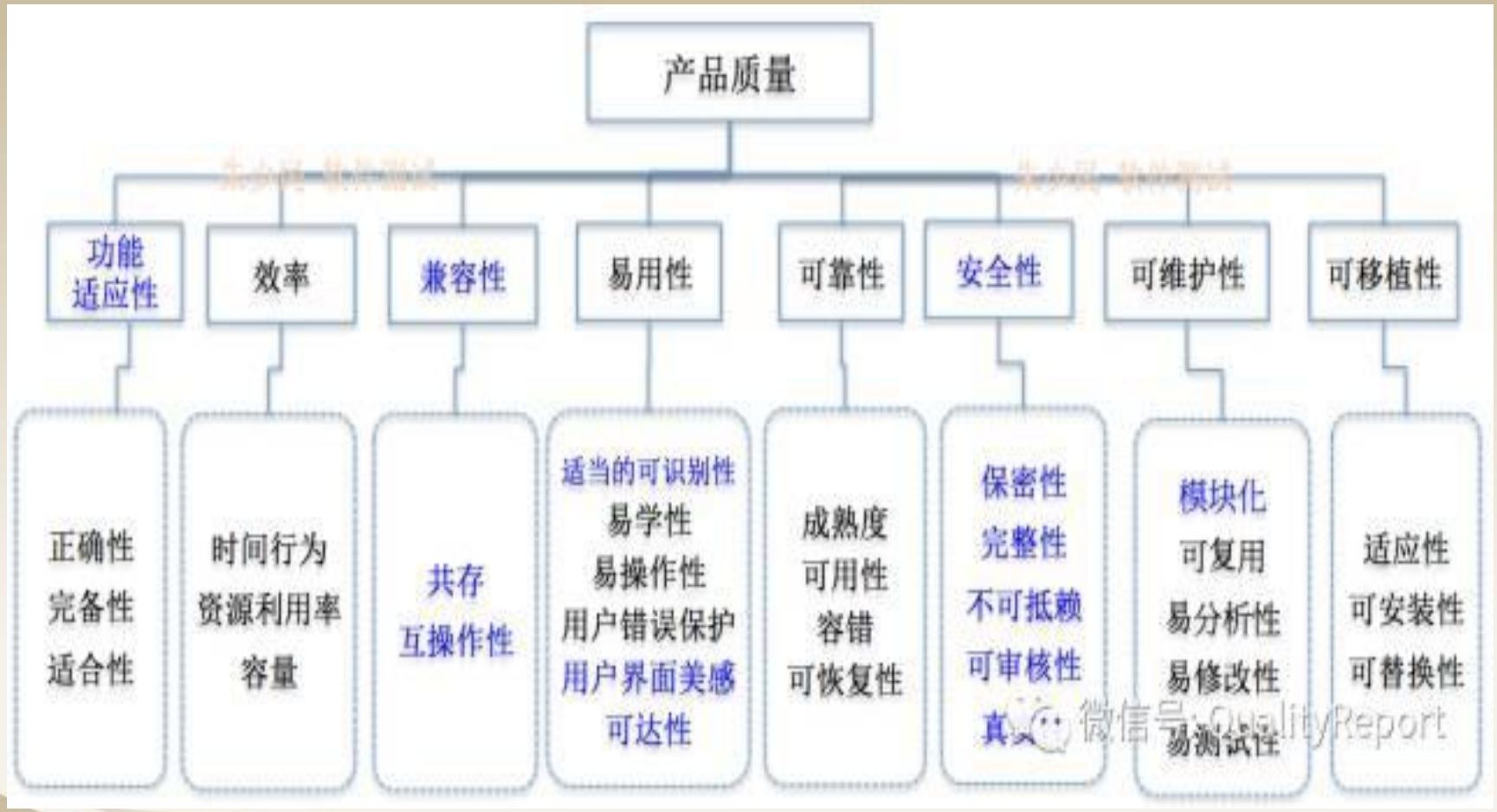
五．掌握性能测试类型

软件的六大质量属性

图4-3 软件产品质量六属性

https://blog.csdn.net/xifeijian/article/details/8542255

软件的六大质量属性



完整诠释软件质量模型ISO 25010： http://www.hezongedu.cn/267.html

认识软件性能

- 软件性能与性能测试：

✓ 软件性能的覆盖面广泛，对一个系统而言，包括执行效率、资源占用、稳定性、安全性、兼容性、可扩展性、可靠性等等。

✓ 性能测试用来保证系统运行后的性能可以满足用户需求。性能测试在软件质量保证中起重要作用。

认识软件性能

- 用户眼中的软件性能

- ✓ 响应快！
- ✓ 运行稳！
- ✓ 无错误！

认识软件性能

- 开发人员视角的软件性能

架构是否合理	系统架构
数据库设计是否存在问题	数据库设计
代码是否存在性能方面的问题	代码
系统中是否有不合理的内存使用方式	代码
系统中是否有不合理的线程同步方式	设计与代码
系统中是否有不合理的资源竞争	设计与代码

认识软件性能

- 系统管理员视角的软件性能

系统管理员关心的问题	软件性能描述
服务器的资源使用状况合理吗	资源利用率
系统是否能够实现扩展	系统扩展性
系统最多支持多少用户的访问， 支持多少业务处理量	系统容量
系统可能的瓶颈在哪里	系统扩展性
更换哪些设备能提高系统性能	系统扩展性
是否支持7*24的业务访问	系统稳定性

认识软件性能

- 性能测试的角度

1. 服务器硬件性能
2. 根据需求和历史数据制定性能目标
3. 建立性能通过模型
4. 对开发代码框架和硬件框架进行性能分析
5. 针对开发发布版本的基准测试
6. 执行软件性能验收及稳定性测试
7. 生产环境的配置和优化
8. 制定性能测试的测试用例设计
9. 制定性能测试的场景
10. 协调各部门配合
11. 特定的性能分析

案例分享

- 美国医保网站故障

奥巴马14日紧急召开记者会，称已听到怨声载道，保证医保网站在月底前完成修复， 解决 所有软件及硬件问题。他对先前宣称民众现有保单不会受新法案影响的言论道歉， 又承认 在“医保交易所”投保不会像在苹果公司iTunes购买音乐一样简单。

奥巴马医保网站 healthcare.gov上线后出现了一系列严重问题。《纽约时报》报导称，出现问题要部分要归因于它决定使用一种非主流的数据库管理软件——不是来自微软、IBM和甲骨文的主流数据库软件，而是选择了MarkLogic的Nosql数据库产品，查询语言是xquery，目标文件格式是XML。

MarkLogic的产品并不是不优秀，而是它的使用方式不同于主流产品，学习曲线非常陡。政府官员坚持使用这款产品或许是因为这家公司的政府关系不错。

MarkLogic

未经过POC测试

案例分享

- 北京奥运售票网站

系统平台 实际流量超预设8倍

为此次门票销售提供技术系统平台的是北京歌华特玛捷票务有限公司透露，此次票务官网的流量容量是每小时100万次，但承受了每小时800万次的流量压力，所以系统在启动不久就出现了处理能力不足的问题。从30日上午9点正式开始售票到中午12点，3个小时内，票务网站被浏览次数达到2000万次。这与他们此次所提供的100万次/小时流量相差甚远。

“不停地刷新网页，也是造成网络拥堵的原因之一。”杨力说，不少网民在无法正常登录后便不断刷新，“这就相当于一名申购者变成了若干名申购者了，无形中增大到了网站流量。从技术角度讲，网站的流量几乎成几何倍增长，导致其他申购者无法登录”。

此外，在票务技术系统上，票务官网、中国银行指定代售网点以及呼叫中心（即订票热线）用的都是一个售票系统平台，该系统会自动确认订单并按先后顺序进行处理。由于昨天申购的人数庞大，加上网民不停地刷新页面，最终造成奥运门票无法正常销售。

能力规划不足
并发测试不充分

案例分享

- 12306


中国铁路客户服务中心
www.12306.cn
系统运行异常错误:


登录名:
用户名／邮箱
密码:
忘记用户名/密码?
验证码:
请点击下图中所有的 沙盘
刷新
验证码如何使用?
登录
快速注册

为什么进行性能测试

- 测试目的
  - 能力验证
  - 能力规划
  - 性能提升
  - 系统可用性

为什么进行性能测试

- 能力验证

➢ 容量是否满足要求 ➢ 响应时间是否满足要求

➢ 系统是否可以稳定运行

为什么进行性能测试

- 能力规划
  - 探索性的测试（POC测试）
  - 了解系统性能以及获得扩展性能

为什么进行性能测试

- 性能提升
  - 发现系统性能瓶颈
  - 通过性能调优解决性能瓶颈
  - 对比产品升级前后的性能表现

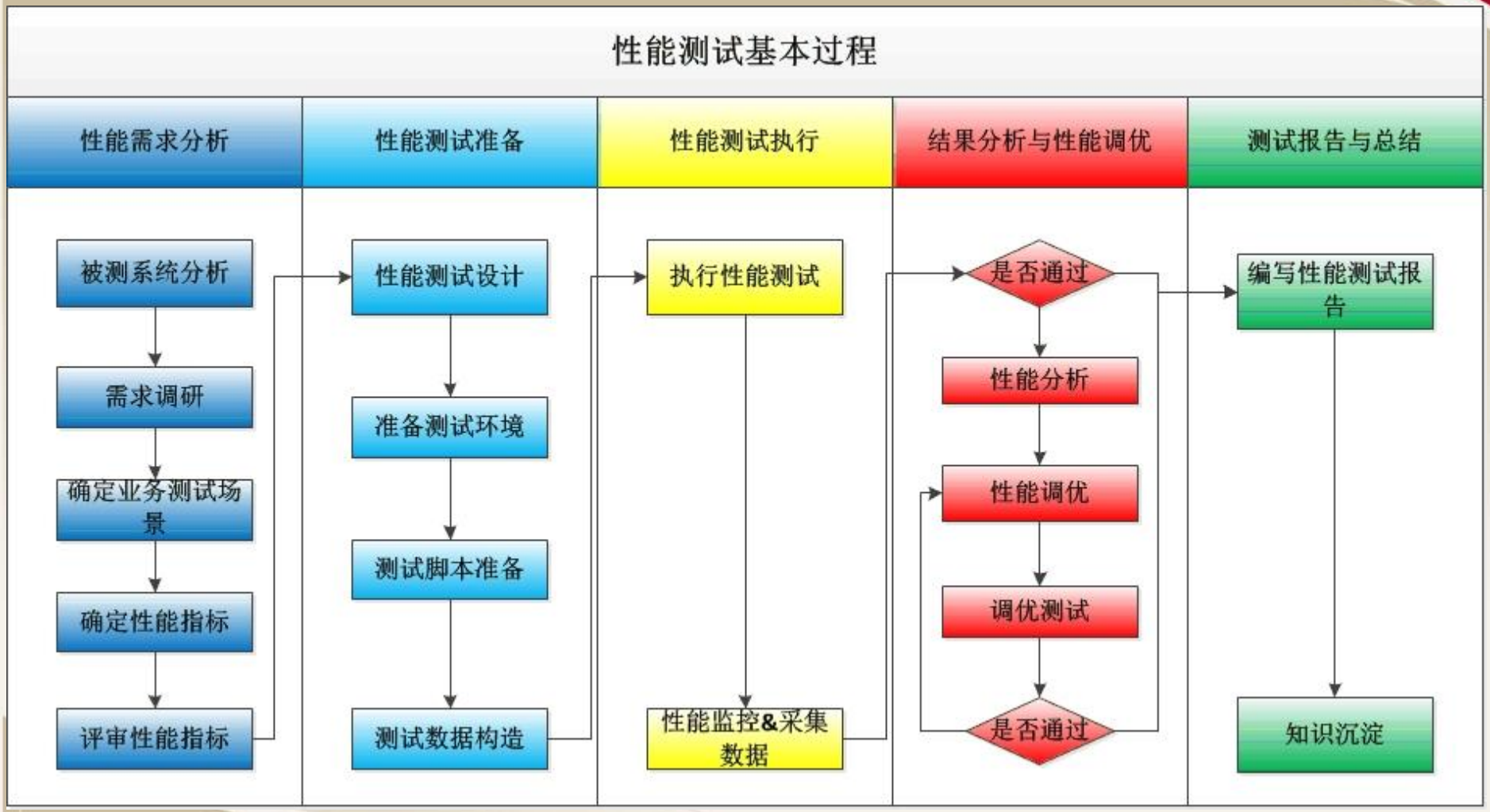
为什么进行性能测试

- 系统可用性
  - 评价系统的可用性如何
  - 测试系统高可用机制是否完备健全
  - 评估系统恢复时间和商业风险

职责描述

角色	职责
客户、产品经理	1.负责提出性能测试需求的人员或组织，包括公司内部客户或外部客户的技术支持、实施、咨询组织 2.配合测试组制定测试需求、测试目标等，一般提供唯一接口人协助落实性能测试项目工作。
项目经理/性能测试经理	1.负责性能测试体系的裁剪和执行，使之符合性能测试项目的实际情况，保证性能测试工作的有效进行。 统筹整个项目的支持实施。 2.负责项目进度、质量、成本及风险
性能测试分析师	1.在项目经理的组织下，负责性能测试的需求分析，测试方案、测试案例、测试场景等的设计工作。 2.分析系统的瓶颈及调优建议。 3.协助编写测试报告 4.可以由项目经理兼任。
性能测试工程师	1.负责性能测试的测试脚本开发、参数化数据准备工作 2 负责监控部署 3.负责任务执行、结果搜集及瓶颈初步定位。 4. 协助编写测试报告工作

性能测试基本过程


性能测试基本过程
性能需求分析
性能测试准备
性能测试执行
结果分析与性能调优
测试报告与总结
被测系统分析
需求调研
确定业务测试场景
确定性能指标
评审性能指标
性能测试设计
准备测试环境
测试脚本准备
测试数据构造
执行性能测试
性能监控&采集数据
是否通过
性能分析
性能调优
调优测试
是否通过
编写性能测试报告
知识沉淀

性能需求分析

性能需求分析是整个性能测试工作开展的基础，如果连性能的需求都没弄清楚，后面的性能测试工具以及执行就无从谈起了。 在这一阶段，性能测试人员需要与PM、DEV及项目相关的人员 进行沟通，同时收集各种项目资料，对系统进行分析，确认 测 试的目标。并将其转化为可衡量的具体性能指标。

测试需求分析阶段的主要任务是分析被测系统及其性能需求，建立性能测试数据模型，确定性能测试范围，确定合理性能指标，并进行评审；

性能测试准备

主要包括：编写测试方案、测试策略、设计场景、编写脚本、准 备测试环境，构造测试数据，环境预调优等；

针对系统的特点设计出合理的测试场景。
为了让测试结果更加准确，这里需要很细致的工作。如建立用户模型，只有知道真实的用户是如何对系统产生压力，才可以设计出有代表性的压力测试场景。这就涉及到很多信息，如用户群的分布、各类型用户用到的功能、用户的使用习惯、工作时间段、 系统各模块压力分布等等。只有从多方面不断的积累这种数据， 才会让压力场景更有意义。最后将设计场景转换成具体的用例。

测试数据的设计也是一个重点且容易出问题的地方。生成测试数据量达到未来预期数量只是最基础的一步，更需要考虑的是数据的分布是否合理，需要仔细的确认程序中使用到的各种查询条件 ，这些重点列的数值要尽可能的模拟真实的数据分布，否则测试 的结果可能是无效的。

性能测试准备

预调优指根据系统的特点和团队的经验，提前对系统的各个方面做一些优化调整，避免测试执行过程中的无谓返工。比如一个高并发的系统，10000人在线，连接池和线程池的配置还用 默认的，显然是会测出问题的。

执行性能测试

执行阶段工作主要包含两个方面的内容：一是执

行测试用例模型，包括执行脚本和场景；
其次测试过程监控，包括测试结果、记录性能业务指标和资源占用情况

结果分析与性能调优

发现问题或者性能指标达不到预期，及时的分析定位，处理后重复测试过程。性能问题通常是相互关联相互影响的，表面上看到的现象很可能不是根本问题，而是另一处出现问题后引起的反应。这就要求监控收集数据时要全面，从多方面多个角度去判断定位。调优的过程其实也是一种平衡的过程，在系统的多个方面达到一个平衡即可。

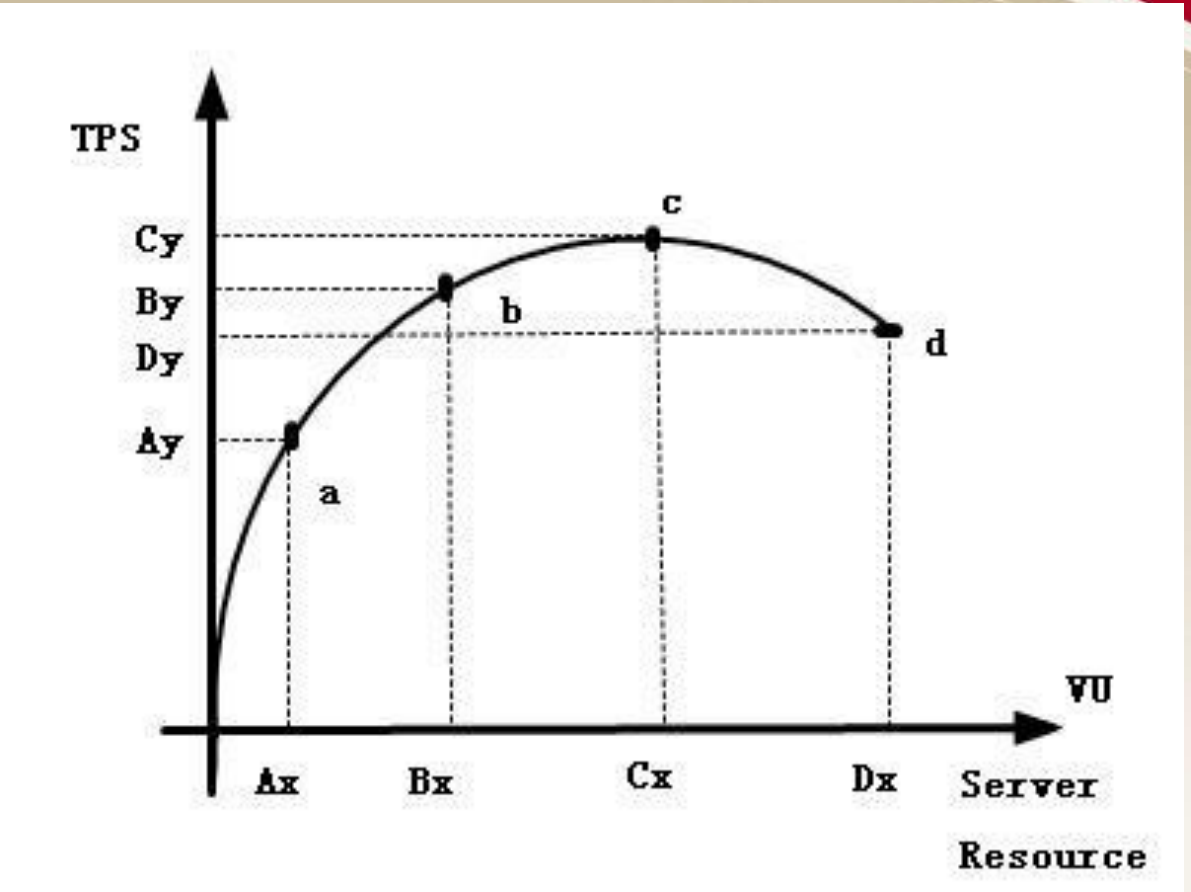
性能报告与总结

编写性能测试报告，阐明性能测试目标、性能结果、测试环境、 数据构造规则、遇到的问题和解决办法等。并对此次性能测试 经验进行总结与沉淀。

上面所有内容中，如果排除技术上的问题，性能测试中最难做 好的，就是用户模型的分析。它直接决定了压力测试场景是否 能够有效的模拟真实世界压力，而正是这种对真实压力的模拟， 才使性能测试有了更大的意义。可以说，性能测试做到一定程 度，差距就体现在了模型建立上。

至于性能问题的分析、定位或者调优，很大程度是一种技术问题，需要多方面的专业知识。数据库、操作系统、网络、开 发都是一个合格的性能测试人员需要拥有的技能，只有这样， 才能从多角度全方位的去考虑分析问题。

性能测试模型



说明：

a点：性能期望值

b点：高于期望，系统资源处于临界点 c

点：高于期望，性能处于拐点

d点：超过负载，资源不够用，系统处于崩溃

性能测试类型

性能测试的定义为：在一定的负载情况下，系统的响应时间等特性是否满足特定的性能需求。

性能测试是指通过模拟生产运行的业务压力量和使用场景组合，测试系统的性能是否满足生产性能要求。

目的：验证系统是否有其宣称具有的能力。
特点:对系统性能已经有了解的前提,对需求有明确的目标，并在已经确定的环境下进行的。

关注的是系统性能是否和具体的性能需求相一致，而当系统性能超过性能需求的时候，系统的表现并不是测试人员关心的重点。

负载测试

是指对系统不断地增加压力或增加一定压力下的持续时间，直到系统的某项或多项性能指标达到安全临界值，例如某种资源已经达到饱和状态等

目的：找到系统处理能力的极限。了解系统的性能容量，或是配合性能调优来使用。

特点：对系统性能已经有了解的前提，对需求有明确的目标，并在已经确定的环境下进行的。

- 产出：

✓ 1.得出线下系统最大TPS。

✓ 2.得出线下系统最大TPS时**硬件资源利用率。** ✓

3.**得出线下系统极限并发数。**

压力测试（强度测试）

压力测试是评估系统处于或超过预期负载时系统的运行情况。压力测试的关注点在于系统在峰值负载或超出最大载荷情况下的处理能力。

目的：检查系统处于大压力性能下时，应用的表现。 特点：一般通过模拟负载等方法，使得系统的资源使用达到较 高的水

关注点:发现功能测试不能发现的非功能性缺陷。

- 产出：

✓ 得出线下系统崩溃点的TPS。 ✓

得出线下系统崩溃时资源使用率

✓ 得出线下系统极限并发数

稳定性测试（可靠性测试）

在给系统加载一定业务压力（性能测试压力或者参考 80/20 原则）的情况下，使系统运行一段时间，以此检测 系统是否稳定。

目的：主要目的是验证是否支持长期稳定的运行。

关注系统稳定性。

产出：

长时间测试中系统稳定不宕机

不测试业务场景无性能问题

某业务出现错误时系统可稳定持续运行

系统稳定状态下的资源利用、连接池、TPS、响应 时间、DB 健康情况等数据

容量测试

容量测试的目的是通过测试预先分析出反映软件系统应用特征的某项指标的极限值(如最大并发用户数、数据库记录数等)，系统在其极限状态下没有出现任何软件故障或还能保持主要功 能正常运行。

容量测试还将确定测试对象在给定时间内能够持续处理的最大负载或工作量。

基本原理是通过不同的负载测试找出系统在资源饱和，无故障的情况下的最大性能指标，如：并发，吞吐流量等。

产出:

1.获取系统在拐点时的性能数据 2.获取系统性能点下降时的性能数据

容量测试在资源占用和以及其他应用方面都有设定的指标。比如：CPU ，memory ，Disk usage ，load average等。那么这 些指标如何设定。一般采用80/20原则。如：CPU不超过80%

配置测试

配置测试主要是针对硬件而言，了解各种不同环境对系统性能影响的程度，从而找到系统各项资源的最优分配原则。主要目的是了解各种不同因素对系统性能影响的程度，从而判断出最值得进行的调优操作。

基准测试

在系统无任何负载压力情况下，对系统施加单个用户，为了获 取单用户的各业务响应时间作为性能基准数据的测试 基准测试是指在一定的软件、硬件及网络环境下，模拟一定数 量的虚拟用户运行一种或多种业务，将测试结果作为基线数据， 在系统调优或系统评测的过程中，通过运行相同的业务场景比 较测试结果，确定调优的结果是否达到预期效果或者为系统的 选择提供决策数据。

基准测试一般基于配置测试，通过配置测试得到数据，并将这个数据作为基准来比较每次调优后的性能是否有所改善。

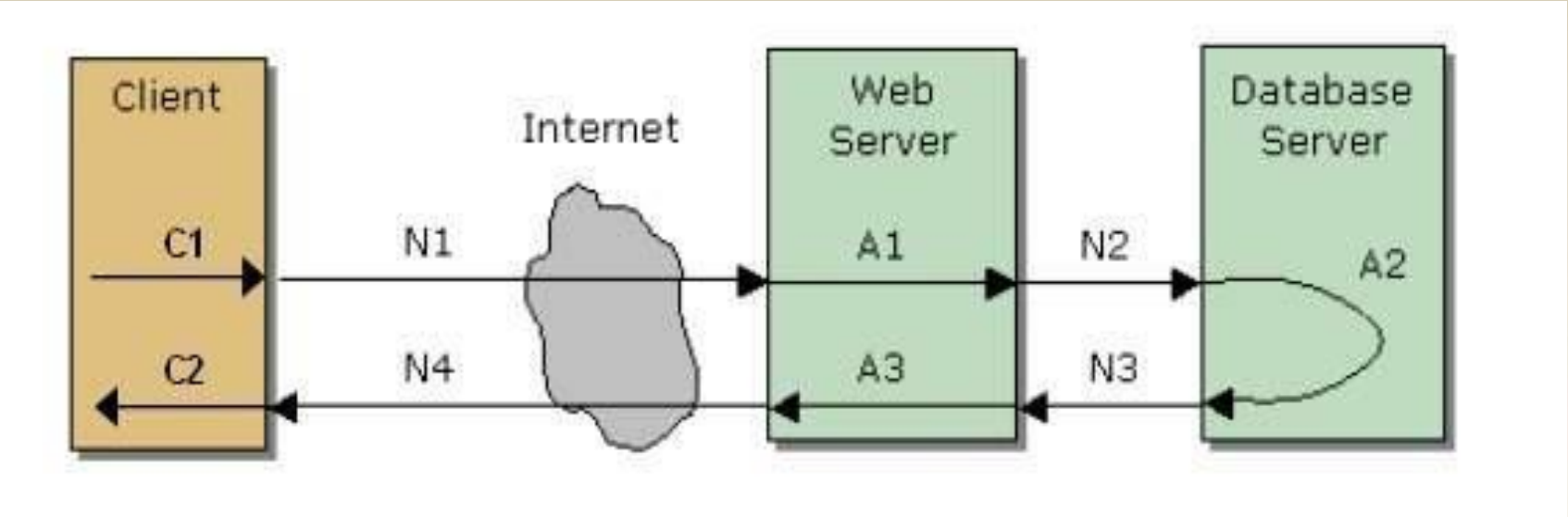
软件性能的核心概念 - 响应时间

- 什么是响应时间

指的是客户发出请求到得到响应的整个过程的时间

软件性能的核心概念 - 响应时间

- 什么是系统响应时间



软件性能的核心概念 - 响应时间

- 网络传输时间：N1+N2+N3+N4
- 应用服务器处理时间：A1+A3 •

数据库服务器处理时间：A2

- 响应时间：N1+A1+N2+A2+N3+A3+N4

软件性能的核心概念

1. 负载：模拟业务操作对服务器造成压力的过程，比如模拟100个用户进行发帖。

2. TPS：每秒完成的事务数，通常指每秒成功的事务数，性能测试中重要的综合性性能指标。一个事务是一个业务度量单位， 有时一个事务会包含多个子操作，但是为了方便统计，我们会 把这多个子操作计为一个事务。比如一笔电子支付操作，在后 头系统中可能会经历会员系统、账务系统、支付系统、会计系 统、银行网关等。但对于用户来说只想知道整笔支付花费了多 长时间。

3. RT/ART：响应时间/平均响应时间，指一个事务花费多长时间完成（多长时间响应客户请求），为了使这个响应时间更具代 表性，会统计更多的响应时间然后取平均值，即得到了事务平 均响应时间（ART），为了方便大家通常会直接用RT来代替 ART，ART与RT是代表同一个意思。

4. PV（Page View）:每秒用户访问页面的次数，此参数用户分析平均每秒有多少用户访问页面。

软件性能的核心概念

5. Vuser虚拟用户（ Virtual user ）:模拟真实业务逻辑步骤的虚 拟用户，虚拟用户模拟的步骤都被记录在虚拟用户脚本里。 Vuser脚本用于描述Vuser在场景中执行的操作。

6. Concurrency并发，并发分为狭义和广义两类。狭义的并发， 即所有的用户在同一时刻做同一件事情或操作，这种操作一遍针 对同一类型的业务，或者所有用户进行完全一样的操作，目的是 测试数据库和程序对并发操作的处理。广义的并发，即多个用户 对系统发出了请求或者进行了操作，但是这些请求或者操作可以 是不同的。对整个系统而言，仍然有很多用户同时进行操作。狭 义并发强调对系统的请求是完全相同的，多适用于性能测试、负 载测试、压力测试、稳定性测试场景；广义并发不限制对系统的请求操作，多适用于混合场景、稳定性测试场景。

软件性能的核心概念

7. 场景（Scenario）：性能测试过程中为了模拟真实用户的业务 处理过程，在LR中构建的基于事务、脚本、虚拟用户，运行设置、 运行计划、监控、分析等的一系列动作的集合，称之为性能测试 场景。场景中包含了待执行脚本、脚本组、并发用户数、负载生 成器、测试目标、测试执行时的配置条件等。

8. 思考时间（Think Time）:模拟正式用户在实际操作时的停顿 间隔时间。从业务的角度来讲，思考时间指的是用户在进行操作 时，每个请求之间的间隔时间。在测试脚本中，思考时间体现为 脚本中两个请求语句之间的间隔时间。

9. 标准差（Std.Deviation）:该标准差根据数理统计的概念得来， 标准差越小，说明波动越小，系统越稳定，反之，标准差越大， 说明波动越大，系统越不稳定。包括响应时间标准差、TPS标准 差、Running Vuser标准差、Load标准差、CPU资源利用率标准 差、Web Resources标准差等。举例响应时间标准差。

Thank You !