

性能测试基础 (PDF测试岗位课程)





第一部分 性能测试理论

第二部分 性能测试过程

第三部分 性能测试流程



第一部分 性能测试理论-性能定义

什么是性能

系统实现其功能的能力,对性能进行描述可以从宏观和微观两个层面

- 宏观上:

系统能稳定运行, 高并发访问量下系统不宕机, 一定数量的促销或 抢购活动系统处理完需要的时间, 系统能够支撑的访问量, 系统每秒 能够处理的交易或者事务数, 系统的容量, 系统能否支撑未来一段时间内访问量增加等。

- 微观上:

每笔交易或者每个事务的资源开销, 包括CPU, 磁盘的IO, 网络传输, 服务器的连接数, 各种线程锁, 线程池的使用情况, JVM heap的使用, 内存的分配, 回收, 对象占用内存的大小, 缓存的规则, 条目大小, 缓存的大小, 命中率与未命中率, 接口调用的响应时间, 系统间交互的响应快慢等等



第一部分 性能测试理论-性能指标

衡量性能的好坏，需要使用一些性能指标来描述，以下是一些最常使用的性能指标：

■ 响应时间(Response Time)

请求或者某个操作从发出的时间与收到服务器响应的时间的差值，一般统计是事务的响应时间，响应时间是衡量系统性能的一个很重要的性能指标

■ 吞吐率(TPS Transaction Per Second)

系统每秒钟能够处理的交易或事务的数量，一般统计的是每秒通过的事务数，TPS也衡量系统性能的一个很重要的性能指标

➤ 响应时间和TPS两个非常重要的性能指标，任何系统的性能都可以用这两个指标来描述，针对不同的应用系统会衍生出一些别的指标，但都是从这两个指标转化而来的。

事务：自定义的某个操作或者是一组请求的集合。例如在登录页面，输入完账号密码，从点击登录按钮开始到跳转到新的页面且页面加载完全，这一步操作，我们定义为一个事务，对于打开一个页面，从浏览器中输入URL地址到页面加载完全，我们把这一过程定义为一个事务，它包括若干个HTTP请求，所以事务也是一组请求的集合。



第一部分 性能测试理论-性能指标

■ 资源开销 (SD Server Demand)

每个交易或者事务对系统资源的消耗，是一个可量化的概念，用来衡量不同交易或者事务对资源的消耗程度。例如对CPU的消耗程度用消耗的CPU秒数来衡量，其计算逻辑为 $(1 * \text{CPU个数} * \text{CPU利用率}) / \text{TPS} = \text{每个事务或者交易的CPU开销}$ 。其他的磁盘IO以及网络传输有类似的计算方法

■ 并发用户(Concurrency)

真实用户的相邻操作之间会有一定的间隔时间(称之为思考时间)，所以并发用户有绝对和相对之分。狭义的并发是某个时间点同时向服务器发出请求的并发用户数。广义上的并发是一段时间内向服务器发出请求的并发用户数。

例如易购的订单流程包括登录，浏览产品页面，加入购物车，去结算，确认商品清单，确认地址信息，使用券，提交订单，去支付。100个人同时在网站上下订单，但因为每个人执行订单流程中的步骤有快有慢，所以在同一时间点向服务器发出请求的是不会有100个的，会远小于100这个值，我们假设为10.那么这个100是相对的并发用户数，10为绝对的并发用户数。



第一部分 性能测试理论-性能指标

■ PV (Page View)

即页面浏览量，或点击量；用户每1次对网站中的每个网页访问均被记录1次。用户对同一页面的多次访问，访问量累计。PV是衡量电子商务网站性能容量的重要指标。PV统计有全天PV,平均每小时PV以及高峰1小时的PV

■ UV (Unique Visitor)

UV(独立访客), 访问网站的一台电脑客户端为一个访客。00:00-24:00内相同的客户端只被计算一次。UV统计也会有全天UV,平均每小时UV和高峰1小时的UV

PV和UV是衡量web站点的两个重要的指标。PV/S由TPS通过一定的模型转化为PV。例如若把一个完整的页面定义为一个transaction,则TPS就可以等同于PV/S。
PV于UV之间存在一个比例，PV/UV可以理解为平均每个人浏览的页面个数。这个比值会有一些波动，促销期间的PV/UV比会略高于平时的PV/UV。所以用PV来衡量web站点的容量更准确。



第一部分 性能测试理论-性能指标

■ Hit/S

每秒的Hit数，一个hit为一个HTTP请求，性能测试过程中一般不请求静态的资源(js、css、图片等)，所以Hit/s一般为动态请求。

■ 订单转化率

访客访问网站时，把访客转化成网站的常驻用户进而再提升成网站的消费用户，而由此产生的消费率就是订单转化率，它是衡量电子商务网站运营水平的重要指标。计算方法是一段时期内(一般为1小时)的有效订单数(订单状态为C+M)/访问人数(UV)



第一部分 性能测试理论-性能测试类型

性能测试

通过自动化的测试工具模拟多种正常、峰值以及异常负载条件来对系统的各项性能指标进行测试

■ 测试类型

针对不同的测试目的有以下的几种常见的测试类型：

- **性能测试:** 系统在正常负载的情况的各项性能指标,即通过调整,找到合适的负载,使系统的资源的利用率处于中等的情况下,采集系统的各项指标
- **负载测试:** 系统在不同的负载的情况的性能表现,可以得到系统在不同负载下的性能变化趋势,寻求性能的拐点。例如其他条件相同,分别测试系统在20,50,100并发用户下的各项性能指标,找到其变化的规律,找到系统的能达到的最大TPS,统计对应的响应时间和资源消耗。
- **压力测试:** 系统在高负载的情况下的性能表现,寻找系统能够承受的最大负载以及对应的系统吞吐率



第一部分 性能测试理论-性能测试类型

➤ 基准测试:

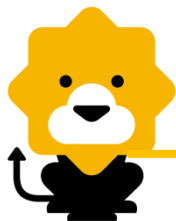
针对确定的测试系统，代码版本执行的测试，采集性能指标，作为后期的版本对比

➤ 稳定性测试:

以正常负载或者稍高于正常负载施加于系统，进行长时间的测试，检测系统能够稳定的运行，以及系统的各项性能指标会不会随着时间发生变化。

➤ 扩展性测试:

通常用于新系统，新环境的搭建，通过先测试单台服务器的处理能力,然后逐渐增加服务器数量，测试集群环境下的单台服务器的处理能力是否有损耗。



第一部分 性能测试理论-场景区分

业务场景和测试场景区别

■ 业务场景

业务流程，具体的业务操作的总和。描述的是具体的用户行为，可以通过对用户的行为进行分析得到，也可以根据以往的数据对业务场景进行估计。业务场景对性能测试非常重要,是性能测试场景的设计的重要来源。

■ 测试场景

测试场景是对业务场景的模拟，测试场景应该尽可能的贴近真实的业务场景，但是有时由于条件所限制，会做一些调整和特殊的设置。



第一部分 性能测试理论-场景区分

单场景与混合场景区别

■ 单场景

测试场景中只涉及到单个的业务流程，目的是测试系统的单个业务在响应时间正常，系统资源利用正常的情况下所能达到的最大的TPS，*测试系统的单个业务的容量。*

■ 混合场景

测试场景中涉及到多个业务流程，且各业务流程在混合的业务流程中占的比重会不同。目的是为了测试系统的混合业务在响应时间正常，系统资源利用正常的情况下所能达到的最大的TPS，测试系统的混合业务的容量。



第一部分 性能测试理论-混合场景

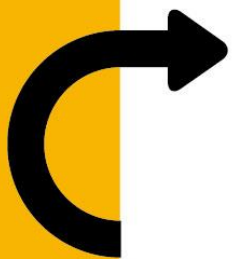
■ 业务模型或者访问模型

混合场景里含有多个业务流程，但是每个业务流程所占的比重会不一样，针对每个业务以及业务的比重，称之为业务模型，对于web站点，我们称之为访问模型

业务流程	占比
浏览首页	10%
浏览产品页	40%
浏览三级页面	20%
关键字搜索	25%
订单流程	5%

一个简单的业务模型/访问模型

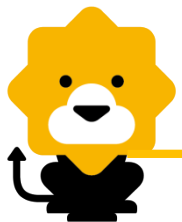




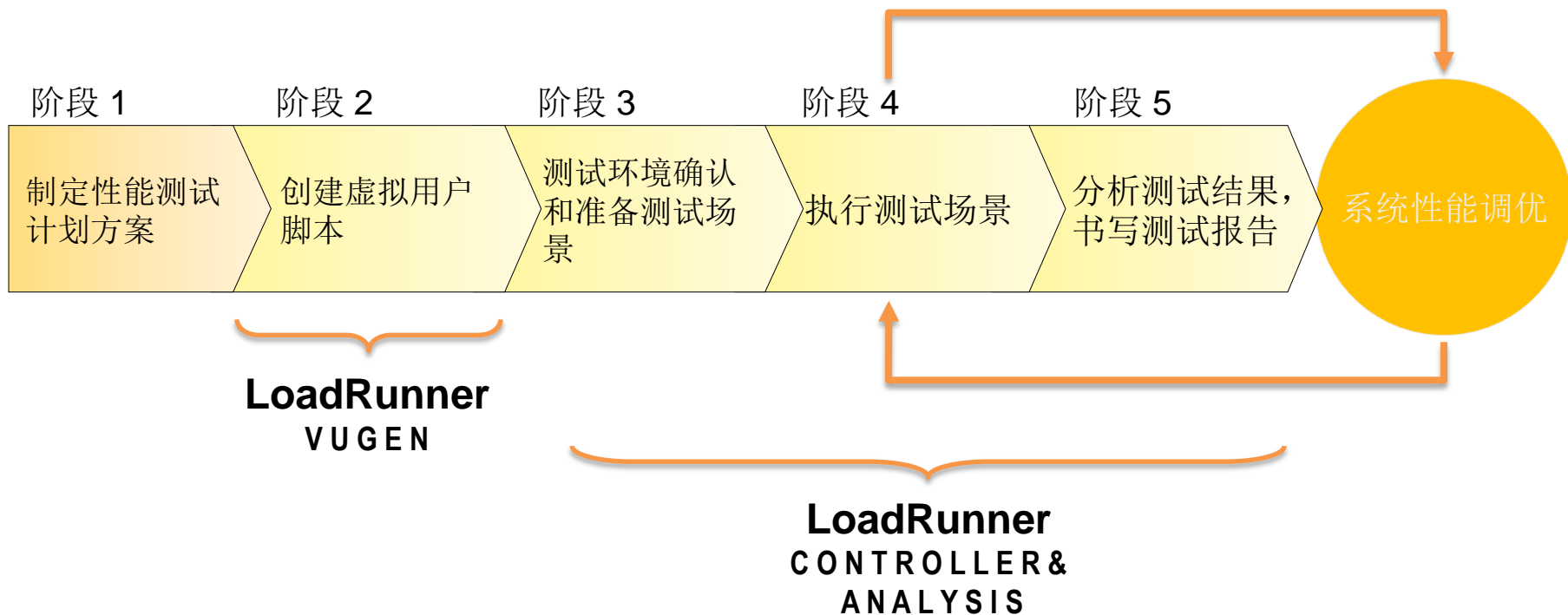
第一部分 性能测试理论

第二部分 性能测试过程

第三部分 性能测试流程



第二部分 性能测试过程



NOTE: 这是LR的一个典型工作流程。



第二部分 性能测试过程-计划方案

制定性能测试计划方案

■ 制定计划方案前需要做的事情

熟悉业务流程，收集性能测试的目的，与开发人员，业务人员，架构师，有经验的性能测试工程师沟通，找到性能点

熟悉被测试系统的架构，与其他系统的交互过程，系统的硬件配置信息，软件配置信息，对性能测试有重要影响的关键配置信息需要明确的列举出来

明确项目的上线时间点，目前项目的进度，可调配的资源包括环境和性能测试人员。



第二部分 性能测试过程-计划方案

制定性能测试计划方案

性能测试计划方案是性能测试的指导，是后续一序列测试活动的依据，完整的性能测试计划方案需要填写以下的几个部分：

- **测试概述** 对整个测试项目进行整体的描述
- **测试目的** 计划中必须有明确的测试目的，它是性能测试退出的判断条件
- **测试环境** 明确的测试环境的软硬件信息，若有生产环境，除要有生产环境的详细信息外，还需要有环境的差异分析
- **测试数据** 明确对测试数据的要求，例如可重复使用，参数的数量级，分布等
- **测试场景** 最初设计的测试场景的细节，包括每个测试场景的设置，测试目的，对环境的调整，对测试数据的特殊要求等
- **执行计划** 最初设计的测试场景的执行时间计划
- **计划变更** 记录整个测试项目中的未按照计划执行的部分
- **参与人员** 项目的参与人员以及每个人员的职责和分工
- **风险控制** 预期的风险和不可控的因素以及它们对测试项目进度的影响



第二部分 性能测试过程-创建脚本

创建虚拟用户脚本

虚拟用户的脚本是执行测试的前提条件,对整个性能测试的结果的准确性有举足轻重的影响。脚本的制作需要遵循一定的规范.良好的虚拟用户脚本至少需要满足以下几点要求:

- 虚拟用户脚本模拟的业务流程与真实用户的行为应该尽可能的保持一致, 不能有过大的偏差。
- 脚本有简要的描述, 一些关键的执行步骤有注释、说明
- 事务的定义准确, 即事务的起始边界和结束边界位置正确, 同时需要对事务的通过与失败进行明确的检查和判断, 确保场景运行时事务的统计是准确的。
- 合理进行参数化, 是否参数化以及参数的分布, 参数的取值和更新方式, 参数的数量级等都会影响性能测试结果的准确性
- 脚本中要明确是否需要做关联以及关联的位置合理准确, 有些请求处理逻辑中的检查比较弱, 虽然脚本能够运行成功, 但服务器会抛exception。
- 测试数据准确, 有效, 满足性能测试的要求。避免因测试数据的不合理影响性能测试的结果
- 脚本中的参数和变量的使用要合理。
- 脚本中加入一些必要的控制逻辑, 使脚本健壮, 稳定运行。

.....



第二部分 性能测试过程-测试环境

确认测试环境

在测试环境搭建好后，需要确认测试环境，需要做以下的一些事情：

- 收集并确认测试环境的软硬件信息
- 确认部署的代码版本。
- 测试环境各服务器是否部署监控程序，性能测试人员是否具备监控权限
- 所有服务器的进程是正常对外提供服务
- 确认系统是否具备执行性能测试的条件(可适当用小负载测试下系统)
- 确认系统无外界干扰，性能测试执行过程中确保能独占该环境
- 确认服务器所在的操作系统的状态正确，包括CPU,内存使用，磁盘空间剩余是否足够等
- 对于数据库服务器，确保数据库的基础数据量满足测试需求，数据库中的表若有重大的改变，确认执行过RUNSTAT操作

.....



第二部分 性能测试过程-测试场景

设计和确认测试场景

测试场景的设置对性能测试的结果影响很大。执行测试之前务必确认场景的设置是否正确。以下为测试场景设置中通常需要确认的检查点：

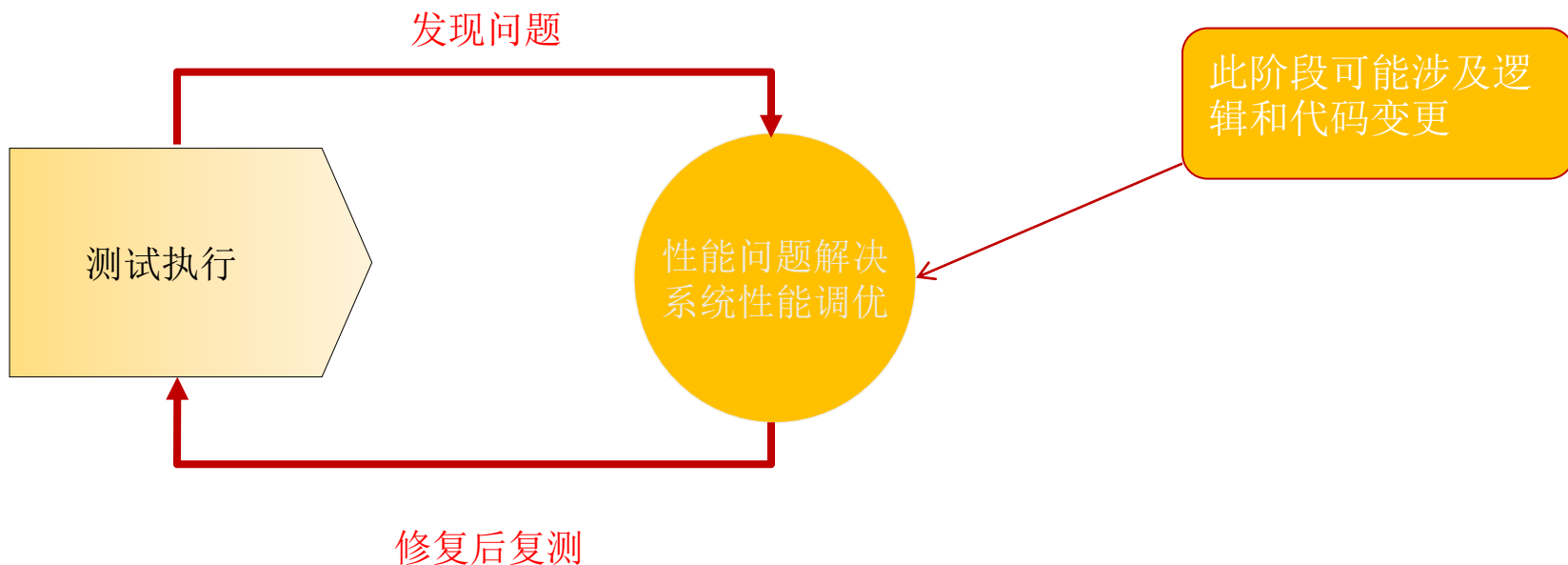
- Schedule中虚拟用户的Ramp up, Duration, Ramp dump 设置正确，对于设定一段时间后自动运行或者到某个时间点自动运行的场景确认点击了 start scenario，场景进入倒计时
- 场景中使用的agent确认能够连通，agent的资源空闲，无外界的干扰。Agent的C盘的磁盘空间足够，temp目录有足够的空间。Agent的host配置正确，确保请求发送到指定的系统。
- 场景中若使用到了集合点，确认集合点的释放规则按需求设置
- 确认场景的RTS(若是由脚本创建场景的，脚本的RTS会初始化场景的RTS。)设置正确，run logic, log已关闭，Download non HTML resources设置正确，Browser simulation 中的选项设置正确。
- 执行测试前根据checklist对测试场景进行确认



第二部分 性能测试过程-执行测试

执行测试场景

性能测试执行是一个发现问题->解决问题，复测->发现新问题的迭代的过程。需要开发人员，系统运维人员，DBA，架构师等共同参与的分析问题，解决问题的过程。





第二部分 性能测试过程-性能调优

性能调优验证

性能调优是一个复杂的过程，系统性能数值，能否达到系统预期调优结果。对系统和应用程序进行调优，以提高响应时间、吞吐量和资源利用率的性能指标。

■ 不同视角的软件性能体现

- 用户视角
 - 相应时间
- 系统视角
 - 并发用户数量
 - 用户操作模式
 - (每秒点击数)
- 调优视角
 - 服务器资源情况
 - 应用服务器资源情况
 - 数据库资源情况
 - 应用时间消耗分布



第二部分 性能测试过程-性能调优

分析与优化

➤ 服务器

CPU
内存
IO

➤ 数据库

缓存命中
索引
单条SQL性能
数据库线程数
数据库连接池数

➤ 中间件

jvm内存
日志
Full GC频率

➤ 网络

吞吐量
吞吐率

➤ 监控工具

用户执行情况
场景状态
事务响应时间
成功率
TPS等

➤ 压力机资源

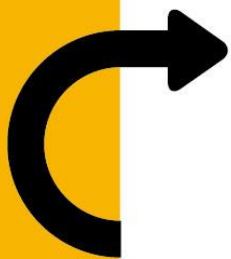
CPU
Memory
网络
磁盘空间



第二部分 性能测试过程-测试报告

整理测试结果和书写测试报告

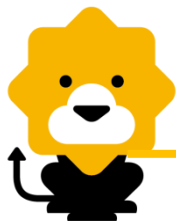
性能测试达到测试目的，则性能测试结束，性能测试人员整理测试过程中的脚本，阶段结果，数据，整理问题的发现和解决过程，书写性能测试报告。最终对外提供性能测试报告作为性能测试项目的终结



第一部分 性能测试理论

第二部分 性能测试过程

第三部分 性能测试流程

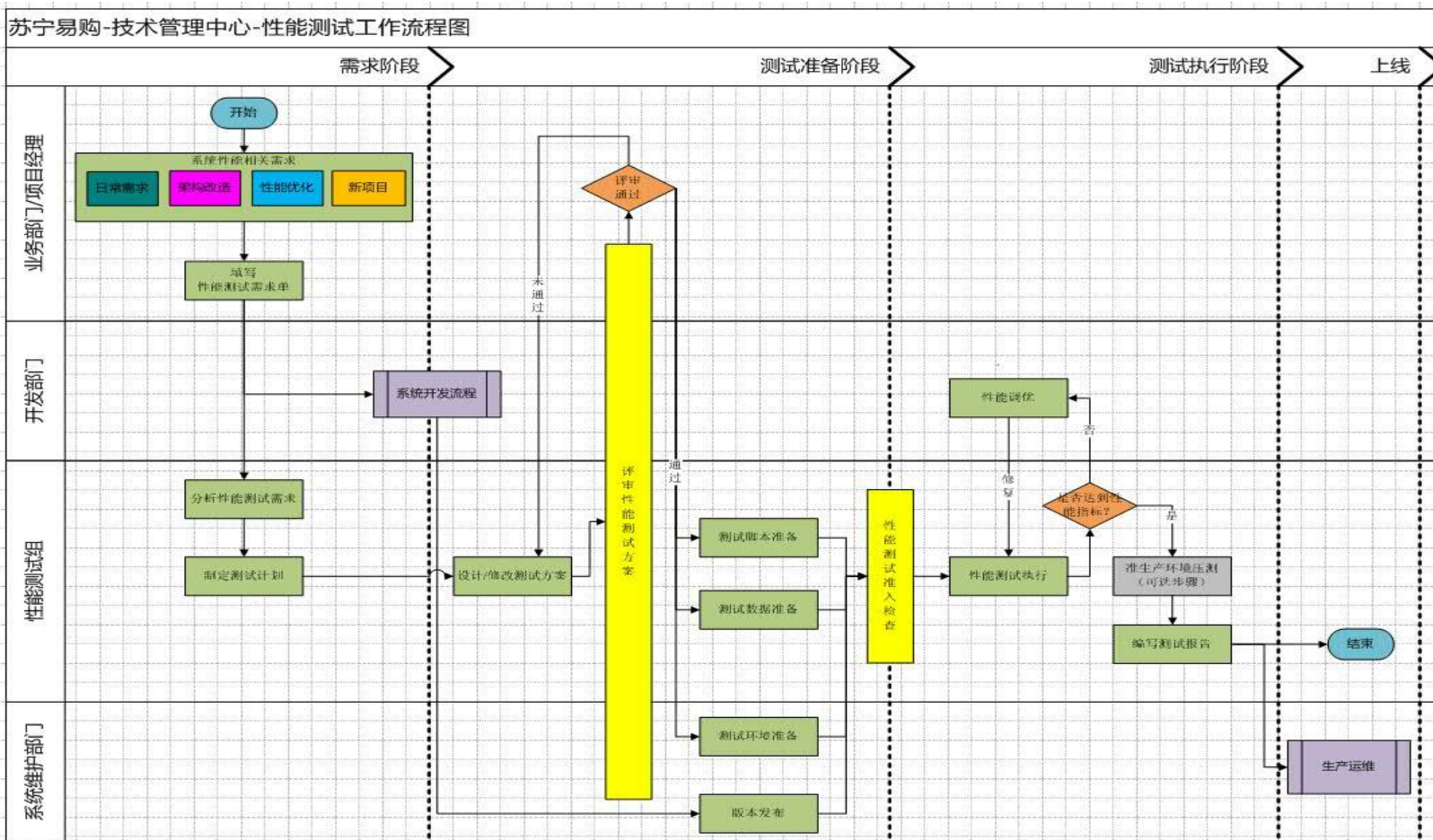


第三部分 性能测试流程

苏宁易购性能测试流程



苏宁易购性能测试
图_v0.1_2012-5-4





第三部分 性能测试流程-参与人员

流程参与人员

业务部门/项目经理：

性能测试需求的提出方，需要明确的提出性能测试的目标，负责讲解业务流程和操作

开发人员：

具体项目或者功能的开发实施人员，参与性能问题的诊断以及优化

性能测试组：

性能测试的准备以及性能测试执行的实施者

系统运维人员：

负责日常系统运维工作。在具体的性能测试项目中负责测试环境的搭建，系统调优的方案的具体实现，系统的监控以及日常的运维。



第三部分 性能测试流程-需求阶段

需求阶段

需求来源

性能测试的需求来源主要有四个：日常需求，架构改造，性能优化，新项目。这与目前苏宁易购的四类项目：维护型，优化型，标准型，新产成品型并不冲突，只是更细程度的划分。



第三部分 性能测试流程-需求阶段

需求提出

新功能的上线，架构的调整，性能优化点，版本的变更若有性能测试需求，则先需要填写性能测试需求单

◆ 需求提出的时间点

性能测试需求的提出至少在系统上线或者期望执行测试前**3-5天**提出。对于新系统，新的项目，需要在立项时，需求设计的阶段若有性能测试需求就要向性能测试组提出书面的性能测试需求。对于临时提出的性能测试需求，除非获得高优先级，否则会根据性能测试的目前工作进展情况排入周工作计划。

◆ 需求单中必须明确以下的几部分

1) 需求基本信息

- 业务联系人
- 开发联系人
- 计划上线时间
- 测试时间要求



第三部分 性能测试流程-需求阶段

需求提出

2) 业务需求概述

- 业务需求概述
- 预期峰值业务场景
- 性能测试的目的和意义
- 是否对其他业务有影响

3) 预期业务性能指标

- 用户访问量
- 页面响应时间
- 订单量
- 订单转化率



第三部分 性能测试流程-测试准备

测试准备阶段

在制定好性能测试计划后，在环境和功能基本稳定可行的情况下，可进入性能测试的准备阶段，主要包括测试脚本，测试数据，测试环境的准备以及版本发布等

方案设计

制定了性能测试计划的同时，需要设计详细的方案，即按照什么样的思路和策略去测试，详细方案中需要设计具体的测试的场景以及测试场景的执行顺序，每个场景的重点关注点等。

方案评审

性能测试组人员制定了初步的性能测试计划和方案后，与架构师，项目经理/业务部门共同交流，讨论，评估测试方案的设计是否合理，若不合理，则需要进一步修改知道三方都认同该测试方案为止。测试方案评审通过以后进入具体的测试准备步骤。



第三部分 性能测试流程-测试准备

测试准备阶段

测试脚本

性能测试工程师开发性能测试的脚本

测试数据

性能测试工程师熟悉业务流程，确认需要的测试数据，在业务人员的配合下准备好测试数据

测试环境

性能测试工程师在获得环境信息和基本的监控权限后，确认系统的架构，软硬件配置信息

版本发布

由运维管理中心的系统管理工程师来进行版本的发布



第三部分 性能测试流程-测试执行

执行阶段

•准入检查

性能测试需要满足一定的条件才能介入，性能测试人员按照checklist逐个检查条件是否满足。若满足条件，则允许进入性能测试阶段。

•测试执行

性能测试准入检查通过以后，性能测试人员按照设计的测试方案，测试场景进行测试，或者在系统调优之后进行复测，测试过程中记录测试过程中发现的问题，收集整理测试过程中的分析材料，测试结束后收集并整理测试结果。

性能调优

性能测试执行过程中遇到性能问题时，性能测试人员需要记录性能问题，并收集详细的分析材料，调优过程中架构师，系统运维人员，开发人员，DBA需配合分析问题，提出优化修改的建议。性能测试人员需要记录问题的分析过程以及解决方案。在性能问题修复后转入测试执行步骤验证优化后的效果。



第三部分 性能测试流程-测试执行

执行阶段

准生产环境压测(可选)

针对有些性能测试环境架构相对比较复杂，性能测试环境测试通过后，为保证准生产环境在正式对外后能正常提供服务，减少上线的风险，由性能测试人员申请在准生产环境进行较小压力的浏览查询相关的测试。上报项目经理，由项目经理协调领导确认是否需要进行准生产环境的压力测试。

测试报告

经过测试执行以及性能调优的迭代过程后，系统的性能指标满足测试目标，测试执行阶段结束。性能测试人员书写性能测试报告。



第三部分 性能测试流程

上线前阶段

性能测试人员提交性能测试报告，并告知项目的参与人员。

Thanks!

