性能测试流程与实战

薛亚斌@土司阿哈

个人简介

个人简介

- 在神州数码、亚信、用友、京东等企业从事开发、项目经理、架构师、测试总监等岗位。
- 从零搭建飞信服务端分层测试体系，并搭建国内较早的接口测试平台。
- 亲历京东金融移动端日活三十万到千万的全过程，从零建立金融移动端质量保障体系，主导七次大促活动。
- 公众账号“土司阿哈”维护人，发表文章180余篇，40多万字。
- 在TID、TOP100、MTSC等在大会发表演讲30多次。
- 中国软件测试经理联盟发起人之一。

PART ONE

01

基础概念

• • •

生活中那些系统问题

北京奥运票务系统

票务系统已经做了多次压力测试,票务系统每小时将能处理3万张门票的销售,以及承担每小时100万次以上的网上浏览量;实际承受了每小时800万次的流量压力，每秒钟从网上提交的门票申请超过20万张；系统在启动不久就出现了处理能力不足的问题

12306购票系统

12306的高峰访问是10亿页面访问量，集中在早8点到10点，每秒访问量在高峰时上千万。12306根本无法面对“春运”期间的瞬间海量高并发，一度出现用户无法登陆、访问速度过慢以及频繁报错等现象，引起怨声一片。

淘宝双11

淘宝双11凌晨秒杀时，因访问量较大出现很多用户提交不了的情况。

性能测试目的

- **能力验证：主要验证软件是否满足规定的性能指标要求。**
  - ✓TPS 描述方式："某系统能否在A条件下具有B能力？"
  - ✓要求在已确定的环境下运行
  - ✓需要根据典型场景设计测试方案和用例
- **能力规划：主要测试软件在某种条件下运行的性能状况。**
  - ✓ 描述方式：“某系统能否支持未来一段时间内的用户增长” 或 “应该如何调整系统配置，是系统能够满足增长用户数的需要”
  - ✓ 是一种探索性的测试
  - ✓ 可被用来了解系统的性能以及获得扩展性能的方法
- **性能调优：通过性能测试找出软件的性能瓶颈，分析出缺陷的原因，并进行针对性的性能优化，以改进软件性能。**
  - ✓ 确定基准环境、基准负载和基准性能指标
  - ✓ 调整系统运行环境和实现方法，执行测试
  - ✓ 记录测试结果，进行分析、响应时间、网络、超时概率、稳定性、CPU、Memory、Load、I/O）

性能测试的重要性

质量重要组成部分

性能测试是软件质量保证的重中之重，功能存在缺陷只是单个功能无法使用，而一旦性能出现问题，极容易导致严重的生产事故

用户眼中良好形象

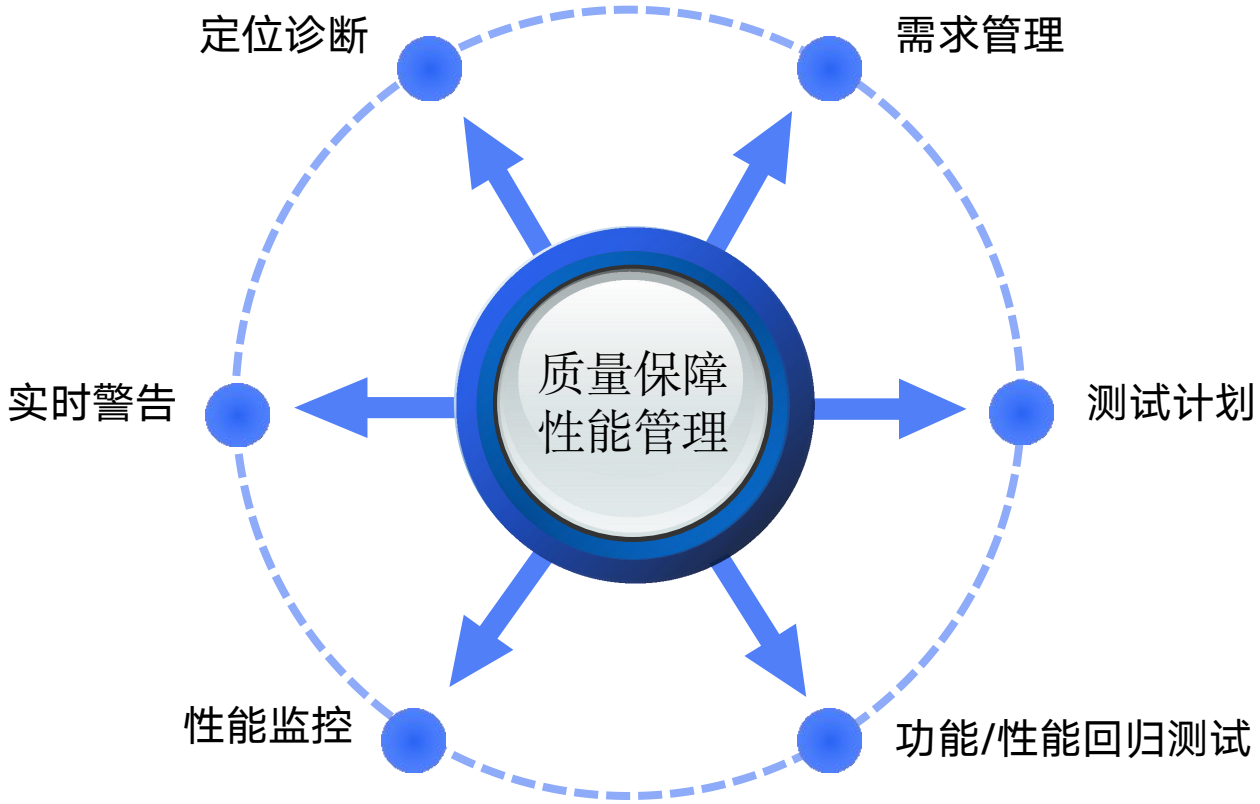
显示慢甚至苦苦等待最后却得到系统忙无法访问的提示，远比界面布局差、功能使用不方便而给用户留下的印象恶劣的多

节约成本重要手段

根据性能测试的分析结果可以精确的判断需要多少个服务器，服务器上需要多少CPU和内存，租用多少带宽的专线等等，不会因为无根据的拍脑袋确定导致多花钱而浪费了资源。

性能测试介入时机

性能测试通常是在功能测试进入末尾阶段或开展一阶段后进行，其执行时间较功能测试时间短，但通常也需要较长时间的准备工作。



性能测试的局限性---为什么测试不准

- 性能测试有限性导致测不准
  - 没有测试到的场景和业务
  - 软件异常流程
  - 硬件、网络环境不一样
  - 客户端性能问题（特别是移动端）
- 用户行为变化导致的测不准
- 用户数据增长导致的测不准

性能测试一些概念-测试指标

1. TPS
2. 响应时间
3. 错误率
4. CPU
5. 内存
6. 网络

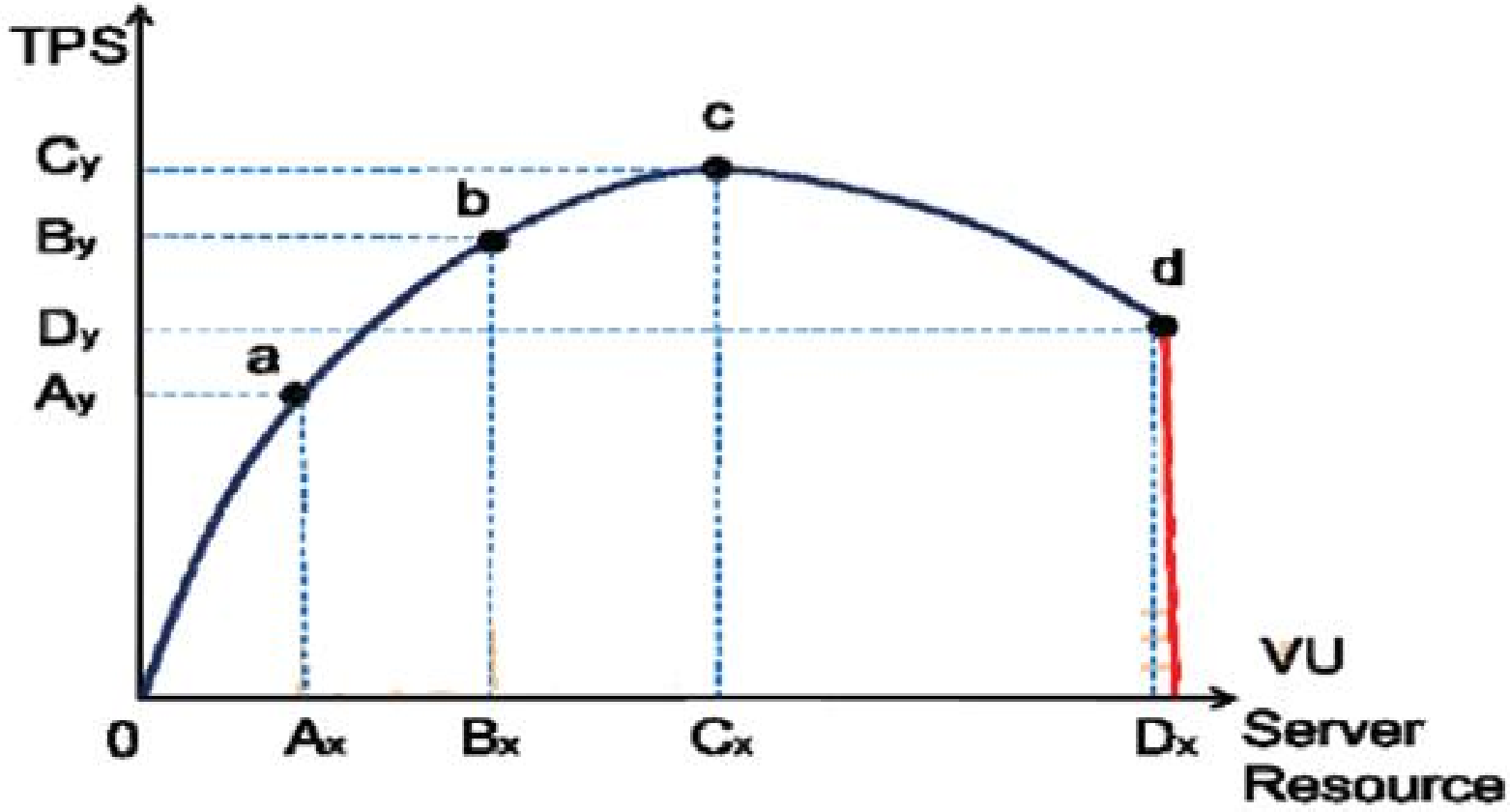
性能测试类型---测试模型

a点：性能期望值

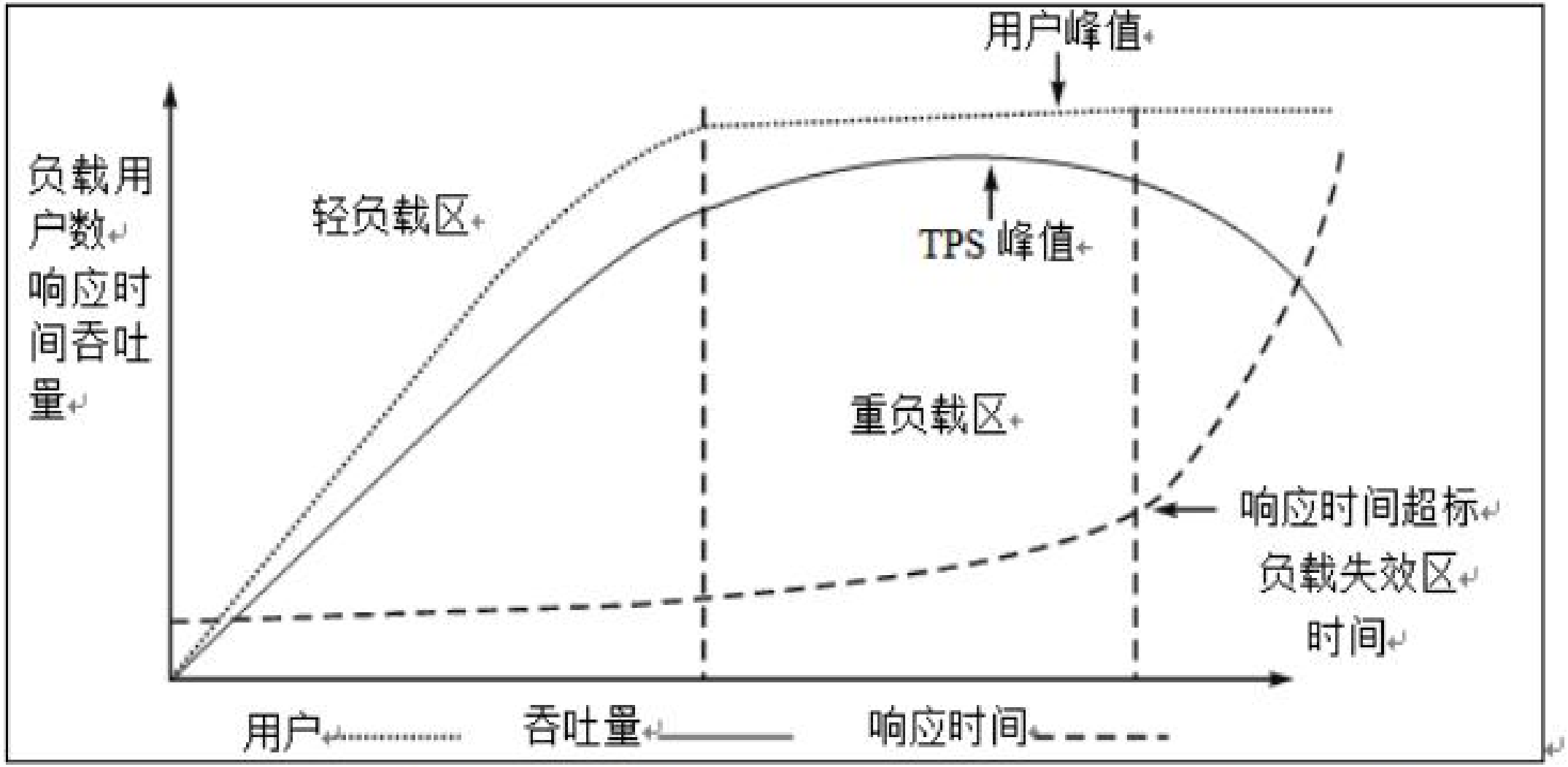
b点：高于期望，系统资源处于临界点

c点：高于期望，拐点

d点：超过负载，系统崩溃



性能测试流程---拐点模型


用户峰值
负载用户数
响应时间
吞吐量
轻负载区
TPS 峰值
重负载区
响应时间超标
负载失效区
时间
用户
吞吐量
响应时间

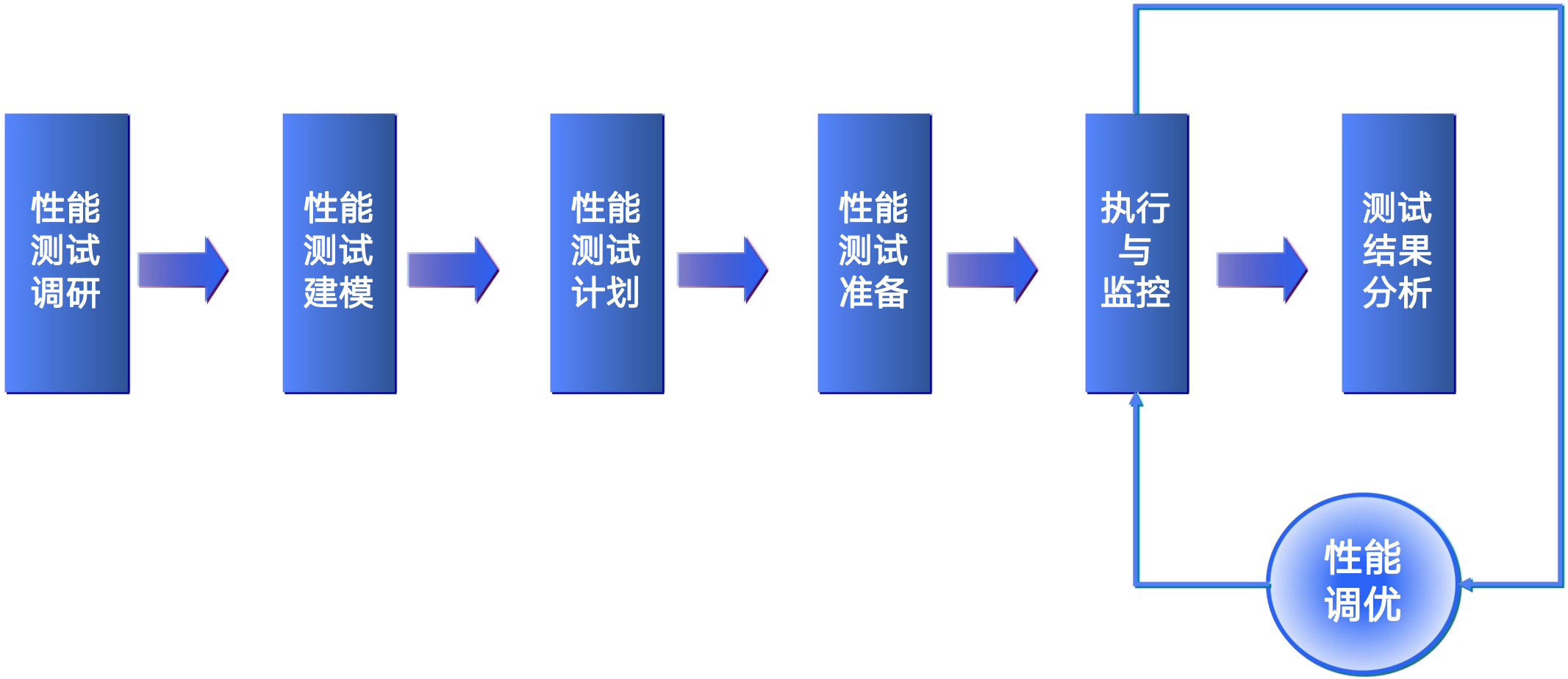
PART TWO

02

测试流程

• • •

性能测试流程



需求调研---测试业务分析

场景设计策略

业务运营

1. 根据最终用户真实使用行为和场景使用的频率来设计场景；
2. 从业务重要性、业务规律以及未来增长趋势设计场景。

技术运营

1. 根据业务架构、技术架构、部署架构、网络架构等设计场景；
2. 根据业务接口调用频率、业务调用链路以及调用时间分布来确定设计场景。

线上问题

1. 根据线上问题分类归集分析设计测试场景；
2. 根据用户问题反馈设计测试场景；
3. 根据线上问题修复策略设计测试场景

测试经验

1. 借鉴行业经验（银行、电信）设计测试场景；
2. 借鉴互联网全链路测试设计测试场景；
3. 根据行业特点以及之前测试经验设计测试场景。

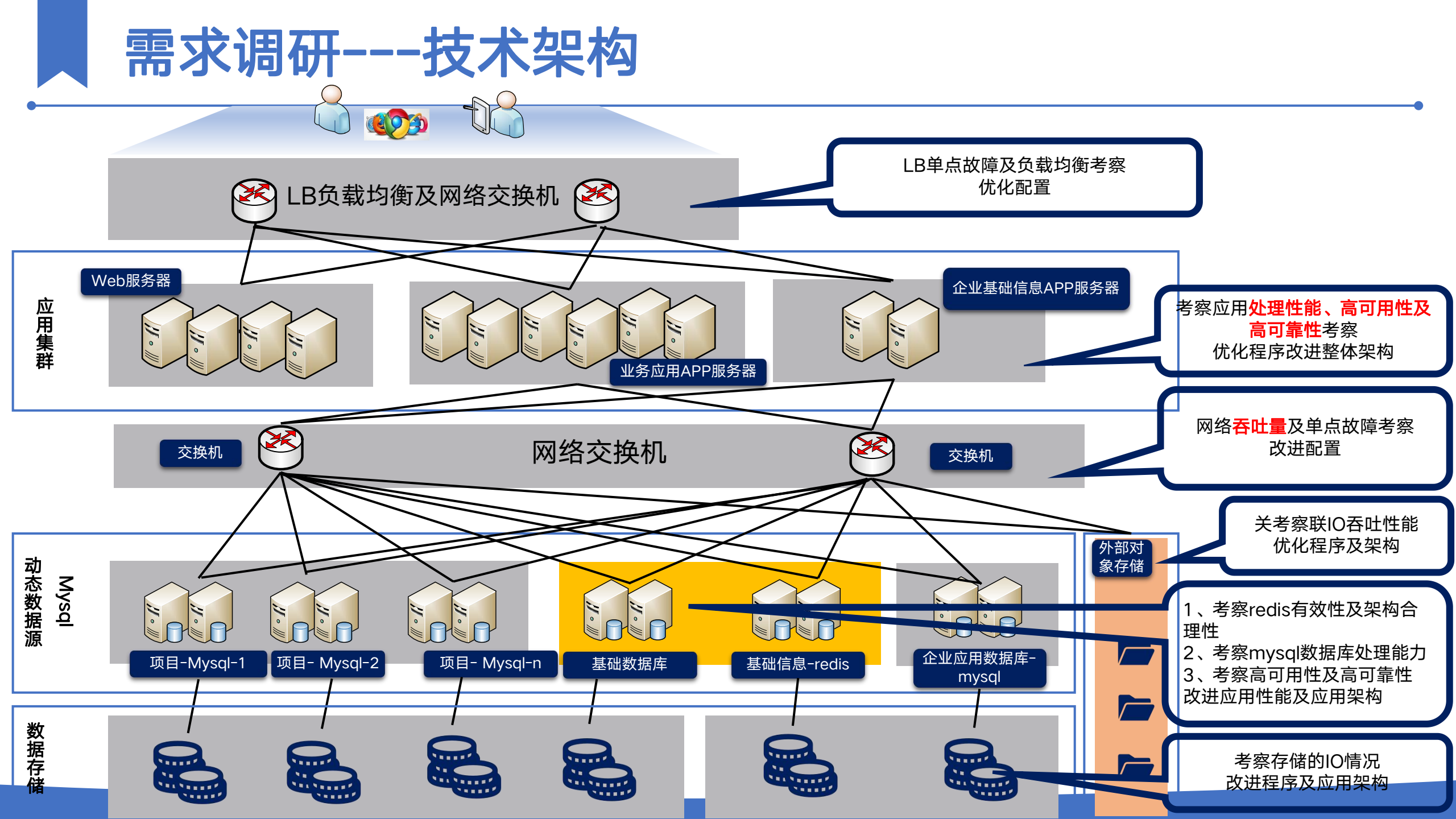
需求调研---业务分析

- 用户量访问较大的功能，应该优先纳入性能测试场景
- 与金钱相关比较重要的场景，应该优先纳入性能测试场景
- 影响业务主流程的场景，应该优先纳入性能测试场景
- 开发人员认为可能存在性能问题的场景，应该优先纳入性能测试场景
- 应该考虑综合场景，防止线程争用导致现场思锁以及数据库死锁
- 应该做稳定性测试场景测试，防止长时间运行导致内存泄漏情况发生

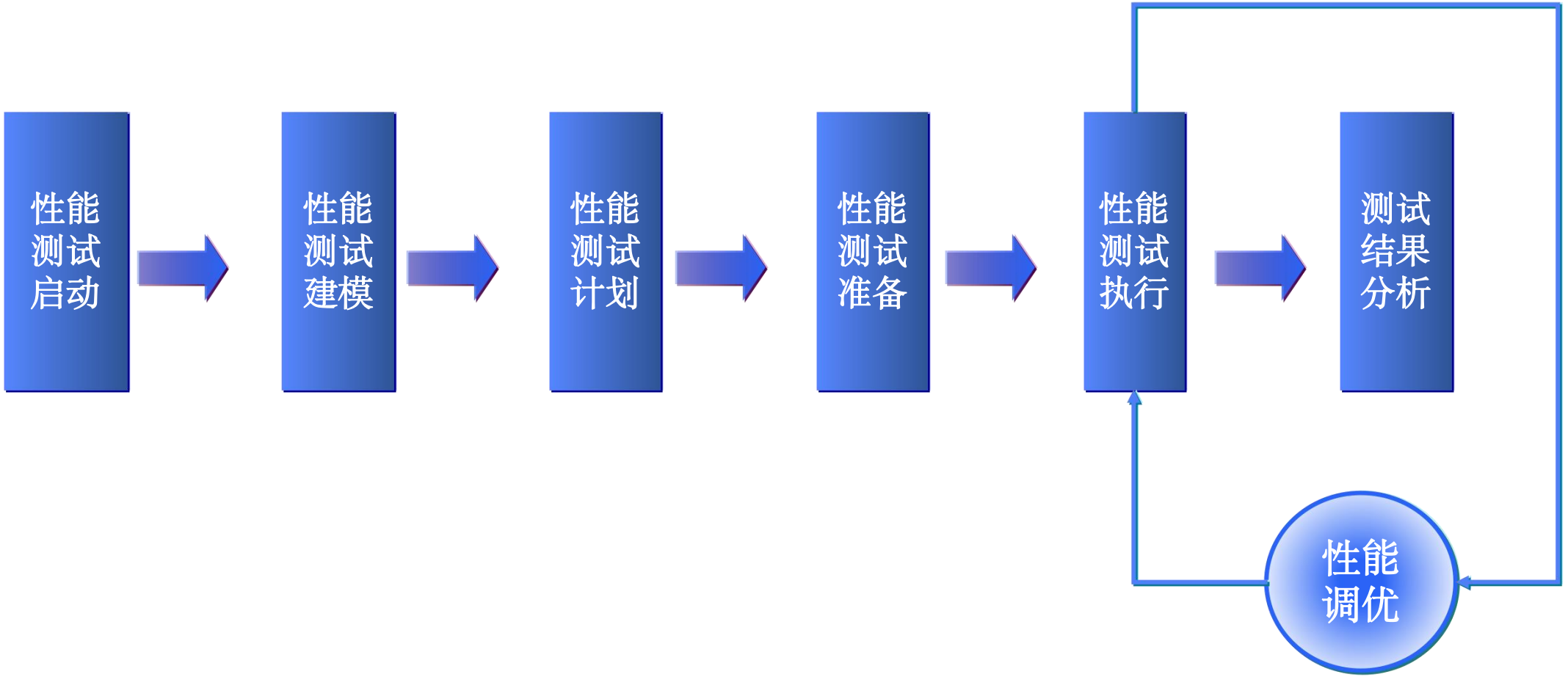
需求调研---系统架构分析

- **上下游依赖关系：**性能测试业务上游依赖业务，及下游发送业务，如结算页上游业务为添加购物车，下游业务为接单服务
- **数据流向：**性能测试业务数据如何流转，如测结算页是从客户端请求，到应用服务，最后保存到数据库中。
- **部署情况：**性能测试业务使用到的服务器配置以及在线上的实际部署情况，如结算页用docker是多少C多少内存以及网络带宽等，在廊坊和马驹桥各部署了多少台服务器，以及应用参数。

需求调研---技术架构


LB负载均衡及网络交换机
LB单点故障及负载均衡考察
优化配置
应用集群
Web服务器
业务应用APP服务器
企业基础信息APP服务器
考察应用处理性能、高可用性及高可靠性考察
优化程序改进整体架构
交换机
网络交换机
交换机
网络吞吐量及单点故障考察
改进配置
Mysql
动态数据源
项目-Mysql-1
项目- Mysql-2
项目- Mysql-n
基础数据库
基础信息-redis
企业应用数据库-mysql
外部对象存储
关考察联IO吞吐性能
优化程序及架构
1、考察redis有效性及架构合理性
2、考察mysql数据库处理能力
3、考察高可用性及高可靠性
改进应用性能及应用架构
数据存储
考察存储的IO情况
改进程序及应用架构

性能测试流程



业务建模---业务建模两种方式

- 用户行为模型
  - ✓描述高峰时期用户行为特点(瞬时模型)
  - ✓优点：对与大多数性能测试工具而言实现简单
  - ✓缺点：用户行为较难统计分析
- 系统业务模型
  - ✓描述高峰时期系统业务特点(时段模型)
  - ✓优点：相对于用户行为模型而言较容易获得
  - ✓缺点：设置复杂，需要较高的工具技能

性能测试流程---用户行为分析

- **前提条件**：理发店有 3 个理发师，2 个洗头工，3 个工作座椅，2 个洗头座椅，6 个等待座椅，剪发、护理之前需先洗头，顾客最大忍受等待时间 2 小时。

服务编号	服务项目	耗时(min)	执行人	费用(剪发/护理费用包含洗发)	店铺效益(H)
S1	洗发	10	洗头工	/	30元
S2	剪发	20	理发师	30元	90-5*3=75元
S3	护理等	50	理发师	90元	108-5=103元

  - 【系统分析】由前提条件可以分析得出:
  - 本店最大顾客接待量为3+2+6=11位，即2位洗头，3位剑法或护理，6位等待；
  - 若顾客同时进店，则最大容纳顾客量为2+6=8位，即2位洗头，6位等待；
  - 剪发护理均由理发师完成，则店铺最多剑法顾客场景是：3为正在剪发/护理，2位洗头，6位等待，合11位。

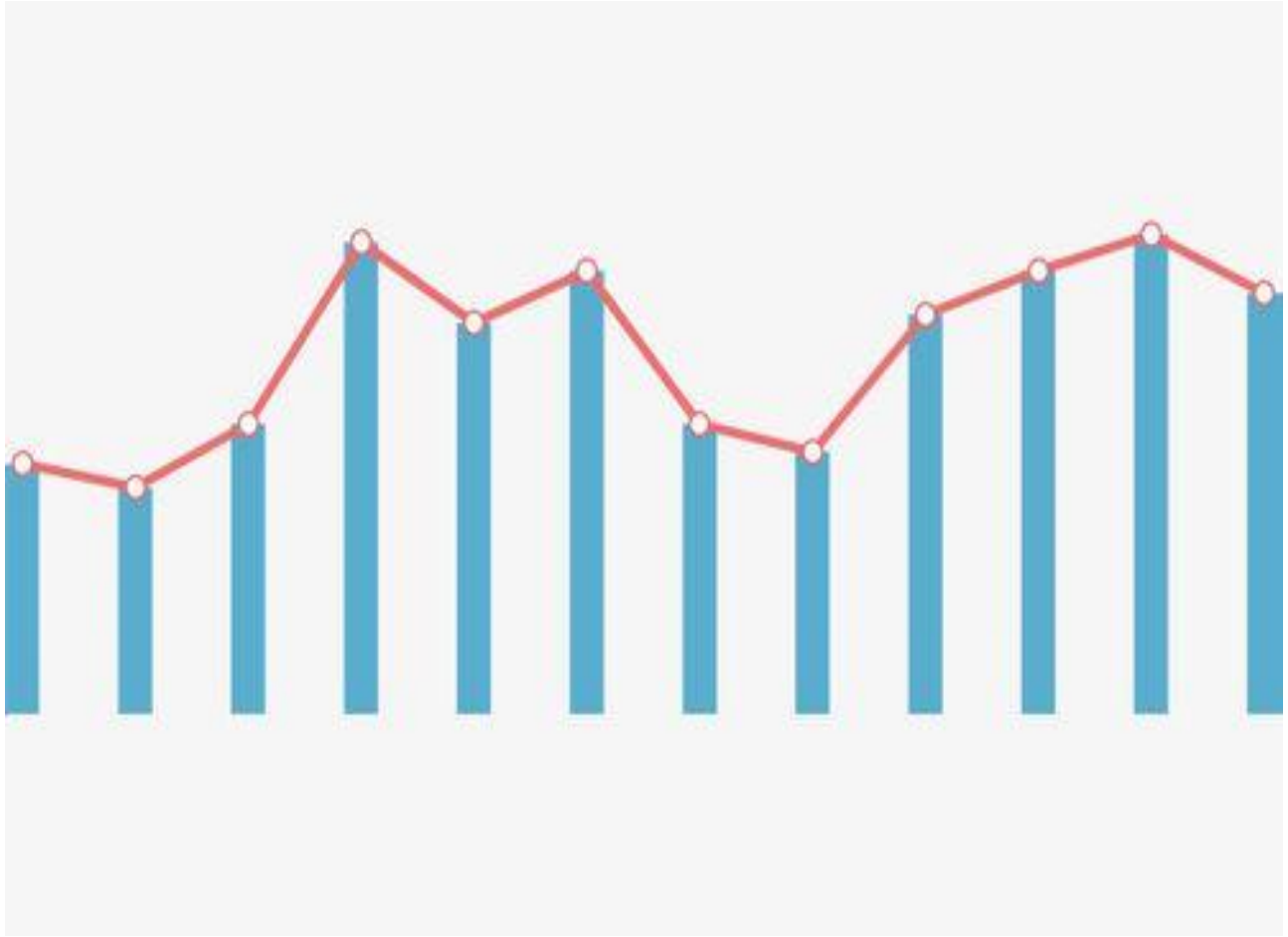
测试建模---系统业务模型

系统业务模型，主要定义测试完成哪些业务？完成速率？核心是建立与生产环境一致的压力测试模型；建立业务模型的一般原则。

- 最好是实际用户行为的统计，如果没有借助同类软件的用户行为统计，再没有，根据有经验人员的预估。
- 把用户所有可能使用业务按使用频繁程度排序，频繁程度越高就越应该纳入测试场景。
- 查看业务消耗计算资源的程度，预计消耗程度越大的越应该纳入测试场景

测试建模---系统业务模型

- 通过获取生产环境高峰时段的交易以及交易量，建立仿真的业务模型；
- 收集高峰月、天、小时的交易量；
- 收集高峰小时按交易量排名的业务列表，取前十个左右的交易。
- 收集线上性能问题聚聚的业务场景



性能测试建模---用户行为分析与指标模型

多少用户（WHO）在什么时间或者持续多长多久（When），在多大的数据量的基础上（How much），完成了什么业务（What），最终需要关注怎样的指标（How）

- 系统总容量达到日委托6000万笔，成交9000万笔
- 系统处理速度每秒7300笔，峰值处理能力达到每秒10000笔
- 实际股东帐号数3000万

思考	最佳并发用户数需求： 最大并发用户数需求： 基础数据容量： 业务数据容量：

测试建模---性能测试指标模型

- 用户支撑能力测试/用户体验测试
  - ✓ 调研用户使用方式，确定Think Time
  - ✓ 基准测试确定各交易基准执行时间
  - ✓ 调整Interval时间
  - ✓ 风险
- 系统处理能力测试
  - ✓ 调研系统处理量需求
  - ✓ 基准测试确定各交易基准执行时间
  - ✓ 调整Interval时间
  - ✓ 风险

性能测试建模---建模的方法依据

- 开发过程相关文档
- 相似项目性能需求
- 业界公认标准
- 用户使用模型
- 80/20原则
- 业务分布图
- 系统日志

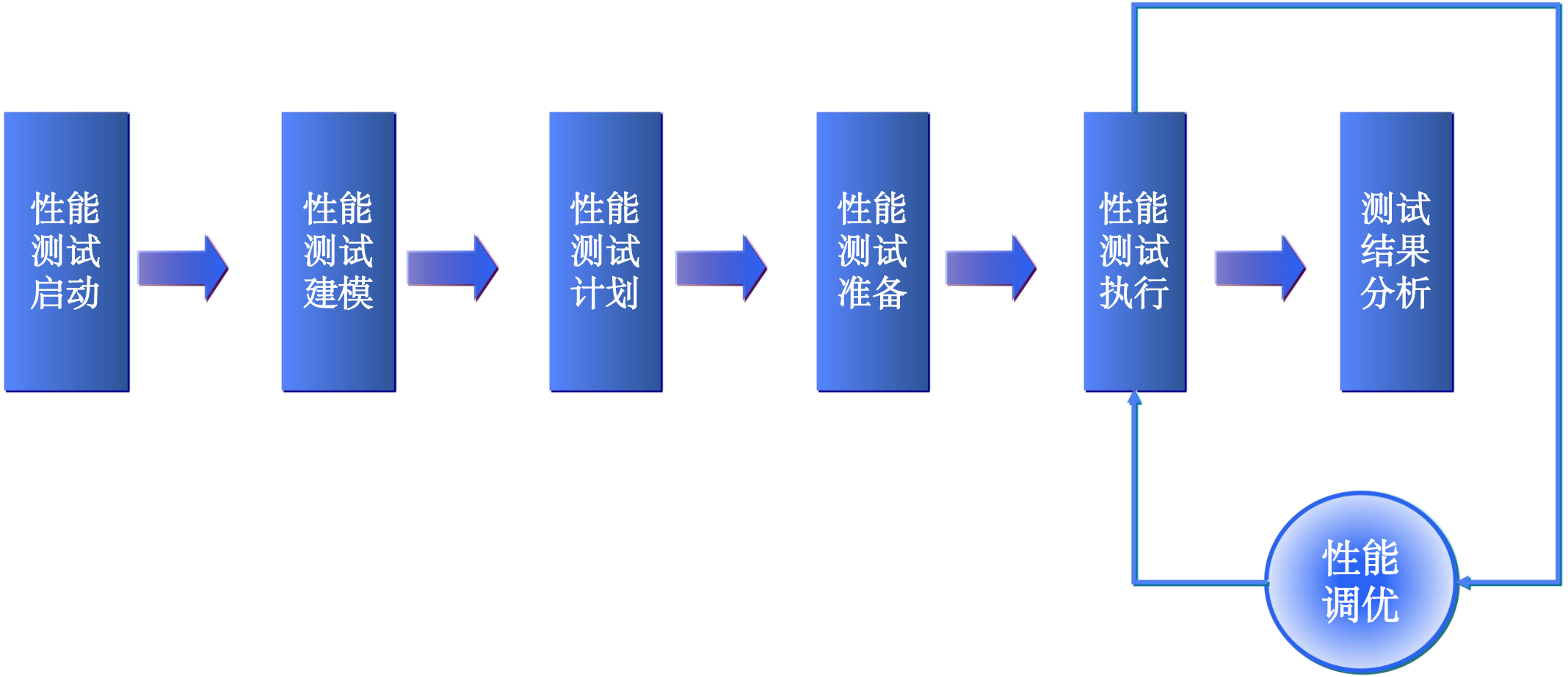
性能测试流程---用户体验与感知

- 2s以内，用户感觉良好
- 2-5s，用户感觉可以接受
- 5-8s，用户会觉得烦躁，可能频繁刷新页面
- 超过8s，用户无法忍受，直接离开

测试建模---测试指标一些经验

- CPU限制： vmstat user+sys超过80%，有一定压力时60%~80%属于正常，正常情况下25%；
- 磁盘I/O限制：vmstat iowait超过40%；
- 应用磁盘限制: iostat tm_cat超过70%
- 虚存空间： lsps -a 分页空间的活动率超过70%

性能测试流程



测试方案---测试计划&实施方案

- 测试经理根据测试启动阶段得到的测试模型编写测试方案和测试计划
- 用于指导整个测试实施过程

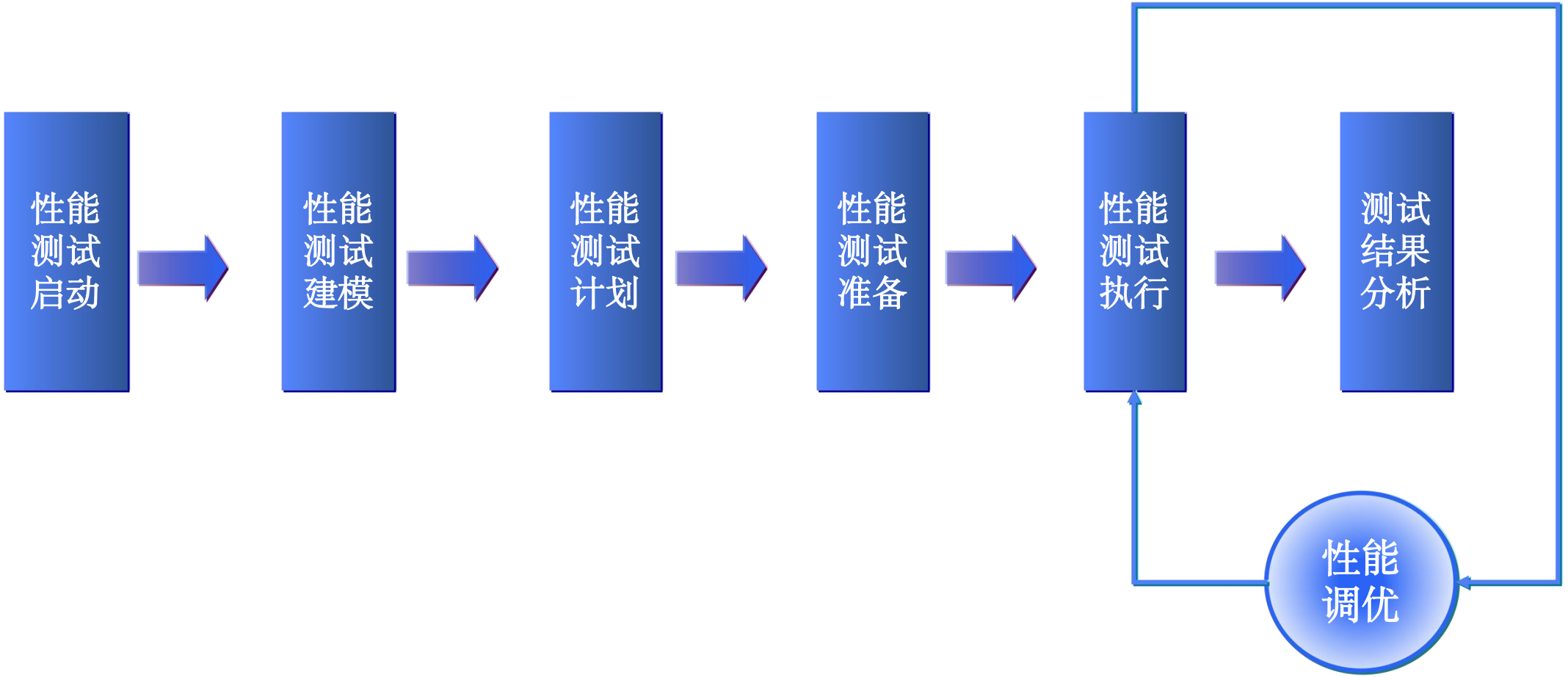
测试方案---关键文档

- 《性能测试方案》
- 《性能测试计划》
- 《典型交易列表》
- 《性能测试需求表》

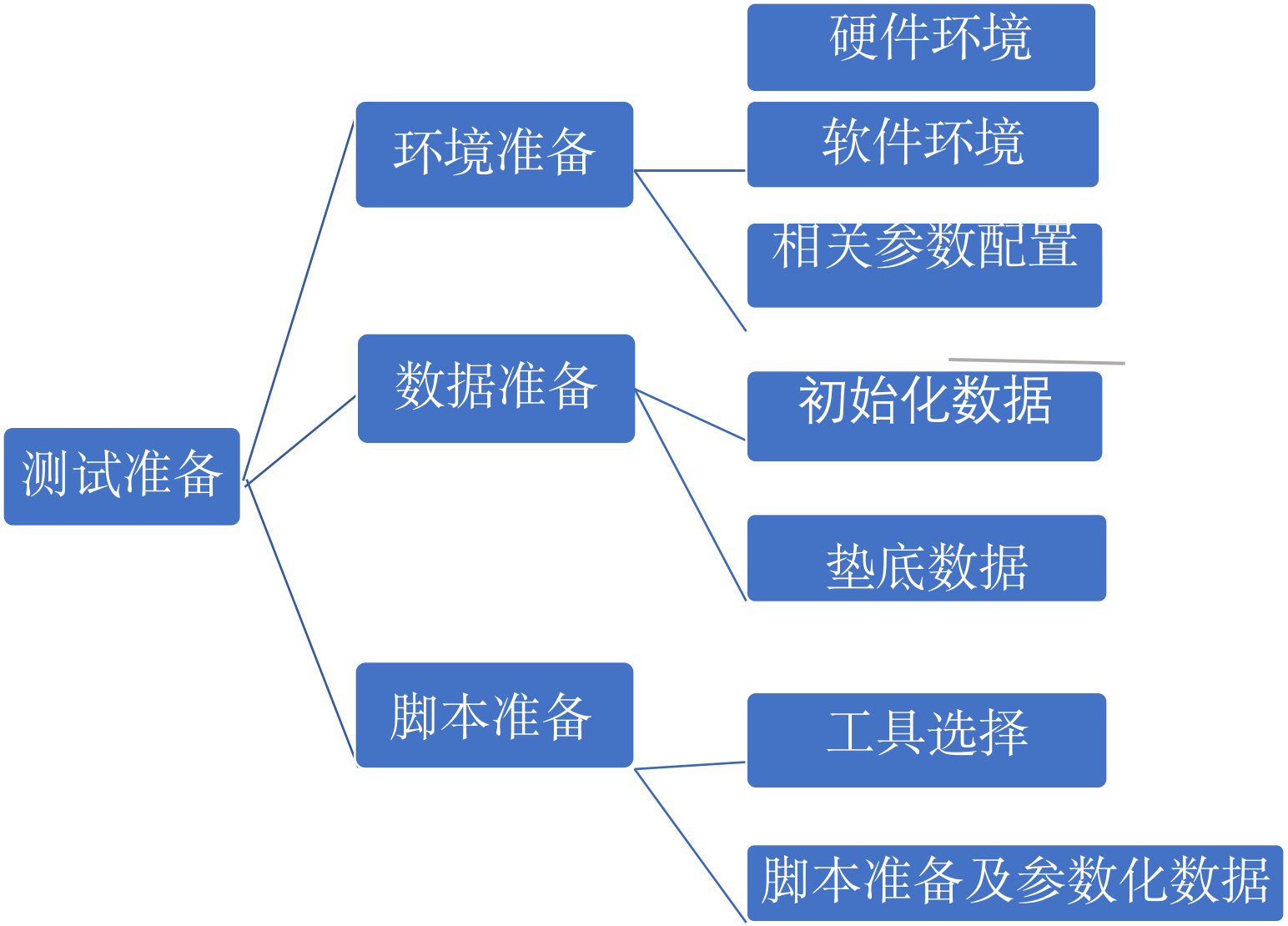
测试方案---方案的关键点

- 测试目的（清晰明确，没有歧异）。
- 测试范围（交易列表，路径图等）。
- 性能指标（要可测量，量化指标最好给出具体数值，无法定量的给出说明）。
- 数据量 （给出具体数值和参考依据，无法定量的给出说明）。
- 测试环境（分为网络、硬件、软件和拓扑图）。
- 测试工具和监控工具及其相关环境。
- 风险控制（风险描述、严重程度、规避办法、负责人等明确清晰）
- 测试策略（符合项目实际情况，具有可执行性）
- 挡板（根据情况可裁减）
- 时间戳（根据情况可裁减）
- 角色分工（无遗漏，职责描述清楚）
- 测试执行过程等（符合项目实际情况，没有遗漏，具备可执行性）

性能测试流程



性能测试工作准备


测试准备
环境准备
硬件环境
软件环境
相关参数配置
数据准备
初始化数据
垫底数据
脚本准备
工具选择
脚本准备及参数化数据

测试环境准备---环境需要

- 硬件环境不一致
- 软件环境版本不一致
- 软件配置项不一致
- 网络环境配置项不一致
- 集群环境不一致

测试环境准备---基础数据（铺地数据）

基础数据 （铺地数据）：性能测试之前，测试环境中已有的数据总和，基础数据的原则

- 必须调查或者预测出数据库表中每个表应该有多少行的数据。
- 而且数据取值要实际情况一样丰富。
- 数据长度和实际情况相同
- 基础数据决定着数据库server的cpu、内存、io使用或其他多种资源的使用逻辑。

测试环境准备---业务数据

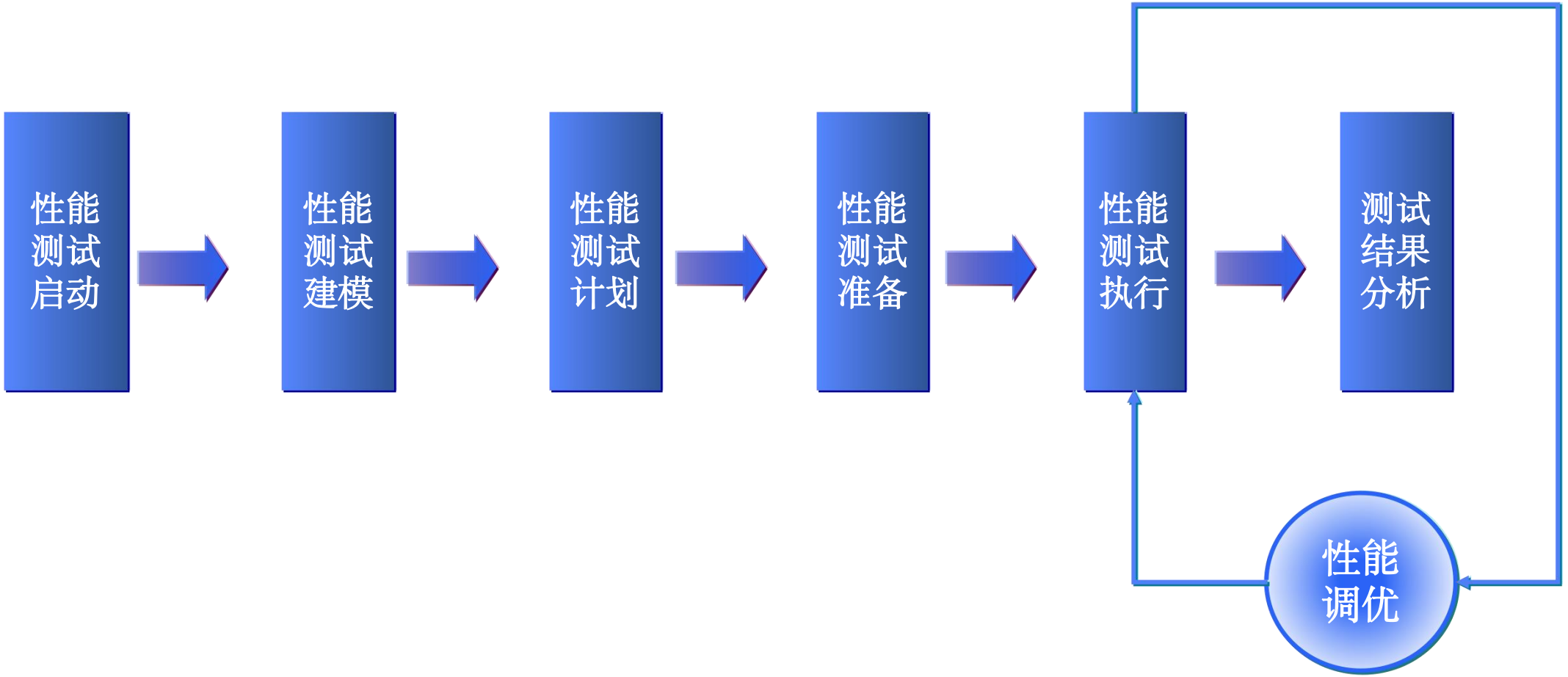
测试数据（业务数据）： 从基础数据中选取的，参与到性能测试中的数据，选择原则。

- 在应用合理范围内随机挑选数据、挑选足够的量。
- 挑选数据的方式通常影响数据库的内存和io，有状态服务的cpu和内存，线程、锁等。

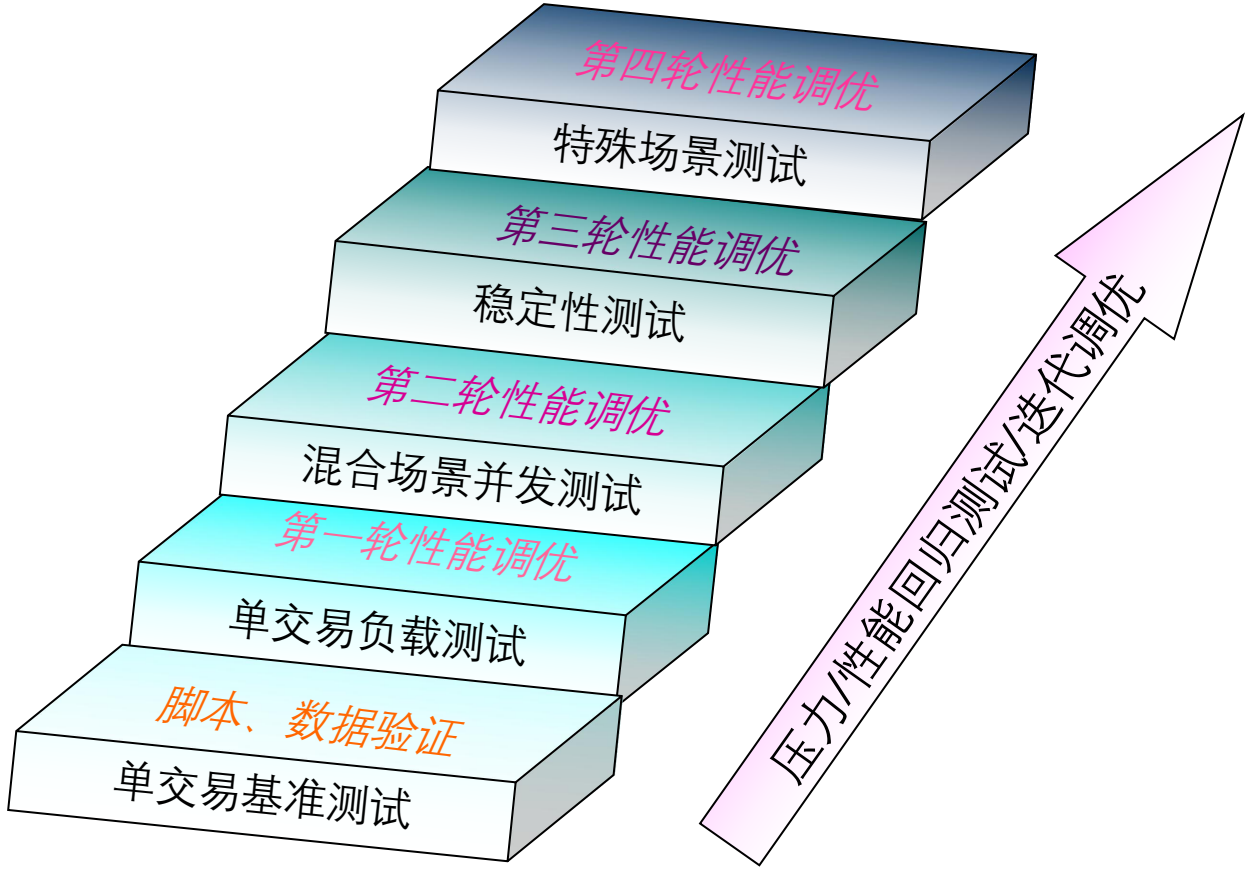
测试脚本编写---测试脚本

- 项目组(业务&开发)：
  - 提供脚本录制时所需数据
  - 解决录制过程中出现的系统问题。
- 测试主管单位：
  - 录制脚本并调试，确保脚本能在测试环境中正确运行。
    - 脚本是否能够真实模拟实际操作
      - Think Time
      - Transaction
      - 参数化
      - 集合点
      - 关联
      - 检查点
      - Run time Settings
    - 脚本是否存在并发问题
      - 2X2验证

性能测试流程



测试执行---执行策略



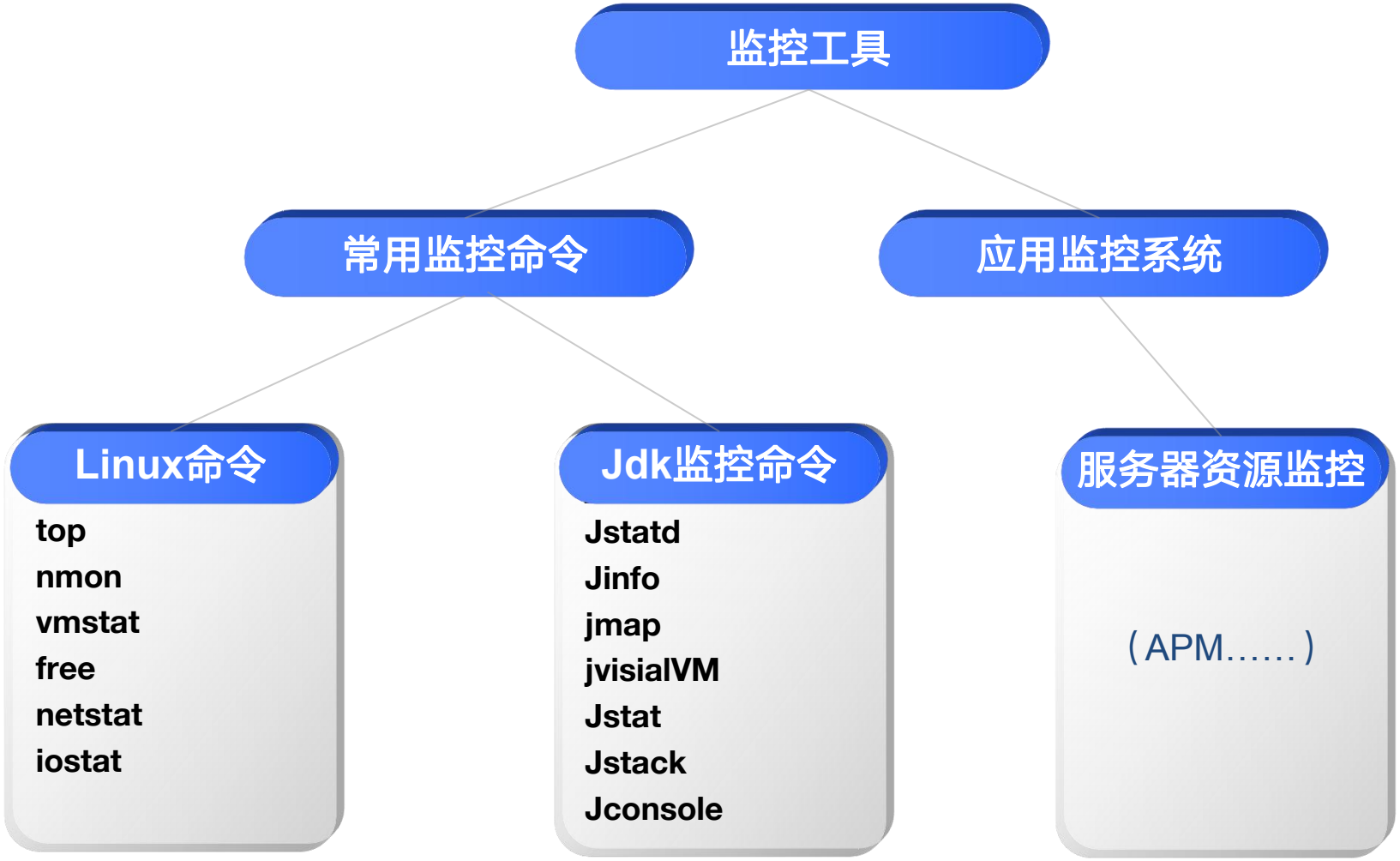
执行记录　结果搜集　数据清理　环境恢复

测试执行---执行过程的关键点

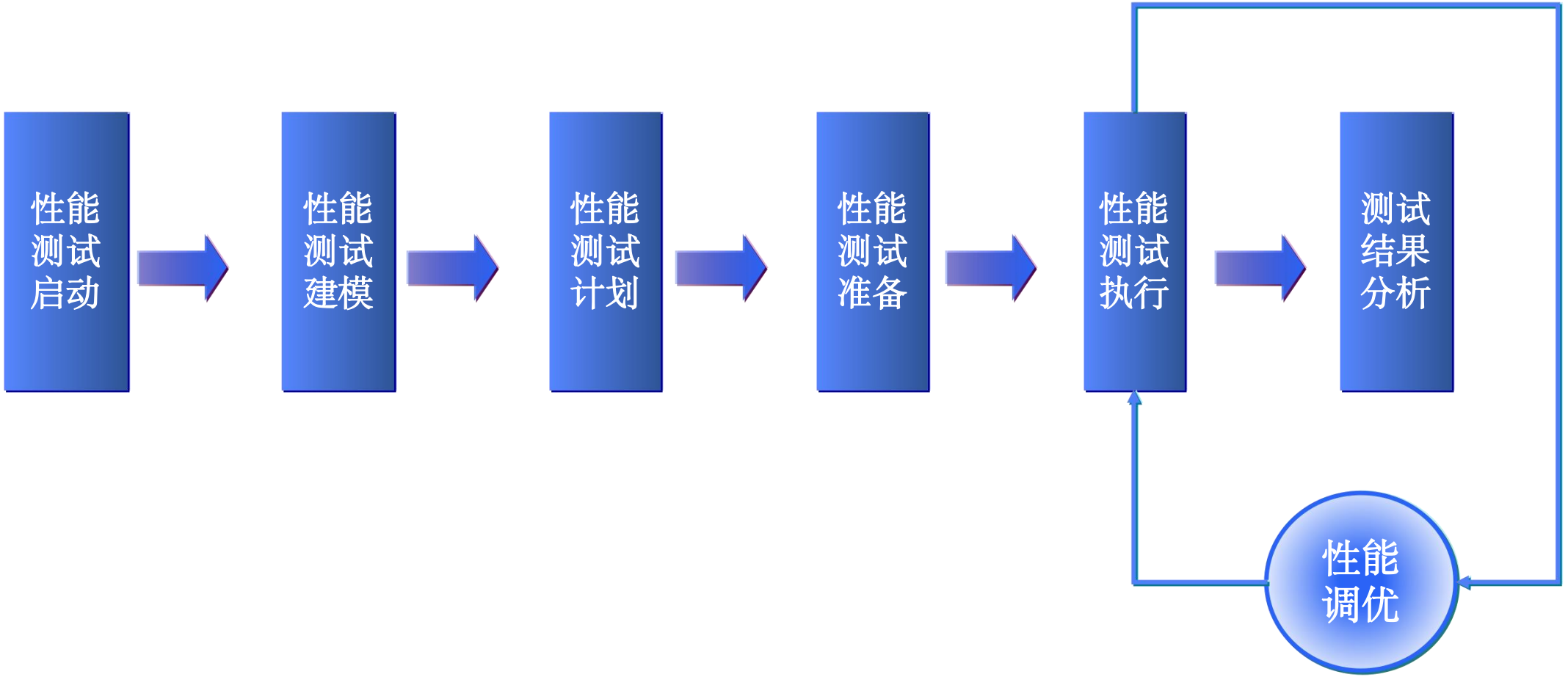
- 执行记录
- 结果搜集
- 数据恢复
- 环境清理

注意：
每次性能测试的结果必须要全部收集、合理命名并保存，否则在测试完成后很可能发现某次测试需要重做……

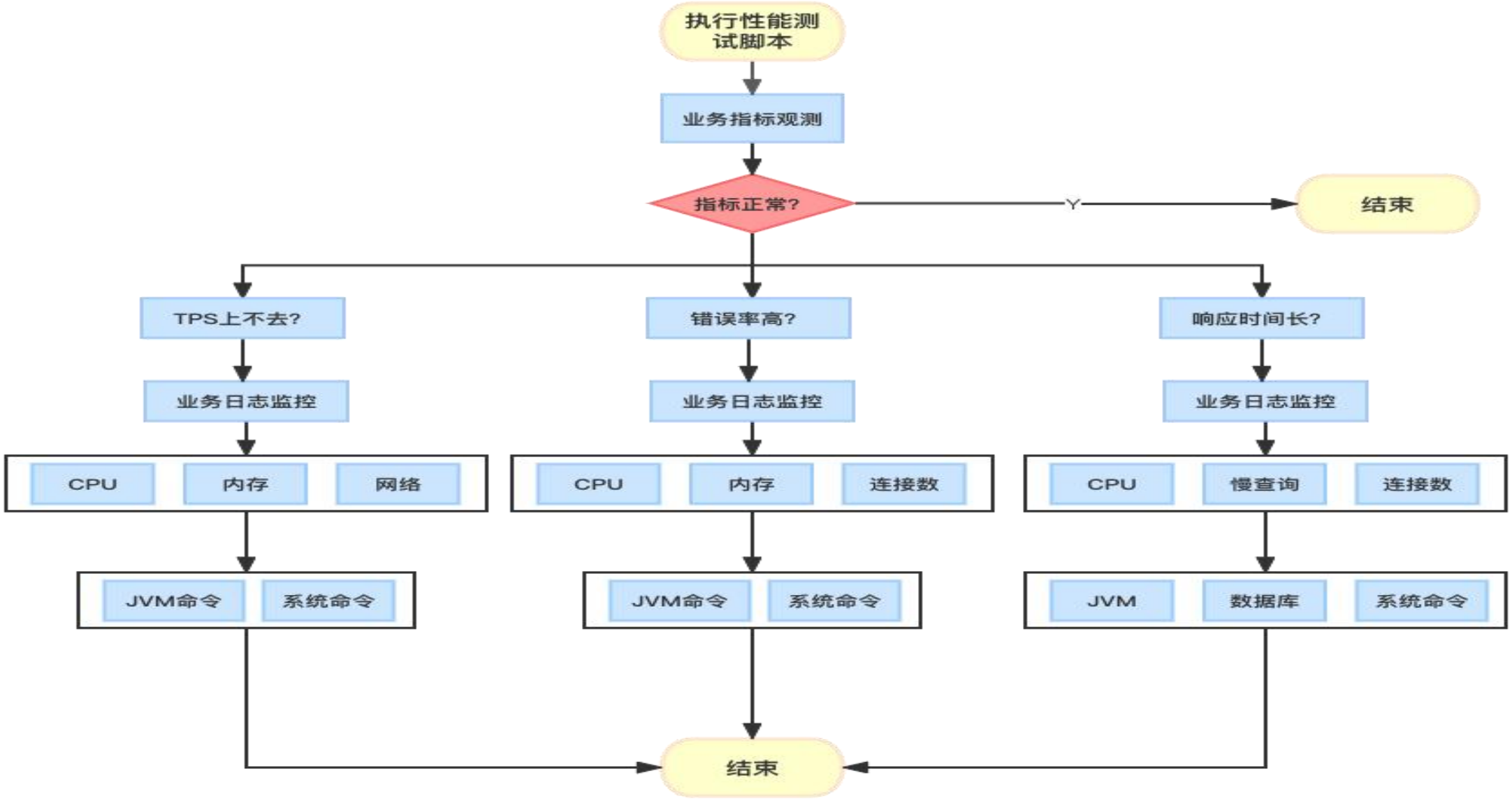
测试执行与调优---测试监控



性能测试流程



测试执行与调优---测试执行监控



测试执行与调优---常见问题分析

服务器性能瓶颈	1、cpu 使用率情况 2、内存 内存使用率 3、磁盘io读写程度、网络使用情况
应用服务性能瓶颈	1、web服务器等应用软件例如tomcat、nginx 2、数据库系统瓶颈，比如索引、慢查询、数据库磁盘io
系统性能瓶颈	1、主要是开发系统代码上的瓶颈例如逻辑处理、日志级别
操作系统的性能瓶颈	1、linux、unix服务器内核参数设置
如何去定位	1、通过性能监控工具查看例如nmon工具、查看jvm使用情况 2、通过查看系统日志

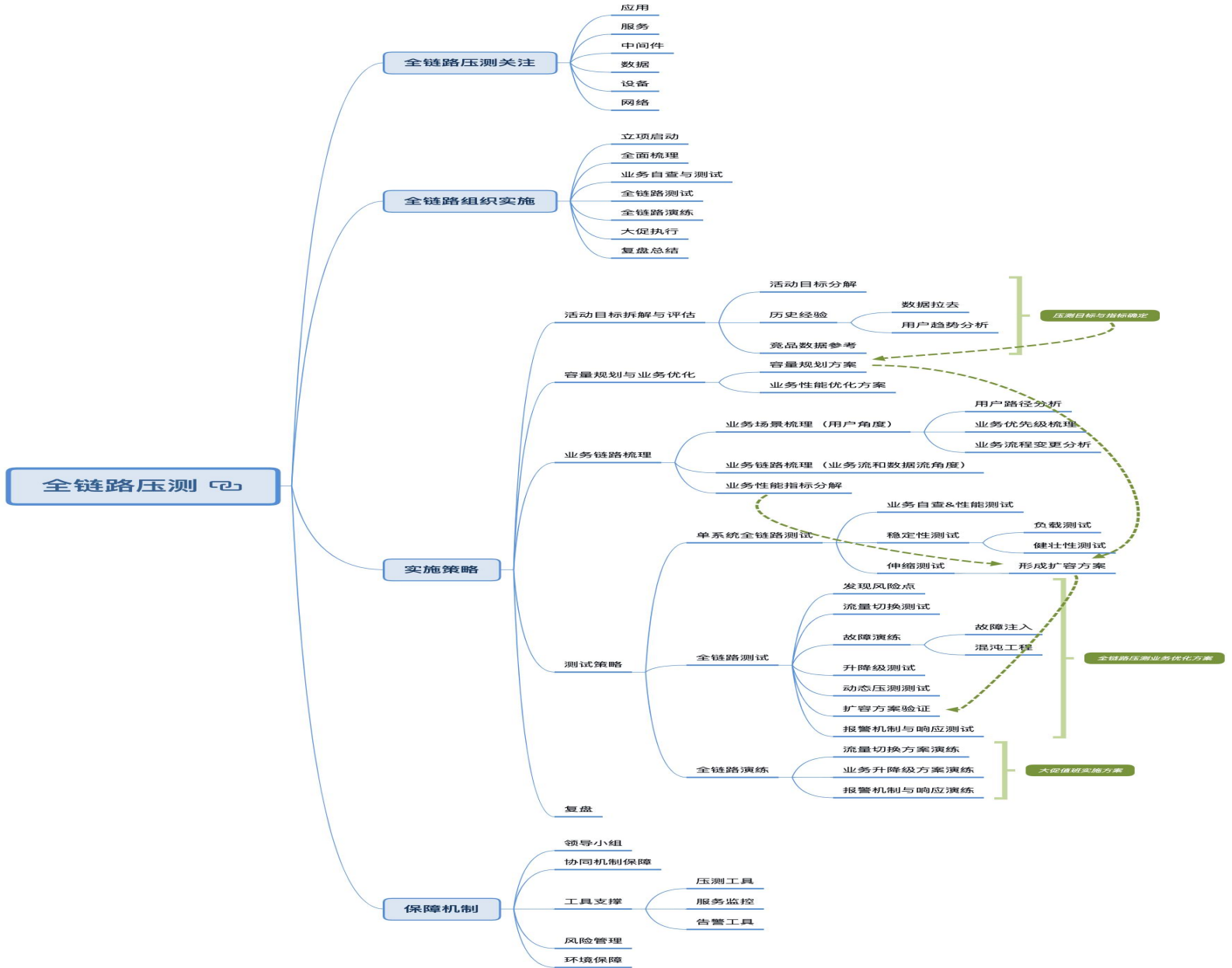
PART THRE

03

项目实践

...

项目管理系统实战-某电商618压测保障



感谢观看