

软件工程实践  
Rational 用户组



uml.org.cn  
UML 软件工程组织

# 性能测试过程和方法

 UML 软件工程组织

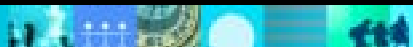
张斌



Evolve by case

# 第一章 性能测试一些基本概念

- 性能测试的定义
- 性能测试的分类
- 性能测试角色定义
- 性能测试的工作原理
- 测试管理流程

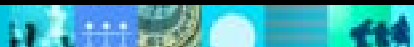


# 性能测试的定义

- 系统的性能是一个很大的概念，覆盖面非常广泛，对一个软件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
- 性能测试用来保证产品发布后系统的性能满足用户需求。性能测试在软件质量保证中起重要作用。

# 性能测试的分类

- 负载测试
- 压力测试
- 负载压力测试



# 负载测试

负载测试是确定在各种工作负载下系统的性能，目标是测试当负载逐渐增加时，系统组成部分的相应输出项，例如通过量、响应时间、CPU负载、内存使用等如何决定系统的性能，例如稳定性和响应等。

负载测试通常描述一种特定类型的压力测试，即增加用户数量以对应用程序进行压力测试。

# 压力测试

压力测试通过确定一个系统的瓶颈或者不能接收的性能点，来获得系统能提供的最大的服务级别的测试。通俗地讲，压力测试是为了发现在什么条件下您的应用程序的性能会变得不可接受。

# 负载压力测试

负载压力测试是性能测试的重要组成部分，负载压力测试包括：

- 并发性能测试（重点）
- 疲劳强度测试
- 大数据量测试

# 负载压力测试

- 并发性能测试

考察客户端应用的性能，测试的入口是客户端  
并发性能测试的过程，是一个负载测试和压力测试的过程。即逐渐增加并发虚拟用户数负载，直到系统的瓶颈或者不能接收的性能点，通过综合分析交易执行指标、资源监控指标等来确定系统并发性能的过程。并发性能测试是负载压力测试中的重要内容。



# 负载压力测试

## ■ 疲劳强度测试

通常是采用系统稳定运行情况下能够支持的最大并发用户数或者日常运行用户数，持续执行一段时间业务，通过综合分析交易执行指标和资源监控指标来确定系统处理最大工作量强度性能的过程。

疲劳强度测试案例制定的原则是保证系统长期不间断运行的业务量，并且应该尽量去满足该条件。

# 负载压力测试

## ■ 大数据量测试

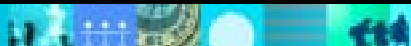
### ▶ 大数据量测试的两种类型

#### ▪ 独立的数据量测试

针对某些系统存储、传输、统计、查询等业务进行大数据量测试

#### ▪ 综合数据量测试

和压力性能测试、负载性能测试、并发性能测试、疲劳性能测试相结合的综合测试方案



# 性能测试角色定义和人员组成

- 测试角色
  - 需求分析人员
  - 案例设计人员
  - 测试执行人员
  - 测试分析人员

# 性能测试角色定义和人员组成

- 人员组成
  - 开发人员
  - 测试人员
  - 用户代表
  - 系统人员



# 性能测试工作原理

通过对现实各种操作模拟，去获取系统和应用各项的性能指标，去分析软件是否满足预期设定的结果，总结为如下6个字：

- 模拟
- 监控
- 分析

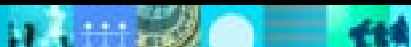
# 模拟

- 如何通过程序模拟现实中各种操作
  - 多线程程序去模拟应用操作关注点
  - 测试数据关联性的策略
  - 系统环境的模拟
  - 如何降低模拟程序所消耗的资源
  - ○ ○ ○ ○ ○ ○

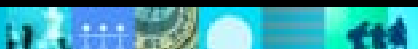
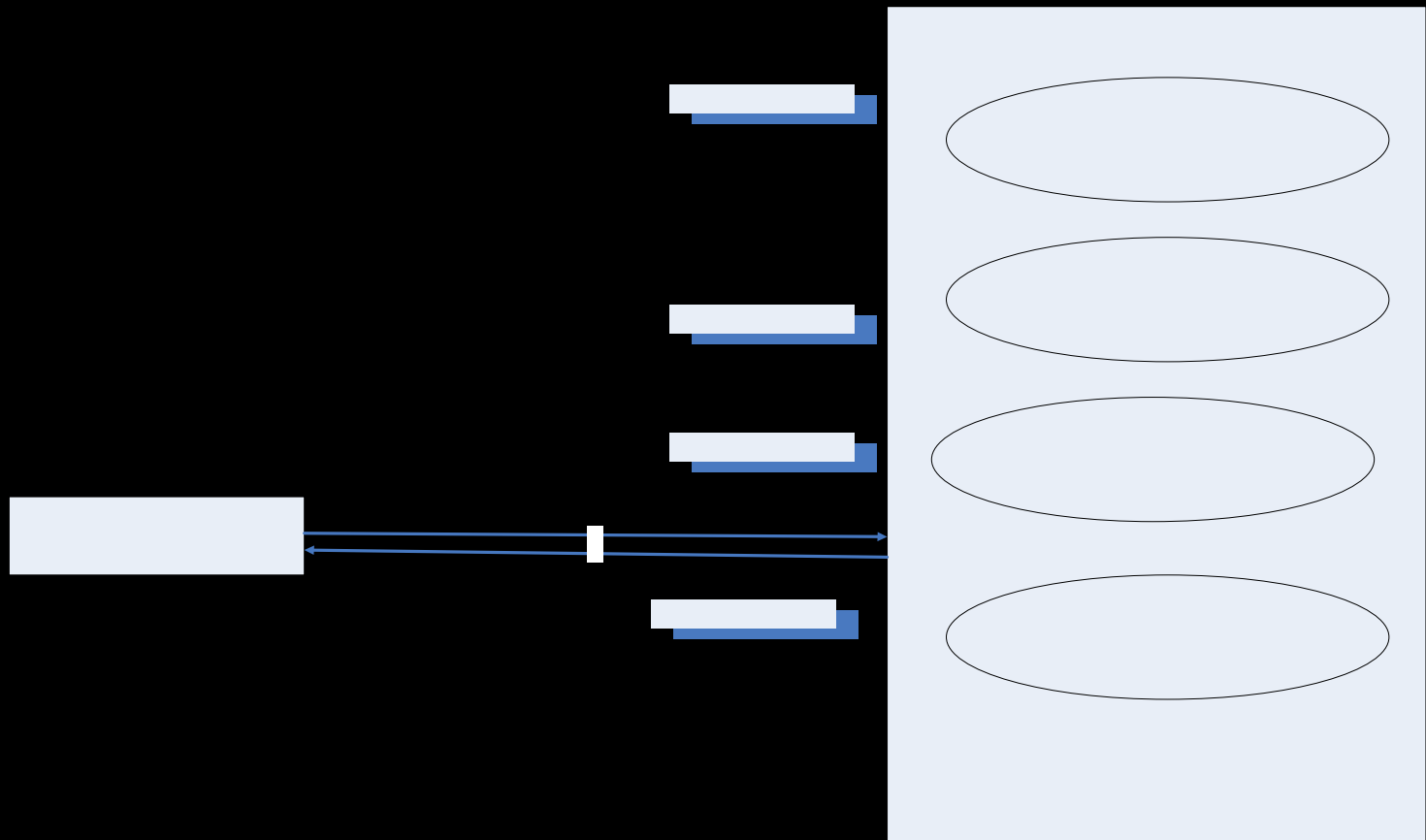


# 模拟

- 实现模拟的两类思路
  - 从某单一应用角度去设计
  - 从测试平台角度去设计

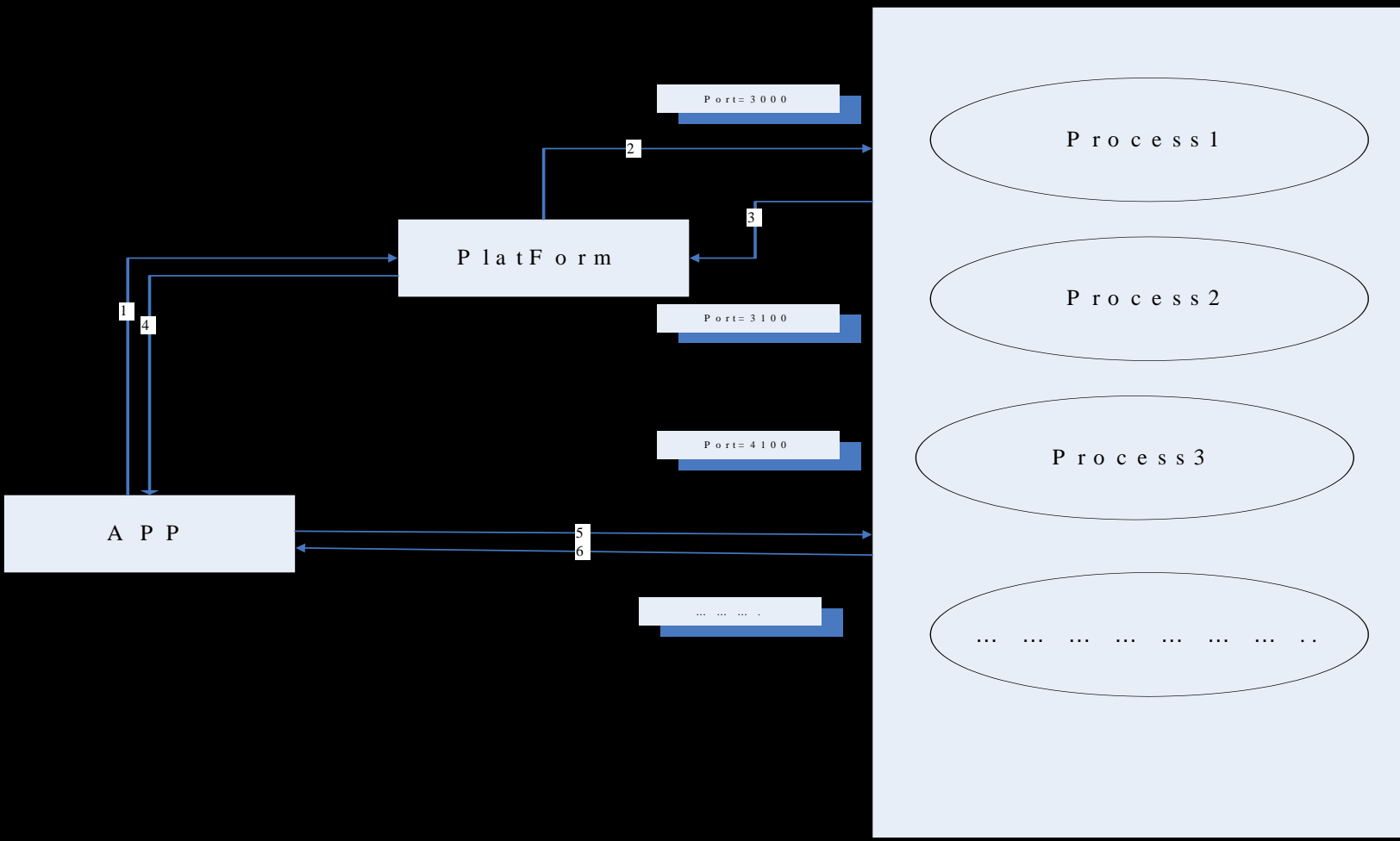


# 应用角度



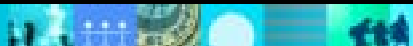


# 平台角度



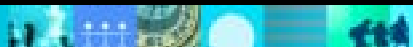
# 监控

- 应用性能指标的监控
- 系统性能指标的监控



# 监控

- 监控设计关注点
  - 如何实时获取到测试过程中所需各种参数
  - 如何实现在监控过程中对应用或者系统性能的负面影响



# 分析

- 如何通过一定的方法去组合各种监控参数
  - 数据的关联性
  - 利用已经有的各种数学模型（Matlab）等
  - ○ ○ ○ ○ ○ ○
- 如何通过各种分析模型快速的定位问题参数，从而降低分析难度

# 测试管理流程

- 测试管理要素

- 测试过程

- ▶ 技术过程、管理过程、支持过程

- 测试人员及组织

- 测试工作产品

- ▶ 测试计划、测试说明书、测试案例、测试报告、问题报告

# 测试过程

## ■ 测试过程

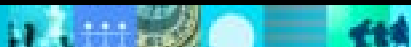
- ▶ 定义和定制所需要的测试过程
- ▶ 满足测试过程所需要的资源和条件
- ▶ 实施确定的测试过程
- ▶ 测量和分析测试过程的有效性和效率
- ▶ 进行基于度量的测试过程的持续改进

# 测试人员及组织

- 测试人员及组织
  - ▶ 选择合适的测试人员
  - ▶ 使测试人员能够按测试计划完成测试任务
  - ▶ 与相关方进行沟通，协同工作
  - ▶ 建立有效的软件测试团队

# 测试工作产品

- 测试工作产品
  - ▶ 检查和评审测试工作产品
  - ▶ 测量和分析测试对象 - - 开发的软件产品，收集质量分析和产品放行决策所需要的数据

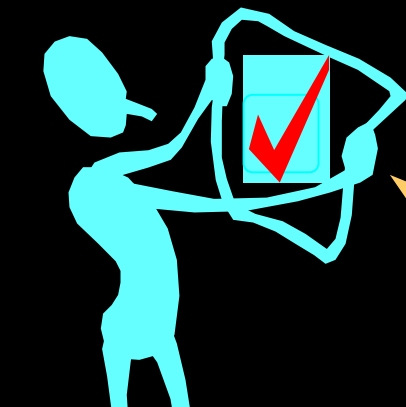
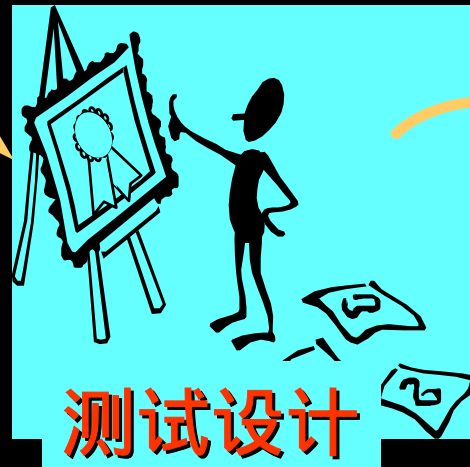




## 第二章 性能测试过程

- 测试生命周期
- 测试计划
- 测试设计
- 测试开发与测试执行
- 测试分析与评估

# 测试生命周期



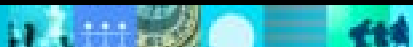
测试评估

# 测试计划 - 内容

- 目标
- 测试项（对象）
- 测试类型
- 测试范围
- 测试过程
- 资源需求（硬件、软件、人力）

# 测试计划 - 内容

- 进度安排
- 测试开始、结束准则
- 测试记录
- 测试的评估



# 测试计划 - 测试需求

目的:

挖掘出可能造成系统瓶颈的因素，并为后面案例设计的准确性提供保证。在此过程应关注如下几点：

- 应用配置需求
- 系统配置需求
- 用户使用情况需求

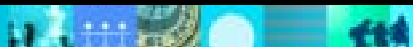
# 应用配置需求

- 应用系统整体框架
- 涉及到哪些第三方组件
- 应用层与数据库层的接口
- 应用数据库的配置



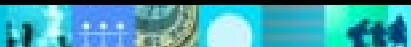
# 系统配置需求

- 用户客户端的配置
- 客户端与服务器端交互的网络配置
- 应用服务器/数据库服务器硬件配置
- ○ ○ ○ ○ ○ ○



# 用户使用操作需求

- 用户的分布情况
- 用户操作模块的频繁度
- 用户操作数据的特点
- ○ ○ ○ ○ ○ ○





## 案例说明：

- 全球客户关系系统



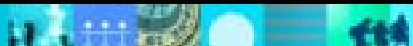
# 性能测试设计

设计出一些合理的场景去验证在需求分析阶段获得的可能影响性能因素是否是造成系统瓶颈的因素。

- 测试策略
- 测试案例
- 测试内容

# 测试策略

- 测试环境
- 测试数据
- 网络环境
- ○ ○ ○ ○ ○



# 测试案例

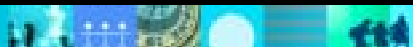
- 虚拟用户的操作步骤要尽量类似与真实用户的操作
- 操作的数据要类同于真实用户实际使用数据，防止应用缓存技术对负载测试带来的负面影响
- 在案例设计时要充分考虑到需求中用户对模块的使用频率

# 测试内容

- 标注案例的测试类型
- 定义出测试过程中各种需要监控的参数
- 列举出主要关注的性能参数指标

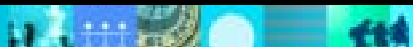


案例：



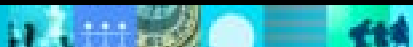
# 测试脚本开发与测试执行

- 测试脚本开发
  - 如何快速的开发出测试脚本
  - 动态数据的关联
  - ○ ○ ○ ○ ○



# 测试执行

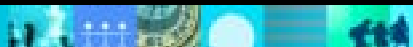
- 性能测试执行的一些“盲区”
- 20%、100%、120%用户的执行策略
- 性能测试执行的系统环境分配
- ○ ○ ○ ○ ○ ○





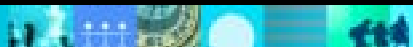
# 测试分析

根据系统监控的参数去分析系统是否存在性能瓶颈和系统瓶颈的定位以及分析造成系统瓶颈的原因



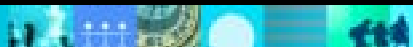
# 测试分析技巧

- 分析人员的组成
- 关注不同时间段的跳跃值
- 系统瓶颈的深度挖掘
- 排除法
- 模型验证法
- ○ ○ ○ ○ ○



# 案例：

- 全球客户关系系统



# UML软件工程组织 使命

“关心您的软件工程实践！”

---

[www.uml.org.cn](http://www.uml.org.cn)

