## 软件性能问题：

**轻则对外不能提供服务**

**重则系统崩溃**

**甚至数据丢失**

## 性能需求：

**计算性能：**

软件系统执行速度，执行一个典型的业务需要花多少时间。

**资源的利用和回收：**

硬件资源：客户端硬件、服务器硬件、网络硬件

软件资源：操作系统、中间件、数据库

服务器内存数量，对整个系统的性能

**启动时间：**

J2EE系统在第一次启动的时候一般会比较慢，因为期间涉及缓存的加载、JSP页面的编译、Java class编译成机器指令等

C/C++程序直接运行的是二进制机器代码，启动速度就要快一些

**伸缩性**

随着并发用户的增加，平均响应时间应逐渐稳定下来

**稳定性**

长时间运行

## 衡量一个软件系统性能的常见指标：

### 1.响应时间（Response time）

（1）服务器端响应时间

指服务器完成交易请求执行的时间，不包括客户端到服务器端的反应（请求和耗费在网络上的通信时间），可以度量服务器的处理能力

（2）网络响应时间

网络硬件传输交易请求和交易结果所耗费的时间

（3）客户端响应时间

客户端在构建请求和展现交易结果时所耗费的时间

对于胖客户端Web应用来说，比如Java applet、AJAX，由于客户端内嵌了大量的逻辑处理，耗费的时间有可能很长，从而成为系统的瓶颈。

客户感受的响应时间=客户端响应时间+服务器端响应时间+网络响应时间

### 2.吞吐量（Throughput）

吞进去的是请求，吐出来的是结果，就是系统的处理能力，指软件系统在每单位时间内能处理多少个事务/请求/单位数据等。

数据库吞吐量指单位时间内，不同sql语句的执行数量；

网络吞吐量指单位时间内在网络上传输的数据数量。

吞吐量的大小由负载（如用户的数量）或行为方式来决定

### 3.资源使用率（Resource utilization）

常见的资源有：CPU占用率、内存使用率、磁盘I/O、网络I/O

### 4.点击数（Hits per second）

指客户端向Web Server发起了多少次http请求计算的。而不是鼠标点击次数。

### 5.并发用户数（Concurrent users）

性能测试就是为了验证这些性能指标是否被满足

性能测试属于软件系统级测试，其最终目的是验证用户的性能需求是否达到。

性能测试还经常用来做：

识别系统瓶颈和产生瓶颈的原因

最优化和调整平台的配置（包括硬件和软件）来达到最高的性能

判断一个新的模块是否对整个系统的性能有影响

## 软件测试的规则：

确定预期输出

必须彻底检查每一个测试结果

穷举测试

## 选择性能测试案例时需要遵循的原则：

基本且常用的：用户使用频率较高的。

对响应时间要求苛刻的：比如，从手机呼叫开始，经过基站、核心网，再到被叫手机铃响，整个系统的处理时间应该在用户能接受的范围内

在性能测试开始之前，一般要搭建模拟生产环境，分析用户规模及使用场景等

性能测试开始的必要条件是软件系统已经处于一个比较稳定的状态，系统架构、主要代码、中间件等都不再有大的变化

## 数据表取数规则：

Sequential：顺序的

Random：随机的

Unique：唯一的（超出范围，LoadRunner会报错，提示数据不够用）

Same Line As<parameter>：与以前定义的参数取同一行。注意：至少其中的一个参数必须是Sequential、Random或Unique

Update value on：

Each iteration：

每次迭代就要取新值（在同一个迭代中，无论读几次参数，获得的都是同一个参数值）

Each occurrence：

只要取一次，就要新值（在同一个迭代中，读一次参数，就要取新值，新值从哪来，由Select next row来规定）

Once：

在所有的循环中都使用同一个值（只取一次，此参数只有一个值）

When out of values（选择数据为Unique时才会用到）

Abort Vuser：中止

Continue in a cyclic manner：继续循环取值

Continue with last value：取最后一个值

Allocate Vuser values in the Controller（选择数据为Unique时才会用到）

Automatically allocate block size：自动分配

Allocate()values for each Vuser：指定一个值

## VU编写脚本最佳实践流程：

录制→回放（单用户单循环）→参数化（单用户多循环）→多线程（多用户单循环）→多线程+参数化（多用户多循环）

## 软件开发-测试V型图

需求分析 系统测试（性能测试）

设计 集成测试

编码 单元测试

## GAME(A)性能测试过程模型：

### G：Goal，目标

**（确定本次测试的目标，选择测试设计策略）**

开始时间：需求获取阶段

输入：性能需求意向

输出：明确的性能测试目标和性能测试策略

常规的性能测试目标有以下几种：

1、度量最终用户响应时间：

查看用户执行业务流程以及从服务器得到响应所花费的时间

2、定义最优的硬件配置

检测各项系统配置（内存、cpu速度、缓存、适配器、调制解调器）对性能的影响

3、检查可靠性

确定系统在连续的高工作负载下的稳定性级别。强制系统在短时间内处理大量任务，以模拟系统在数周或数月的时间内通常会遇到的活动类型。

4、查看硬件或软件升级

执行回归测试，一遍对新旧版本的硬件或软件进行比较。此回归测试的目的不是验证升级版的新功能，而是查看新版本的效率和可靠性是否与旧版本相同

5、确定瓶颈

运行测试以确定系统的瓶颈，并确定哪些因素导致性能下降。如：文件锁定、资源争用、网络过载等。

6、度量系统容量

度量系统容量，并确定系统在不降低性能的前提下提供多少额外容量。确定响应时间曲线的“拐点”。

### A：Analysis，分析

**（分析性能需求，分析系统架构）**

开始时间：需求分析阶段和性能测试启动阶段

输入：性能需求

输出：达成一致的性能指标列表，性能测试案例文档

1、分析性能需求

定义性能测试的内容，细化性能需求

客户、需求分析人员和测试工程师一起起草一个性能需求标准，对此标准获得一致认同。此标准将用户的需求细化、量化，并能在测试中作为判断依据。

测试对象是什么，如：被测系统中有负载压力需求的功能点包括哪些？测试中需要模拟哪些部门用户产生的负载压力？等

系统配置如何，如：预计有多少用户并发访问？用户客户端的配置如何？使用什么样的数据库？服务器怎样和客户端通信？等

应用系统的使用模式是什么，如：使用在什么时间达到高峰期？用户使用系统是采用B/S运行模式吗？网络设备的吞吐能力如何，每节承受多少并发用户？等

得出的性能测试指标标准至少包含测试环境、业务规则、期望响应时间等。

2、分析系统架构

对硬件和软件组件、系统配置以及典型的使用模型有一个透彻的了解。

### M：Metrics，度量

**（场景的定义，事务的定义，虚拟用户）**

开始时间：性能测试设计阶段

输入：细化的性能指标和性能测试案例

输出：和工具相关的场景度量、交易度量、监控器度量和虚拟用户度量等

1、场景的定义，pass/fail的标准

测试场景包含性能测试的宏观信息，有测试环境、运行规则和监控数据等。具体可表现为历史数据记录数、虚拟用户数、虚拟用户加载方式、监控指标等。

2、事务(Transaction)的定义，pass/fail的标准

事务用来度量服务器的处理能力。

3、虚拟用户pass/fail的标准

### E：Execution，执行

**（环境、数据、脚本的准备，场景及监控器的运行，生成报告）**

开始时间：软件测试执行阶段

输入：场景、交易、虚拟用户等设置信息

输出：测试报告

执行测试包含两个工作：

1、准备测试环境、数据和脚本：

测试环境：硬件平台和软件平台

测试数据：包括初始测试数据和测试用例数据两部分。表现为SQL脚本、Excel文件等

测试脚本：用测试工具生成脚本

2、运行场景和监控性能

### (A)：Adjust，调整。

**E执行失败后才进入A阶段，并且涉及的大多是有关开发和系统管理的工作，因此A设为隐式。（应用程序的修改，中间件的调优）**

开始时间：第一轮性能测试结束后，而且没有通过的条件下

输入：测试报告和测试结果数据

输出：性能问题解决方案

调整包含两个意思：应用程序修改和中间件调优

中间件调优可考虑如下因素操作系统调优：

数据库调优、内存升级、cpu数量、代码调优、Cache调优

### GAME(A)模型有两个优势：

1、灵活，每个过程都有自己的关注点，可以根据不同的项目特点增加或删除关注点

2、通用，不依赖于具体的工具

## web系统性能测试整个流程：（GAME(A)模型）

### 了解软件系统背景及架构设计

### 确定性能指标

（用户对本系统要求的性能指标）

总用户数

使用高峰期用户并发量

使用高峰期时间段

各功能最大响应时间

系统计划使用时间，决定数据库中是否要有一定数量的历史数据

### 制定性能测试方案和设计用例

目的：最大程度地模拟真实业务场景，来验证系统的性能指标，并发现可能存在的性能瓶颈

确定业务流程/场景

编写测试用例

准备测试数据

### 录制/编写性能脚本

录制→回放（单用户单循环）→参数化（单用户多循环）→多线程（多用户单循环）→多线程+参数化（多用户多循环）

### 正式运行，获得测试结果和相关监控数据

### 分析测试结果

### 性能调优（若出现性能问题）

### 回归测试

## 做性能测试前需要知道的：

1、测试目的是什么

2、实际的应用服务器、数据库服务器软硬件配置环境

3、实际总用户量是多少，高峰期最大并发用户数以及并发时间段

4、需要测试哪些功能，各功能对数据量的要求，最好可以提供测试用例

5、最好可以提供一个实际库，提供程序