# 性能测试

## 一、相关概念

性能测试方法是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。通俗地说，这种方法就是要在特定的运行条件下验证系统的处理能力。

### 基准测试

准备性测试，帮助验证性能测试脚本能够在性能环境中正确执行。通常使用一个虚拟用户执行单个用例，获得最佳的用例性能效果，然后将其和后续的测试数据做对比。

### 负载测试

这种方法是对一个系统持续不段的加压，看你在什么时候已经超出“预期要求”或系统崩溃。

最典型的性能测试类型。施加足够的负载来达到预期的并发压力，达到目标后不再增加。

目的：验证可用性，并发数，吞吐率和响应时间的性能指标。

### 压力测试

压力测试方法测试系统在一定饱和状态下，例如CPU、内存在饱和使用情况下，系统能够处理的会话能力，以及系统是否稳定会出现错误。尝试探测应用或者应用的支撑基础设施某些部分的极限能力。

目的：找到被测系统的容量天花板。

会一直增加负载，直到应用的部分功能不能正常工作。

### 并发测试

并发测试通过模拟用户并发访问，测试多用户并发访问同一个应用、同一个模块或者数据记录时是否存在死锁或其者他性能问题。

### 可靠性测试

目的：为了发现那些只有长时间运行才能发现的问题。典型的例子包括内存泄漏或者某些用例没有预见地执行次数的上限。

必须有合适的监控系统。

发现问题常常表现为响应时间缓慢增加，或某时刻系统突然变得不可用。

需要将问题发生时刻的负载数据和监控数据关联后进行分析。

### 性能冒烟测试

性能冒烟测试：只测试那些被代码变更影响到的性能测试用例。

### 隔离测试

目的：定位和排查问题。

通常会对已经定位产生性能问题的用例多次重复执行。

## 二、性能指标

可用性。（可用时长）

响应时间。（用户发起请求到应用响应完全到达用户客户端所消耗的时间）

吞吐率。（某些面向应用的事件的发生速率）

资源利用率。（对某种资源理论容量的使用百分比）

### 思考时间

思考时间指的是用户在与应用进行交互时必然存在的延迟和暂停时间。

### 响应时间

从用户发起请求到应用响应完全到达用户客户端所消耗的时间。整个过程分三个部分：**呈现时间，数据传输时间和系统处理时间。**

一般测试工具都屏蔽响应的呈现过程，只是模拟多用户并发请求，计算用户得到响应的时间，不会将服务器的每个响应做客户端渲染呈现。

对于数据传输的问题，这也是我要强调的性能测试要在局域网中进行，在局域网中一般不会受到数据带宽的限制。所以，可以对数据的传输时间忽略不计。

### 吞吐率

单位时间内网络上传输的数据量，也可以指单位时间内处理客户请求数量。它是衡量网络性能的重要指标，通常情况下，吞吐率用“字节数/秒”来衡量，当然，你可以用“请求数/秒”和“页面数/秒”来衡量。

### TPS (Transaction Per second)

每秒钟系统能够处理事务或交易的数量，它是衡量系统处理能力的重要指标。

**点击率（Hit Per Second）**

点击率可以看做是TPS的一种特定情况。点击率更能体现用户端对服务器的压力。TPS更能体现服务器对客户请求的处理能力

需要注意的是，这里的点击不能简单的看作鼠标的一次“单击”操作，也许一次“单击”操作中，客户端可能向服务器发现多个HTTP请求。

### 并发用户数

虚拟并发用户数：从性能测试工具角度来看的活跃虚拟用户数。

应用并发用户数：活跃的虚拟用户，活跃指的是已经登录或者正在访问被测系统的虚拟用户。

虚拟并发用户数=应用并发用户数+退出用户数

### 服务器资源利用率

CPU利用率

内存使用率

I/O（磁盘和网络）

### RBI（rapid bottleneck identify）

是Empirix公司提出的快速识别系统性能瓶颈的方法。该方法基于以下事实。

    1. 发现的80%系统的性能瓶颈都由吞吐量制约；

    2. 并发用户数和吞吐量瓶颈之间存在一定的关联；

    3. 采用吞吐量测试可以更快速定位问题。

通过不断增加并发用户数和吞吐量观察系统的性能瓶颈。然后，从网络、数据库、应用服务器和代码本身4个环节确定系统的的性能瓶颈。

## 三、难点

性能测试的难点？

* 需求分析
* 场景设计
* 性能诊断调优
* 环境搭建和模拟

### 如何衡量可用性

可用性通常采用应用对于终端用户的可用时长来衡量。

## 四、性能测试流程

1. 获取非功能需求，确定性能目标
2. 构建性能测试环境
3. 编写性能测试脚本
4. 构建性能测试场景
5. 执行性能测试和分析
6. 测试结果分析和报告