结合最近听到的一堂培训课，整理性能测试的定义，前后端性能测试的常见指标，以及通过什么方法和工具来测试。

性能测试的定义

转：观察系统在一个给定的环境和场景中的性能表现是否与预期目标一致，评判系统是否存在性能缺陷，并根据测试结果识别性能瓶颈，改善系统性能的完整的过程

性能测试的分类

基准测试：单用户，发单次请求，产出基准性能数据

负载测试：多用户，用户数渐增，持续同时发同一业务请求，产出最大TPS

压力测试：多用户，资源使用饱和，持续同时发同一业务请求，产出系统瓶颈或使用极限

混合场景测试：多用户，资源使用不饱和，持续同时发不同业务请求，验证系统稳定性

性能测试的指标

响应时间：用户从客户端发出请求，并得到响应，以及展示出来的整个过程的时间。

加载速度：通俗的理解为页面内容显示的快慢。

内存占用：APP的内存开销。

电量：APP的耗电量。

流量：APP所消耗的流量

并发用户数：同一时间点请求服务器的用户数。

吞吐量（TPS）：Transaction Per Second, 每秒事务数。在没有遇到性能瓶颈时：TPS=并发用户数\*事务数/响应时间。

错误率：失败的事务数/事务总数。

资源使用率：CPU占用率、内存使用率、磁盘I/O、网络I/O。

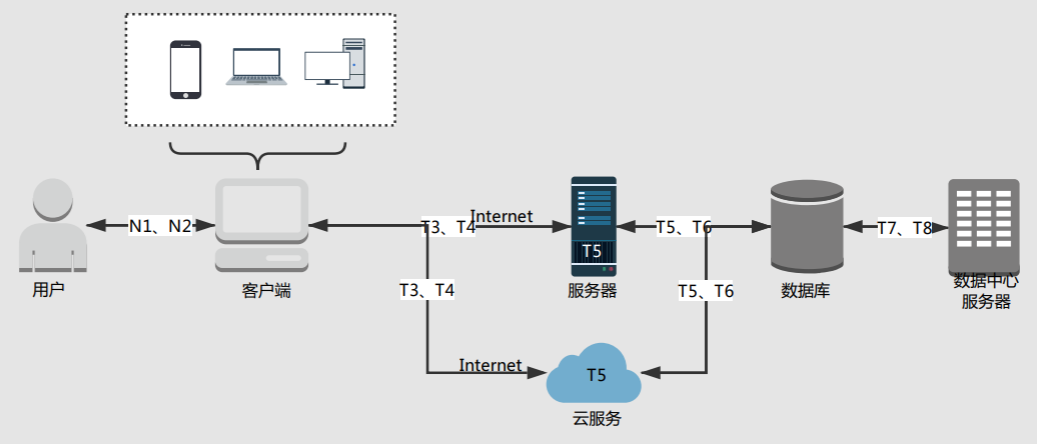
最佳并发用户数：用户无等待时间，资源无浪费时的并发数。

最大并发用户数：被所有用户接受的最大响应时间时的并发数。

响应时间

指的是客户发出请求到得到响应的整个过程的时间。在某些工具中，请求响应时间通常会被称为TTLB（Time to laster byte），意思是从发起一个请求开始，到客户端收到最后一个字节的响应所耗费的时间。所以也可以理解成，响应时间＝网络响应时间+应用程序响应时间。

因此在大部分公司的项目实际运作中，会把性能测试分为两部分，APP 前端的响应时间、后端接口请求和返回的时间，即分别是系统级性能测试和接口级性能测试。



网络传输时间：T3+T4+T5+T6

应用服务器处理时间：T5+T7+T8

数据库服务器处理时间：T7+T8

响应时间=N1+N2+T3+T4+T5+T6+T7+T8

那么什么是合理的响应时间呢？

互联网上对于用户响应时间，有一个普遍的标准，2－5－10原则。

详细来说，就是：

2秒之内得到响应，会认为系统响应的很快。

5秒之内得到响应，会认为系统响应的速度还不错。

10秒之内得到响应，会认为系统响应的速度很糟糕。

超过10秒还未得到响应，会认为系统是没有响应的。。

加载速度

通俗的理解，可以将加载速度视为页面内容显示的快慢。拿Google搜索的例子来说，从用户输入搜索内容按下enter键，到看到搜索出来的内容，这个过程的快慢就是加载速度。假设选中一个内容点击，跳转到一个网页，网页的内容显示出来能让用户看见的过程，也是加载速度。

早些年Amazon曾经做过一个统计：网页加载时间每延长1秒钟，一年将减少16亿美元的营收。

一般有哪些方式可以改善加载速度带来的用户体验呢？

l 减少HTTP重复请求

性能黄金法则：只有10%~20%的最终用户响应时间花在了下载HTML文档上，其余的80%~90%时间花在了下载页面中的所有组件上。因此，改善响应时间最简单的途径就是减少HTTP请求的数量，并且去除不必要的重复请求。

l 使用CDN

HTTP请求和响应的时间会受到离 web 服务器距离的影响。如果用户离应用程序的web服务器离用户更近，那么多个HTTP请求的响应时间将缩短。

CDN（内容发布网络）是一组分布在多个不同地理位置的Web服务器，可以选择网络阶跃数最小的服务器，或者具有最短响应时间的服务器，用于更加有效地向用户发布内容。

l 减少下载的资源

比如通过压缩图片的方式，减少图片的大小，缩短下载的时间。另外可以通过比对客户端与服务端差异的方式，快速展示本地的缓存资源，减少同样内容的重复下载。

CPU

在Linux系统下，CPU利用率分为用户态、系统态、空闲态，分别表示CPU处于用户态执行的时间，系统内核执行的时间，和空闲系统进程执行的时间。平时所说的CPU利用率是指：CPU执行非系统空闲进程的时间 / CPU总的执行时间。

CPU可能出现的问题是，持续CPU占用较高、设备发热、使用非常卡顿、程序卡死。

什么情况下会消耗CPU 呢？

就是大量的运算。比如某个Activity或者方法有一直不停的运算消耗CPU（比如：不停止的while 或者for 循环）

一般从哪些指标监控CPU情况呢？

l 设备的应用在空闲时间，CPU的占用情况

l 应用使用时，CPU的占用走势，持续变化

l CPU的占用峰值

内存占用

Android系统中，每个APP进程除了同其他进程共享(shared dirty)外，还独用私有内存(private dirty)，通常使用PSS(=私有内存+比例分配共享内存)来衡量一个APP的内存开销。

移动设备的内存资源有限，因此为每个APP进程分配的私有内存也是有限制的。APP 的内存常见问题有内存占用过高、内存泄露，以及内存溢出。

内存泄漏：程序在向系统申请内存分配后，使用后未释放。

内存溢出：程序向系统申请的内存空间超出了系统本身的内存，会出现崩溃，也就是客户端的carsh。

电量

Android的很多特性都比较耗电(屏幕、GPS、唤醒机制、CPU、连网等的使用)。

流量

目前的网络类型包含2G\3G\4G\wifi，其中还有不同运营商的区分。APP 使用过程中，常见的网络流量严重消耗的原因主要有，，调用响应慢，调用失败等各种情况。

通常从哪些指标去衡量流量消耗的状态是否正常呢？

l 应用首次启动流量提示；

l 应用处于后台，连续运行2小时的静默流量；

l 应用处于前台，高负荷运行时的流量峰值。

一般有哪些原因导致流量被大量消耗呢？

l 资源太多

l 图片太大

l 重复请求

l 日志上传

l 埋点数据

Crash和ANR

Crash的原因一般有：空指针、内存泄漏、数组越界、调用了高版本的API。

Android应用程序，如果主线程(即UI线程)在超时间内对用户输入时间没有处理完毕，就会出现Application Note Responding弹出框，用户需要选择等待或者强制关闭来杀死进程。

FPS

FPS，就是动画帧率。帧就是指动画或视频的“画面”，1幅画就叫做“1帧”，帧数就是在1秒钟时间里传输的图片的量，也可以理解为图形处理器每秒钟能够刷新几次，通常用FPS（Frames Per Second）表示。

每一帧都是静止的图象，快速连续地显示帧便形成了运动的假象，高的帧率可以得到更流畅和逼真的动画，因此每秒钟帧数 (FPS) 越多，显示出来的动作就越流畅。

那么什么是合理的FPS呢？

帧率达到60FPS以上，人眼主观就感受不到差别了。所以一般以60FPS作为衡量标准，即要求每一帧刷新的时间小于16ms，这样才能保证滑动中平滑的流畅度。

————————————————

版权声明：本文为CSDN博主「alice\_tl」的原创文章，遵循 CC 4.0 BY-SA 版权协议，转载请附上原文出处链接及本声明。

原文链接：https://blog.csdn.net/alice\_tl/article/details/79967932