非功能性测试

张程
Email：bootan@cqu.edu.cn
QQ:80463125

认识性能测试

为什么要进行性能测试

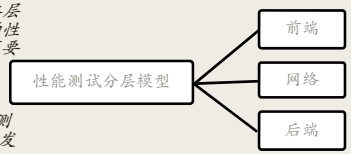
- 境内公众启动第二阶段奥运会门票预售。然而，为了让更多的公众实现奥运梦想的“先到先得，售完为止”的销售政策适得其反，公众纷纷抢在第一时间订票，致使票务官网压力激增，承受了超过自身设计容量8倍的流量，导致系统瘫痪。
- 上午9点，预售一开始，公众提交申请空前踊跃。北京奥运会官方票务网站的浏览量在第一个小时内达到800万次，每秒从网上提交的门票申请超过20万张；票务呼叫中心热线的呼入量超过了380万人次。由于瞬间访问数量过大，技术系统应对不畅，造成很多申购者无法及时提交申请。
- 北京奥运会官方票务网站的浏览量在第一个小时达到800万次，每秒从网上提交的门票申请超过20万张。
- 当时的新闻解释：
  - *从昨天上午8点左右开始，就有不少网民登录票务官网排队等待申购门票。据了解，从上午9点正式开始售票到中午12点的3小时内，票务网站的浏览次数达到2000万次。这与此次所提供的100万次/小时的流量相差甚远。*
  - *不停地刷新网页，也是造成网络拥堵的原因之一。杨力说：“不少网民在无法正常登录后便不断刷新，这就相当于一名申购者变成了若干名申购者，无形中增大了网站流量。从技术角度上讲，网站的流量几乎成几何倍数增长，导致其他申购者无法登录。”*

性能测试定义

- 性能测试工程师的主要目标就是确保系统能够在一定的硬件、软件环境下达到一定的性能指标。
- 性能测试的定义为：
  - *在一定的负载情况下，系统的响应时间等特性是否满足特定的性能需求。*
- 什么是负载?
  - *对于基于网络架构（如C/S架构或者B/S架构）的系统，当众多终端用户对系统进行访问时，用户越多，那么服务器需要处理的客户请求也越多，从而形成负载。*

性能测试分层模型

- 前端层
  - *前端层主要是指用户看到的页面,比如电商网站的首页、移动APP的各个页面,这些页面都是用户最关心的*
- 网络层
  - *任何系统都可以粗略地分成客户端、网络和服务器端,其中网络是连接前后端的命脉,网络质量的好坏对系统性能也有很大的影响。在性能测试中可能遇到的情况大致可分为两种,一种是测试服务器在不同网络状况的大流量下的表现(一般接触得比较少);另一种则要求将压力机和服务器最好放在同一网段,不然压力无法完整地到达后端,系统性能可能会在网络层就被拖垮,这样就没法较为准确地评测服务器端的性能情况了。如果测试的是移动APP端,那么可能还要考虑在不同网络状态下的测试。*
- 后端层
  - *不论是Web端还是移动APP端,在后端层实施性能测试的方法都是类似的,都是模拟大量客户端请求发送给后端层,同时监控后端服务器的处理能力。*



性能指标

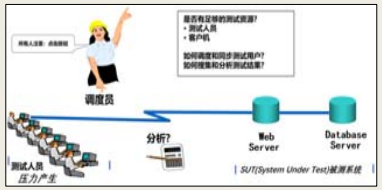
响应时间	• 响应时间反映完成某个业务所需要的时间。 • 在性能测试中通过事务函数来实现对响应时间的统计
吞吐量	• 吞吐量反映单位时间内能够处理的事务数目 • 在性能测试工具中,吞吐量也称为TPS (Transaction Per Second,每秒事务数),即在单位时间内能完成的事务数量
服务器资源占用	• 服务器资源占用反映了负载下系统资源的利用率 • 服务器资源的占用率越低,说明系统越优秀

性能测试的流程

LOADRUNNER

LoadRunner介绍

- 准备足够的资源：50名测试人员,每人有一台计算机以进行操作支持。
- 备一名“嗓音足够大”的指挥人员统一发布号令，以调度测试人员对系统进行同步测试，每位参与测试的人员需要注意力集中,在听到指挥员“开始测试”的号令后进行“理论上的同时”操作(每个人反应速度不好控制,所以称为理论上的“同时”)。
- 在50个测试人员“同时”执行操作后,对每台计算机上的测试数据和服务器中的测试数据进行搜集和整理。
- 在缺陷被修复后,要开展回归测试(在软件发生修改之后重新执行先前的测试，以保证修改的正确性),即需要再执行1~4步直到满足性能需求为止。



LoadRunner介绍

- 性能测试工具都要解决下面的共性问题：
  - *通过协议模拟用户行为。*
  - *模拟大量用户。*
  - *具备数据采集与整理分析的能力。*
  - *以LoadRunner工具为例，LoadRunner工具通过三大组件来解决上述问题：*
  - *虚拟用户脚本生成器（Virtual User Generator）：通过录制、编辑测试脚本来模拟用户行为；*
  - *压力调度控制台（Controller）：创建场景、运行场景、监控场景、收集测试数据；*
  - *压力结果分析器(Analysis)：测试结果分析。*

LoadRunner脚本示例

- 标准的电子商城,拥有Web端和WAP端(标准的H5),也就是说既可以通过PC机浏览器来访问电子商城,也可以通过手机端来访问电子商城。
- 电子商城实训项目拥有大部分商城应有的功能,比如
  - *注册*
  - *登录*
  - *搜索*
  - *下单*
  - *支付*
  - *与第三方支付系统对接*
  - ……

- 需求分析

- 脚本开发
- 1. 登录脚本
  - *登录业务是人们最为熟悉的业务,一般的登录界面在正确输入用户名和密码后就可以登录，稍微安全点的登录功能还会要求输入验证码。那么第一个问题来了,对于有验证码的登录功能的测试该怎么处理? 一般的处理方法有下面几种:*
  - *利用各种先进技术去识别。比如利用OCR技术来识别验证码。但现在的验证码都比较复杂，干扰因子很多并不好识别，所以没特殊需求可以放弃这种方法。*
  - *对于做性能测试而言验证码的影响其实并不大,可以直接找开发人员协助屏蔽掉系统中的验证码即可。*
  - *如果被测系统已经上线,那么直接屏蔽验证码对系统的影响就比较大了,这时可以设计一个"万能验证码"，也就是在系统后端设计一个确定的一串字符,只要用户输入这串字符,不论现在的验证码是什么,系统都认为是正确的。这样就可以比较妥善地解决验证码的问题。*

- 脚本开发
- 2. 浏览单品页脚本
  - *浏览单品页业务其实就是访问一个商品的详情页,浏览器模拟用户向服务器发送一个GET请求，一般软件会通过一个类似ID的参数来区分不同的商品页。*
- 3.搜索脚本
  - *在商城购物时经常会搜索商品，搜索时经常会用到汉字作为关键词进行搜索，使用LoadRunner录制搜索业务时因为汉字关键词的出现可能会在录制生成的脚本中产生乱码。*

解决这个问题，关键是要把本地的 GBK 编码的汉字转换成 UTF-8 编码格式的信息，可以尝试使用 LoadRunner 中自带的lr_convert_string_encoding 函数对中文进行 UTF-8 转码, 该函数具体语法如下：

```
int lr_convert_string_encoding(const char *sourceString, const char *fromEncoding, const char *toEncoding, const char *paramName)
```

该函数有 4 个参数，含义如下：

- sourceString：被转换的源字符串。
- fromEncoding：转换前的字符编码。
- toEncoding：要转换成为的字符编码。
- paramName：转换后的目标字符串。

- 脚本开发
- 4.下单支付脚本
  - *所谓的下单支付就是读者熟知的购买和付款，这个业务可能会出现这样一个问题：下单是在电子商城进行，而支付却需要用到其他机构的支付系统。这个问题可以归结为这类问题：需要测试的系统A与系统B有交互,而系统B不在测试人员的控制范围内,导致测试无法进行。*
  - *碰到这样的情况怎么办?*
  - *一般的解决方法是利用mock技术（mock测试技术就是在测试过程中，对于某些不容易构造或者不容易获取的对象，用一个虚拟的对象来创建以便测试的测试方法），通俗点解释就是构建一个虚拟的Service来自动返回所需要的响应。*

```
lr_start_transaction("加入购物车");
web_custom_request("flow.php",
    "URL=http://127.0.0.1/shop/flow.php?step=add_to_cart",
    "Method=POST",
    "Resource=0",
    "RecContentType=text/html",
    "Referer=http://127.0.0.1/shop/goods.php?id={goods_id}",
    "Mode=HTML",
    "Body=goods=[{\"quick\":1,\"spec\":[ ],\"goods_id\":{goods_id},\"number\":\"1\",\"parent\":0
}]",
    LAST);
lr_end_transaction("加入购物车",LR_AUTO);
```

使用LoadRunner完成H5网站的脚本开发

- H5的优势至少有下面几点:
  - *逐步推动标准的统一化。*
  - *多设备跨平台。*
  - *自适应网页设计。*
  - *即时更新。*
  - *对于SEO（Search Engine Optimization，搜索引擎优化）很友好。*
  - *大量应用于移动应用程序和游戏。*
  - *提高可用性和改进用户的友好体验。*
- 使用H5网站的URL就可以在LoadRunner中完成脚本录制。

场景设计精要

- 有两种常见的创建场景的方式：
- （1）单场景
  - *单场景仅对某个业务或某个接口进行单点的测试，主要是为发现单点可能存在的性能问题。类似于“水桶原理”中只要提高最短的那个板就可提升整桶的装水能力，实际中可能会出现只要提高某个单点的性能就能提高系统整体性能的情况。*
- （2）混合场景
  - *实际用户使用场景时，在同一时间用户可能会进行不同的操作，因此测试时也必须模拟这种情况，即模拟多业务的混合场景。*

性能测试分析思路-观察现象

- 对于现象观察的准确度会直接影响后续的推理分析，只有现象抓准了才能事半功倍
- 在性能测试中一般通过监控系统、Log日志或者命令进行现象的监控。这里的现象主要是指页面的表现、服务器的资源表现、各类中间件的健康度、Log日志、各类软件的参数、各类数据库的健康度等。
- 互联网公司中一般常用的监控方式有如下几种：
  - *综合监控系统，比如，Zabbix、Nagios、Open-falcon等。*
  - *专项监控系统，比如，专门监控数据库的MySQLMTOP,Spotlight等。*
  - *命令监控，比如，Linux命令、Shell脚本等。*
  - *软件自带的Console 监控台。*
  - *自主研发的监控系统。*
  - *云监控平台，比如，听云APM、OneAPM等。*

观察现象 → 层层递进 → 缩小范围

性能测试分析思路-层层递进

性能测试分析思路-缩小范围

- 一定范围内可分析的点基本都是固定的，可以在分析问题时适当缩小分析范围从而简化问题的复杂度，提高分析的逻辑性。以Tomcat相关容器为例，包括但不限于以下需要分析的点：

1. Tomcat参数的配置。比如，运行模式、Max Threads等参数。
2. Tomcat部署方式。单点或集群负载。
3. 服务器。Tomcat部署所在的服务器是否存在瓶颈，比如，内存太小等。
4. JVM。各个内存代的分配，GC垃圾回收机制等。
5. 代码。不合理的逻辑代码或未被释放的对象引用等。

移动APP非功能测试

移动APP启动时间测试

- 用户体验角度的APP启动时间
- 启动时间对于一款APP来说是一个比较重要的指标，用户都不愿意花时间来等待一款APP慢吞吞的启动。
- 下面将以如何获取用户体验角度的启动时间为例进行讲解。一般地，启动时间的测试需要考虑以下两种场景：
  - （1）冷启动。手机系统中没有该APP的进程，即首次启动该APP。
  - （2）热启动。手机系统中有该APP的进程，即APP从后台切换到前台。

- 常见的APP启动时间测试方法包括但不限于如下几种：
  - 通过adb命令，比如，adb logcat、adb shell am start、adb shell screenrecord等。
  - 代码里打点。
  - 高速相机。
  - 秒表，看到这个一定会有读者朋友偷偷笑，但事实上确实有时候只能这样做，就连某些巨头互联网公司的一些测试团队也是通过这种方式来做的。

使用adb获得APP启动时间示例

- 1. adb简介
  - adb工具即Android Debug Bridge（安卓调试桥）tools。它就是一个命令行窗口，用于通过电脑端与模拟器或者真实设备交互。Android软件测试开发工作者常用adb工具来安装卸载软件、管理安卓系统软件、启动测试、抓取操作日志等。
- 2.adb shell screenrecord命令
  - 下面使用Android4.4(API level 19)以上版本的系统中提供的adb shell screenrecord的命令，通过录制并分析视频来得到启动时间。
  - 命令格式：adb shell screenrecord [options]<filename>
  - 命令示例：adb shell screenrecord /sdcard/demo.mp4
  - 命令解释：使用screenrecord进行屏幕录制，录制结果存放到手机SD卡中，视频格式为mp4，默认录制时间为180s，之后对保存好的视频进行分析。
  - 可在下面的网站中查看更多adb命令的用法：
    - http://adbshell.com/commands/adb-shell-screenrecord。

使用adb获得APP启动时间示例

- 3.实现步骤
  - *把待测手机连上电脑，进入cmd命令窗口，输入录制命令开始录制。*
  - *待APP完全启动后，按下<Ctrl>+C结束视频录制。*
  - *使用adb pull/sdcard/demo.mp4 d:\record命令，导出视频到D盘的record文件夹。*
  - *使用按帧播放的视频软件打开该视频并进行播放分析（比如，KMPlayer）。*
  - *当在视频中看到ICON变亮时可以作为开始时间，等待APP完全启动后的时间作为终止时间，后者减去前者就是用户体验角度的APP启动时间了。*
  - *但是这个测试方法也有一些限制，大致有如下几个：*
  - *某些设备中可能无法录制；*
  - *在录制过程中不支持转屏；*
  - *声音不会被录制下来；*
  - *如果手机中有其他APP在运行，则会对启动时间产生一定的干扰。*

移动APP流量测试

- 流量是指连网设备在网络上所产生的数据流量，流量是一个数字记录。这里所讲的流量主要是关注用户层面的流量。
- 一般来说APP流量的测试需要考虑以下两种场景：
  - *（1）活动状态，即用户直接操作APP而导致的流量消耗。*
  - *（2）静默状态，即用户没有操作APP，APP处于后台状态时的流量消耗。*
- Android系统下流量的测试方法包括但不限于如下几种：
  - *（1）通过Tcpdump抓包，然后利用Wireshark分析。*
  - *（2）查看Linux流量统计文件。*
  - *（3）利用类似DDMS的工具查看流量。这种方式非常方便，容易上手，数据直观。*
  - *（4）利用Android API进行统计。通过Android API的TrafficStats类来统计，该类提供了很多不同方法来获取不同角度的流量数据。*

APP流量测试示例

- 以大部分公司的测试工程师经常使用的方法——查看Linux流量统计文件为例进行讲解。
- 以test.apk这个APP为例，统计其消耗流量的步骤如下：
  - *（1）通过ps|grep com.android.test命令获取pid；*
  - *（2）通过cat /proc/{pid}/status命令获取uid，其中{pid}替换为上一步获取的pid值；*
  - *（3）通过cat /proc/uid_stat/{uid}/tcp_snd命令获取发送的流量(单位：byte)，其中{uid}替换为上一步获取的uid值；*
  - *（4）通过cat /proc/uid_stat/{uid}/tcp_rcv命令获取接收的流量(单位：byte)，其中{uid}替换为第二步获取的uid值。*
- 通过上面的步骤可以大致知道test.apk应用消耗的流量了。这里需要注意的是该方法有一个弊端，那就是其统计出来的是一个总数据，不能提供更多纬度的统计。

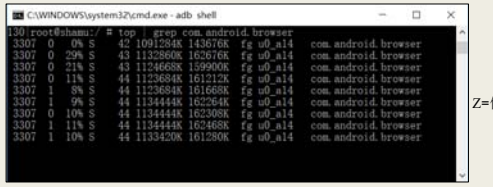
移动APP CPU测试

- APP的CPU测试一般需要考虑以下两种场景：
  - *（1）活动状态，即用户直接操作APP时的CPU占用率。*
  - *（2）静默状态，即用户没有操作APP，APP处于后台状态时的CPU占用率。*
- 测试APP的CPU占用率的方法包括但不限于如下几种：
  - *（1）第三方工具。比如，腾讯GT、网易Emmagee、阿里易测、手机自带监控等。这类工具使用起来简单、容易上手，并且可以产生易读性较高的报告，是初学者和小型测试团队的首选。*
  - *（2）dumpsys命令。类似：adb shell dumpsys cpuinfo | grep {PackageName}。*
  - *（3）top命令。类似：adb shell top | grep {PackageName}。*
  - *其中（2）和（3）得出的数据可能会不一样，但这是正常的，因为这两者在底层的计算方法是不一样的。在使用这两种方式的时候，也可以把数据保存到Excel中，然后利用Excel的图表功能绘制出一张CPU的变化曲线图。*

APP的CPU占用率测试示例

- 下面使用top命令来讲解如何查看手机的浏览器软件所消耗的CPU，命令如下

```
adb shell top | grep com.android.browser
```



使用top命令查看APP的CPU占用率

图中各字段含义大致如下：

- 第一列 PID：进程ID。
- 第二列 PR：优先级。
- 第三列 CPU：瞬时CPU占用率。
- 第四列进程状态：R=运行，S=睡眠，T=跟踪/停止，Z=僵尸进程。
- 第五列 THR：程序当前所用的线程数。
- 第六列 VSS：虚拟耗用内存。
- 第七列 RSS：实际使用物理内存。
- 第八列 UID：进程所有者的用户ID。
- 第九列 Name：进程名称。

移动APP电量测试

- 电量测试是评估APP消耗电量快慢的一种方法。电量测试方法很少，但需要测试的场景却比较多，手机在不同使用场景下消耗的电量肯定会不一样。
- 电量测试中需要考虑的测试场景包括但不限于以下几种：
  - *（1）待机状态。包括无网络待机、WiFi待机、4G待机等。*
  - *（2）活动状态。即不断地进行某些场景的操作，除了常规操作外，还应该包括看视频、灭屏下载、唤醒等。*
  - *（3）静默状态。即打开APP之后并不操作，让APP在后台运行。*
- 相对于其他项目的测试，电量测试的方法比较少，一般常见的电量测试方法包括但不限于以下几种：
  - *（1）通过硬件进行测试。比如，耗电量测试仪、腾讯的电量宝等。*
  - *（2）通过adb shell dumpsys batterystats命令。该命令只能在Android 5.0以上的系统中使用。Android 6.0对该命令进行了一些优化，可以得出更加详细的数据。*
  - *（3）第三方工具或者云测平台。手机系统内部也有一个自带的电量统计这个工具可以分别从软件和硬件角度看到耗电百分比。*