**【APP测试前瞻】**

当前移动互联网已经过了草创时期，各家公司都在抢占市场和用户；那么，就看哪一家做的更精致（交互做的更好，崩溃出现的更少）。

Android用户也许会经常碰到以下的问题：

**1. 稳定性问题** —— ANR /Crash等问题；ANR是指当Android系统监测到应用程序在5秒内没有响应输入的事件或广播在10秒内没有执行完毕时抛出无响应提示。Crash是指当应用程序出现错误时导致程序异常停止或退出的情况。

**2. 性能问题**—— 应用启动慢（首次/非首次启动应用，进入应用特别慢），CPU不足/内存泄露（应用使用过程中，越来越卡），应用耗电快（应用后台开着，手机很快没电）等问题。

注：卡顿与ARN的问题。卡顿简单的来说，就是手机没有及时响应、页面延迟，出现丢帧的现象，或者点击无响应。绝大多数的卡顿，稍等片刻系统就会恢复正常，但假如超过5S，就可能会引发手机ANR警告。

APP测试涉及各个方面，而稳定性和性能测试一般在功能测试完成后开展。如何应用自动化测试场景，如图1所示：

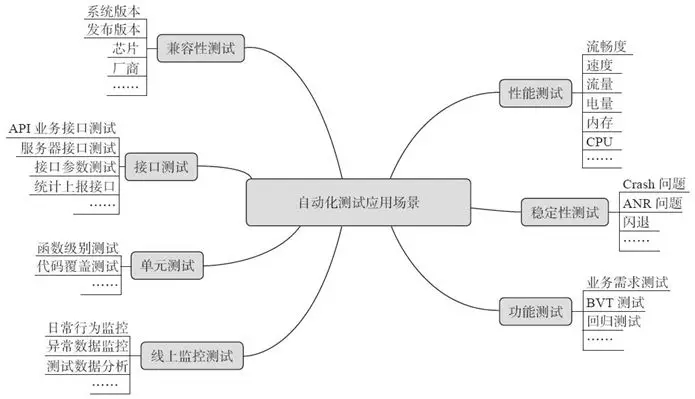


图1 自动化测试应用场景

**自动化框架的选择**

**1. autoMonkey框架**：Android平台一般都会考虑使用系统自带的Monkey工具进行测试，此工具既易上手也实用，但运用起来有非常多的讲究和技巧，简单的Monkey工具不一定能完成使命，在测试中也需要花费心思去对Monkey进行封装改造（还可以修改原生的Monkey），以满足测试需求。

**2. MBT（Model based testing）**：基于模型的测试。建模过程，通常是指对被测系统的某些方面的描述，以及对被测系统预期行为的抽象描述。FSM（有限状态机）用途最广泛，一般用于黑盒测试的建模。MBT与Monkey测试理论的结合，即放一只“猴子”在FSM模型中随机漫游，开展充分的稳定性测试。

本文是针对APP完成首轮功能测试后，使用autoMonkey框架进行稳定性和性能测试，从而提高产品的稳定性和留存率。知识结构分为四个部分，如图2所示：



图2 autoMonkey框架原理与应用

**一、Monkey基础知识与测试场景**

在Android的官方自动化测试领域有一只非常著名的“猴子”叫Monkey，这只“猴子”一旦启动，就会让被测的Android应用程序像猴子一样到处乱跑。我们常用这只“猴子”对被测程序进行压力测试，检查和评估被测程序的稳定性。

Android官方对这只“猴子”的描述：Monkey是Google提供的一个命令行工具，可运行在模拟器或实际设备中。它可以在规定的次数范围内做任何随机的操作，随机操作包括点击、滑动、Application切换、横竖屏、应用关闭等等，用户能做的操作都可以模拟，从而对正在运行的应用程序进行压力测试，目的是看设备多长时间会出现异常，并观察系统的稳定性和容错性能。

**Monkey测试出crash的bug等级永远为1，版本发布前，Monkey跑出的结果中crash要为0。**

**业内标准：最终发布版本前，Monkey跑完的总次数应为25W次，其结果里不允许有nullPointException出现。**

Monkey程序是Android系统自带的，其启动脚本是位于Android系统的/system/bin目录的Monkey文件，其jar包是位于Android系统的/system/framework目录的Monkey. jar文件。用户主要是通过adb 命令来启动Monkey。Monkey在运行时，会根据命令行参数的配置，生成伪随机的事件流，并在Android设备上执行对应的测试事件。

按照选定的不同级别的反馈信息，在Monkey中还可以看到其执行过程报告和生成的事件。

**Monkey的弊端**

 无法准确地得知bug的复现步骤

 发送的用户事件流的有效性成问题，很多是无效操作

**1．常规的稳定性测试**

**测试目的：**

**希望通过Monkey测试来模拟用户长时间的随机操作，检查被测APP是否会出现异常（应用崩溃或者无响应）。**

**测试脚本：**

adb  shell  monkey  -p  $package\_name  --hprof  --pct-touch  40  --pct-motion  25  --pct-appswitch  10 --pct-rotation  10  --pct-majornav  10  -s 10000  --throttle 500  --ignore-crashes  --ignore-timeouts  -v -v -v  500000

这个Monkey测试的命令比较复杂，**主要是对一些操作事件做了限制，从而减少Monkey伪随机流的无效操作**。这体现在以下几个方面。

**1）使用-p参数来制定测试APP的包名（Package）**

因为被测APP是一个特定的Android应用程序，需要指定被测APP的包名。指定包名后，Monkey会根据包名找到对应的APP，并启动其main activity，然后执行Monkey测试。

**技巧：**

查找应用包名的方法有很多，这里简单列举几个常用方法：

（1）通过pm命令查看。

在命令行窗口输入：

>adb shell pm list package -3

列出手机上所有第三方的APP包名，在列表中找到要测试的APP包名；

（2）通过查看APK源码下的AndroidManifest.xml文件；

（3）通过aapt dump badging $package.apk命令查看；

（4）通过adb logcat抓取当前Android机运行的App的包名；

（5）使用adb shell dumpsys window | findstr mCurrentFocus命令查看当前运行的包名。

**2）使用--pct-xxx参数限制Monkey执行的事件类型和占比**

测试的目的是希望模拟用户操作，因此需要让Monkey执行的事件尽可能地接近用户的常规操作，这样才可以最大限度地发现用户使用过程中可能出现的问题。因此需要对Monkey执行的事件百分比做一些调整。

**触摸事件和手势事件是用户最常见的操作**，所以通过--pct-touch和--pct-motion将这两个事件的占比调整到40%与25%；

目标应用包含了多个Activity，为了能覆盖大部分的Activity，所以通过--pct-appswitch将Activity切换的事件占比调整到10%；

被测应用之前在测试中出现过不少横竖屏之间切换的问题，这个场景也必须关注，因此通过--pct-rotation把横竖屏切换事件调整到10%；

通过--pct-majornav，调整“主要”导航事件的百分比调整到10%（这些导航事件通常引发图形界面中的动作，如：5-way键盘的中间按键、回退按键、菜单按键）。

**3）使用-s参数来指定命令执行的seed值**

Monkey会根据seed值来生成对应事件流，同一个seed生成的事件流是完全相同的。**指定seed值，是为了测试发现问题时，便于进行问题复现。**

**4）使用--throttle参数来控制Monkey每个操作之间的时间间隔**

**指定操作之间的时间间隔，一方面是希望能更接近用户的操作场景，正常用户操作都会有一定的时间间隔；另一方面也是不希望因为过于频繁的操作而导致系统崩溃，尤其是在比较低端的手机上执行测试时。**因此通过--throttle设置Monkey每个操作固定延迟0.5秒。

**5）使用--ignore-crash和--ignore-timeouts参数使Monkey遇到意外时能继续执行**

在执行Monkey测试时，会因为应用的崩溃或没有响应而意外终止，所以需要在命令中增加限制参数--ignore-crash和--ignore-timeouts，让Monkey在遇到崩溃或没有响应的时候，能在日志中记录相关信息，并继续执行后续的测试。是否使用这两个参数在后续Monkey测试策略详细介绍。

**6）使用-v指定log的详细级别**

Monkey的日志输出有3个级别：默认的是level 0， -v -v日志级别为level 1， -v -v -v日志级别为level 2。日志的级别越高，越详细。为了方便问题的定位，将日志级别设置为level2。

**7）使用--hprof在出现内存问题时，存储一份内存“快照”**

另，在常规的稳定性测试中，虽然可以自定义各种事件的操作占比，但毕竟是随机事件流。在实际测试过程中，难免会遇到Monkey点了我们不希望它点击的地方，比如误点了工具栏导致网络断开的情况等。当WiFi断开时，使用APP“Auto WiFi Toggle”每隔5分钟检测网络状态并自动连接。

**2. 结合辅助命令，获取更多信息（性能测试）**

常规测试只要记录下Monkey日志，再分析Monkey日志检查是否有异常即可。**但是，很多时候，测试除了想知道执行过程是否有异常，还需要能获取执行过程中的一些详细信息或性能数据，比如想知道在Monkey执行过程中是否存在内存泄漏或内存溢出，需要获取内存信息。**此时，我们需要借助一些辅助的命令来获取更多信息。下面列举Monkey测试中常用的辅助命令，使用方法也非常简单，只要在执行Monkey的同时，另起一个CMD命令行窗口输入对应命令执行即可。

**1）记录logcat日志信息：**

adb shell logcat -v time > logcat.txt

**2）获取内存信息：**

adb shell dumpsys meminfo $package\_name or $pid

**3）获取CPU信息：**

adb shell top -n 1 |findstr $package\_name or $pid

**4）获取电量信息：**

adb shell dumpsys battery

**5）获取GPU信息：**

GPU信息命令：adb shell dumpsys gfxinfo $package\_name or $pid

要获取GPU信息，在“开发者选项”中打开“GPU呈现模式分析”，如图3所示：



图3 GPU呈现模式分析

**6）获取流量信息：**

adb shell cat/proc/uid\_stat/$uid/tcp\_rcv

adb shell cat/proc/uid\_stat/$uid/tcp\_snd

**技巧：如何获取被测应用的UID**

步骤1：查看APP的进程ID（PID）

adb shell ps | grep $package\_name

步骤2：查看APP的用户ID（UID）

adb shell cat /proc/$pid/status