

APP 性能测试(启动速度、内存、CPU、FPS、GPU、耗电量)

文章目录

- 启动速度
- 内存
- CPU
- FPS (应用的使用流畅度)
- GPU 渲染
- 耗电量

启动速度

测试内容

- ✧ 冷启动速度
- ✧ 热启动速度
- ✧ 完全启动速度
- ✧ 有网启动速度
- ✧ 无网启动速度
- ✧ 主要测试冷启动和热启动

测试标准

测试标准：冷启动时间不超过 1.5s, 热启动不超过 1s

测试方法

方法 1：通过 FFmpeg 录屏拆帧

- ✧ 官网：<https://ffmpeg.zeranoe.com/builds/>
- ✧ FFmpeg 常用命令：<https://www.jianshu.com/p/91727ab25227>

```
✧ ffmpeg [global_options] {[input_file_options] -i input_url} ...  
    {[output_file_options] output_url} ...
```

✧

```
✧ ffmpeg -i [输入文件名] [参数选项] -f [格式] [输出文件]
```

✧

- ✧ 参数选项:
- ✧ (1) `-an`: 去掉音频
- ✧ (2) `-vn`: 去掉视频
- ✧ (3) `-acodec`: 设定音频的编码器, 未设定时则使用与输入流相同的编解码器。音频解复用一般在后面加 `copy` 表示拷贝
- ✧ (4) `-vcodec`: 设定视频的编码器, 未设定时则使用与输入流相同的编解码器, 视频解复用一般在后面加 `copy` 表示拷贝
- ✧ (5) `-f`: 输出格式 (视频转码)
- ✧ (6) `-bf`: B 帧数目控制
- ✧ (7) `-g`: 关键帧间隔控制 (视频跳转需要关键帧)
- ✧ (8) `-s`: 设定画面的宽和高, 分辨率控制 (`352*278`)
- ✧ (9) `-i`: 设定输入流
- ✧ (10) `-ss`: 指定开始时间 (`0:0:05`)
- ✧ (11) `-t`: 指定持续时间 (`0:05`)
- ✧ (12) `-b`: 设定视频流量, 默认是 `200Kbit/s`
- ✧ (13) `-aspect`: 设定画面的比例
- ✧ (14) `-ar`: 设定音频采样率
- ✧ (15) `-ac`: 设定声音的 Channel 数
- ✧ (16) `-r`: 提取图像频率 (用于视频截图)
- ✧ (17) `-c:v`: 输出视频格式
- ✧ (18) `-c:a`: 输出音频格式
- ✧ (18) `-y`: 输出时覆盖输出目录已存在的同名文件
- ✧
- ✧

✧ `-vcodec` 设定视频的编码器，未设定时则使用与输入流相同的编解码器

方法 2：使用 adb

- ✧ `adb shell am start -W packageName (包名) /ActivityName (活动名) (绝对路径, 首个 Activity)`
- ✧ `adb shell am start -W (包名) / (活动名)`

方法 3：使用 charles 抓包（查看 duration）

内存

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

测试点

- ✧ 空闲状态：切换至后台或者启动后不做任何操作，消耗内存最少
- ✧ 中强度状态：时间偏长的操作应用
- ✧ 强度状态：高强度使用应用，可以跑 monkey 来测试（通常用来测内存泄漏）
- ✧ 内存泄漏：指应用里的内存一直没有释放，内存一直增加，系统内存一直减少

关注点

- ✧ 退出某个页面后，内存是否有回落
- ✧ 进行某个操作后，内存是否增长过快
- ✧ 旧版本和新版本比较
- ✧ 新版本和竞品比较

测试方法

方法 1：使用 adb 命令

`adb shell dumpsys meminfo packageName`

获取当前活动的包名和 actively (`adb shell dumpsys window | findstr mCurrentFocus`)
(mCurrentFocus—当前焦点)

关注参数

native heap alloc (JNI 层的内存分配)

dalvik heap alloc (java 层的内存分配)

pss: 应用真正占据的内存大小

注意: 如果前两个值一直增长, 应用程序可能出现了内存泄漏

方法 2: 使用性能测试工具 emmagee

Emmagee 是网易开发的一款测安卓应用性能的测试 apk

使用方法

安装到 emmagee 到手机上, 启动

选择需要测试性能的应用启动

被测应用界面会展示内存、cpu、电流、流量等数据

stop test 后, 本地 sd 卡中会保存一份性能测试数据 ((保存地址: /sdcard/Emmagee/*****.csv 文件))

可以通过 excel 将数据转化为图表, 更直观的查看各性能指标的数据

方法 3: 使用 AndroidStudio 自带 CPU 和内存检测功能 – Android Monitor

方法 4: 内存检测工具 DDMS -->Heap

CPU

主要关注的是 cpu 的占用率

测试点

- ✧ 在空闲时间 (切换至后台) 的消耗, 基本没大应用使用 cpu
- ✧ 在运行一些应用的情况下, cpu 已占 50% 的情况下, 观察应用程序占用 cpu 的情况
- ✧ 在高负荷的情况下看 cpu 的表现 (cpu 应该是在 80% 以上)

关注点

- ✧ 和自身 app 的上个版本对比
- ✧ 和竞品对比
- ✧ 自身 app 各个 activity 对比

测试方法

方法 1：使用 adb

adb shell top -m cpu |grep packageName (查看某个软件的 cpu 占用率)

adb shell top -m 10 -s cpu (查看 cpu 占用前 10 的应用)

top cpu 参数:

-m 显示最大数

-s 按指定行排序

-t 显示进程名称

-n 在退出前刷新几次

-d 刷新闻隔

adb shell dumpsys cpuinfo |grep 包名 (一段时间的平均值)

方法 2：使用第三方工具 Emmagee、GT 等

方法 3：使用 androidstudio 自带的检测工具 android monitor

FPS (应用的使用流畅度)

fps 是指画面每秒传输的帧数，每秒钟帧数越多，所显示的动作就会越流畅，一般来说安卓设备的屏幕刷新率为 60 帧/秒，要保持画面流畅不卡顿，要求每一帧的时间不超过 $1000/60=16.6\text{ms}$ ，否则就会出现跳帧、画面卡顿

关注点

高于 16ms 的帧率 (即卡顿率)

测试方法

方法 1：adb 命令

打开手机：开发者选项->GPU 呈现模式分析->在 adb shell dumpsys gfxinfo

操作要测试的 app

在 cmd 窗口输入 adb shell dumpsys gfxinfo 包名

得到一个矩阵数据，计算矩阵中帧率大于 16 的点所占比例，即为卡顿比

Draw: 表示在 Java 中创建显示列表部分中，OnDraw()方法占用的时间。

Process: 表示渲染引擎执行显示列表所花的时间，view 越多，时间就越长。

Execute：表示把一帧数据发送到屏幕上排版显示实际花费的时间。

Draw + Process + Execute = 完整显示一帧，这个时间要小于 16ms 才能保证每秒 60 帧

Janky frames：丢帧率

方法 2：直接使用开发者选择自带的图标

打开手机：开发者选项->GPU 呈现模式分析->在屏幕上显示为条形图

操作要测试的 app

绿色的线是 16ms 的分隔线，可以直接看出来流畅度

方法 3：使用第三方工具 Emmagee、GT 等

方法 4：使用 androidstudio 自带的检测工具 android monitor

GPU 渲染

GPU 渲染是指在一个像素点上绘制多次（超过一次），过度绘制对动画性能的影响是极其严重的，如果你想要流畅的动画效果，那么一定不能忽视过度绘制。

测试指标

- ✧ 控制过度绘制为 2x
- ✧ 不允许存在 4x 过度绘制
- ✧ 不允许存在面积超过屏幕 1/4 的 3x 过度绘制

测试方法

方法 1：使用手机的开发者选项

- 打开手机：开发者选项->调试 GPU 过度绘制->显示过度绘制区域
- 打开被测的应用，进行操作
- 颜色深的区域为过度绘制的地方
 - 原色：无过度绘制
 - 蓝色：绘制一次
 - 绿色：绘制两次
 - 浅红：绘制三次（可以优化了）
 - 深红：绘制四次（必须优化）

耗电量

测试应用对电量的消耗前需要对手机本身的电量消耗有个大概了解, 然后再启动待测试 app 看看消耗的电量增加了多少, 取差值

测试点

- ✧ 测试手机安装目标 apk 前后待机功率有无明显差异
- ✧ 常见使用场景中能够正常进入待机, 待机电流在正常范围内
- ✧ 长时间连续使用无异常耗电现象

测试方法

- ✧ 先关闭所有的应用, 再打开被测 app
- ✧ 方法 1: 使用第三方测试工具: Emmagee、GT 等, 只需要测试的电流静置一晚, 待机
 - 电流在正常范围内即可。一般是被测应用对比待机电流 $\leq 2\text{mA}$ 。
- ✧ 方法 2: 使用 adb 命令
 - `adb shell dumpsys batterystats |grep packageName`

改变手机电池状态

- ✧ 手机连接电脑, 默认为充电状态
- ✧ 切换手机电池为非充电状态 `adb shell dumpsys battery set status 1`

获取电量消耗信息

- ✧ 获取整个设备的电量消耗信息: `adb shell dumpsys batterystats | more`
 - ✧ 获取某个 apk 的电量消耗信息: `adb shell dumpsys batterystats com.Package.name | more`
-