

异步赠书：9月重磅新书升级，本本经典 SDCC 2017之区块链技术实战线上峰会 程序员9月书讯 每周荐书：ES6、虚拟现实、物联网（评论送书）

Android 常用的性能分析工具详解：GPU呈现模式, TraceView, Systrace, HirearchyViewer

2015-12-24 21:04

12321人阅读

评论(0)

版权声明：本文为博主原创文章，未经博主允许不得转载。

此篇将重点介绍几种常用的Android性能分析工具：

一、Logcat 日志

选取Tag = ActivityManager，可以粗略地知道界面Displaying的时间消耗。当打印一串log如下：

```
[html]
01. I/ActivityManager: Displayed xxx.xxx.xxx/TestActivity: +1s
```

第一个时间表示系统接受到打开的intent到TestActivity界面显示出来的时间1.272秒

关闭





核电池



十大电吹风



程序员薪资



呼叫中心系统



新款手机



阿米巴经营模



橡胶止水带



密集架

- Android 进程常驻、进程 (15515)
- 如何获取Android Recycl (12646)
- Android 常用的性能分析 (12282)
- Android 如何获取最顶端 (11700)
- Android LLVM-Obfuscate (11446)
- Java 进程间文件锁FileLc (9686)
- Android 如何判断CPU是 (8995)
- Android 如何监控当前Fo (8988)
- Git remote 同步远程仓库 (8985)
- Android Plugin 插件化技 (7664)

第二个时间特殊情况下才会有。例如这种调用流程：A->B(在onCreate立刻finish掉自己，直接跳转到C)->C(TestActivity)，它包含了B页面onCreate处理以及finish的时间。这个时间如果过长，会导致用户点击跳转后，页面还停留在原来界面，延迟一段时间再跳转。这种体验很差，用户会觉得卡顿并容易点击多次。很多应用程序的入口页面，都设计成这种情况的跳转。

在此我们只讨论第一个时间，在这个时间内系统做了以下事情：ActivityManager做一些task相关的处理 -> 应用内创建Activity实例 -> Activity的onCreate->Activity的onResume->布局的初始化->首次绘制的一些东西，这个流程是指一个Activity在一个task初次创建的情况。如果是clearTop等流程，可能还涉及其他Activity的finish，自己onNewIntent等流程。

二、GPU呈现模式分析(Peofile GPU Rendering tool)

1. 在adnroid手机上开启这个功能：打开“开发者选项”->“GPU呈现模式分析”->“在屏幕上显示为条形图”



GPU呈现模式用来测量app的帧速率，属于GPU Profile工具的一种。目前安卓基础60fps以滴可以计算出 $1/60 \approx 1.66$ （绘制每帧需要的时间约为16ms）

2. 开启GPU呈现模式之后，当你的应用程序在运行时，你会看到一排柱状图在，直的柱状图代表一帧的渲染，越长的垂直柱状图表示这一帧需要渲染的时间越长，会堆积在一起，这样你就可以观察到这段时间帧率的变化。

关闭



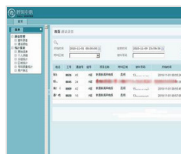
核电池



十大电吹风



程序员薪资



呼叫中心系统



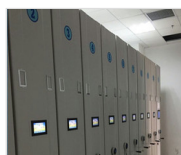
新款手机



阿米巴经营



橡胶止水带



密集架

「万法快速打造你的阅读清单」

- * Android检查更新下载安装
- * 动手打造史上最简单的Recycleview 侧滑菜单
- * TCP网络通讯如何解决分包粘包问题
- * SDCC 2017之区块链技术实战线上峰会
- * 快速集成一个视频直播功能

最新评论

Android 如何 ListView 判断滚动? GNOSOAB: 还有问题,需要考虑listview的条目高度大于屏幕高度,以及只有一个条目的情况;

[plain]

- (1). 绿色水平线代表16ms,要确保一秒内打到60fps,你需要确保这些帧的每一条线都在绿色的16ms标记线之下.任何时候你看到一个竖线超过了绿色的标记线,你就会看到你的动画有卡顿现象产生.
- (2). 蓝色代表测量绘制的时间,或者说它代表需要多长时间去创建和更新你的DisplayList.在Android中,一个视图在可以实际的进行渲染之前,它必须被转换成GPU所熟悉的格式,简单来说就是几条绘图命令,复杂点的可能是你的自定义的View嵌入了自定义的Path.一旦完成,结果会作为一个DisplayList对象被系统送入缓存,蓝色就是记录了需要花费多长时间在屏幕上更新视图(说白了就是执行每一个View的onDraw方法,创建或者更新每一个View的Display List对象).当你看到蓝色的线很高的时候,有可能是因为你的一堆视图突然变得无效了(即需要重新绘制),或者你的几个自定义视图的onDraw函数过于复杂.
- (3). 红色代表执行的时间,这部分是Android进行2D渲染 Display List的时间,为了绘制到屏幕上,Android需要使用OpenGL ES的API接口来绘制Display List.这些API有效地将数据发送到GPU,最总在屏幕上显示出来.
- (4). 橙色部分表示的是处理时间,或者说是CPU告诉GPU渲染一帧的时间,这是一个阻塞调用,因为CPU会一直出接到命令的回复,如果柱状图很高,那就意味着你给GPU太多的工作,太多的负责视图需要OpenGL命令去绘制

3. 如果需要分析具体详细的数据, 需要结合下面的命令来进行:

```
$ adb shell dumpsys gfxinfo your_package
```

执行该命令产生的日志中, 你会发现一个标题为: Profile data in ms. 这一节包含为每个窗口所属应用产生的3列表格. 为数据, 简单的复制表格到你喜欢的电子表格软件中就会生成一个堆叠柱状图表. 下面的图是借用网上他人的测量结果:

关闭





核电池



十大电吹风



程序员薪资



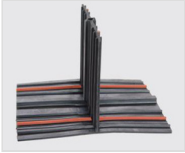
呼叫中心系统



新款手机



阿米巴经营模



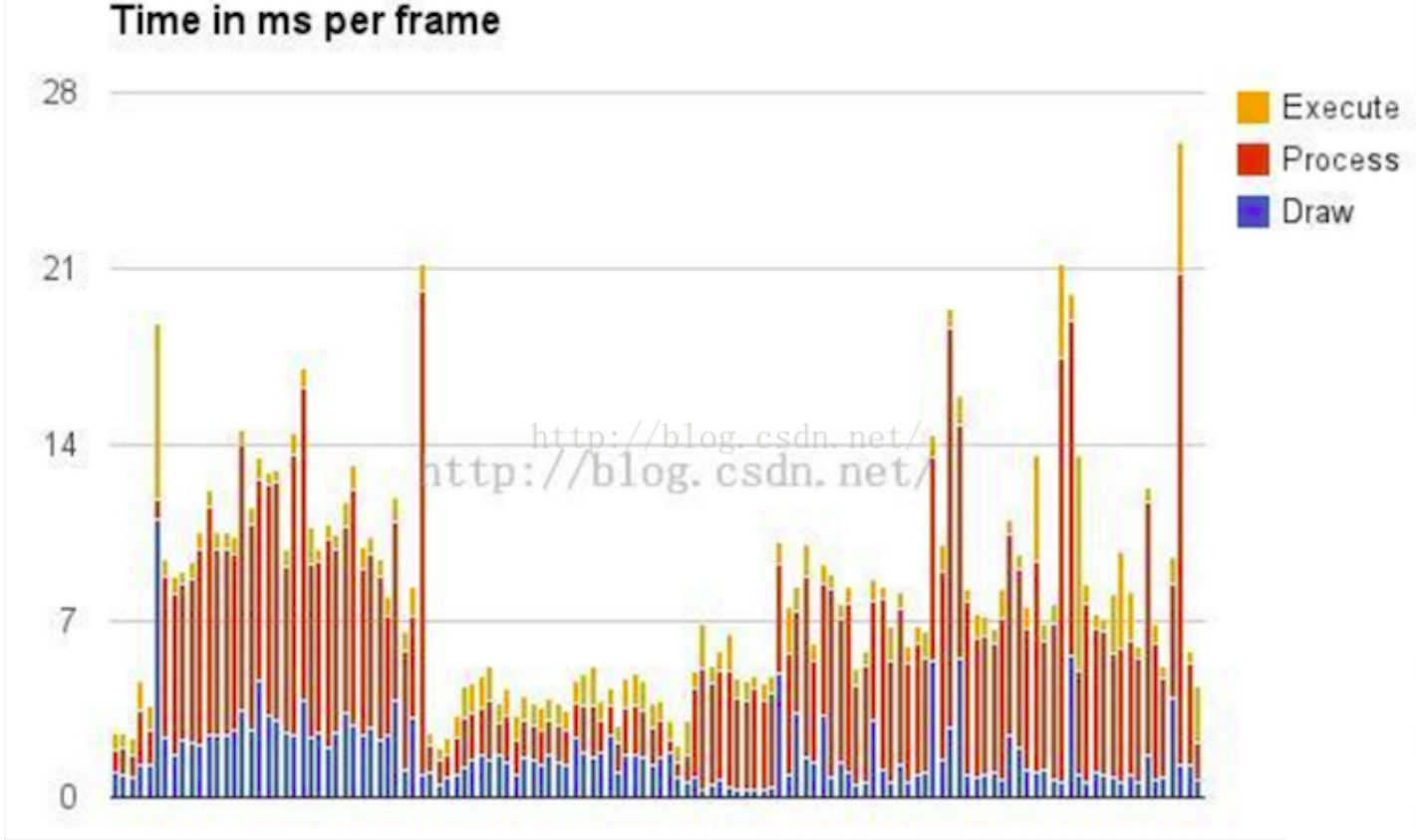
橡胶止水带



密集架

如何获取Android RecyclerView
yx90Chen: @wangbaochu:非常
感谢楼主能够回复, 谢谢交流~

如何获取Android RecyclerView
苍痕: @Cyx77520:对, 你说得
对, 上面这个方法只适合item高
度相同的RecyclerView。但对...



上图每列给出渲染每帧大概需要多长时间:

[plain]

01. (1). Draw 对应于蓝色线:是消耗在构建java显示列表DisplayList的时间,方法,创建或者更新每一个View的DisplayList对象的时间。
02. (2). Process 对应于红色线:是消耗在Android的2D渲染器执行显示列表的命令就越多。
03. (3). Execute 对应于橙色线:是消耗在排列每个发送过来的帧的顺序的时间,这是一个阻塞调用,因为CPU会一直等待GPU发出接到命令的回复。其实可以简
family: Arial, Helvetica, sans-serif;">Process时间 + GPU返回
family: Arial, Helvetica, sans-serif;">GPU<span st
family: Arial, Helvetica, sans-serif;">与CPU通信的时间</spar

关闭

注意：使顺利在60帧，每帧必须小于16毫秒完成。

关于Execute:如果执行耗费了过长的时间,这意味着你是跑在前面的图形管线。 android在运行时可以有3个缓冲区.如果你需要另一个应用程序将阻塞直到其中的一个缓冲区释放出来。两个原因会发生这种情况。第一，你的应用在Dalvik中快速绘制但在GPU显示列表时候消耗了大量时间。第二，你的应用程序花了很长的时间来执行第几帧;一旦管线满了他将无法赶上。

GPU Profile工具能够很好地帮助你找到渲染相关的问题,但是要修复这些问题就不是那么简单了。保持动画流畅的关键就在于让这些垂直的柱状条尽可能地保持在绿线下面,任何时候超过绿线,你就有可能丢失一帧的内容。你需要结合代码来具体分析,找到性能的瓶颈,并进行优化.有时候你可以打开GPU呈现模式来测试所谓的流畅度,让负责设计这个产品的人修改他的设计,以获得良好的用户体验。

三、TraceView

Android自带的TraceView堪比java的性能调优工具，可以方便的查看线程的执行情况，某个方法执行时间、在总体中的占比等，从而定位性能点。

1. 运行TraceView有两种方式

(a).调用Debug类

```
[java]
```

```
01. //在开始调试的地方，如Activity的onCreate函数，添加
02. Debug.startMethodTracing("tracefilename");
03.
04. //结束调试的地方，如Activity的onDestroy函数，添加
05. Debug.stopMethodTracing();
```

之后运行你的app一段时间并退出会在sd卡根目录生成tracefilename.trace这个文件。将日志文件pull到PC端，cd到android sdk tools文件夹内(或绑定sdk tools)，执行adb pull /sdcard/tracefilename.trace即可打开TraceView分析界面，如下



关闭





核电池



十大电吹风



程序员薪资



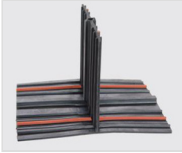
呼叫中心系统



新款手机



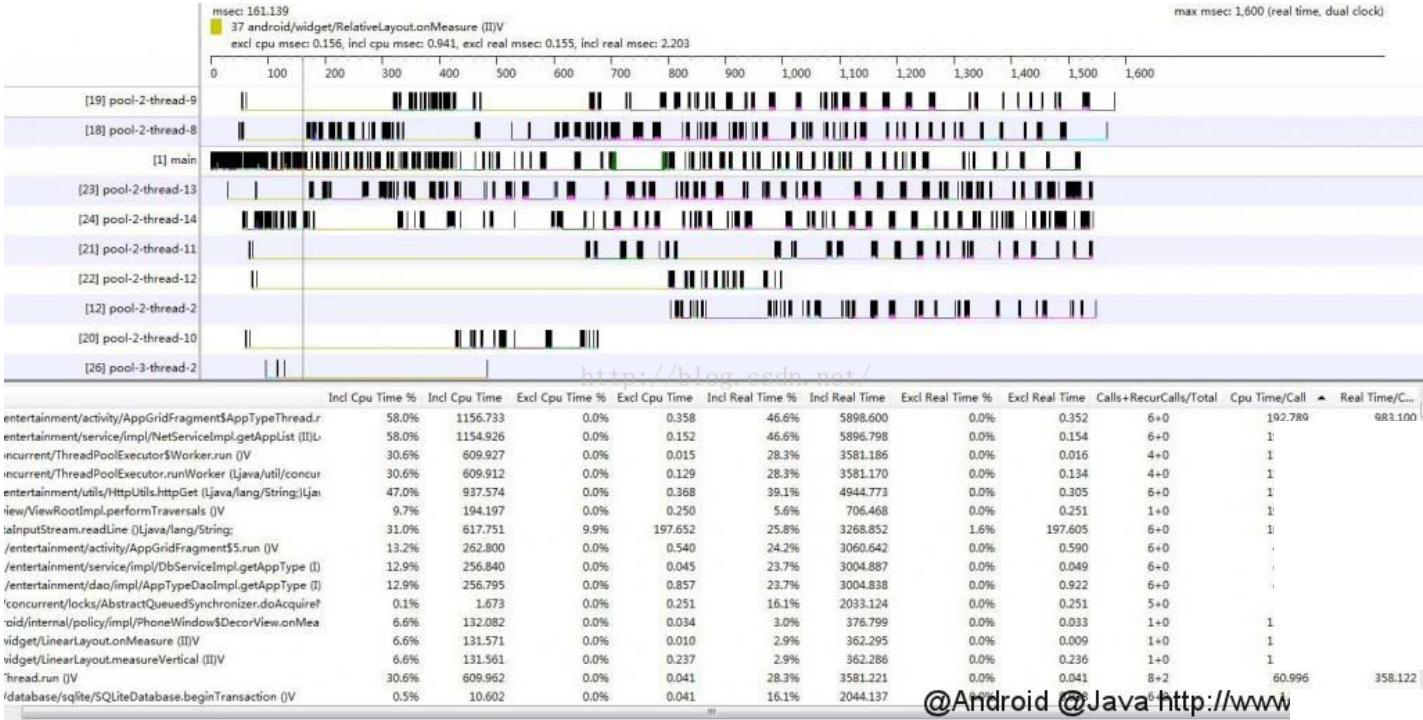
阿米巴经营模



橡胶止水带



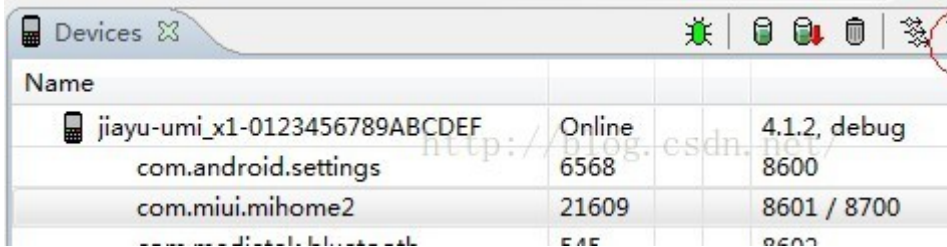
密集架



这种方式可以随意开始和结束调试的位置，所以适合具体代码的性能排查。

(b). 使用DDMs

打开devices窗口，选择某个进程，点击右上角的start method profiling



运行app一段时间后，再点击已变成stop method profiling的该按钮。eclipse会 (File->save as保存数据), 界面同上面。这种方式不需要修改代码，所以对于没时可以的进行全局性能排查。

关闭



2. TraceView界面信息介绍。TraceView界面包括时间面板和方法面板

(1) 时间面板(Timeline Panel)

时间面板展示了每个线程的执行情况，其中的[1]main即为ui主线程。
移动到某个位置可以查看该点对应的方法的执行信息，点击方法面板则会选中相应的方法。
可以左键按住不放选中区域放大局部精细查看，不同方法用不同颜色标注

(2) 方法面板(Profile Panel)

方法面板展示了所有方法的执行情况，点击某个方法可以查看在对应线程上的执行时间区域，并会显示其父方法及子方法。

每个方法包括如下信息列，可点击某列进行排序，从而确定产生性能问题的函数：

- [plain]
01.

Incl Cpu Time
02.

Excl Cpu Time
03.

Incl Real Time
04.

Excl Real Time
05.

Incl Cpu Time%
06.

Excl Cpu Time%
07.

Incl Real Time%
08.

Excl Real Time%
09.

Calls+RecurCalls/Total
10.

Cpu Time/Call
11.

Real Time/Call

所有的Time都是以毫秒计算。每列具体含义及作用如下：

关闭





列名	描述
Name	该线程运行过程中所调用的函数名
Incl Cpu Time	某函数占用的CPU时间，包含内部调用其它函数的CPU时间
Excl Cpu Time	某函数占用的CPU时间，但不含内部调用其它函数所占用的CPU时间
Incl Real Time	某函数运行的真实时间（以毫秒为单位），内含调用其它函数所占用的真实时间
Excl Real Time	某函数运行的真实时间（以毫秒为单位），不含调用其它函数所占用的真实时间
Call+Recur Calls/Total	某函数被调用次数以及递归调用占总调用次数的百分比
Cpu Time/Call	某函数调用CPU时间与调用次数的比。相当于该函数平均执行时间
Real Time/Call	同CPU Time/Call类似，只不过统计单位换成了真实时间

四、Systrace 性能分析的工具

1. Systrace是Android4.1中新增的性能数据采样和分析工具。它可帮助开发者收集Android关键子系统（surfaceflinger、WindowManagerService等Framework部分关键模块、服务）的运行信息，从而帮助开发者更直观的分析系统瓶颈，改进性能。Systrace的功能包括跟踪系统的I/O操作、内核工作队列、CPU负载以及Android子系统的运行状况等。在Android平台中，它主要由3部分组成：

内核部分：Systrace利用了Linux Kernel中的ftrace功能。所以，如果要使用Systrace的话，必须开启ftrace相关的模块。

数据采集部分：Android定义了一个Trace类。应用程序可利用该类把统计信息然后交给数据分析工具。还有一个atrace程序，它可以从ftrace中读取统计信息然后交给数据分析工具。

数据分析工具：Android提供一个systrace.py（python脚本文件，位于Android4.1的system/bin目录下，内部将调用atrace程序）用来配置数据采集的方式（如采集数据的标签、采样率等）并生成一个结果网页文件供用户查看。

从本质上说，Systrace是对Linux Kernel中ftrace的封装。Android 4.1为系统中添加Systrace功能。

2. 如何使用 Systrace

关闭



HUAWEI

「**免费试用 华为云**」

新 / 手 / 建 / 站

高性能云服务器

马上试用



核电池



十大电吹风



程序员薪资



呼叫中心系统



新款手机



阿米巴经营



橡胶止水带



密集架

systrace 可以通过命令启动，或者使用eclipse, android studio启动。systrace 工具你可以在sdk/platform-tools/ 找到，或者在源码里位于 external/chromium-trace 下面。

Systrace在各个平台上的使用流程基本是一样的：

(1)手机准备好你要进行抓取的界面

(2)点击开始抓取（命令行的话就是开始执行命令）

(3)手机上开始操作

(4)设定好的时间到了之后，会将生成Trace文件，使用Chrome将这个文件打开进行分析

UsingEclipse

[plain]

01. 1. In Eclipse, open an Android application project.
02. 2. Switch to the DDMS perspective, by selecting Window > Perspectives > DDMS.
03. In the Devices tab, select the device on which to run a trace. If no devices are
04. 3. Click the Systrace icon at the top of the Devices panel to configure tracing.
05. 4. Set the tracing options and click OK to start the trace.

Using Android Studio

[plain]

01. 1. In Android Studio, open an Android application project.
02. 2. Open the Device Monitor by selecting Tools > Android >
03. 3. In the Devices tab, select the device on which to run a trace.
04. 4. Click the Systrace icon at the top of the Devices panel.
05. 5. Set the tracing options and click OK to start the trace.

UsingDevice Monitor

[plain]

01. 1. Navigate to your SDK tools/ directory.
02. 2. Run the monitor program.

关闭





03. 3. In the Devices tab, select the device on which to run a trace. If no devices are listed
04. 4. Click the Systrace icon at the top of the Devices panel to configure tracing.
05. 5. Set the tracing options and click OK to start the trace.

Command Line Usage

你可以通过python systrace.py -h 来查看systrace 的使用帮助

[plain]

01. Usage: systrace.py [options] [category1 [category2 ...]]
02. Example: systrace.py -b 32768 -t 15 gfx input view sched freq
03. Options:
04. -h, --help show this help message and exit
05. -o FILE write HTML to FILE
06. -t N, --time=N trace for N seconds
07. -b N, --buf-size=N use a trace buffer size of N KB
08. -k KFUNCS, --ktrace=KFUNCS
09. specify a comma-separated list of kernel functions to
10. trace
11. -l, --list-categories
12. list the available categories and exit
13. -a APP_NAME, --app=APP_NAME
14. enable application-level tracing for
15. list of app cmdlines
16. --link-assets link to original CSS or JS resource
17. embedding them
18. --from-file=FROM_FILE
19. read the trace from a file (compressed)
20. running a live trace
21. --asset-dir=ASSET_DIR
22. -e DEVICE_SERIAL, --serial=DEVICE_SERIAL
23. adb device serial number

3. 下面，我们通过一个示例来展示Systrace的使用

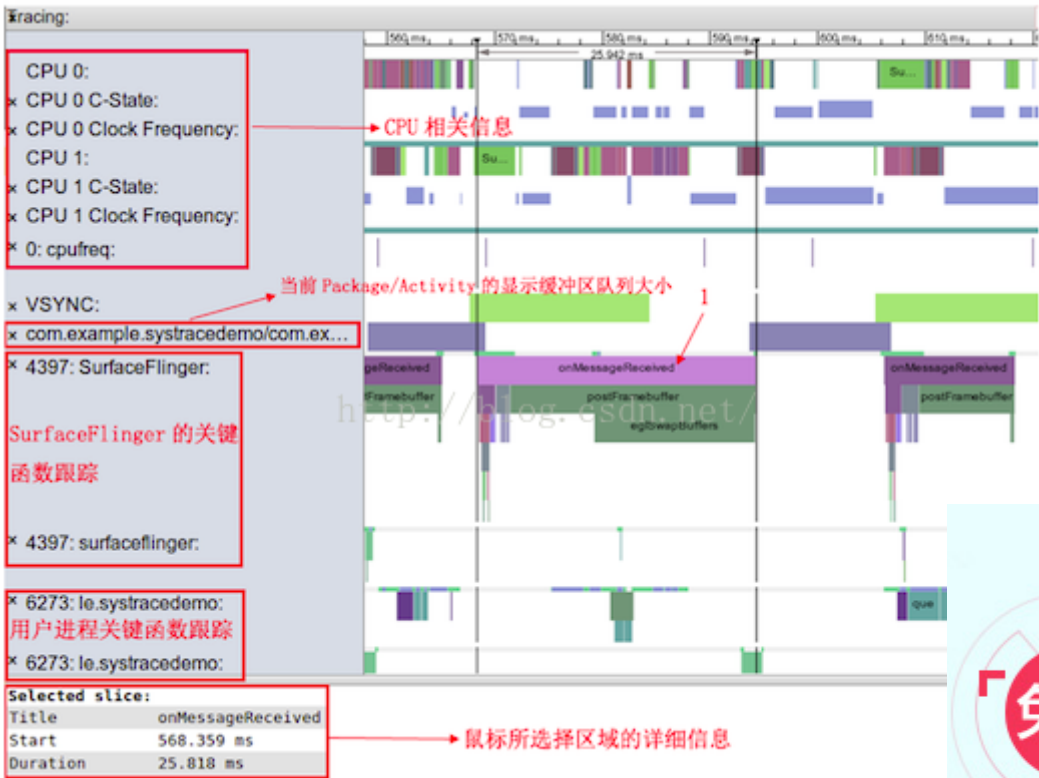
关闭





```
adb root
cd systrace-dir #进入systrace.py所在目录
#设置需要采集数据的标签
python systrace.py -set-tags=gfx #gfx代表Graphic标签
#start和stop由Android toolbox (类似busybox) 程序提供,
#用于启动和关闭init.rc 中定义的service (
#如bootanim, surfaceflinger)
adb shell stop#如果未指定参数, 则默认操作的是surfaceflinger和zygote
adb shell start #重新启动系统服务
#-f: 获取CPU频率信息, -l: 获取CPU负载信息, -i: 获取CPU状态信息
python systrace.py -f -l -i
```

执行上述命令后, 将得到一个名为trace.html的文件 (trace.html是默认文件名, 读者也可在命令行中指定其他文件名)。生成的trace文件需要通过Chrome浏览器打开, 结果如下图：



图中所示的trace.html页面内容和Traceview的Timeline Panel非常类似。图中(1). 由于在systrace.py中指定了-f -l和-i参数, Systrace将生成CPU频率、负载个红框所示。由于笔者所测手机CPU为双核, 故图中有CPU0和CPU1之分。某个核。

关闭



HUAWEI

「**免费试用 华为云**」

新 / 手 / 建 / 站

高性能云服务器

马上试用



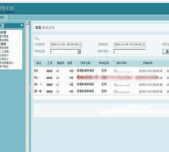
核电池



十大电吹风



程序员薪资



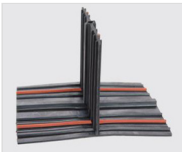
呼叫中心系统



新款手机



阿米巴经营模式



橡胶止水带



密集架

“CPU N”所示行对应于整个测试时间内，某个核上运行的进程信息。

“CPU N C-State”所示行为整个测试时间内，某个CPU状态的变化。C-State取值见下表。

“CPU N Clock Frequency”所示行展示了某个CPU运行的频率。通过点击这一行的色块可以查看某个时间点上CPU N的运行频率。

“cpufreq”：该行所示内容和CPU交互式频率调节器（Interactive Governor）的工作有关。交互式CPU调节器驱动添加了对CPU频率调节事件的跟踪。感兴趣的读者不妨阅读kernel中的 include/trace/events/cpufreq_interactive.h文件以了解更多的信息。

C-state	描述
C-0	RUN MODE，运行模式。
C-1	STANDBY，就位模式，随时准备投入运行
C-2	DORMANT，休眠状态，被唤醒投入运行时有一定的延迟
C-3	SHUTDOWN，关闭状态，需要有较长的延迟才能进入运行状态，减少耗电

(2). VSYNC：显示了每次Tick Tack的时间大概都在16ms左右

(3). 图中 com.example.systracedemo/com.example.systracedemo.MainActivity所示为应用程序占用显示 Tick-Tack情况。如果使用时间超过16ms，将导致界面显示迟滞等现象。

(4). 图中SurfaceFlinger行展示了其函数调用的CPU耗时情况（如箭头1所指，SurfaceFlinger中的 onMessageReceived函数的运行信息）。

(5). 图中最下部的方框中，详细显示了当前鼠标在时间线中选择的部分（即SurfaceFlinger中的 onMessageReceived）的详细信息。

你还可以用如下键盘操作：

- [plain]
01.

Key Description
02.

w Zoom into the trace timeline.
03.

s Zoom out of the trace timeline.
04.

a Pan left on the trace timeline.
05.

d Pan right on the trace timeline.
06.

e Center the trace timeline on the current mouse location.
07.

g Show grid at the start of the currently selected task.

关闭



HUAWEI

「**免费试用 华为云**」

新 / 手 / 建 / 站

高性能云服务器

马上试用



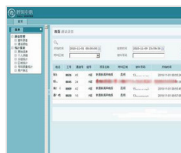
核电池



十大电吹风



程序员薪资



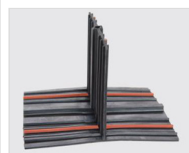
呼叫中心系统



新款手机



阿米巴经营



橡胶止水带



密集架

Android 常用的性能分析工具详解：GPU呈现模式, TraceView, Systrace, HierarchyViewer - 我的专栏 - CSDN博客

08. Shift+g Show grid at the end of the currently selected task.
09. Right Arrow Select the next event on the currently selected timeline.
10. Left Arrow Select the previous event on the currently selected timeline.
11. Double Click Zoom into the trace timeline.
12. Shift+Double Click Zoom out of the trace timeline.

找到你要分析的进程ID，分析每个线程在一段时间内具体做什么工作。如果你发现某个方法耗用了很长时间。可以去代码里面搜索去具体的实现。

4. 在自己代码中加入trace

Android framework 里面已经有很多trace的代码。你也可以仿照在自己一些方法里加入trace 跟踪调试。你需要保证 traceBegin 与 traceEnd 一定要成对出现，并且一定要在同一个线程里面。

[plain]

```
01. Trace.traceBegin(Trace.TRACE_TAG_VIEW, "performTraversals");
02. try {
03.     //TODO your work
04. } finally {
05.     Trace.traceEnd(Trace.TRACE_TAG_VIEW);
06. }
```

5. 总体来说，Systrace用法比Traceview要复杂，它支持对CPU、Native进程甚至Kernel线程进行性能数据帮助开发者对整个系统的性能情况进行一个详尽的分析。

Android官方对Systrace也有一些介绍，请读者阅读：

<http://developer.android.com/tools/debugging/systrace.html>

<http://developer.android.com/tools/help/systrace.html>

五、HierarchyViewer

1. 如何使用HierarchyViewer ?

关闭





Hierarchy Viewer是随Android SDK发布的工具，位于Android SDK/tools/hierarchyviewer.bat (Windows操作系统，mac上显示的为hierarchyviewer)，使用起来也是超级简单，通过此工具可以详细的理解当前界面的控件布局以及某个控件的属性（name、id、height等），调试UI界面分析其性能。

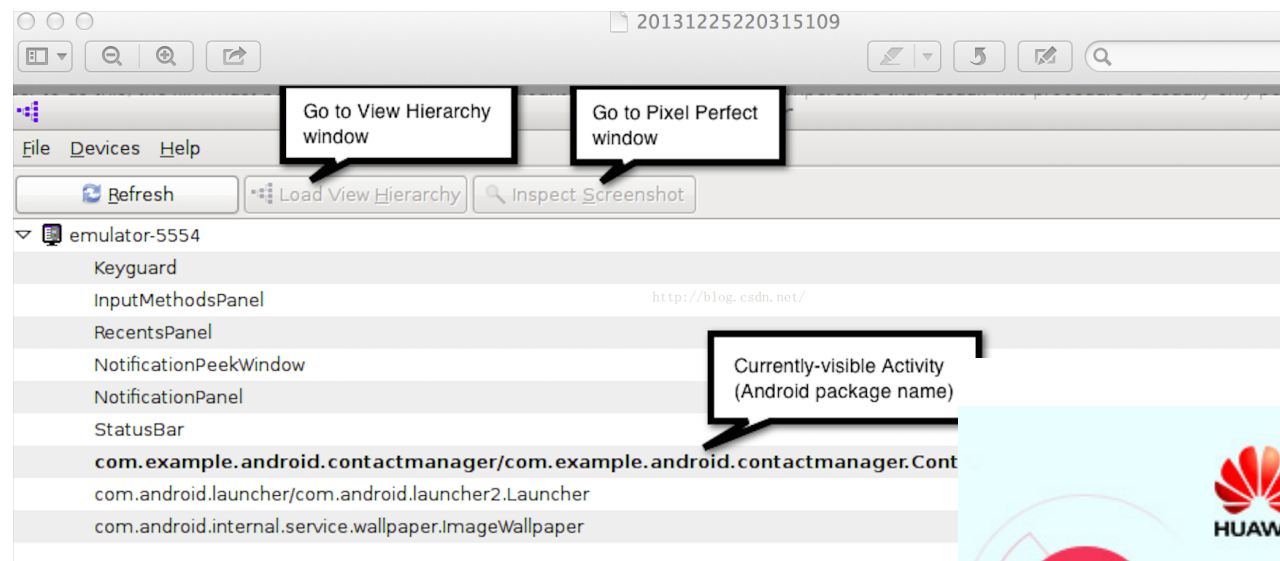
(1) 连接设备真机或者模拟器（真机可能无法连接，我用的2.3，连接上了，没读到内容）

(2) 启动你要观察的应用

(3) 打开Hierarchyviewer，点击hierarchyviewer文件即可。连接后如下图：

Load View Hierarchy: 界面的控件层次，查看界面中各个控件的层次结构关系

Inspect Screenshot: 精确查看模式，开发者可以随意点界面的任意一部分，进行放大或缩小观察以查看界面中各个控件的位置和情况



(4) 点击“Load View Hierarchy”，然后双击下图最上面的Focused Window，这个是当前窗

关闭





核电池



十大电吹风



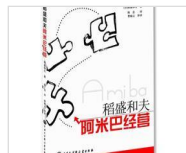
程序员薪资



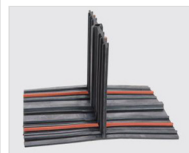
呼叫中心系统



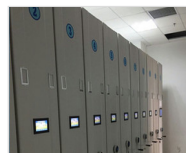
新款手机



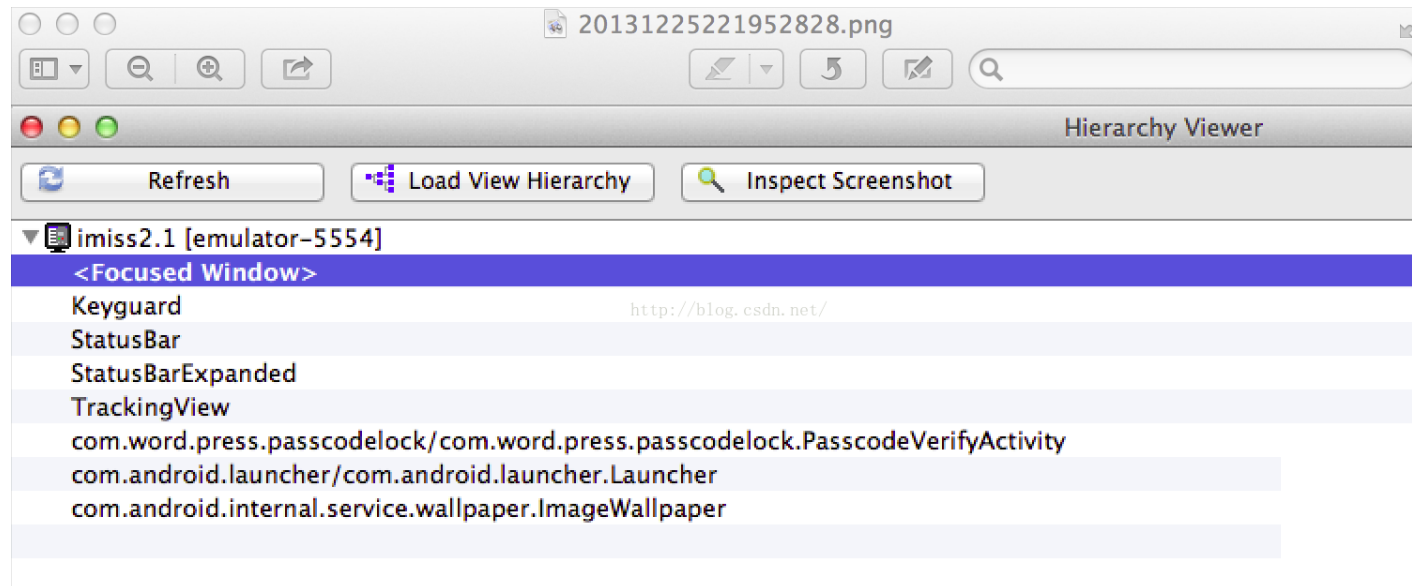
阿米巴经营



橡胶止水带



密集架



(5) 观察层次结构图，这个图有点大，可以拖动。View Hierarchy窗口显示了Activity的所有View对象，选中某个View还可的具体信息，最好选择工具中的Show Extras选项。

关闭



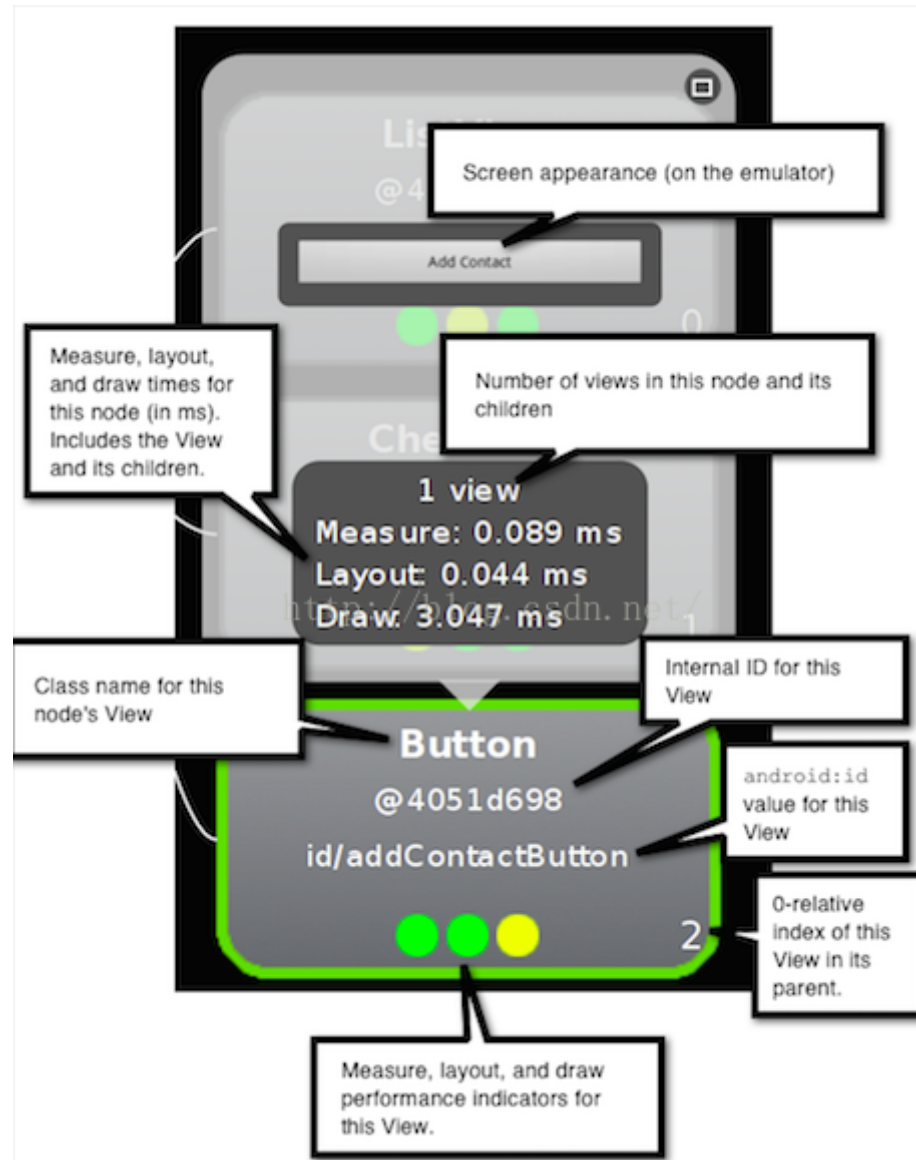


(6) 观察单个view，选择单个view后会出现如下图所示图形：

这里会看到Measure、Layout、Draw的耗时。View节点中带有红色或黄色的点代表速度较定有问题，特别像是ViewGroup对象，View的子节点越多，结构越复杂，性能越差。只要绘制）最慢的View对象是哪个，这样你就能快速确定问题。

关闭





不合理的布局会使我们的应用程序UI性能变慢，HierarchyViewer能够可视化的角度直观地帮助我们优化布局设计。

HierarchyViewer是我们优化程序的工具之一，它是Android自带的非常有用的工具，可以帮助我们是对UI检视的利器。

关闭



核电池



十大电吹风



程序员薪资



呼叫中心系统



新款手机



阿米巴经营模



橡胶止水带



密集架

顶 踩

3

0

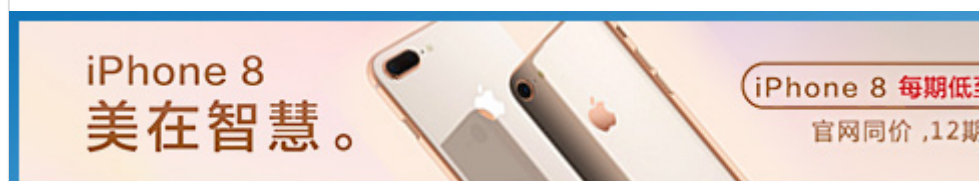
上一篇 Android StrictMode严格模式的使用方法

下一篇 Android Activity页面加载时间性能分析，以及改进要点

相关文章推荐

- Android Activity页面加载时间性能分析，以及改进...
- Python即将成为第一语言
- Android应用程序与SurfaceFlinger服务的关系概述...
- 构建企业级高性能OLAP引擎--董西成
- Android性能测试之fps获取
- JDK9新特性解读
- Android性能分析工具
- 华为工程师，带你实战C++
- Android的multidex带来的性能问题 - 减慢app启动
- Android自定义控件全知道
- android gpu呈现模式分析（文章写得美，仅仅用干
- TensorFlow入门
- RecyclerView加入
- RecyclerView加入
- dumpsys实现原理
- Android开发者选

关闭





核电池



十大电吹风



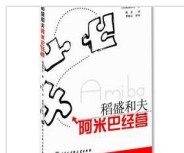
程序员薪资



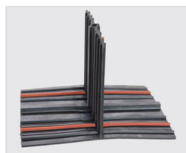
呼叫中心系统



新款手机



阿米巴经营模式



橡胶止水带



密集架

查看评论

暂无评论

您还没有登录,请[\[登录\]](#)或[\[注册\]](#)

* 以上用户言论只代表其个人观点,不代表CSDN网站的观点或立场

[联系方式](#) | [版权声明](#) | [法律顾问](#) | [问题报告](#) | [合作伙伴](#) | [论坛反馈](#)webmaster@csdn.net

400-660-0108

| 北京创新乐知信息技术有限公司 版权所有 | 江苏知之为计算机有限公司 | 江苏乐

17, CSDN.NET, All Rights Reserved



关闭

