


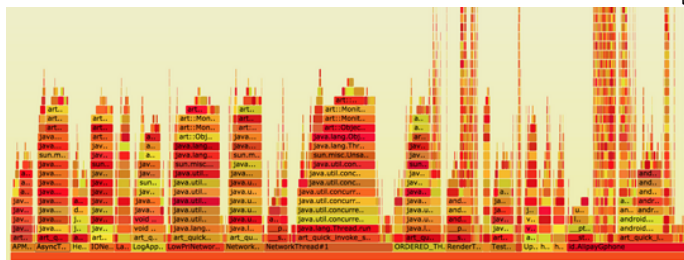


[首页](#) / [社区](#) / 另一个Android性能剖析工具——simpleperf

## 另一个Android性能剖析工具——simpleperf

 yellowbear 发布于7月前 阅读19020次 |  0 人点赞  0 条评论



谈到Android上的性能剖析，很多人会想到TraceView，SysTrace。TraceView擅长Method Tracing，与IDE集成，使用方便，不过有时候它的误差让人无法忍受；SysTrace的牛逼之处在于它可以给出整个系统的一些关键模块的性能信息，因此用途广泛；虽然它也支持分析自定义模块，但是使用起来稍显复杂。

事实上，在AOSP中有另外一个性能剖析工具不为人知，这个安静的小可爱就是今天要介绍的主角——simpleperf。我们先来看看它的介绍：

Simpleperf is a native profiling tool for Android. Its command-line interface supports broadly the same options as the linux-tools perf, but also supports various Android-specific improvements.

**native profiling** ？是不是很心动？



yellowbear

 0人点赞

 收藏

共有0 条评论，0人收藏

我要点评

被 729人关注，获得了178个喜欢

+关注

至于simpleperf的原理，简单来说，现代CPU一般都带有一个叫做性能监视单元（PMU）的组件，这个硬件能够记录诸如cpu周期数、执行的指令数、缓存失效次数等等关键信息；Linux内核对这个硬件做了一层封装，通过`perf\_event\_open`系统调用把接口暴露给用户空间；这就是simpleperf工具的由来。

简介和原理都表明这个工具貌似很牛逼，接下来我们就试一试。

首先把AOSP上的源码下载下来，点我。

simpleperf是一个命令行工具，与systrace不同的是，它的工具集包涵client端和host端；client端运行在Android系统上，负责收集性能数据；host端则运行在你的开发机上，负责对数据进行分析 and 可视化。（这些可执行文件在下载后的bin文件夹的android和win/mac/linux下）整个工具链使用起来相对复杂，所幸simpleperf的创造者提供了一个炒鸡傻瓜的使用脚本，使用起来毫不费力，非常方便！这个脚本就是`app\_profile.py`。

我们把 simpleperf 下载下来之后，首先修改`app\_profile.config`文件中的配置，主要是以下几个字段：



```
# 待分析的app的包名，需要是debuggable的
app_package_name = ""

# 分析的命令，体验的话使用默认即可；更详细的用途看文档
record_options = "-e cpu-cycles:u -f 4000 -g"

# 待分析app的主界面，app_profile.py会通过am start
main_activity = ''
```

然后在你的设备上安装待分析的App，要注意的是这个App必须是debuggable的，在AndroidManifest中把debuggable设置为true即可。接下来，就可以直接通过：

```
python app_profile.py
```

进行分析了，这个脚本帮你完成了simpleperf下载，权限设置，打开app，导出数据等一系列过程；开始分析之后，就可以对App进行操作；在设置好的时间之后就会停止分析，跟SysTrace一样。分析完毕之后，脚本会自动把分析数据从设备上pull下来并存放在当前目录下，叫 perf.data。

接下来就是分析数据了；原始的数据是一个文本文件，长这样：



♡ 0人点赞

🔖 收藏

共有0 条点评，0人收藏

我要点评



```

0.00%      0.00%  ORDERED_THREAD_  17046  18
|
-- art::Mutex::ExclusiveLock(art::Thr
0.00%      0.00%  AsyncTaskExecut  17046  18
|
-- art::Monitor::Lock(art::Thread*)
|
|--52.01%-- art::Mutex::Exclusive
|
--0.54%-- art::Mutex::ExclusiveU

```

嵌套如果过深，基本就看不懂了；所幸我们有另外一个分析脚本！直接运行：

```
python report.py -g
```

会启动一个GUI显示分析得到的数据，这个GUI使用python的tk写的，实话说，长得比较丑。。

简单观察分析图，可以知道，的确支持native profiling；gui和TraceView差不多，比较直观，但是指标没有TraceView多。另外，这个工具由于硬件直接支持，对性能的影响非常小；光是这一点就好顶赞了。



♡ 0人点赞

🔖 收藏

共有0 条评论，0人收藏

我要点评



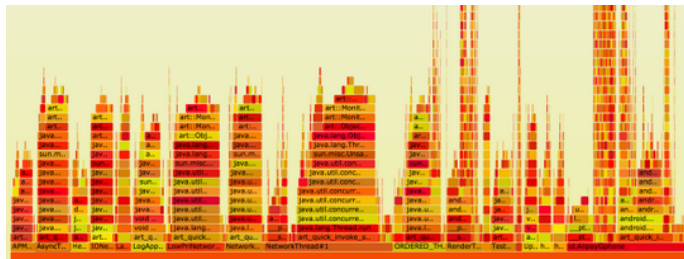
你以为这就完了？No！这玩意儿还支持 \*\*火焰图\*\*！如果你不知道火焰图为何物，建议看看这篇文章 [动态追踪技术漫谈](#)。

然后，要绘制火焰图，我们需要借助 FlameGraph 这个工具；clone下来之后，里面的脚本就可以直接使用了。（perl脚本，windows系统需要安装Perl语言的支持）

接下来我们执行命令：

```
$python report_sample.py >out.perf
$stackcollapse-perf.pl out.perf >out.folded
$./flamegraph.pl out.folded > graph.svg
```

我们用Chrome浏览器打开这个 `graph.svg` 即可得到火焰图：



通过火焰图，我们能非常直观地看到性能的瓶颈，对于分析性能问题很有帮助；相信这个图已经说明一切了。

当然 simpleperf 远非这么简单，我只是带大家尝尝鲜；对它感兴趣的，可以执行 `simpleperf -h` 结合文档去探索它的奥妙；祝大家玩得开心~



♡ 0人点赞

🔖 收藏

共有0 条点评，0人收藏

我要点评



查看原文： 另一个Android性能剖析工具——simpleperf

⬅ 上一篇

下一篇 ➡

需要 [登录](#) 后回复方可回复, 如果你还没有账号你可以 [注册](#) 一个帐号。

#### 🌐 最新项目

Android Clean Architecture Components样板

SwitchButton一个扁平化多项选择器tabs，切换按钮

自定义进度条，仿淘宝抢购进度，progressbar

KMedia|视频|浮窗|全屏|横竖屏切换|字幕|拖动|手势调整|AB...

ExtendEditText一个EditText扩展，支持一键清除、内容匹配

CircleProgress 一个环形进度条

Android常用工具类

实现RecyclerView，NestedScrollView滑动标题栏渐变

#### 🏠 相关主题



💗 0人点赞

🔖 收藏

共有0 条点评，0人收藏

我要点评

