

## Profile 工具系列之四：simpleperf

原创

2017年05月22日 08:41:23

标签：profile / simpleperf / Android

600



0

### 关于 simpleperf

- simpleperf 是 Google 随 NDK 一起发布的一款 profile 工具，从 NDK r13 开始
- 官方文档请参考 Google 在 NDK 里放的 README

### HelloWorld 步骤

- 把 simpleperf 可执行程序 push 到手机上。simpleperf 在 NDK-r13b 的 `simpleperf/android/` 里。根据被测手机的 CPU 选择对应版本：

```
1 adb push simpleperf /data/local/tmp/  
2 adb shell chmod 777 /data/local/tmp/simpleperf
```

- 启动被测程序，ps 出它的进程 ID
- 用 simpleperf 进行 profile：

```
1 adb shell /data/local/tmp/simpleperf record -p 3972 --duration 30
```

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！



old\_man

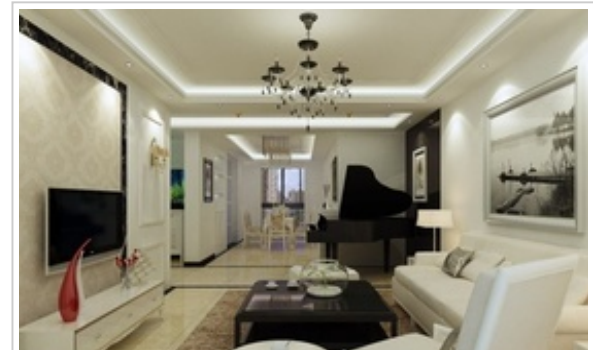
原创  
11粉丝  
2喜欢  
0评论  
0

等级：博客 2

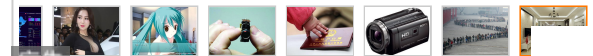
访问量：4449

积分：150

排名：112万+



### 室内装修效果图大全



他的最新文章

更多

登录

注册

```
1 simpleperf E 04-19 15:09:29 4109 4109 record_file_writer.cpp:47] failed t
```

- 用 -o 参数指定输出文件解决问题：

```
1 data/local/tmp/simpleperf record -p 3972 --duration 30 -o /sdcard/perf.data
```

- 得到警告如下：

```
1 simpleperf I 04-19 15:17:47 4299 4299 environment.cpp:504] Access to kern
2 If possible, please do `echo 0 >/proc/sys/kernel/kptr_restrict` to fix this
```

- 暂且不去理会这个警告，30 秒过后采集结束，得到如下输出：

```
1 simpleperf I 04-19 15:18:19 4299 4299 cmd_record.cpp:346] Samples recorded
```

- 看起来一切正常，把结果 report 一下看看：

```
1 adb shell /data/local/tmp/simpleperf report -i /sdcard/perf.data -n --sort
```

- 得到结果如下：

```
1 simpleperf W 04-19 15:31:17 4564 4564 dso.cpp:274] /data/local/rvdecApp d
2 simpleperf W 04-19 15:31:17 4564 4564 dso.cpp:335] Symbol addresses in /p
```

Profile 工具系列之二：gperftools

基于 Binder 的跨进程通信以及 Service  
(二)：Java 层

基于 Binder 的跨进程通信以及 Service  
(一)：Native 层

zygote 和 app 的启动过程

## 文章分类

Android	5篇
Profile	4篇

## 文章存档

2017年5月	10篇
2010年3月	2篇

## 他的热门文章

在winXP下返回当前线程所在CPU（多核）的方法

📖 933

Profile 工具系列之四：simpleperf

📖 500

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！

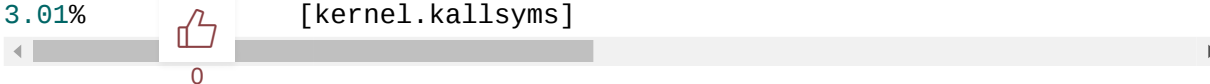
登录

注册

```

5 Event: cpu-cycles (type 0, config 0)
6 Samples: 125526
7 Event count: 43633757353
8
9 Overhead  Sample  Shared Object
10 88.93%    106529  /data/local/rvdecApp
11 8.05%     10560  /system/lib/libc.so
12 3.01%      0     [kernel.kallsyms]

```



- 其中的 `--sort dso` 用来指定结果显示哪些列，我们这里只写了一个 `dso`（即 dynamic shared object），所以只显示一列“Shared Object”，而且按 `dso` 分类，结果也只有三行而已。
- 如果不加 `--sort`，默认显示这几列：Command，Pid，Tid，Shared Object，Symbol，相当于：

```
1 --sort comm,pid,tid,dso,symbol
```

- `-n` 参数用来显示 Sample 那列，表示该行共命中了多少个 Sample，加不加随意。
- 可以看到，百分之 88.93 的时间都耗费在了我们的被测程序上，这是预期中的。
- 看一下我们 app 内部，那些函数占的比重最大：

```
1 adb shell /data/local/tmp/simpleperf report -i /sdcard/perf.data --dsos /da
```

- 结果如下：

```
1 impleperf W 04-19 15:57:34 5046 5046 dso.cpp:274] /data/local/rvdecApp do
```

404

Profile 工具系列之三：oprofile（Android，未成功）

387

Profile 工具系列之一：gprof

364

vs.net 2005, 没有找到MSVCR80D.dll的解决方案

233

JNI 原理进阶

156

基于 Binder 的跨进程通信以及 Service（一）：Native 层

135

欢迎使用CSDN-markdown编辑器

133

zygote 和 app 的启动过程

119

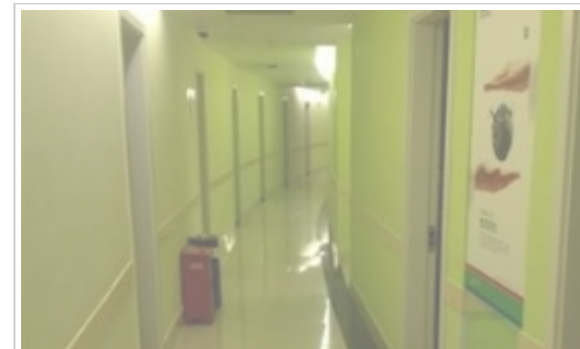
```

5 Event: cpu-cycles (type 0, config 0)
6 Samples: 106529
7 Event count: 38804869540
8
9 Overhead  Sample  Symbol
10 5.06%      5373    rvdecApp[+24380]
11 4.57%      4890    rvdecApp[+24420]
12 1.43%      999      rvdecApp[+13a44]
13 1.01%      0        rvdecApp[+21f94]
14 0.94%      999      rvdecApp[+20188]
15 ...

```

- 其中的 `-dsos` 是 simpleperf 的 5 个 filter 之一，意思是按照指定的 dynamic shared objects 进行过滤，只显示参数指定的 dso 里面的结果。全部 5 个 filter 是：

- `-comms`：按 command 过滤，比如：`--comm rvdecApp`
- `-pids`：按 pid 过来
- `-tids`：按 tid（线程id）过滤
- `-dsos`：按库文件/可执行文件名过滤
- `-symbols`：按函数名过滤，比如：`--symbols "RVComFunc::getPUMVPredictor(RefBlockInfo*, unsigned int, int, int, unsigned int)"`，注意函数里有空格的，需要用双引号引起来。
- 可以看到，结果里没有函数名字。那是因为我们的 rvdecApp 是没有符号表的版本。我们用带符号表的 app 进行分析即可。
- 带符号表的 app 可执行文件可以在 obj 目录下找到。把它 push 到手机上，覆盖原来的可执行文件。



## 个人公寓出租



## 联系我们



请扫描二维码联系客服

✉ webmaster@csdn.net

☎ 400-660-0108

🗣 QQ客服 🗣 客服论坛

关于 招聘 广告服务 百度

©1999-2018 CSDN版权所有

京ICP证09002463号

经营性网站备案信息

网络110报警服务

中国互联网举报中心

北京互联网违法和不良信息举报中心

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！

登录

注册

- 还用刚才的命令：

```
1 l /data/local/tmp/simpleperf report -i /sdcard/perf.data -n --dsos /data/lo
```

- 得到如下结果：

1	Overhead	Symbol
2	10.45%	NEON_ChromaInterpolate4_H00V00_8
3	5.94%	NEON_LumaInterpolate4_H03V03
4	5.10%	RVComFunc::DBFShiftedProcess8x8(unsigned char**, int*, un
5	3.55%	RVComFunc::deblockCUTree(TCBDataTree*, unsigned char**, u
6	3.49%	RVComFunc::reconstructInterPredictors(TCUData*, unsigned
7	3.35%	RVComFunc::deriveDBFStrengthFUbyMotionInfo(unsigned char*
8	2.98%	Decoder::parseBitStream_FrameNew()
9	2.79%	NEON_DBF_EdgeFilter4_Vertical
10	2.52%	RVComFunc::DBFShiftedProcessFu(unsigned char**, int*, uns
11	2.36%	(anonymous namespace)::decode_gen_vlc(unsigned long const
12	...	

- 不覆盖被测程序也可，但是要明确告诉 simpleperf 去哪个可执行里读取符号表和debug信息，不然 simpleperf 默认是去 record 时的那个可执行文件里读的。这个参数是：--symfs，它后面跟一个目录。
- 比如我们刚才的被测程序是 /data/local/rvdecApp，我们把带有符号表的 rvdecApp 放到 /sdcard/data/local/rvdecApp，然后用 --symfs /sdcard 即可。注意，必须把 rvdecApp 放到一个含有 /data/local/ 的目录里，因为执行的时候是全路径，simpleperf 认为 /data/local/rvdecApp 整个是可执行文件。

## 1. 用 simpleperf 启动被测进程

- 可以用 simpleperf 启动被测进程。而不必先把被测进程启动，然后 ps 出进程号再采集。其命令行格式如下：

```
1 adb shell /data/local/tmp/simpleperf record -o /sdcard/a.log /data/local/rvdecApp
```



- 其中 `/data/local/rvdecApp` 是被测 app 的可执行文件，后面跟的都是该 app 的启动参数
- 如此启动的 simpleperf，将在被测进程运行结束后停止采集并退出。

## 2. cache-misses



- 默认采集的是 `cpu-cycles`，所以我们默认得到的结果都是“CPU使用率”



```
1 As the perf event is cpu-cycles, the overhead can be seen as the percentage
```

- 用 `-e` 参数可以指定要采集哪些 event，我们用 `-e cache-misses` 即可以采集 cache-miss

```
1 adb shell /data/local/tmp/simpleperf record -o /sdcard/a.log -e cache-misses
```

## 其他

- 不使用 `-duration` 参数指定采集时间，当被采集进程退出时，simpleperf 不会自动停止采集，而是会报告如下的log：



0

目前您尚未登录，请 [登录](#) 或 [注册](#) 后进行评论

## 笔记



shichaog 2017年08月02日 11:06 346

1.tensorflow源码完整下载  
git clone --recurse-submodules https://github.com/tensorflow/tensorflow.git参数--...



## Profile 工具系列之三：oprofile（Android，未成功）

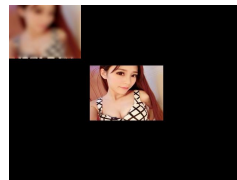
写在前面 本文研究了一下用 oprofile 在 Android 做 profile 之后，最终弃用 原因之一是编译内核比较麻烦（oprofile 需要内核的配合）原因之二是从 Android N ...



zsl\_oo7 2017年05月28日 23:18 387

## 技术外文文献看不懂？教你一个公式秒懂英语

不背单词和语法，一个公式学好英语



加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！

登录

注册

## Simpleperf介绍



tq08g2z

2017年08月17日 10:38

121

什么是simpleperf Simpleperf是Android平台的一个本地层性能分

析工具。它的命令行界面支持与linux-tools perf大致相同的选项，但是它还支持许多Android特有的改...

## Profile 工具系列之四：simpleperf



zsl\_oo7

2017年05月29日 08:41

600

关于 simpleperf simpleperf



0

google 随 NDK 一起发布的一款 profile 工具，从 NDK r13 开始 官方文档请参考 google

e 在 NDK 里放的 REA...

## Linux下的系统性能调优工具——Perf



dreamcoding

2012年07月24日 22:09

8120

参考以下链接文章整理



-- Linux下的系统性能调优工具，第 1 部分 Perf -- Linux下的系统性能调优工具，第 2

部分 感谢原作者，让我知道了性能调优也是如此有趣，L...



## IT人一分钟秒懂升职加薪的奥义

动动鼠标就能实现的新购满额减再返最高6000



## “Ceph浅析”系列之四Ceph的结构



qq\_24916117

2015年12月30日 20:42

591

本文将从逻辑结构的角度对Ceph进行分析。 4.1 Ceph系统的层次结构

Ceph存储系统的逻辑层次结构如下图

所示[1]。 自下向上，可以将Ceph系统...

## HDU 不容易系列之(4)——考新郎（组合错排）

题目：http://acm.hdu.edu.cn/showproblem.php?pid=2040 思想基础：全部三素错排参见http://blog.csdn.net/liuziyan

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！

登录

注册





liuweiyuxiang 2016年04月03日 23:43 807

## (OK) 编译 cm-13-kiwi for (华为 荣耀 5X) - 成功 - and 编译输出

(OK) 编译 cm-13-kiwi for (华为 荣耀 5X) - 成功 - and 编译输出



ztguang 2017年03月09日 17:16 2444



0

## GOMOBILE写Android和iOS移动端SDK



winlinvip 2017年03月09日 11:05 2451

GOMOBILE设置GO环境以及GOPATH，参考GO环境设置。GOMOBILE资料参考官方WIKI，这个WIKI貌似对SDK和NDK没有说清楚。Android+NDK下载和安装Android...



## 示波器基础系列之四——关于示波器的触发功能（下篇）

上篇中我们谈到了触发的一些基本概念。下篇我们首先总结下触发功能的含义，然后对各种触发方式做简单解释。

触发功能：示波器的触发功能主要有两点，第一，隔离感兴趣的事件。第二，同步波形...



sfrysh 2014年07月14日 16:54 944

## 深入理解Spring系列之七：web应用自动装配Spring配置

在《深入理解Spring系列之一：开篇》的示例代码中使用如下方式去加载Spring的配置文件并初始化容器。ApplicationContext applicationContext = new Cla...



tianrui 2017年08月10日 20:42 314

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！

登录

注册

## 3月福利：免费体验40+阿里云产品>

阿里云，阿里巴巴集团旗下云计算品牌，全球卓越的云计算技术和服务提供商。



## Android 性能优化 知必会



qq\_23892379

2018年02月28日 13:21

14

文章目录 优化心得和经验 响应速度 流畅度 内存 图形栈 虚拟机 系统框架 进程管理 调试工具 硬件相关 编程语言 Kernel 我辈楷模 做了这么久性能相关的工作,也接触了不少模块,说实话要做...



## Android 7.1CTS 问题总结



qq\_30427341

2017年04月28日 11:53

1431

CTS测试案例 状态 [CTS 7.1 r2] om.android.cts.devicepolicy.DeviceOwnerTest #testHardwarePro...



## C#游戏编程：《控制台小游戏系列》之《三、游戏输入模块》

一、输入模块结构 游戏输入模块包括对鼠标和键盘两种输入设备的处理，此模块主要为游戏框架提供鼠标操作和键盘操作能力，这些功能均以事件形式提供从而达到模块间消息传递的目的。 ...



fa6507

2013年01月26日 20:30



3093

## [大洋] Unity3D架构系列之- FSM有限状态机设计一至四

Unity3D架构系列之- FSM有限状态机设计一 我们在游戏开发中经常面临架构设计问题，在蛮牛问答里面也有好多朋友问关于架构方面的问题，在这里我就将一些经常使用的游戏开发方面的知识跟大家...



hany3000

2015年02月14日 08:12



1237

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！

登录

注册

## 快来看，SATA秘籍惊现江湖！！！！



zhuzongpeng

2017年07月26日 11:37

📖 326

耗时半月，终于将SATA系列专题连载完毕。SATA专题总共

分为六个部分，14篇文章，一万余字。有时候对某个方面的理解，从读懂到写出来确实需要花费精力。在写这个SATA专题的过程中，古猫也是成长不少，之...

## 强化学习系列之[👍]型无关的策略学习



bbbeoy

2018年01月16日 10:27

📖 30

文章目录 [隐藏] 1. 一些前置话题 2. MC Control 3. SARSA 4. Q Learning 5. 做点实验 5.1. 算法稳定性 5.2. 贪婪策略...



## 室内装修效果图



装修效果图

百度广告



## windows程序员进阶系列：《软件调试》之四：断点和单步调试

windows程序员进阶系列：《软件调试》之四：断点和单步调试 断点和单步调试是两个经常使用的调试功能，也是调试器的核心功能。在第一篇文章中曾简单介绍了下，本文我们将会对它们进行更详细的介绍...



ithzhang

2013年02月03日 14:56

📖 15025

## 常用profile工具使用--采样和检测



feixiang\_john

2012年08月29日 14:33

📖 4326

做编解码开发少不了对代码的分析,下面分析常用的profile工具. 常用profile工具使用 1 概述1.1 profile工具的作用 profile工具一般用于收...

加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！

登录

注册

## hdu 2045 不容易系列之(3)——LELE的RPG难题

是一道递推找规律的题，首先易知 $f(1)=3; f(2)=6; f(3)=6; f(4)=18$ ; 现在考虑 $n>3$ 的情况，若第 $n-1$ 个格子和第一个格子不同，则为 $f(n-1)$ ；若第 $n-1$ 个格子和第1个格子相...



u013008291 2014年10月14日 22:07 1216



0



加入CSDN，享受更精准的内容推荐，与500万程序员共同成长！

登录

注册