



## Q博士的专栏

客户端三年，服务端1年，学习的路上

目录视图

摘要视图

RSS 订阅

【公告】关于开启用户注册及登录手机短信验证的通知

CSDN日报20170417 ——《程序员最好有 社保》

博客搬家，有礼相送

### Android性能专项测试之Batterystats

2015-10-14 11:24

7328人阅读

评论(3)

收藏

举报

分类：

测试[Android性能] ( 14 )

版权声明：本文为Doctorq原创文章，未经博主允许不得转载。

目录(?)

[+]

Batterystats & Battery Historian Walkthrough

Battery Historian Charts

Android应用的耗电量统计

SupplimentState

Viewing Battery Use Data

## batterystats使用条件

- 5.0以上系统
- battery-historian脚本

## batterystats的启动

启动统计Battery数据的话，首先要初始化batterystats数据

```
1 adb kill-server
2
3 > adb devices
4
5 > adb shell dumpsys batterystats --reset
```

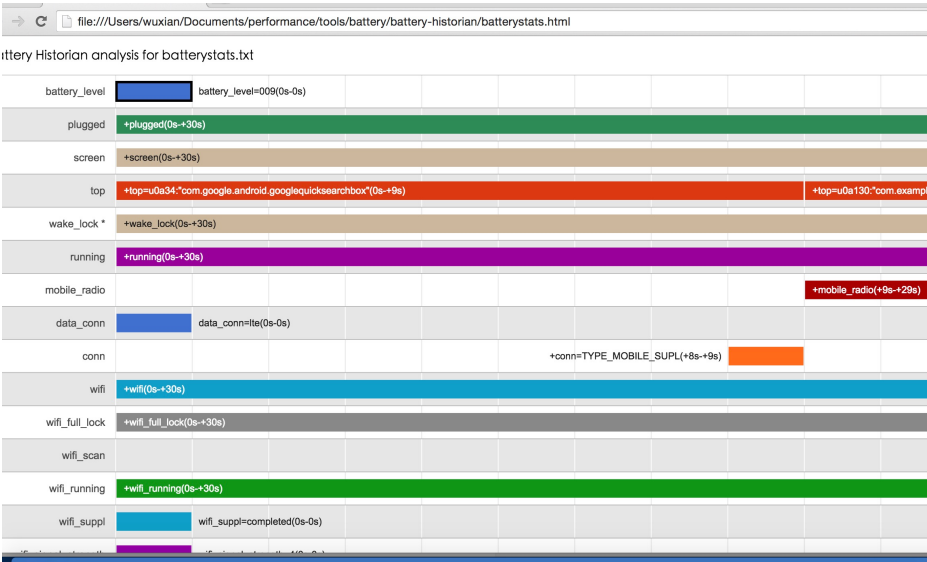
上面的操作执行完毕后，拔掉手机，操作你的App，操作完成后，重新连接手机，执行下面的命令，收集Battery数据:

adb shell dumpsys batterystats > batterystats.txt

得到这些数据后，这个时候使用我们的 battery-historian 来生成我们可见HTML报告:

Python historian.py batterystats.txt > batterystats.html

得到的报告如下:



## Battery Historian指标

个人资料



DoctorQ

关注

发私信



访问：1016293次

积分：15084

等级：BLOG > 7

排名：第621名

原创：479篇      转载：1篇

译文：9篇      评论：512条

联系方式

如果想了解我更多，请点击



文章搜索

🔍

博客专栏



Android安全专项测试

文章：8篇

阅读：21035



接口测试

文章：9篇

阅读：25299



Android性能专项测试

文章：11篇

阅读：52281



react-native试玩

文章：36篇

阅读：74695



UI Testing in Xcode7

文章：8篇

阅读：13603



gradle

文章：31篇

阅读：67408



Appium之android平台的源码分析

文章：17篇

阅读：34668



Cts框架解析

文章：24篇

阅读：48999

在[Android性能专项测试之battery-historian试用](#) 一文中已经介绍了一些属性，但是目前又增加很多，下面来看看新加属性的意义：

属性	意义
gps	gps是否开启
sync	是否跟后台同步
mobile_radio	是否开启radio
wake_reason	唤醒原因
phone_in_call	进行通话

## batterystats信息过滤

### Battery History

电量相关数据的时间序列：

```
1 Battery History (0% used, 152 used of 256KB, 5 strings using 256):
2   0 (9) RESET:TIME: 2015-10-09-19-50-15
3   0 (2) 009 status=charging health=good plug=usb temp=324 volt=3683 +running +wake_lock +wifi_full_lock +audio +screen +plugge
4   0 (2) 009 user=0:"0"
5   0 (2) 009 userfg=0:"0"
6   +189ms (2) 009 volt=3660
7   +216ms (2) 009 volt=3688
8   +8s387ms (2) 009 +gps conn=3:"CONNECTED"
9   +9s387ms (2) 009 -top=u0a34:"com.google.android.googlequicksearchbox"
10  +9s387ms (2) 009 -gps +top=u0a130:"com.example.android.sunshine.app"
11  +9s569ms (2) 009 +mobile_radio conn=3:"DISCONNECTED"
12  +10s217ms (2) 009 volt=3577
13  +17s294ms (2) 009 volt=3619
14  +20s216ms (2) 009 volt=3576
15  +21s530ms (1) 009 +wifi_scan
16  +22s801ms (1) 009 -wifi_scan
17  +29s842ms (1) 009 -mobile_radio
18  +30s252ms (2) 009 volt=3616
```

### Per-PID Stats

每一个进程执行了多长的时间

```
1 Per-PID Stats:
2 PID 0 wake time: +396ms
3 PID 868 wake time: +1s682ms
4 PID 1402 wake time: +539ms
5 PID 0 wake time: +6s473ms
6 PID 868 wake time: +3ms
7 PID 1723 wake time: +166ms
8 PID 4841 wake time: +11m1s479ms
9 PID 19942 wake time: +271ms
10 PID 868 wake time: +42ms
11 PID 21564 wake time: +172ms
12 PID 15488 wake time: +67ms
13 PID 4841 wake time: +6s512ms
14 PID 4841 wake time: +6s512ms
15 PID 18217 wake time: +126ms
16 PID 3876 wake time: +228ms
17 PID 868 wake time: +20s440ms
18 PID 16547 wake time: +23ms
19 PID 22121 wake time: +288ms
20 PID 22146 wake time: +59ms
21 PID 16424 wake time: +485ms
```

### Statistics since last charge

```
1 Statistics since last charge:
2 System starts: 0, currently on battery: false
3 Time on battery: 0ms (0.0%) realtime, 0ms (0.0%) uptime
4 Time on battery screen off: 0ms (0.0%) realtime, 0ms (0.0%) uptime
5 Total run time: 34s 134ms realtime, 34s 134ms uptime
6 Start clock time: 2015-10-09-19-50-15
7 Screen on: 0ms (--%) 0x, Interactive: 0ms (--%)
8 Screen brightnesses: (no activity)
9 Connectivity changes: 2
10 Mobile total received: 0B, sent: 0B (packets received 0, sent 0)
11 Phone signal levels: (no activity)
12 Signal scanning time: 0ms
13 Radio types: (no activity)
14 Mobile radio active time: 0ms (--%) 0x
15 Mobile radio active adjusted time: 0ms (--%)
16 Wi-Fi total received: 0B, sent: 0B (packets received 0, sent 0)
17 Wifi on: 0ms (--%), Wifi running: 0ms (--%)
18 Wifi states: (no activity)
19 Wifi supplicant states: (no activity)
20 Wifi signal levels: (no activity)
```

```
21 Bluetooth on: 0ms (--%)
22 Bluetooth states: (no activity)
23
```

### Estimated power use (mAh)

目前只能算比较粗略的估计，不能作为实际消耗数据，期待google更为精准的数据

```
1 Estimated power use (mAh):
2 Capacity: 3220, Computed drain: -2444.11370879, actual drain: 0.00000000
3 Idle: 24.7
4 Uid u0a46: -0.01197222
5 Uid u0a100: -0.04600000
6 Uid u0a29: -0.07444444
7 Uid u0a118: -0.14044444
8 Uid u0a104: -0.73288889
9 Uid 1000: -2.12172492
10 Uid u0a14: -3.36807500
11 Uid u0a99: -3.78076692
12 Phone calls: -19.54370833
13 Cell standby: -107.36537361
14 Screen: -2331.64616000
```

### Per-app mobile ms per packet

radio唤醒时间除以数据包发送次数，该时间越小代表性能越好，一个高效的App会将所有的流量转化为batches。

### All partial wake locks

所有程序中wakelock，关注持续时间和数量

## 6.0的更新

试用了下6.0的batterystats，现在信息比5.0的更加全面，不但详细列举了不同app的耗电量，还把app中各个子模块的耗电量也输出了：

```
1 Estimated power use (mAh):
2 Capacity: 3220, Computed drain: 1163, actual drain: 451-644
3 Screen: 361
4 Uid u0a99: 189 ( cpu=13.2 wake=0.000408 radio=102 wifi=6.35 gps=67.8 )
5 Uid u0a94: 127 ( cpu=6.99 wake=0.00528 radio=118 wifi=0.451 gps=0.181 sensor=0.836 )
6 Uid 0: 111 ( cpu=30.2 wake=5.13 radio=75.0 wifi=0.242 )
7 Cell standby: 79.8 ( radio=79.8 )
8 Uid u0a95: 63.7 ( cpu=10.9 wake=1.59 radio=50.0 wifi=0.0566 camera=1.15 )
9 Uid 1000: 61.1 ( cpu=59.9 wake=0.642 wifi=0.100 sensor=0.525 )
10 Uid u0a98: 22.2 ( cpu=0.261 wake=0.00395 radio=21.9 wifi=0.0817 )
11 Uid u0a100: 19.0 ( cpu=10.4 wake=0.00502 radio=7.90 wifi=0.624 gps=0.00150 sensor=0.0554 )
12 Uid u0a35: 14.3 ( cpu=14.0 wake=0.158 wifi=0.0804 )
13 Uid u0a15: 14.1 ( cpu=3.18 wake=1.72 radio=8.71 wifi=0.466 )
14 Uid u0a108: 13.9 ( cpu=3.23 radio=10.6 wifi=0.0228 )
15 Uid u0a103: 12.2 ( cpu=2.79 wake=0.00260 radio=8.11 wifi=0.811 gps=0.291 sensor=0.187 )
16 Uid u0a110: 10.8 ( cpu=0.910 radio=9.92 wifi=0.00682 )
17 Uid u0a105: 10.6 ( cpu=5.59 radio=4.95 wifi=0.0642 )
18 Idle: 9.40
```

顶

2

踩

0

- 上一篇 Android性能专项测试之Heap Snapshot工具
- 下一篇 Android性能专项测试之耗电量统计API

#### 我的同类文章

测试[Android性能]（ 14 ）					
• 心向百度	2016-04-12	阅读 3909	• Android内存泄漏检测-LeakCanary	2015-10-23	阅读 2915
• Android性能专项测试之耗电量统计API	2015-10-19	阅读 5829	• Android性能专项测试之Network monitor	2015-10-09	阅读 3192
• Android性能专项测试之GPU Monitor	2015-10-09	阅读 5320	• Android性能专项测试之Systrace工具	2015-10-08	阅读 4598
• Android性能专项测试之MAT	2015-10-05	阅读 2289	• Android性能专项测试之TraceView工具(...	2015-09-30	阅读 2810
• Android性能专项测试之Heap Snapshot...	2015-10-09	阅读 4432	• Android性能专项测试之Allocation Trac...	2015-09-26	阅读 5334
更多文章					

猜你在找

移动手机APP测试从零开始（初...  
HTML5开发手机App之：HTM...  
HTML5开发手机App之：CSS...  
hbuilder开发移动app视频教程,...  
Android开发高级组件与框架—...

Android-NDK错误 undefined ...  
android-opencv undefined r...  
性能测试分析  
集成 TerichDB 的 MongoDB ...  
集成TerichDB的SSDB性能测试

查看评论

YangRou\_Chuan

3楼 2016-09-18 16:35发表

batterystats.bin 在代码中哪创建

qq\_35312033

2楼 2016-06-14 16:45发表

请教下，dumpsys batterystats |grep Uid 取出来的值；其中的uid怎么和app在系统分配的uid对应起来？

Mind-Hacks

1楼 2015-10-14 2

在CSDN首页发现自己学校的学长，学长加油！！！！

发表评论

用户 名：

haijunz

评论内容：

提交

\* 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

核心技术类目

全部主题

Hadoop

AWS

移动游戏

Java

Android

iOS

Swift

智能硬件

Docker

OpenStack

VPN

Spark

ERP

IE10

Eclipse

CRM

JavaScript

数据库

Ubuntu

NFC

WAP

jQuery

BI

HTML5

Spring

Apache

.NET

API

HTML

SDK

IIS

Fedora

XML

LBS

Unity

Splashtop

UML

components

Windows Mobile

Rails

QEMU

KDE

Cassandra

CloudStack

FTC

coremail

OPhone

CouchBase

云计算

iOS6

Rackspace

Web App

SpringSide

Maemo

Compuware

大数据

apttech

Perl

Tornado

Ruby

Hibernate

ThinkPHP

HBase

Pure

Solr

Angular

Cloud Foundry

Redis

Scala

Django

Bootstrap

公司简介 | 招贤纳士 | 广告服务 | 联系方式 | 版权声明 | 法律顾问 | 问题报告 | 合作伙伴 | 论坛反馈

网站客服

杂志客服

微博客服

webmaster@csdn.net

400-600-2320 | 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 | 江苏乐知网络技术有限公司

京 ICP 证 09002463 号 | Copyright © 1999-2016, CSDN.NET, All Rights Reserved



4 of 4

2017年04月18日 15:14