移动端专项测试

性能测试

性能是衡量APP质量的一个重要指标

APP性能测试常见指标：

> 内存、CPU、流量、电量、启动速度、流畅度等

性能测试关注点

> APP使用时对CPU、内存的占用情况； APP使用时是否流畅等
>
> APP使用时，电量流量的消耗情况； APP的启动时间是否过长

1 GT调测平台

1.1 GT简介：

GT（随身调） Android版是腾讯 MIG 专项测试组自行研发的APP 随身调测平台，它是直接运行在手机上的“集成调测环境”(ITE,

Integrated Test Environment)，之所以叫“集成调测环境”，是因为仅用 GT即可独立完成如下针对 AUT 的测试工作：

基础性能测试：手机整机或者手机上安装的任何一个 APP 的 CPU、内存、网络流量、流畅度/帧率、电量等基础性能指标的实时展示、历史数据采集及 excel 格式存储、曲线绘制等。

日志查看：APP 的 Logcat 日志查看，便于直接用手机现场定位 APP 功能异常、crash。

网络数据包抓包：直接用手机抓包保存成 pcap 文件，下载到 PC 后用 Wireshark 查看。

1.2 GT安装：

GT 有两种版本：

（1） 可独立安装的 GT（APK，IOS无该版本），像普通APP一样安装。

（2） GT SDK：将GT的SDK嵌入到被调测的应用的工程里

1.3 GT使用：

注意：GT使用时，部分功能需要有root权限。

启动后点击永久记住选择 -- 点击允许

（1） 进入AUT界面，选择被测程序和测试指标。


GT
被调试的应用:
刷新
running
拉勾教育
com.lagou.education
选择测试应用程序
内存指标
内存助手:
off
PSS
Private Dirty
其他指标
CPU
jiffies
Net
WebView App
Xposed Installer
监控指标
GT

上图五项指标:

内存指标：

PSS： 实际使用物理内存

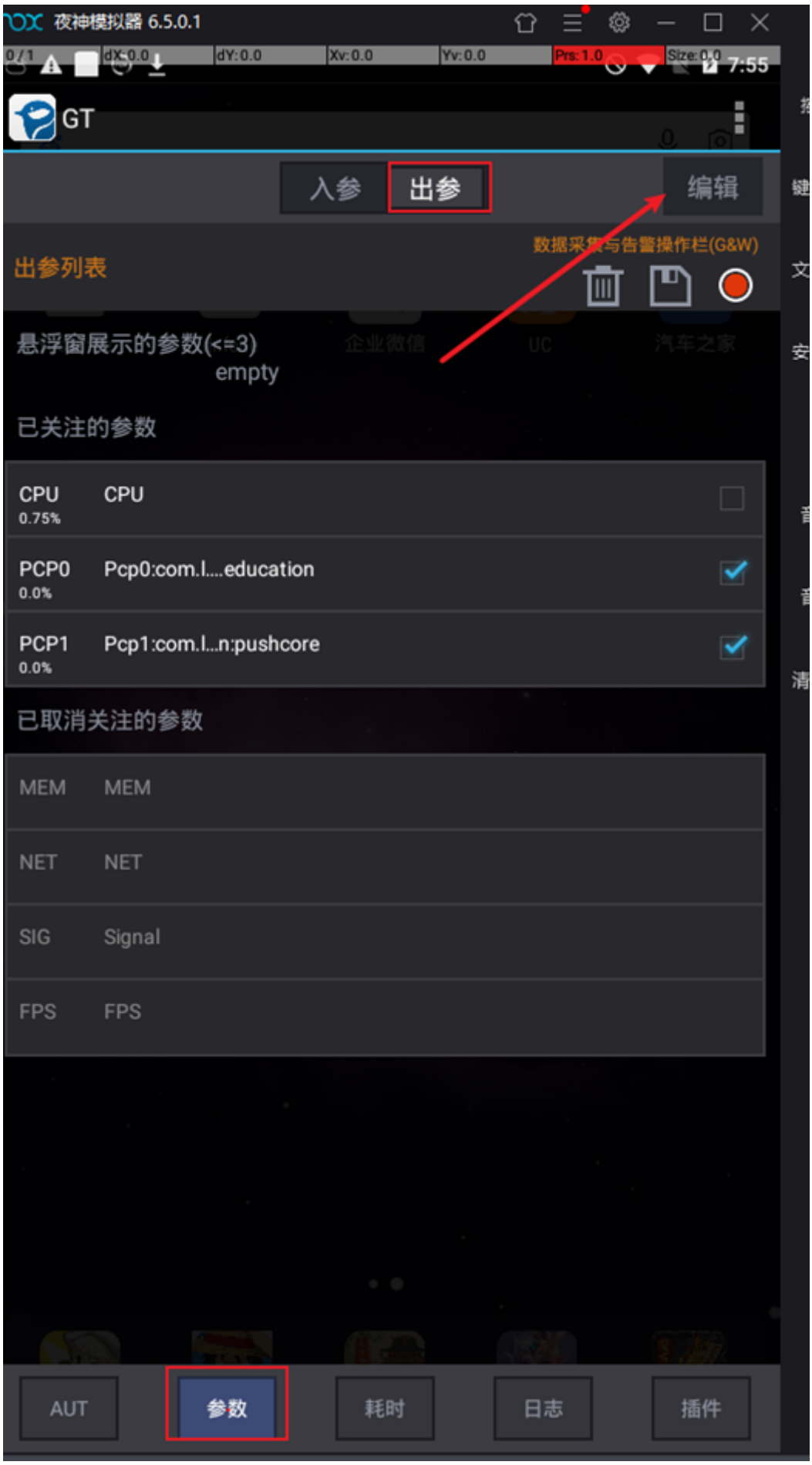
Private Dirty：APP私有内存

其他指标：

CPU：当前cpu使用率

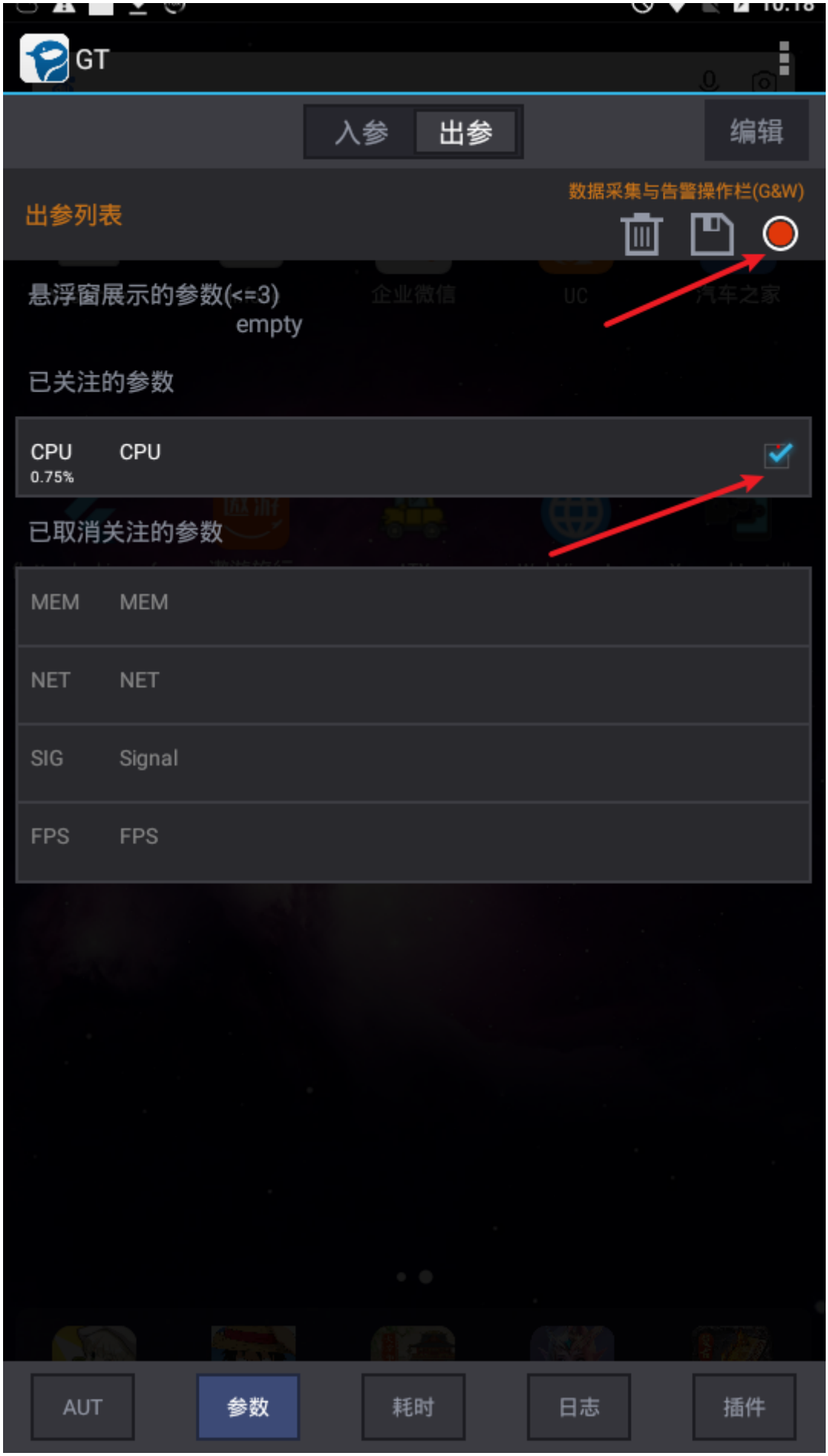
Net：整个手机产生的流量

jiffies：cpu时间片，用来记录系统子开机依赖过了多少tick，一定时间占用的jiffles反应进程cpu的消耗



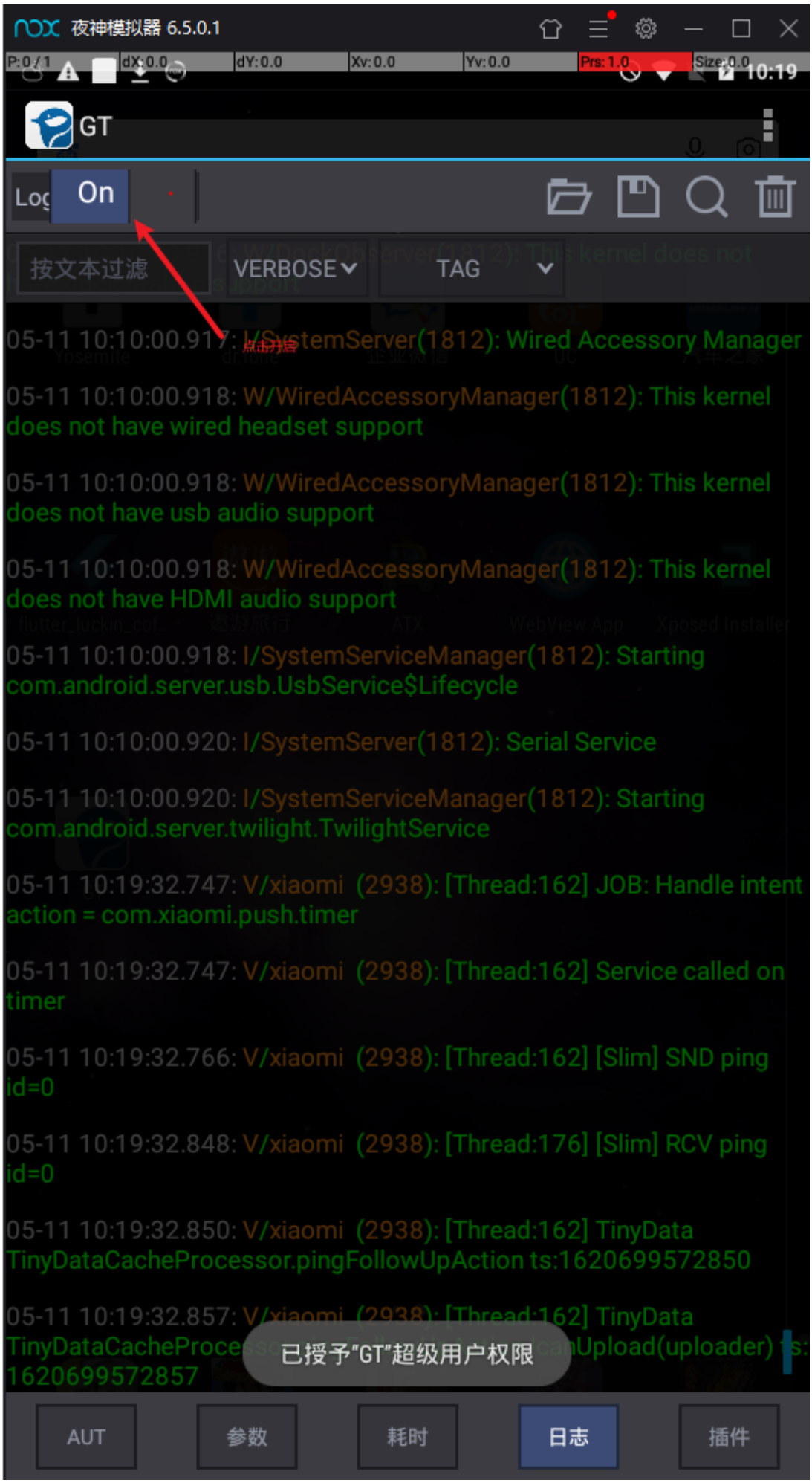
设置参数，点击右上角的“编辑”按钮，然后选中想测试的参数将其拖拽到已关注区域

点击“完成”按钮，勾选已关注的参数，点击右上角的红点即可开始监控



执行测试之前可以打开日志抓取，方便发生问题时定位。

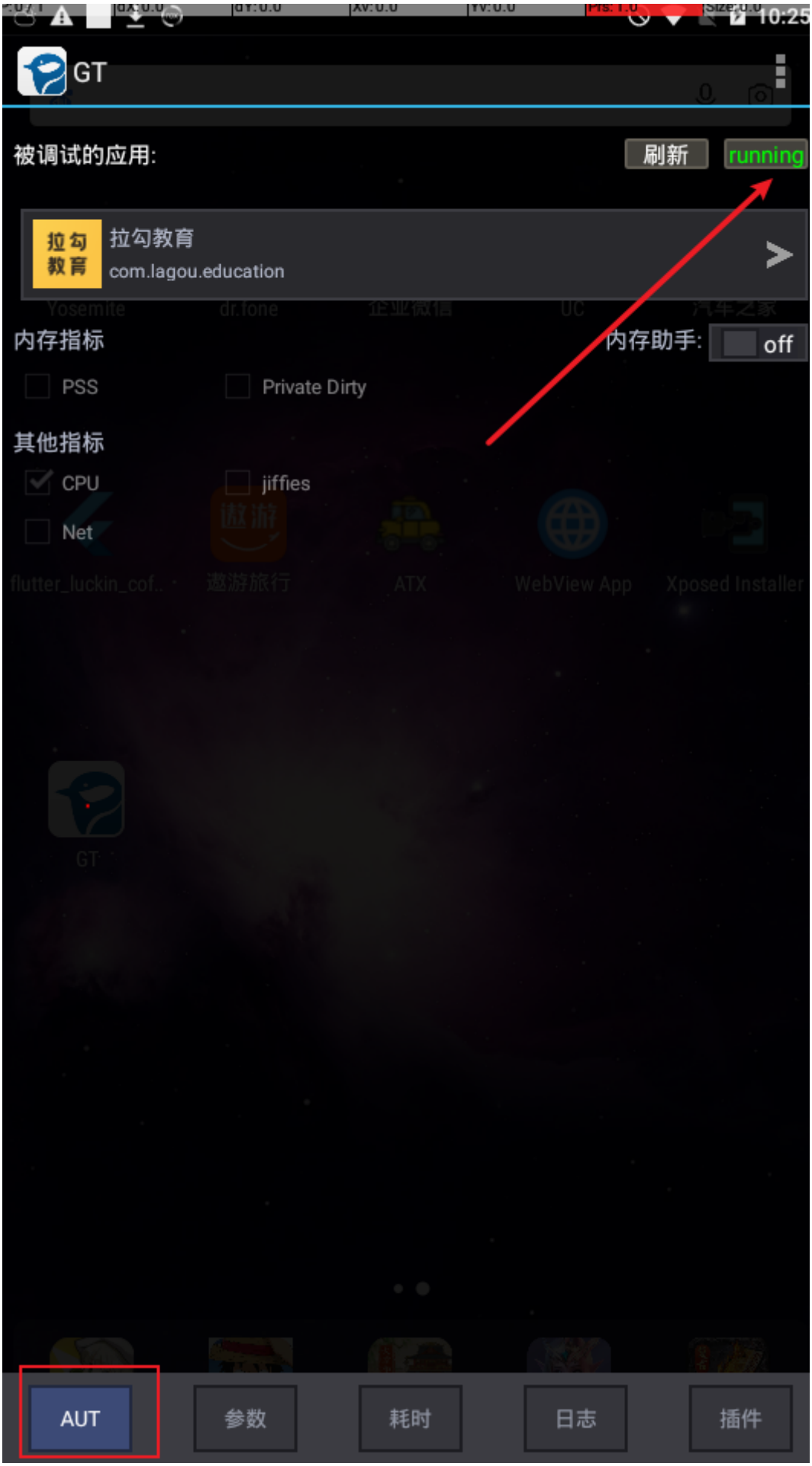

P:0/1
dX:0.0
dY:-119.9
Xv:0.0
Yv:0.959
Prs:1.0
Size:0.0
10:15
GT
入参
出参
完成
出参列表
拖拽
悬浮窗展示的参数(<=3)
empty
已关注的参数
CPU
CPU
已取消关注的参数
MEM
MEM
NET
NET
SIG
Signal
FPS
FPS
AUT
参数
耗时
日志
插件


夜神模拟器 6.5.0.1
P:0/1
dX:0.0
dY:0.0
Xv:0.0
Yv:0.0
Prs:1.0
Size:0.0
10:19
GT
On
按文本过滤
VERBOSE
TAG
05-11 10:10:00.917: I/SystemServer(1812): Wired Accessory Manager
05-11 10:10:00.918: W/WiredAccessoryManager(1812): This kernel does not have wired headset support
05-11 10:10:00.918: W/WiredAccessoryManager(1812): This kernel does not have usb audio support
05-11 10:10:00.918: W/WiredAccessoryManager(1812): This kernel does not have HDMI audio support
05-11 10:10:00.918: I/SystemServiceManager(1812): Starting com.android.server.usb.UsbService$Lifecycle
05-11 10:10:00.920: I/SystemServer(1812): Serial Service
05-11 10:10:00.920: I/SystemServiceManager(1812): Starting com.android.server.twilight.TwilightService
05-11 10:19:32.747: V/xiaomi (2938): [Thread:162] JOB: Handle intent action = com.xiaomi.push.timer
05-11 10:19:32.747: V/xiaomi (2938): [Thread:162] Service called on timer
05-11 10:19:32.766: V/xiaomi (2938): [Thread:162] [Slim] SND ping id=0
05-11 10:19:32.848: V/xiaomi (2938): [Thread:176] [Slim] RCV ping id=0
05-11 10:19:32.850: V/xiaomi (2938): [Thread:162] TinyData TinyDataCacheProcessor.pingFollowUpAction ts:1620699572850
05-11 10:19:32.857: V/xiaomi (2938): [Thread:162] TinyData
1620699572857
已授予"GT"超级用户权限
AUT
参数
耗时
日志
插件

点击开始监控按钮后，打开被测APP应用，悬浮窗会出现在你要测试应用上面

显示数据为收集数据

回到AUT页面点击running


10:25
GT
被调试的应用:
刷新
running
拉勾教育
com.lagou.education
内存指标
内存助手:
off
PSS
Private Dirty
其他指标
CPU
jiffies
Net
AUT
参数
耗时
日志
插件

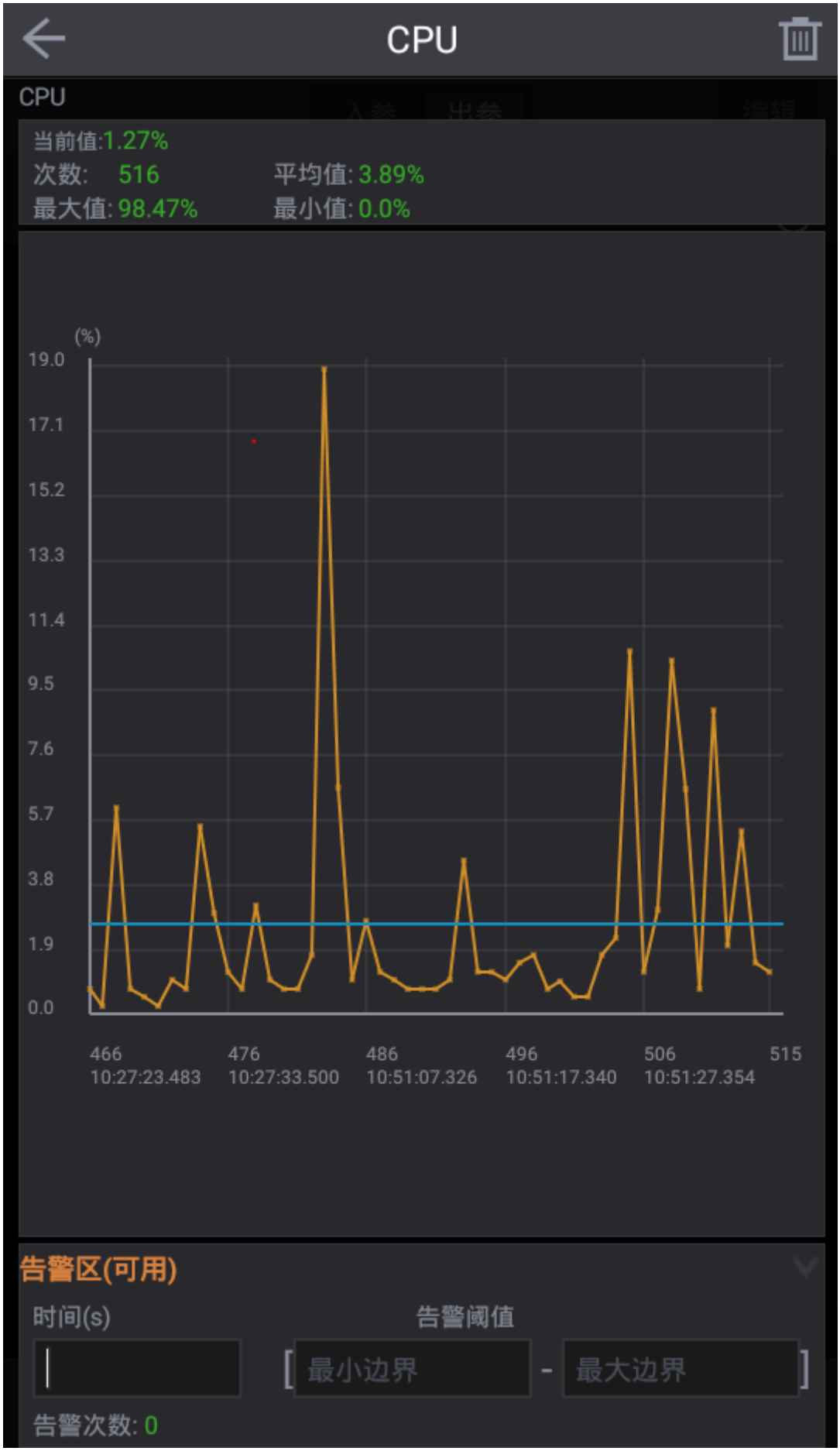
GT被测应用的运行界面

回到GT工具中 点击录制停止

点击数据查看数据

查看数据采集结果


P: 0 / 1
dX: 0.0
dY: 0.0
Xv: 0.0
Yv: 0.0
Prs: 1.0
Size: 0.0
10:32
GT
入参
出参
编辑
出参列表
数据采集与告警操作栏(G&W)
清除数据
悬浮窗展示的参数(<=3)
empty
已关注的参数
保存收集数据，存放文件夹
CPU
CPU
2.51%
99+
PCP0
Pcp0:com.l....education
0.24%
99+
PCP1
Pcp1:com.l...n:pushcore
0.0%
64
已取消关注的参数
点击查看指标图形
MEM
MEM
NET
NET
SIG
Signal
FPS
FPS
AUT
参数
耗时
日志
插件


CPU
CPU
当前值:1.27%
次数: 516
平均值: 3.89%
最大值: 98.47%
最小值: 0.0%
(%)
19.0
17.1
15.2
13.3
11.4
9.5
7.6
5.7
3.8
1.9
0.0
466
10:27:23.483
476
10:27:33.500
486
10:51:07.326
496
10:51:17.340
506
10:51:27.354
515
告警区(可用)
时间(s)
告警阈值
最小边界
-
最大边界
告警次数: 0

查看运行时日志


GT
On
点击启动开关
按文本过滤
VERBOSE
TAG
日志删除
VERBOSE
DEBUG
INFO
WARNING
ERROR
日志过滤
保存文件
tag标记
日志级别
05-11 10:35:39.112: I/UnicodeIME(1942): onFinishInput:
日志内容
05-11 10:35:39.113: I/UnicodeIME(1942): onStartInput
05-11 10:35:39.350: D/AndroidRuntime(6839): >>>>>> START com.android.internal.os.RuntimeInit uid 0 <<<<<<
05-11 10:35:39.354: D/AndroidRuntime(6839): CheckJNI is OFF
05-11 10:35:39.394: W/ (6839): Failed to bind-mount /system/lib/x86/cpuinfo as /proc/cpuinfo: No such file or directory
05-11 10:35:39.394: E/ (6839): Cannot create code cache directory ./code_cache: Read-only file system.
05-11 10:35:39.455: E/memtrack(6839): Couldn't load memtrack module (No such file or directory)
05-11 10:35:39.455: E/android.os.Debug(6839): failed to load memtrack module: -2
05-11 10:35:39.476: D/AndroidRuntime(6839): Calling main entry com.android.commands.settings.SettingsCmd
05-11 10:35:39.499: D/AndroidRuntime(6839): Shutting down VM
AUT
参数
耗时
日志
插件
摇一摇
键位设置
文件助手
安装APK
全屏
音量加
音量减
清理内存

2 CPU

2.1 CPU知识点介绍:

GT工具提供了两个CPU的监控指标：CPU和Jiffies

CPU:

> 整机的CPU使用水平，即当前手机的CPU整体使用率。
>
> 计算公式：在 Linux 系统下，CPU 利用率分为用户态、系统态和空闲态
>
> 用户态：表示 CPU 处于应用程序执行的时间
>
> 系统态：表示系统内核执行的时间
>
> 空闲态：表示空闲系统进程执行的时间。
>
> CPU 使用率 = CPU 执行非系统空闲进程时间 / CPU 总的执行时间

Jiffies:

表示自开机以来，应用程序消耗的CPU时间片的总数，Jiffies代表时间
cpu利用率 = （用户态Jiffies＋系统态Jiffies）/ 总Jiffies

CPU问题产生的影响：

CPU使用长时间处于80%~90%以上

手机发热、耗电量增加

反应变慢、引起ANR（Application Not Responding）

2.2 测试需求：

打开 拉勾教育，专栏点击分类查看，选择测试&运维，CPU指标正常。测试方法：

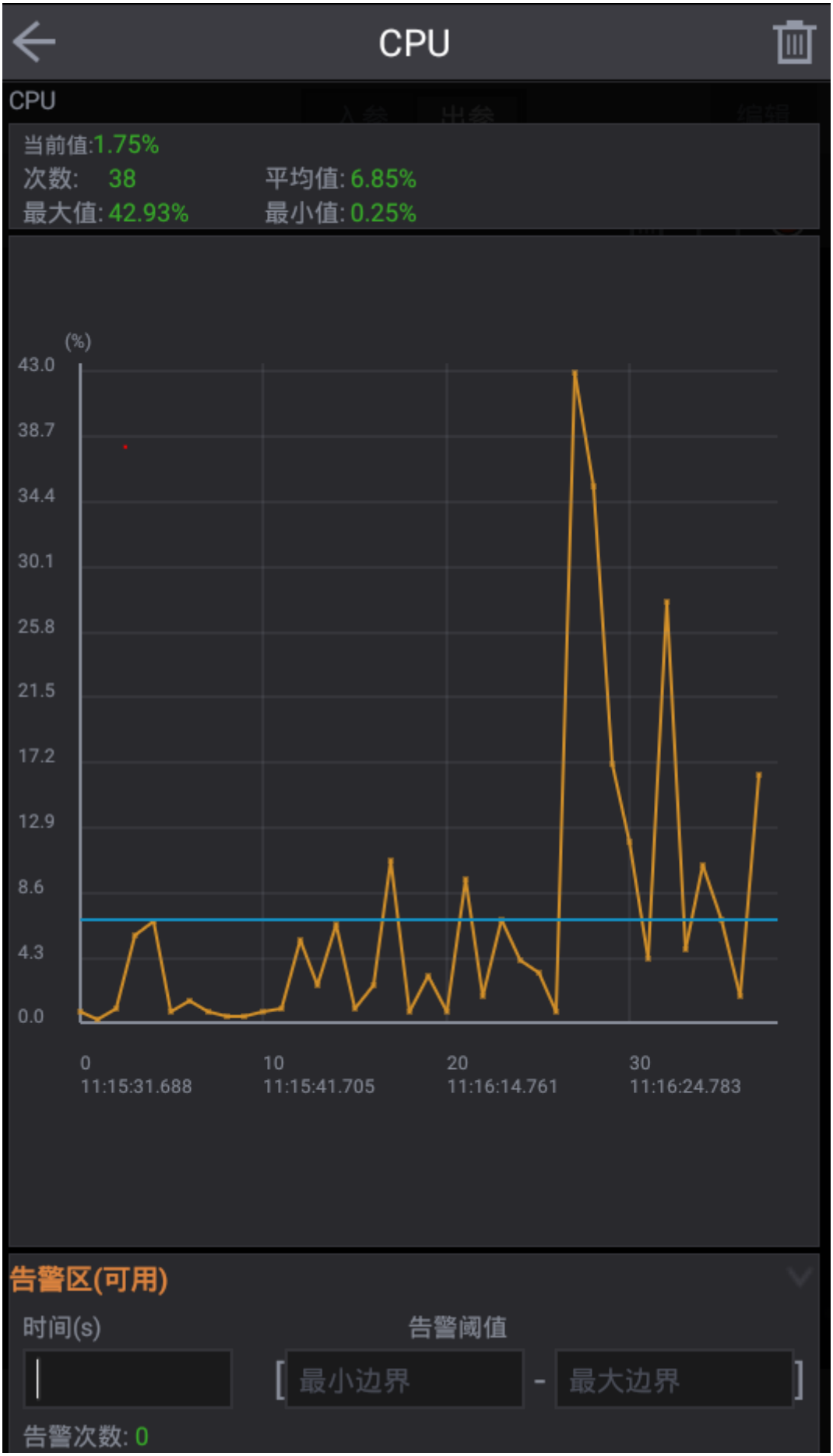
打开GT工具，配置CPU监控指标（可配置告警阈值）

进入 拉勾教育APP，操作上述业务，观察运行时的CPU指标

APP运行时CPU是否有快速飙升

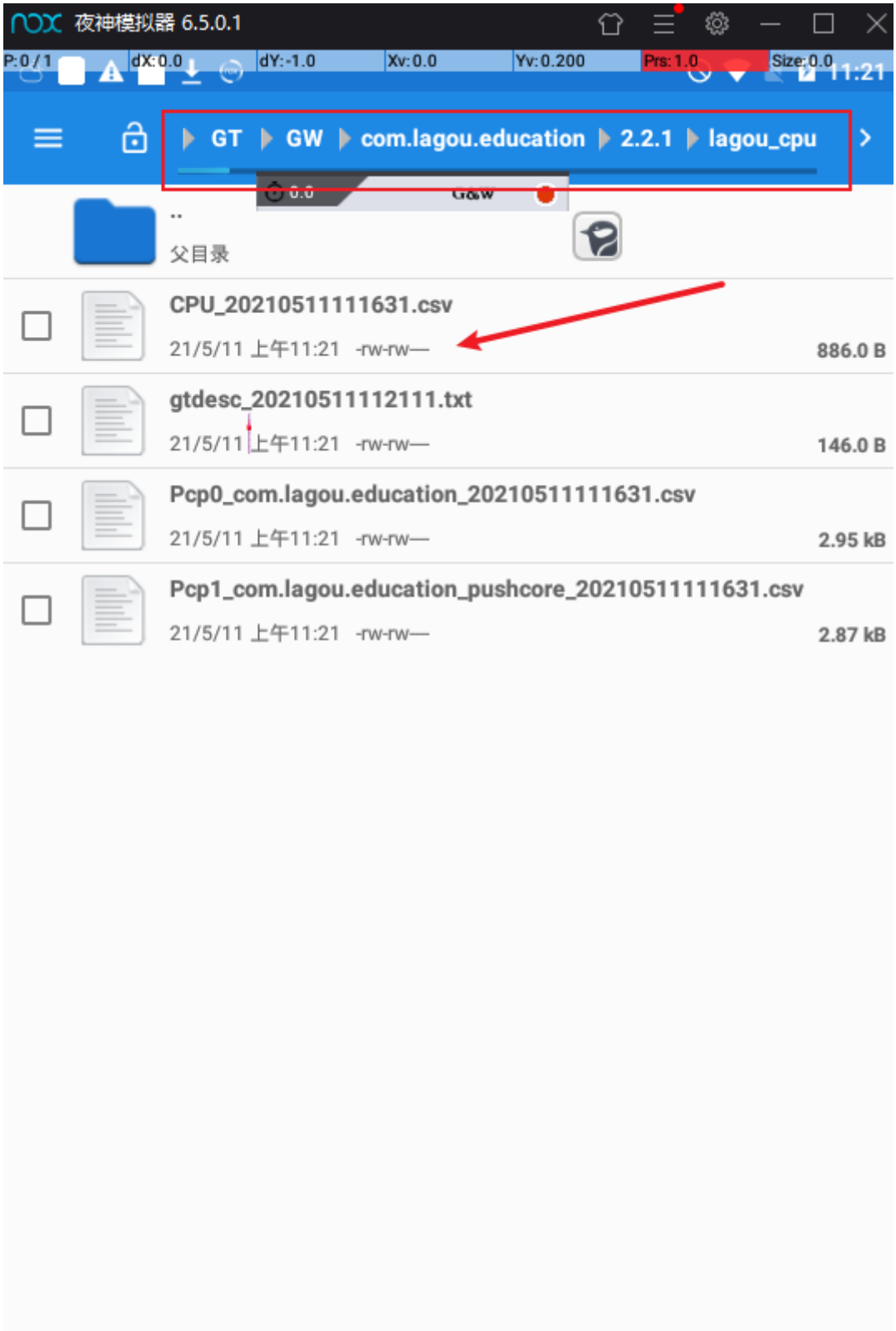
APP运行时CPU是否长时间处于90%以上

查看CPU运行结果



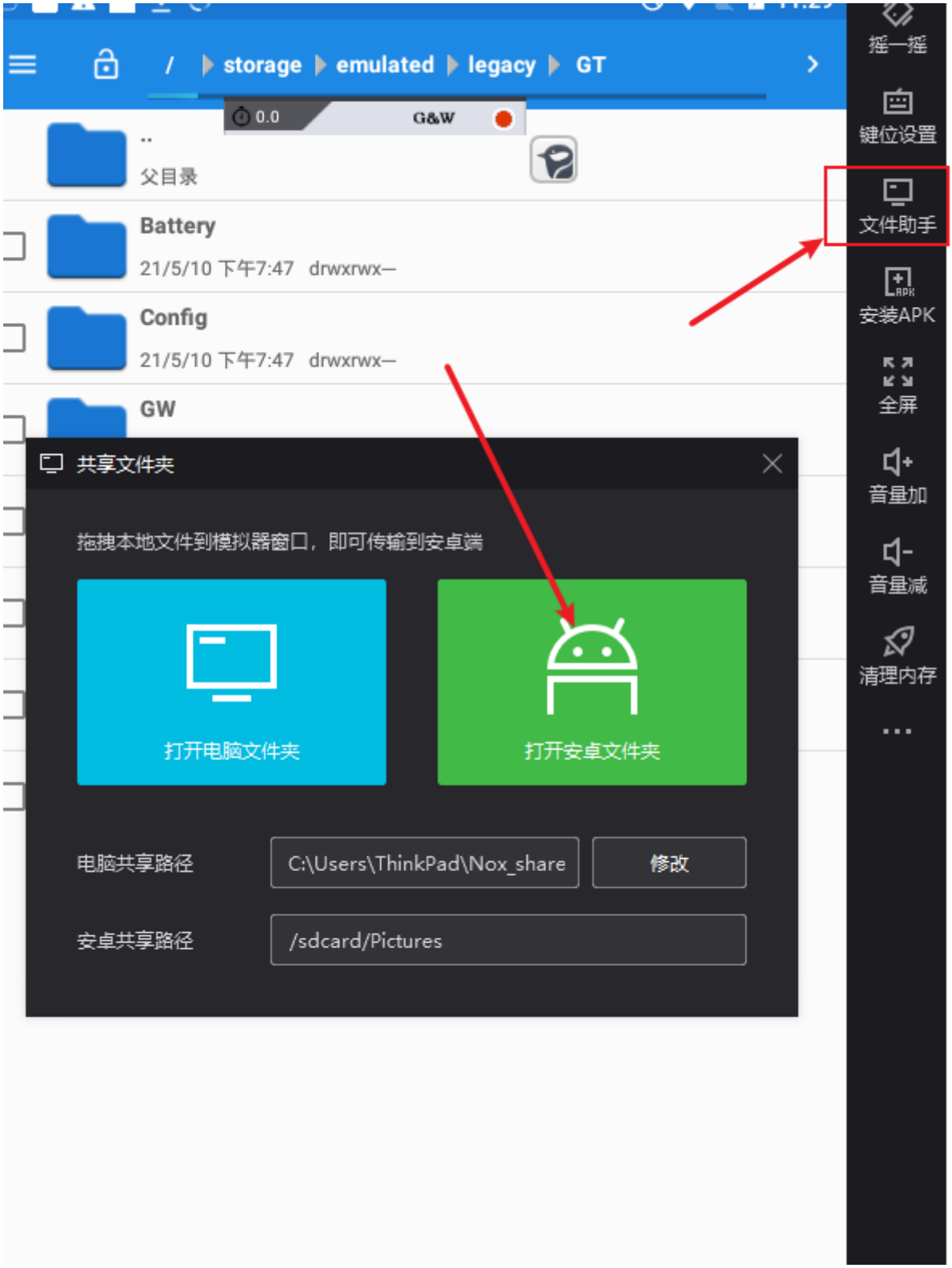
保存CPU详细数据后，可以查看CPU详细的数据统计。

点击保存按钮会保存记录数据到手机本地GT/GW/GW_DATA目录下，后续将数据导入到电脑上，用于分析数据。



需要把生成日志传到pc端

点击文件助手 -- 点击打开安卓文件夹 把对应的日志文件复制 -- 再次打开电脑文件夹



点击电脑文件夹，可以查看到对应日志

key	CPU	
alias	CPU	
unit	(%)	
begin date		2021/5/11
end date		2021/5/11
count		38
min		0.25%
max		42.93%
avg		6.85%
15:31.7		0.75%
15:32.7		0.25%
15:33.7		1.00%
15:34.7		5.77%
15:35.7		6.76%
15:36.7		0.76%
15:37.7		1.51%
15:38.7		0.75%
15:39.7		0.50%
15:40.7		0.50%
15:41.7		0.76%
16:05.7		1.01%
16:06.7		5.55%
16:07.7		2.57%
16:08.7		6.54%
16:09.8		1.00%
16:10.8		2.54%
16:11.8		10.78%
16:12.8		0.75%
16:13.8		3.12%
16:14.8		0.75%
16:15.8		9.53%
16:16.8		1.75%
16:17.8		6.87%
16:18.8		4.15%
16:19.8		3.31%
16:20.8		0.75%
16:21.8		42.93%
16:22.8		35.44%
16:23.8		17.13%
16:24.8		12.02%
16:25.8		4.30%
16:26.8		27.85%

场景： 在空闲时间消耗，cpu占比情况 运行一些其他应用，cpu50% --观察被测应用的cpu情况

高负荷情况cpu80% 被测应用的状态 竞品cup占比情况

3 内存测试

3.1 内存知识点介绍

GT工具提供了两个内存的监控指标：PSS和Private dirty

Private dirty（私有内存）：

进程独占内存，也就是进程销毁时可以回收的内存容量。

PSS（实际使用内存）：

将跨进程共享页也加入进来， 进行按比例计算PSS。这样能够比较准确的表示进程占用的实际物理内存。

3.2 常见内存问题：

内存泄漏：

内存泄露 memory leak，是指程序在申请内存后，无法释放已申请的内存空间，一次内存泄露危害可以忽略，但内存泄露堆积后果很严重，无论多少内存,迟早会被占光。

内存溢出：

内存溢出 out of memory，是指程序在申请内存时，没有足够的内存空间供其使用，出现out of memory。memory leak会最终会导致out of memory!

3.3 内存问题产生的影响：

程序实际使用的内存PSS持续增长

程序出现crash（可能是内存溢出）

测试需求：

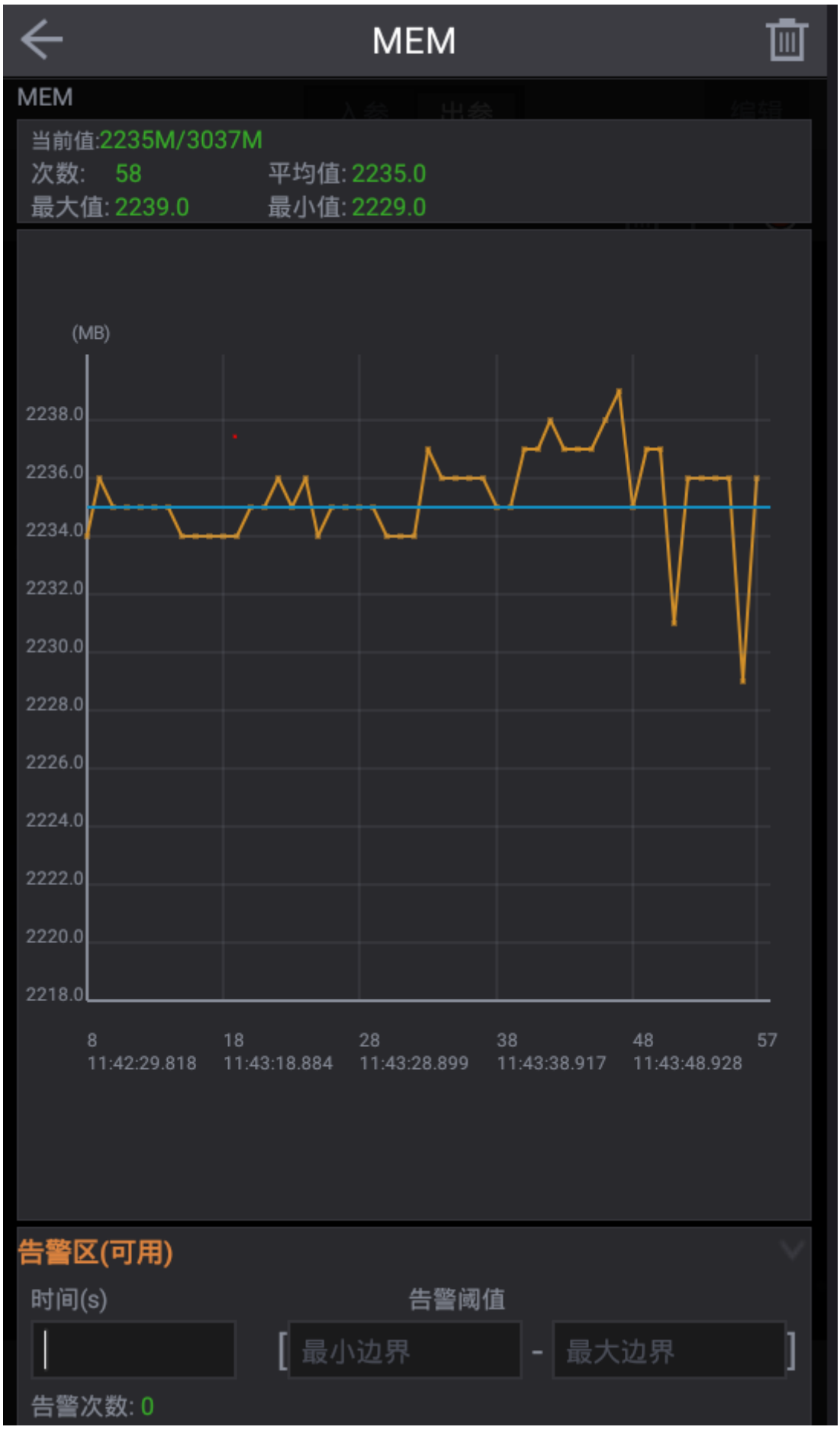
打开 拉勾教育，专栏点击分类查看，选择测试&运维，内存指标正常。测试方法：

打开GT工具，配置内存监控指标（可配置告警阈值）

进入 拉勾教育APP，操作上述业务，观察运行时的内存指标程序实际使用的内存PSS是否持续增长程序是否出现crash（可能是内存溢出）

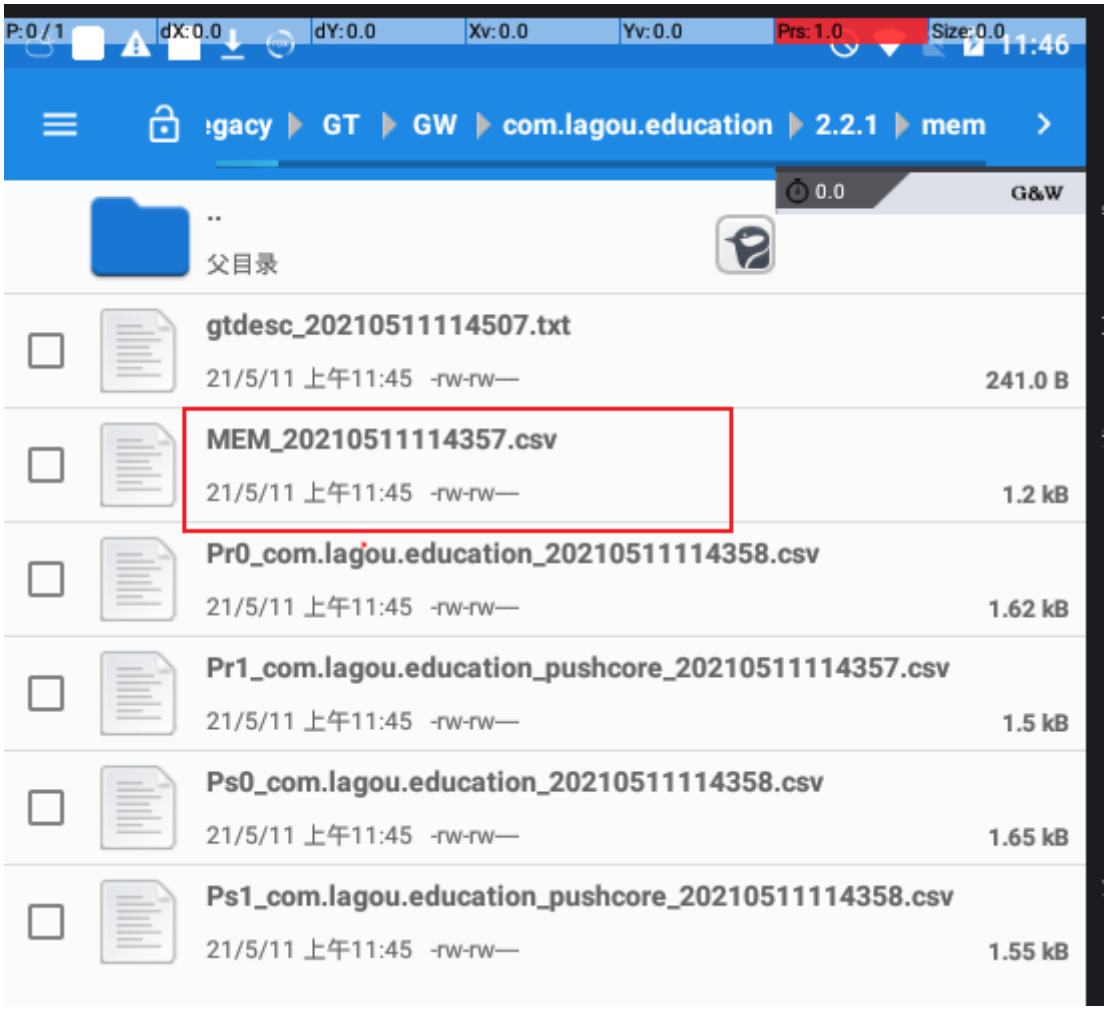
查看内存运行结果

保存内存详细数据后，可以查看内存详细的数据统计。



点击保存按钮会保存记录数据到手机本地GT/GW/GW_DATA目录下，后续将数据导入到电脑上，用于分析数据。

GT可以抓取产品在运行时的日志，方便监控crash log



4 流畅度测试

4.1 流畅度知识点介绍

GT工具提供了流畅度的监控指标：FPS

FPS即Frames per second：GPU在一秒内绘制的帧数

用过flash的人应该知道动画片其实是由一张张画出来的图片连贯执行产生的效果，当一张张独立的图片切换速度足够快的时候，会欺骗我们的眼睛，以为这是连续

的动作。反之类推，当你的图片切换不够快的时候，就会被人眼看穿，反馈给用户的就是所谓的卡顿现象。

4.2 流畅度问题产生的影响：

> 想要让大脑觉得动作是连续的，至少是每秒10-12帧的速度
>
> 想达到流畅的效果，至少需要每秒24帧
>
> 60帧每秒的流畅度是最佳的，我们的目标就是让程序的流畅度能接近60帧每秒

4.3 测试需求：

打开 拉勾教育APP，进入首页，上下滑动动态（下滑5分钟，再上滑5分钟），记录FPS值。

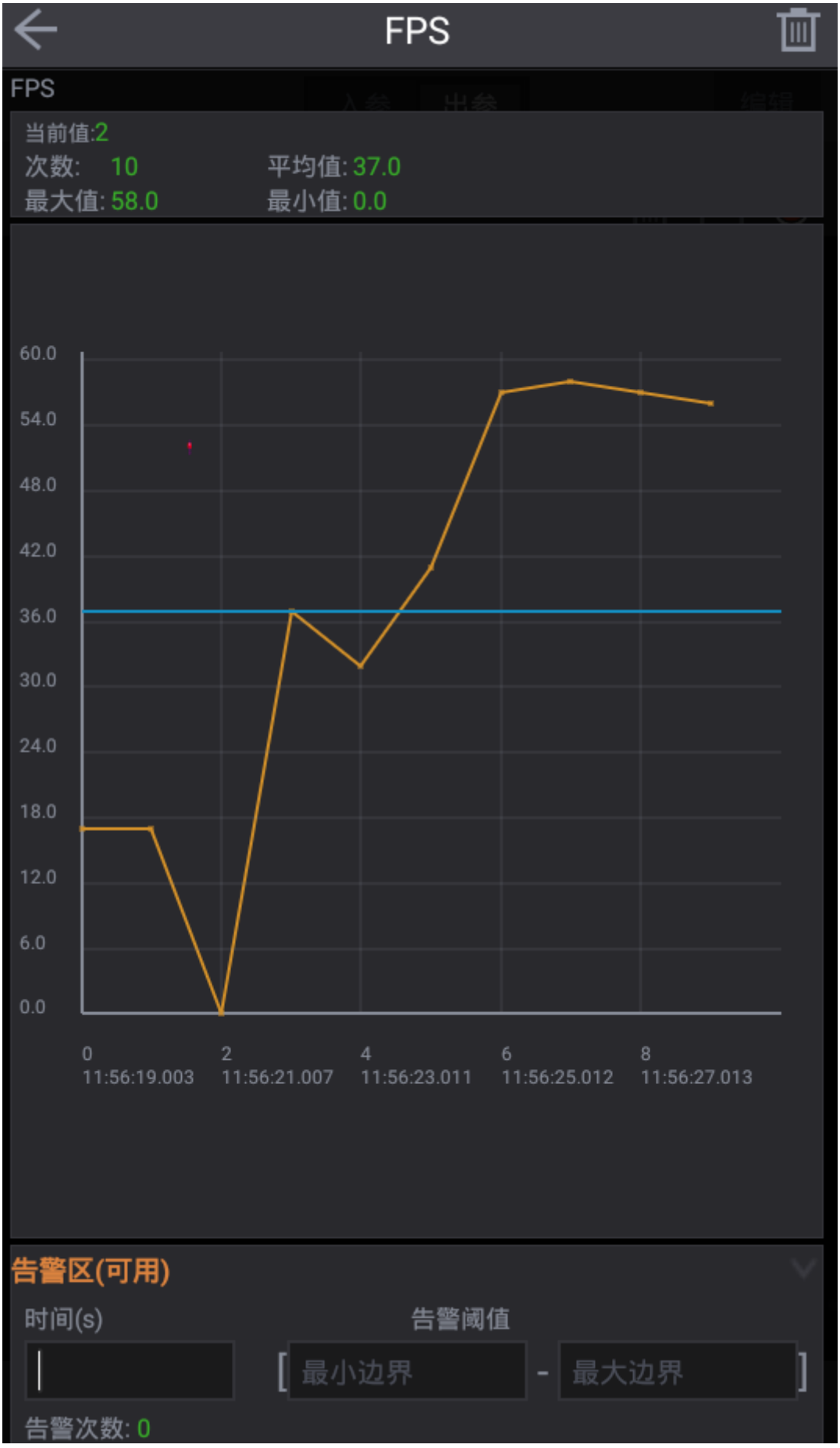
（不同于CPU、内存，当页面多为静态时，FPS值很小是正常的，页面数据多为动态加载时，FPS值比较大，因此我们选择上下滑动动态这个操作来测试）

测试方法：

```
打开GT工具，配置流畅度监控指标（在参数项中配置）

进入 拉勾教育APP，操作上述业务，观察运行时的流畅度指标

查看流畅度运行结果
```



保存流畅度详细数据后，可以查看流畅度详细的数据统计。

key	FPS
alias	FPS
unit	
begin date	2021/5/11
end date	2021/5/11
count	10
min	0
max	58
avg	37
56:19.0	17
56:20.0	17
56:21.0	0
56:22.0	37
56:23.0	32
56:24.0	41
56:25.0	57
56:26.0	58
56:27.0	57
56:28.0	56

点击保存按钮会保存记录数据到手机本地GT/GW/GW_DATA目录下，后续将数据导入到电脑上，用于分析数据。

5 流量测试

5.1 流量知识点介绍：

GT工具提供了流畅度的监控指标：NET

流量：

手机通过运营商的网络访问 Internet，运营商替我们的手机转发数据报文，数据报文的总大小（字节数）即流量，数据报文是包含手机上下行的报文。

5.2 常用流量测试方法：

抓包测试法

```
主要是利用工具 Tcpdump 或者fiddler抓包，导出 pcap 文件，再在 wireshark 中打开进行分析
```

统计测试法

```
获取应用程序收发的数据报文，统计出对应的流量
```

5.3 测试需求：

打开 拉勾教育APP，进入首页进入页面，进行滑动，获取消耗的网络流量

测试方法：

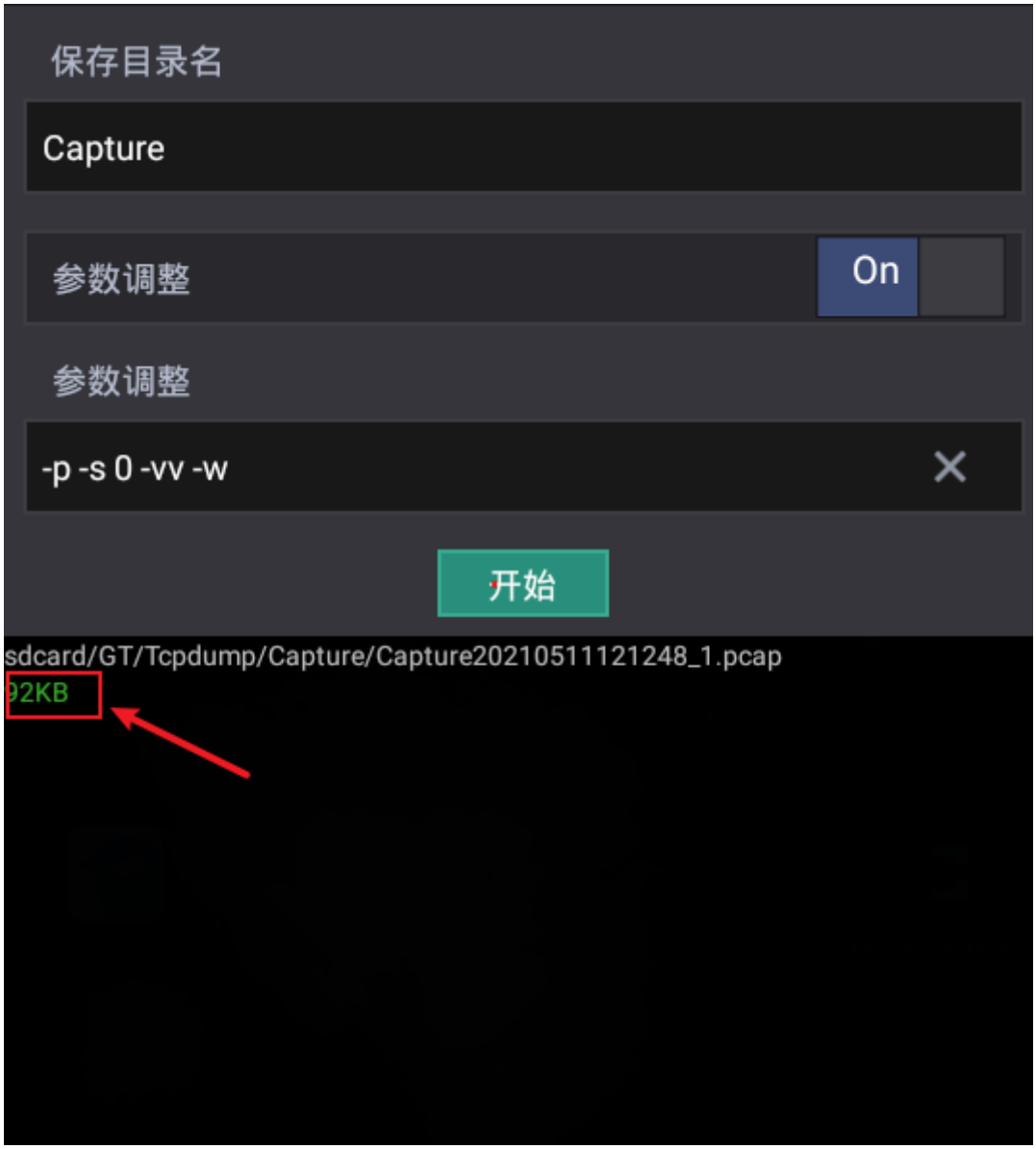
打开GT工具，配置流量监控指标NET（配置流量统计）

进入插件，点击抓包（获取业务测试的报文）

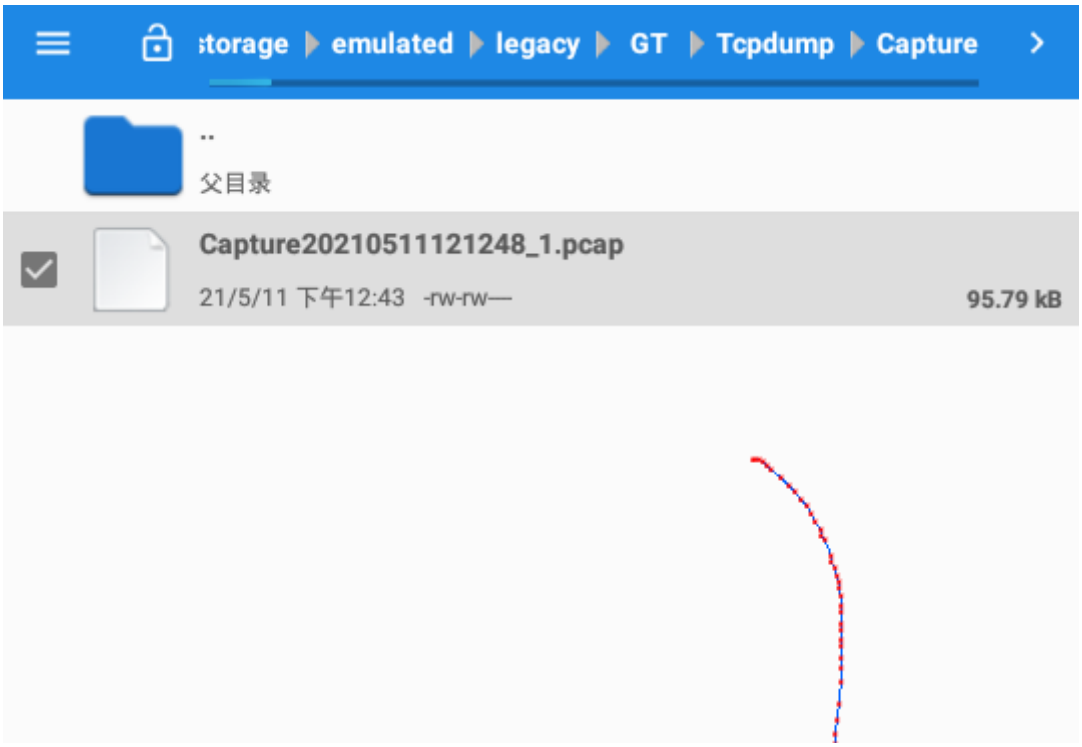
进入 拉勾教育APP，操作上述业务，观察运行时的流量指标


返回
抓包
保存目录名
Capture
参数调整
点击开启
On
参数调整
不需要调整
-p -s 0 -vv -w
停止
sdcard/GT/Tcpdump/Capture/Capture20210511121248_1.pcap
0KB
流量数据保存路径
已授予"GT"超级用户权限
描述:
这是一个抓包插件，直接点击Start即可抓包，点击Stop后保存已抓包的内容到文件。抓包文件可用PC上的Wireshark软件分析。

查看流量统计结果



点击“停止”按钮会保存报文数据为GT/Tcpdump/Capture/XXX.pcap，后续将数据导入到电脑上，用于分析数据。



5.4 adb获取流量方法

由于GT对设备有要求， 这里推荐大家adb命令获取

1、获取目标应用PID

```
C:\Users\ThinkPad>adb shell ps|findstr com.lagou.education
u0_a29    1274  172   1140556 230236 ffffffff b769935b S com.lagou.education
u0_a29    1398  172   565920 78252 ffffffff b769935b S com.lagou.education:pushcore

C:\Users\ThinkPad>
```

2、获取目标应用的UID ， 主要用于表示是哪位用户运行了该程序

```
C:\Users\ThinkPad>adb shell ps|findstr com.lagou.education
u0_a29    1274  172   1140556 230236 ffffffff b769935b S com.lagou.education
u0_a29    1398  172   565920 78252 ffffffff b769935b S com.lagou.education:pushcore

C:\Users\ThinkPad>adb shell cat /proc/1274/status | findstr Uid
Uid:    10029   10029   10029   10029

C:\Users\ThinkPad>
```

3、获取发送流量

```
C:\Users\ThinkPad>adb shell ps|findstr com.lagou.education
u0_a29    1274  172   1140556 230236 ffffffff b769935b S com.lagou.education
u0_a29    1398  172   565920 78252 ffffffff b769935b S com.lagou.education:pushcore

C:\Users\ThinkPad>adb shell cat /proc/1274/status | findstr Uid
Uid:    10029   10029   10029   10029

C:\Users\ThinkPad>adb shell cat /proc/uid_stat/10029/tcp_snd
481952
```

4、获取接受的流量数据

```
C:\Users\ThinkPad>adb shell cat /proc/1274/status | findstr Uid
Uid:    10029   10029   10029   10029

C:\Users\ThinkPad>adb shell cat /proc/uid_stat/10029/tcp_snd
481952

C:\Users\ThinkPad>adb shell cat /proc/uid_stat/10029/tcp_rcv
192175987

C:\Users\ThinkPad>
```

场景的流量优化方法

流量问题不能简单人为是客户端问题

上行: 客户端影响

下行：服务端影响

数据的压缩

不同数格式的采用

控制访问的频次

只获取必要的数

缓存机制

对不同的网络类型设置不同的访问策略

6 电量测试

6.1 电量知识点介绍：

GT工具提供了电量的监控指标：电流，电压，电量跟温度。

电量测试：

就是测试移动设备电量消耗快慢的一种测试方法。一般是用平均电流（电池生产厂家一般都采用mAh来标记电池容量大小，平均电流越小，说明设备使用时间就越长）来衡量电量消耗速度

常见的耗电场景：

定位，尤其是调用 GPS 定位。

网络传输，尤其是非 Wifi 环境

屏幕亮度

CPU 频率

内存调度频度

6.2 测试需求：

调整屏幕亮度为最大

打开 拉勾教育APP，点击首页，开启音频或者视频

使用过程中，间隔几分钟就锁屏或者唤醒屏幕

测试方法：

打开GT工具，进入插件tab，点击耗电数据采集

选择采样频率，屏幕亮度和采集参数。

进入 拉勾教育APP，操作上述业务

测试完成后，回到参数页面，点击停止录制；点击保存，选择路径保保存本次的数据

在上面指定的路径下，找到保存的数据（1个.csv文件）就可以看到刚才测试场景的电量值

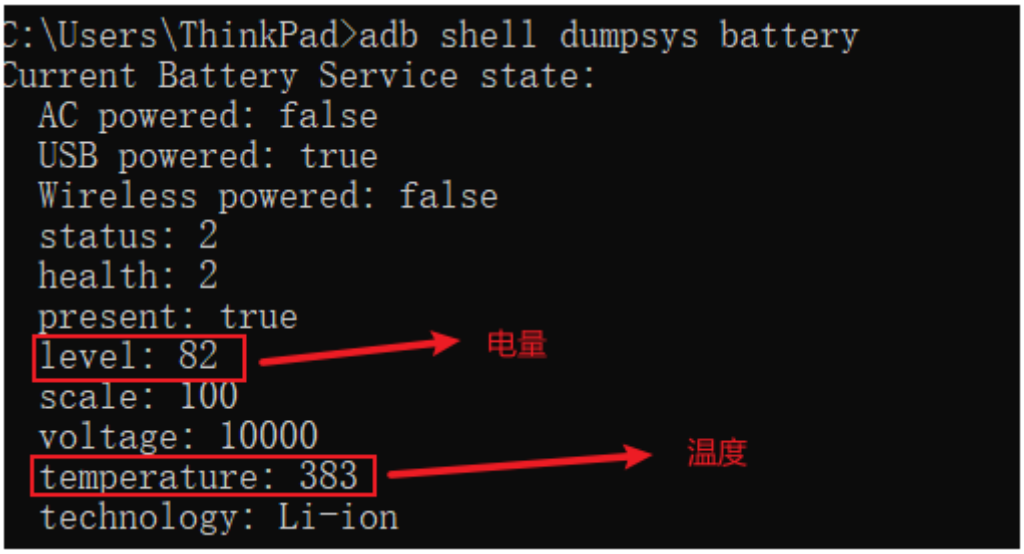
GT电量测试对设备有要求，可以通过adb shell 命令采集电量信息

显示当前电池情况

```
adb shell dumpsys battery

level:电量百分比

temperature：温度 单位0.1摄氏度
```



清除已存在耗电量数据

adb shell dumpsys batterystats --enable full-wake-history

设备耗电量数据重置

```
adb shell dumpsys batterystats --reset
```

查询应用包的uid

```
adb shell ps |find "包名"
```

以下是DOS信息，u0_a77就是uid，需要去除下划线_， 即u0a77

```
C:\Users\ThinkPad>adb shell ps |find "com.lagou.education"
u0_a77    4364  1513  1375908 163924 ffffffff b73e40f5 S com.lagou.education
u0_a77    4466  1513  1023076 83472 ffffffff b73e40f5 S com.lagou.education:pushcore
```

查询电量命令

统计整机电量：

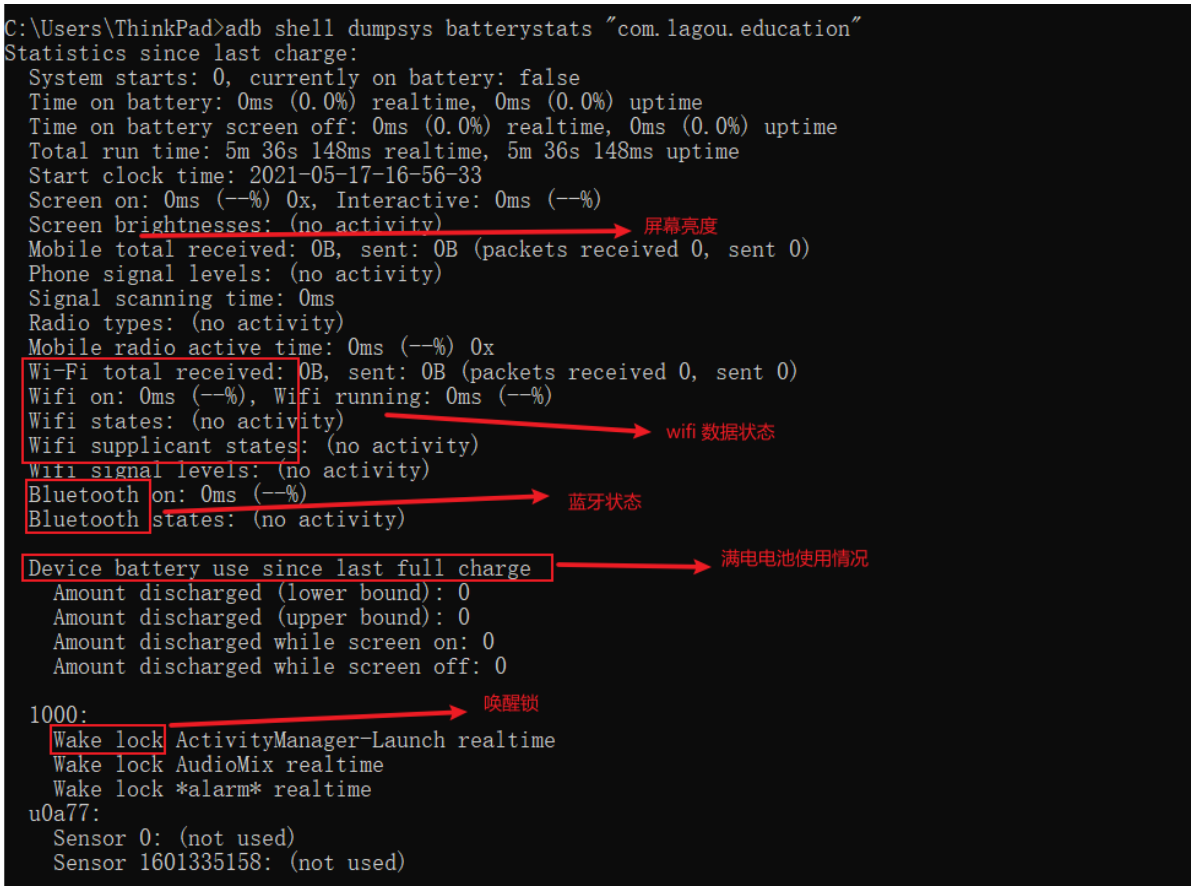
```
adb shell dumpsys batterystats>d:\apk\aa.txt
```

显示电池统计信息

adb shell dumpsys batterystats

统计被测应用电量：

adb shell dumpsys batterystats "com.lagou.education"



6.3 电量的测试结果分析：

我们可以根据不同的业务场景测试出对应的电量消耗数据，但是是否有问题需要对比分析，对比方法：

与基准数据对比。（基准数据来自于产品经理，或者以往数据积累）

横向对比，拉上竞品一起测（目前多数采用这种方法）

同样的网络、手机，相似的测试场景，最后对比我们的产品和竞品在耗电量方面的差距，给出优化建议

7 启动速度测试

7.1 启动速度介绍：

app的启动分为冷启动，热启动
冷启动：
指app被后台杀死后，在这个状态打开app，这种启动方式叫做冷启动。
热启动：
指app没有被后台杀死，仍然在后台运行，通常我们再次去打开这个app，这种启动方式叫热启动。

7.2 测试方法：

使用命令adb shell am start -W -n 包名/Activity名，查看 App 启动耗时

直接启动

```
adb shell  am  start -n  com.lagou.education/.ui.MainActivity
```

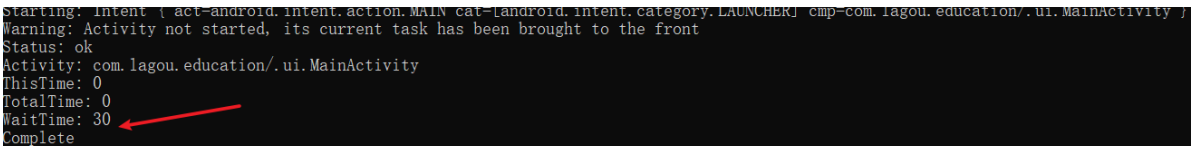
先停止再启动

```
adb shell  am  start -S  com.lagou.education/.ui.MainActivity
```

等待应用开启完成 并记录时间

```
adb shell  am  start -W  com.lagou.education/.ui.MainActivity
```

该命令获取3个关键指标：



ThisTime：表示一连串启动Activity的最后一个Activity启动耗时，一般会<=TotalTime时间。

TotalTime：应用的启动时间，包括创建进程、App初始化、Activity初始化到界面显示。（开发者需要优化的耗时）

WaitTime：前一个应用activity pause的时间+TotalTime

冷启动数据指标

```
C:\Users\ThinkPad>adb shell  am  start -W  com.lagou.education/.ui.MainActivity
Starting: Intent { act=android.intent.action.MAIN cat=[android.intent.category.LAUNCHER] cmp=com.lagou.education/.ui.MainActivity }
Status: ok
Activity: com.lagou.education/.ui.MainActivity
ThisTime: 2500
TotalTime: 2500
WaitTime: 2527
Complete
```

7.3 启动速度的测试结果分析：

同电量测试一样，我们可以测试出APP冷启动和热启动时花费的时长，但是是否有问题需要对比分析，对比方法：

与基准数据对比。（基准数据来自于产品经理，或者以往数据积累）

横向对比，拉上竞品一起测（目前多数采用这种方法）

同样的网络、手机，相似的测试场景，最后对比我们的产品和竞品在启动时长方面的差距，给出优化建议

8 弱网测试

与传统桌面应用不同，移动应用的网络环境比较多样，而且经常出现需要在不同网络之间切换的场景，即使是在同一网络环境下，也会出现网络连接状态时好时坏的情况，比如时高时低的延迟、经常丢包、频繁断线，比较封闭的空间（乘坐地铁、穿越隧道，和地下车库的场景下经常会发生）

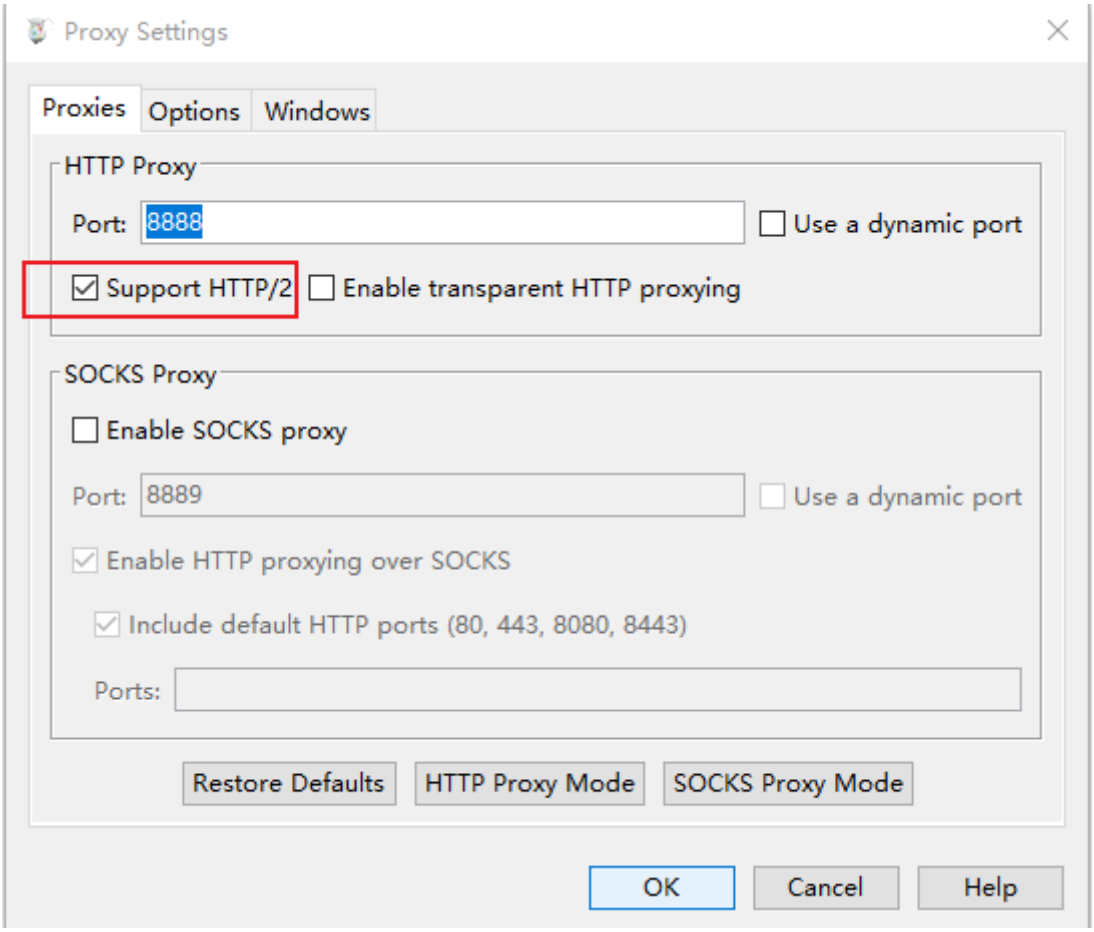
工具推荐

代理抓包工具：charles/fiddler

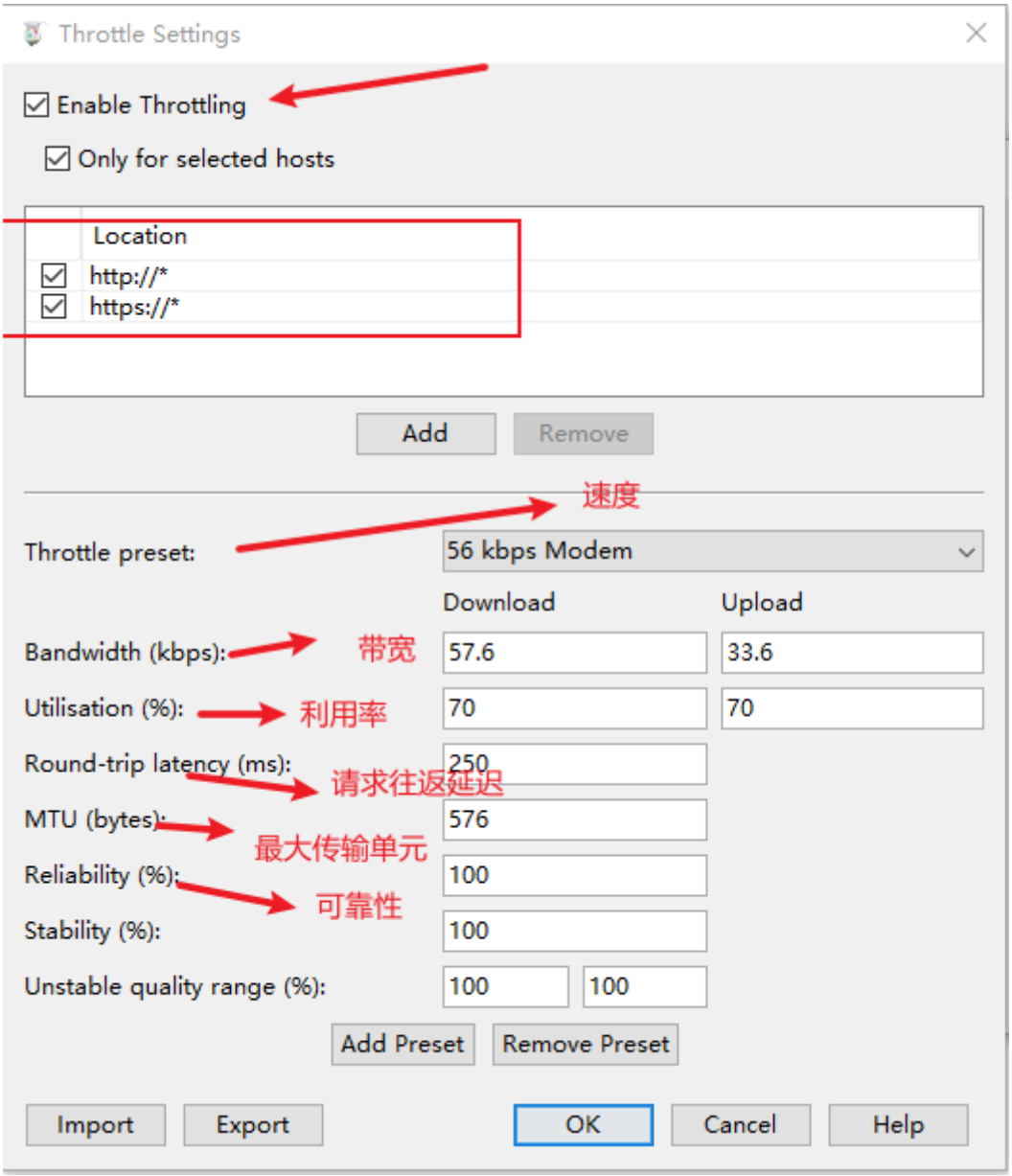
开源移动网络测试工具：Facebook（ATC）

charles工具设置

代理设置


Proxy Settings
Proxies
Options
Windows
HTTP Proxy
Port: 8888
Use a dynamic port
Support HTTP/2
Enable transparent HTTP proxying
SOCKS Proxy
Enable SOCKS proxy
Port: 8889
Use a dynamic port
Enable HTTP proxying over SOCKS
Include default HTTP ports (80, 443, 8080, 8443)
Ports:
Restore Defaults
HTTP Proxy Mode
SOCKS Proxy Mode
OK
Cancel
Help

节流设置

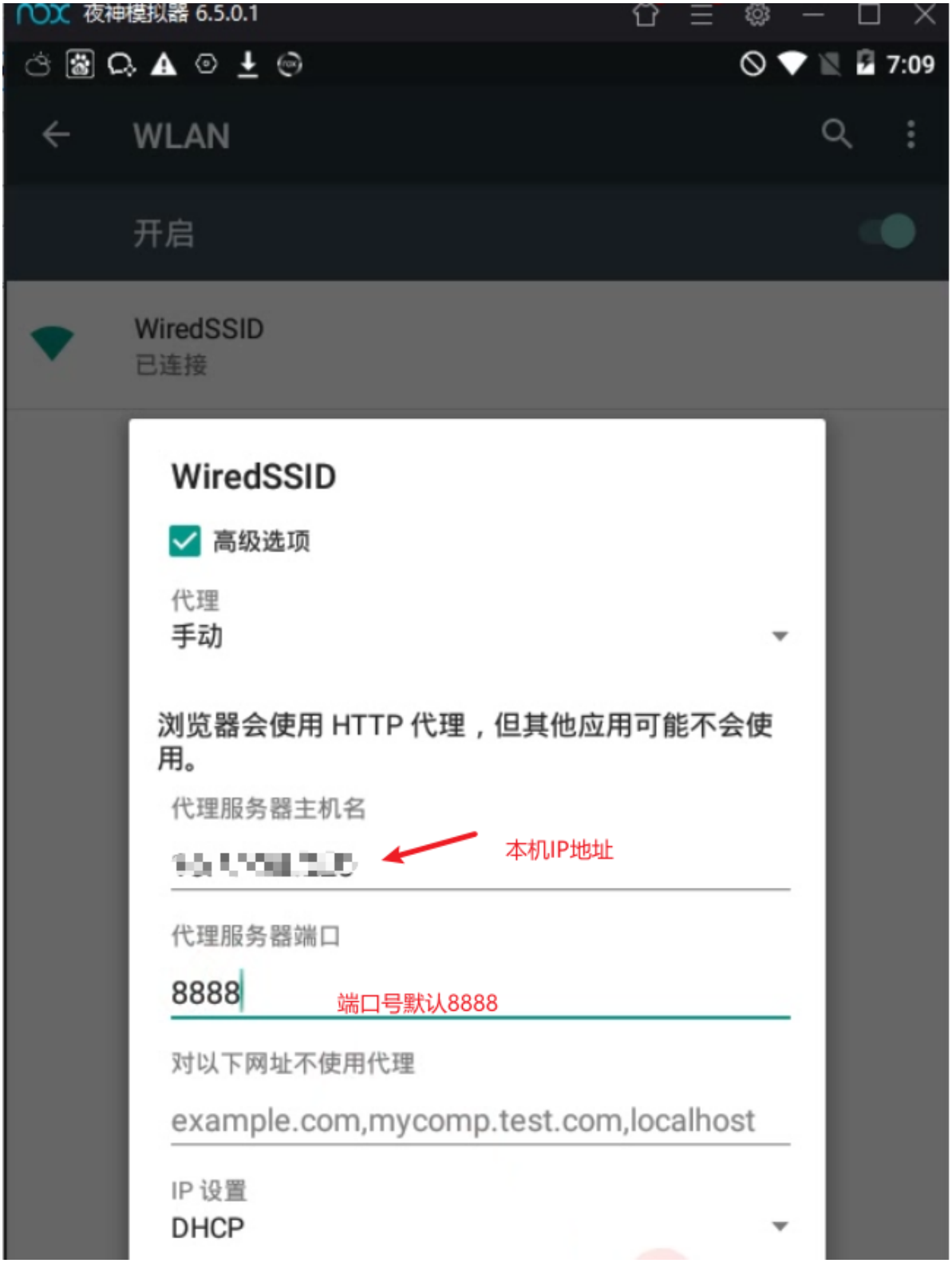


移动设备设置

代理--自动修改手动

服务器地址 -- 本机实际IP

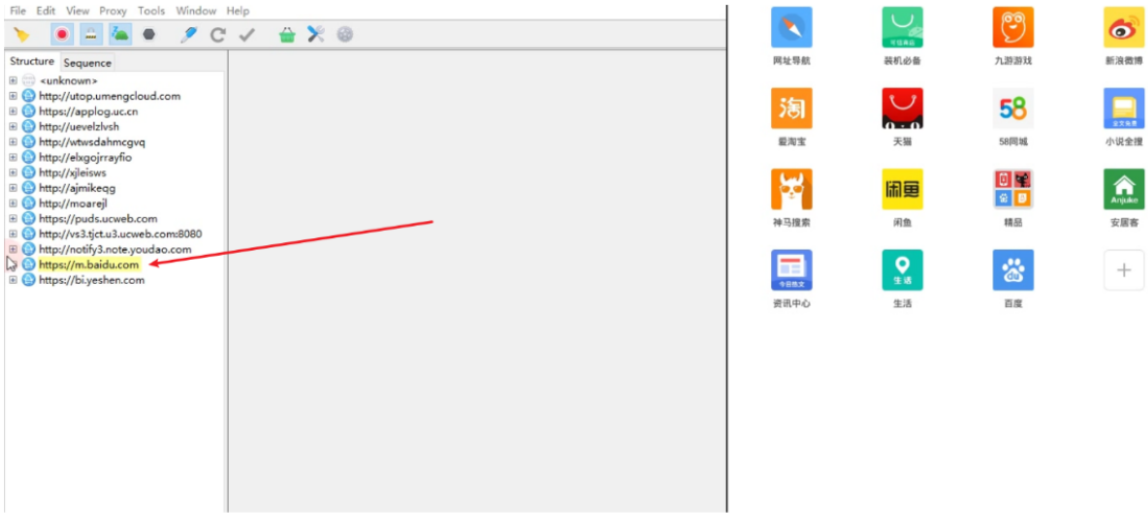
代理端口 ： 8888



启动uc浏览器

点击百度

查询手机百度地址（如图显示）；能够感受页面开启速度缓慢或网页不能开启


File Edit View Proxy Tools Window Help
Structure Sequence
<unknown>
http://utop.umengcloud.com
https://applog.uc.cn
http://uevelzlvsh
http://wtwsdahmcgvq
http://ebxgojrrayfio
http://xjleisws
http://ajmikeqg
http://moarejl
https://puds.ucweb.com
http://vs3.tjct.u3.ucweb.com:8080
http://notify3.note.youdao.com
https://m.baidu.com
https://bi.yeshen.com
网址导航
装机必备
九游游戏
新浪微博
淘宝
天猫
58同城
小说全搜
神马搜索
闲鱼
精品
安居客
资讯中心
生活
百度