

## 安卓端离线数据监控

cpu数据来源可选：（对比后选择**busybox top**）

数据来源	性能（秒）	精度	备注
top	0.57	个位	系统自带
<b>busybox top</b>	0.22	小数1位	前置依赖： adb push busybox /data/local/tmp adb shell chmod 755 /data/local/tmp/busybox
dumpsys cpuinfo	0.04	小数1位	安卓cpuinfo服务提供数据，阶段平均值 <b>受dumpsys meminfo获取影响会被拉高平均值</b>

数据来源选择  
(cpu)

## 内存系统数据来源：

1、总内存 (/proc/meminfo)

2、系统进程内存 (dumpsys meminfo)

注：由于统计信息中也有总内存，且和/proc/meminfo维度不同也做保存

3、进程内存详细信息 (dumpsys meminfo 包名/PID)

注：除了heap内存信息，额外存储views数量，因为和PSS正相关而且引起的变化幅度大

4、线程数，辅助判定内存问题定位 (/data/local/tmp/busybox ps -T)

注：上限42991 (adb shell cat /proc/sys/kernel/threads-max)

5、进程FD总数 (ls -l /poc/(进程pid值)/fd 行数)

注：需root权限；linux限制上限1024 nofiles(descriptors)的值 adb shell ulimit -a 命令获取

6、VSS和RSS内存值，ps命令获取pid时有此两项数据，同时进行保存

注：VSS上限是物理内存+虚拟内存总和，（目前安卓项目的本地存储上通常不分配虚内存）

## 主频系统数据来源:

1、cpu多核主频 (/sys/devices/system/cpu/cpu\*/cpufreq/scaling\_cur\_freq)

注: \*是cpu编号从0开始

2、gpu主频 (/sys/class/kgsl/kgsl-3d0/devfreq/cur\_freq)

注: 这是高通芯片的GPU主频节点

## 温升系统数据来源:

1、sensor描述 (/sys/devices/virtual/thermal/thermal\_zone\*/type)

2、sensor值 (/sys/devices/virtual/thermal/thermal\_zone\*/temp)

## 电量系统数据来源：

- 1、充电状态（dumpsys power 中的mPlugType值）
- 2、电量百分比（dumpsys power 中的mBatteryLevel值）

## 电流系统数据来源：

- 1、高通芯片提供了充放电的电流值，负值进电，正数otg放电  
(/sys/class/power\_supply/battery/current\_now)

**FPS数据来源选择：**（采用1，指定窗口定位场景）

1、指定窗口监控（`dumpsys SurfaceFlinger --latency` 窗口名）

注：窗口名对应SurfaceFlinger 中的名字，安卓7.0后由于分屏的设计带有编号，例如#0

2、指定包名监控（`dumpsys gfxinfo` 包名 `framstats` 安卓5.0及以上版本）

注：基于android设计的统计数据整体评估，无法定位场景，并且不统计SurfaceView数据

**FPS**（原创计算需求，算法实现）：

1、帧数据采样间隔1.6S左右（60帧屏127帧=2.116S，为了连续抓数据和平衡抓取开销）

2、统计样本帧数/样本时间；只有一帧按硬件绘制耗时的结束时间计算

3、Vsync间隔大于500ms算静置，不计算帧率且不输出

4、空数据、vsync=最大正整数数据不处理

5、Vsync帧数据复用，结束时间加首行绘制间隔

## 硬件掉帧比例算法 (jank%)

- 1、帧数据中第三列和第一列的差值，如果大于首行绘制间隔jank帧数+1
- 2、Jank总帧数/样本数据总帧数
- 3、引导优化方向为framework优化SurfaceFlinger绘制性能，如已优化过则主推精简ui布局及图片资源大小

## 帧间隔超过kpi的帧数比例

- 1、按人眼识别卡顿100ms，可定制
- 2、引导优化肉眼可识别的严重卡顿，降低此比例

## 按人眼视觉的当前认知标准判定：

- 1、严重卡顿：帧间隔 $\geq 100\text{ms}$ （按人眼视觉存在100ms的反应时间）
- 2、轻微卡顿： $50\text{ms} \leq \text{帧间隔} < 100\text{ms}$ （游戏帧率可玩底线20帧）
- 3、延迟： $42\text{ms} < \text{帧间隔} < 50\text{ms}$ （视频级体验按网络流24帧）

## 流畅度打分评价（原创计算需求）

- 1、比例分配：平均帧率/60\*60%+kpi/最大两帧间隔\*20%+帧间隔小于kpi帧数比例\*20%（除法比例最大为1，kpi可定制，按人眼识别卡顿是100ms）

## 显示的窗口信息来源:

- 1、FocusedApplication 和 FocusedWindow(`dumpsys input|grep name`)
- 2、显示中的窗口层级信息 (`dumpsys input|grep "visible=true"`)

数据来源选择(窗口显示)



## 依赖文件：

1、设备端需存在/data/local/tmp/busybox，以下命令：  
(busybox可到官网对应cpu架构下载)

```
adb push busybox /data/local/tmp  
adb shell chmod 755 /data/local/tmp/busybox
```

2、监控脚本monitor.sh

**脚本执行：** adb shell

```
sh /data/local/tmp/monitor.sh "$monitor_folder" "$monitorWindow"  
"$monitorPackages" 5 ${meminfo_type=1} &
```

参数说明：

monitor\_folder = 监控结果文件夹名

monitorWindow = fps监控窗口，不抓取为空""

monitorPackages = 额外抓取heap、views、threads、FD信息的进程，多个用|间隔，不抓取为空""

5 = 5秒间隔

meminfo\_type = 1，抓所有进程PSS额外抓指定进程详细信息，0则只抓配置进程内存信息

**监控停止：** adb shell touch /data/local/tmp/stop

## 获取结果：

`adb pull /data/local/tmp/$monitor_folder` pc本地目录位置

## 生成报告：

`python monitor.py完整路径 $monitor_folder完整路径`

## 注：

- 1、需要安装python环境和pandas库
- 2、如果有需要，我可以提供windows的exe版本，由py文件和报告模板打包而成
- 3、本地文件查看浏览器支持
  - (1) 火狐，about:config 中 `privacy.file_unique_origin` 属性false

## 自 Firefox 68 本地文件安全性的改变

当用户在 Firefox 67 和更早版本中使用 `file:///` URI 打开页面时，页面来源被定义为打开页面的目录。同一目录及其子目录中的资源均被视为具有相同的来源，符合 CORS 同源规则。

为响应 [CVE-2019-11730](#)，Firefox 68 及更高版本中使用 `file:///` URI 定义打开页面的来源唯一。因此，同一目录或其子目录中的其他资源不再满足 CORS 同源规则。这个新的表现通过 `privacy.file_unique_origin` 这一首选项控制，默认启用。

- (2) chrome，由于安全权限不支持，需要打开权限启动后查看  
(`start chrome.exe --allow-file-access-from-files`)

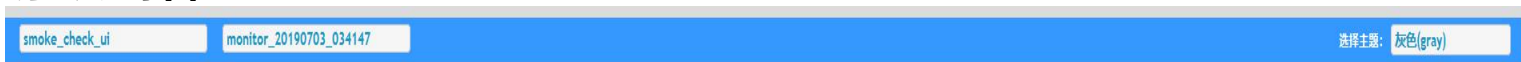
## 模板文件：

- |                |    |               |
|----------------|----|---------------|
| 1、 Report.html | —— | 报告的html布局模板   |
| 2、 “head” 文件夹  | —— | html依赖的js和css |
| 3、 “ data ”    | —— | 结果数据的js文件     |

## 报告的原理关键点：

- 1、 html、 js、 css都是文本文件， 模板化设计
- 2、 预期存储数据支持离线本地和在线查看。由于离线文件加载限制， 生成为js文件支持动态加载， 按选择更新数据变量
- 3、 数据的js文件内容有python脚本处理数据后生成
- 4、 可交互趋势图展示基于Highcharts框架设计实现
- 5、 报告模板按照数据动态切换无数据的图关闭显示

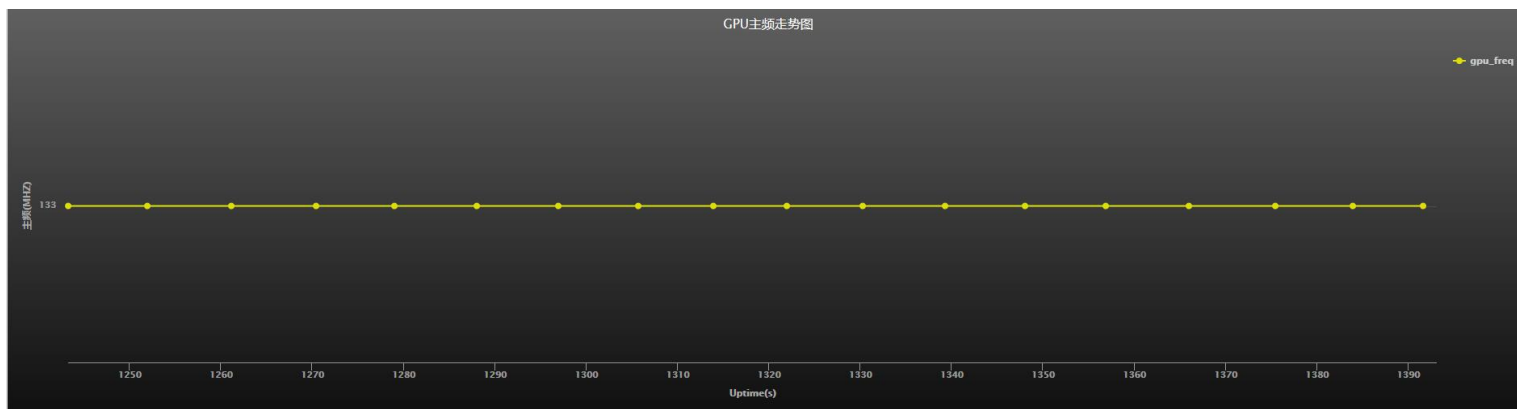
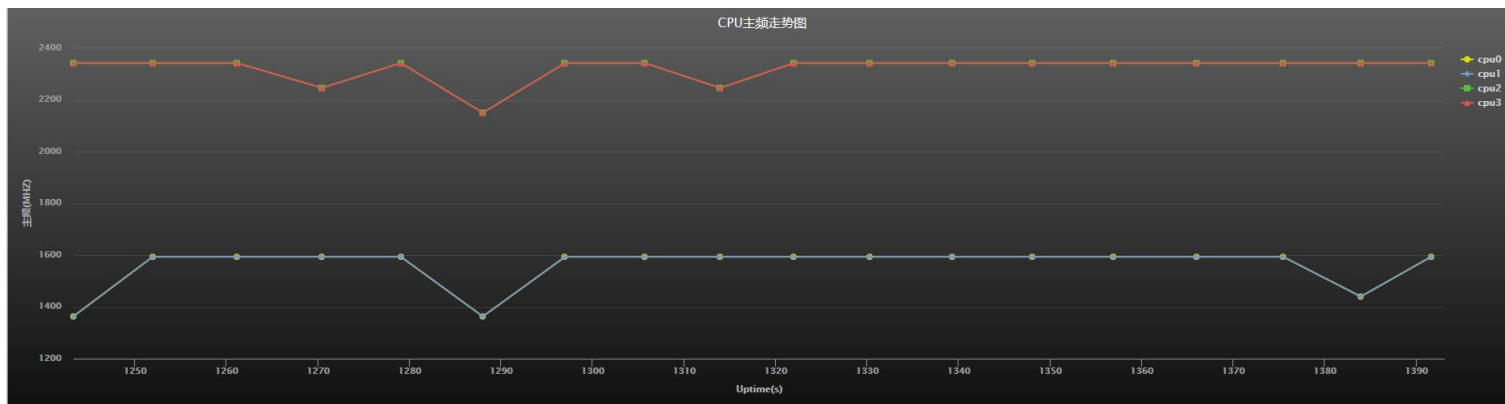
## 浮动选择栏：



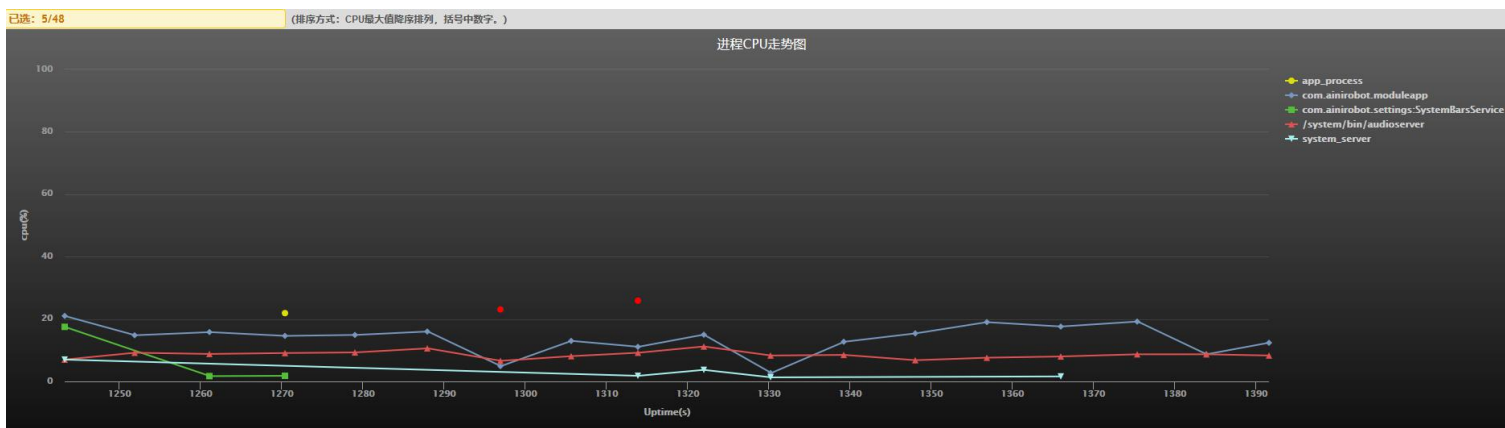
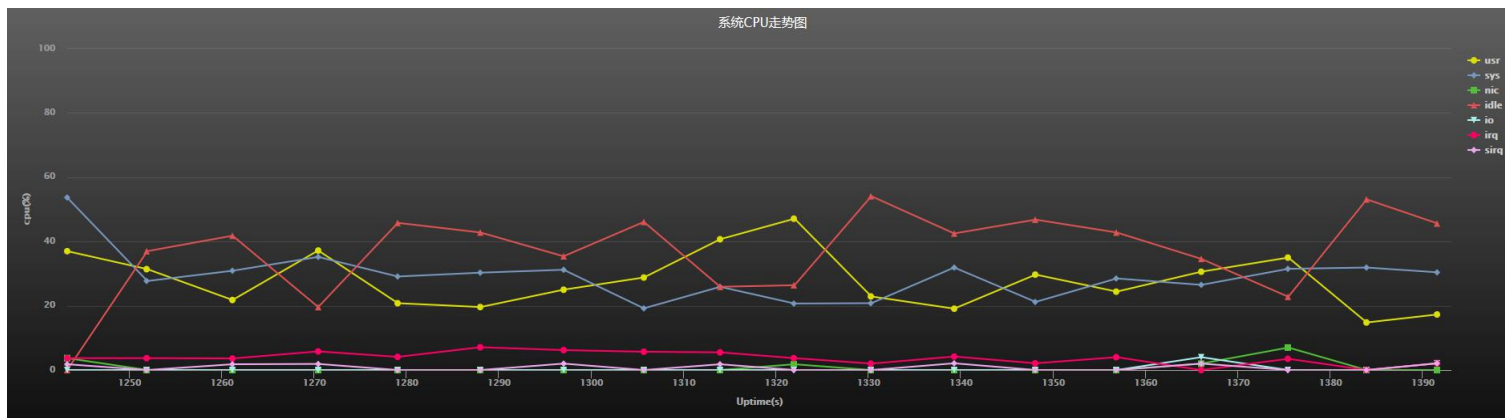
- 1、随页面滚动浮动
- 2、case名（左侧第一个）为测试结果csv所在文件夹的上级目录名
- 3、测试结果名（左侧第二个）为csv所在目录名，选择后更新数据（数据较大情况需稍等）
- 4、右侧可更改Highcharts支持的官方主题，可按报告需求更改配色

## 报告介绍（浮动栏）

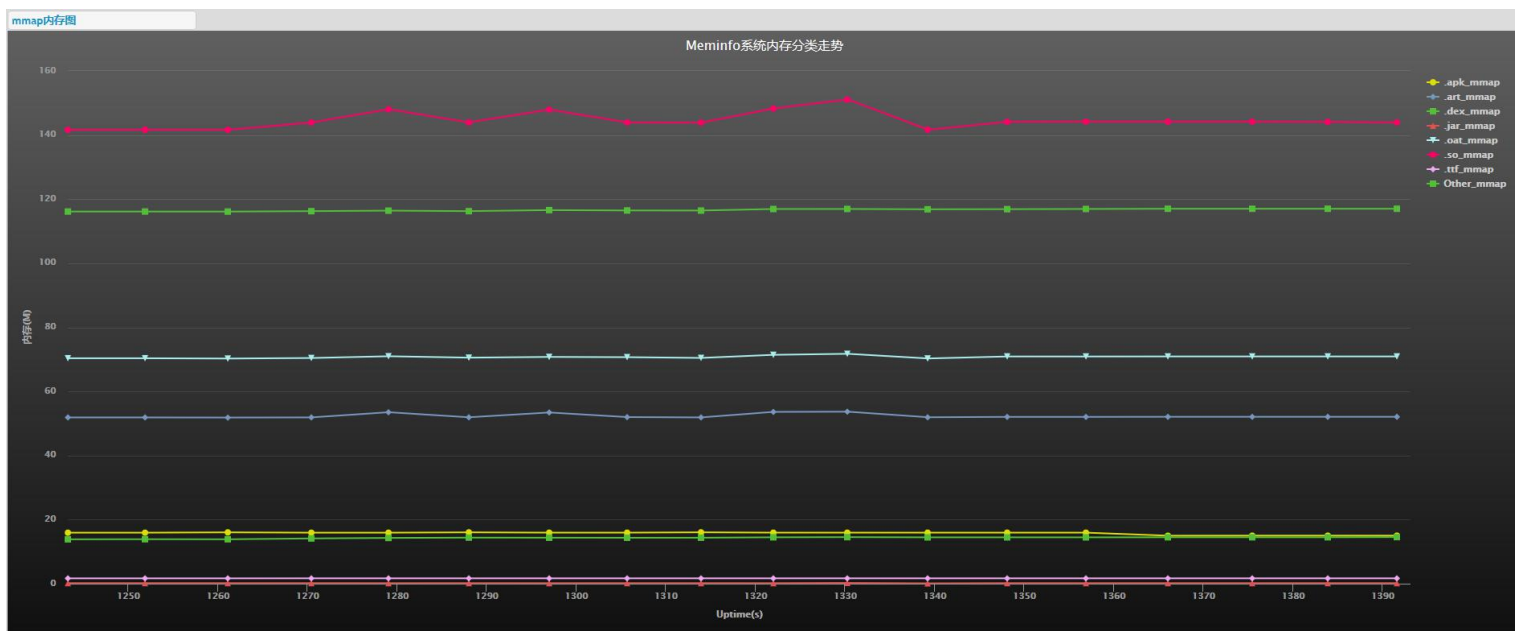
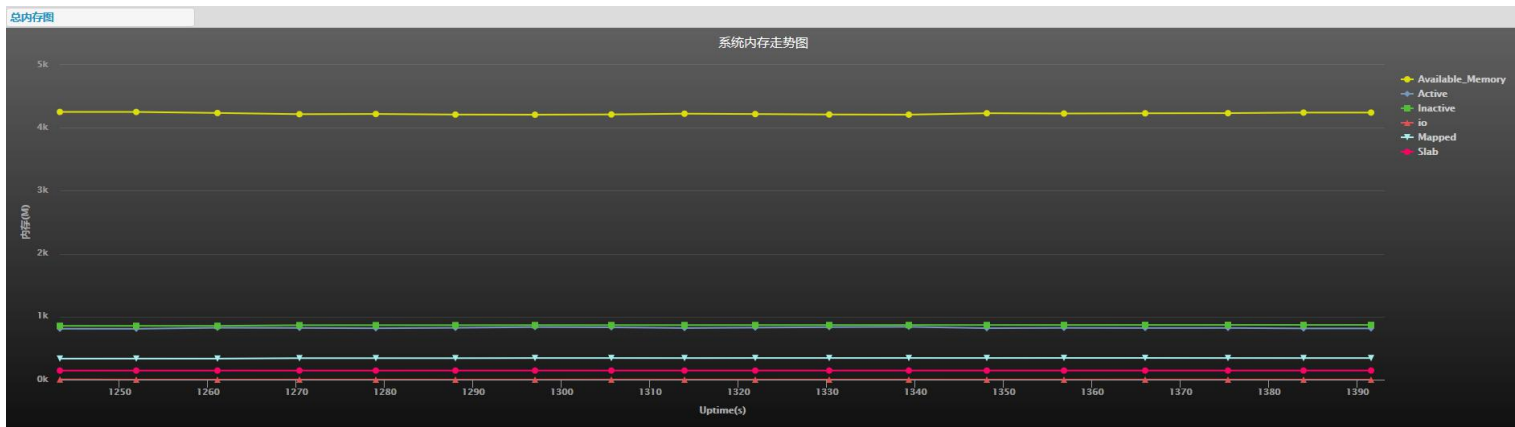
# 报告介绍（主频）



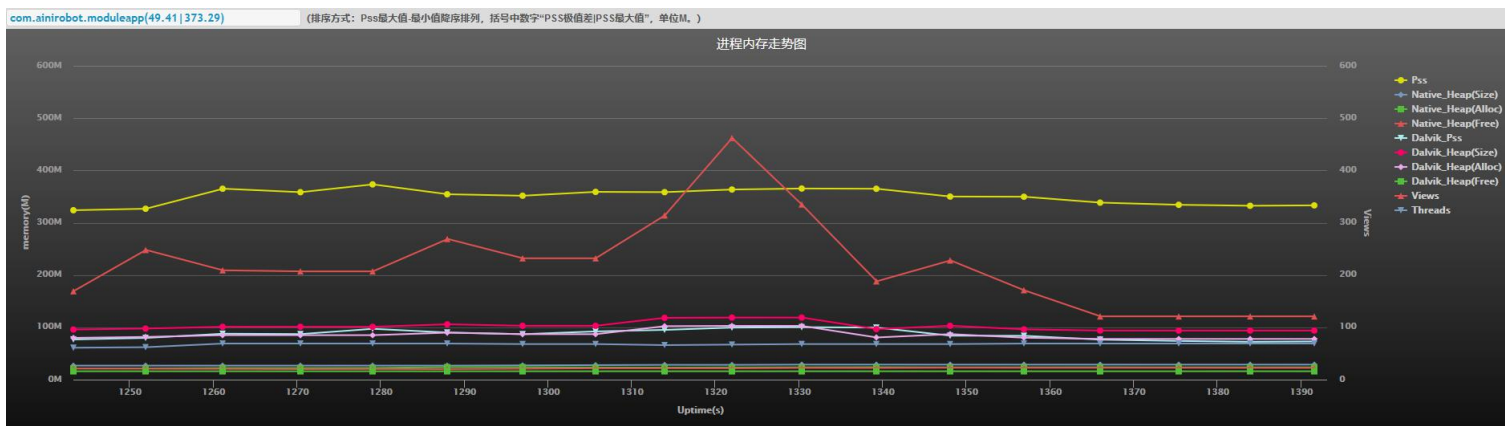
# 报告介绍 (cpu)



# 报告介绍（总内存）



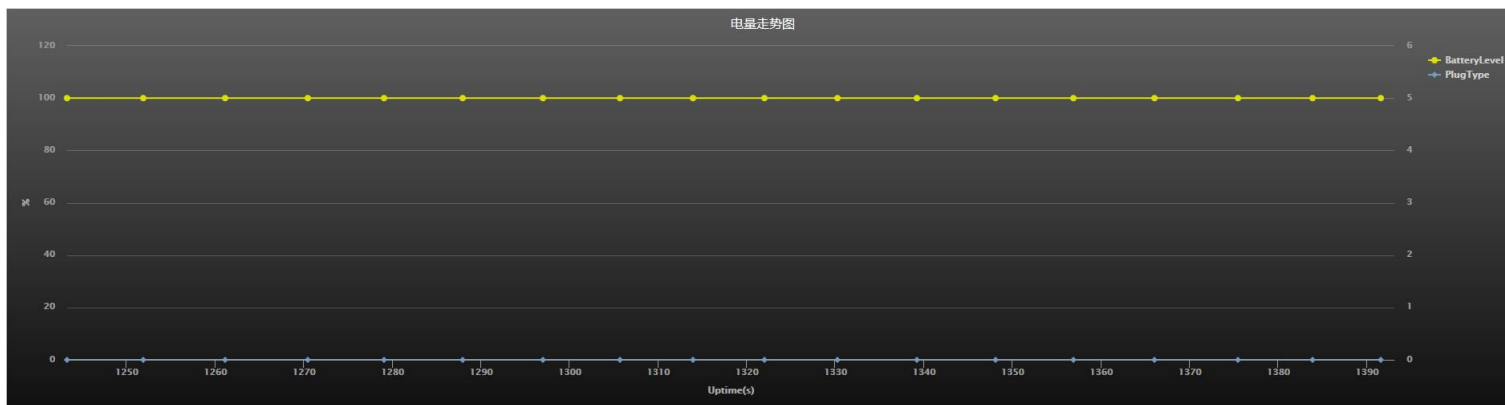
# 报告介绍 (进程内存)



注: 红点代表进程PID变化, 即进程重启过



# 报告介绍 (电量和帧率)



注：帧率包括配置的窗口和系统动效，选择框选取切换

# 报告介绍（温升）



注：温升sensor没有浮点数，保留了源数据格式，按type定义查看