

title: GACXXX\_A88\_AVNT\_SE\_211202.R5.49\_R周版本整体性能报告, 以及各项性能问题暴露

author: 欧杰, 莫国权

## 1 背景

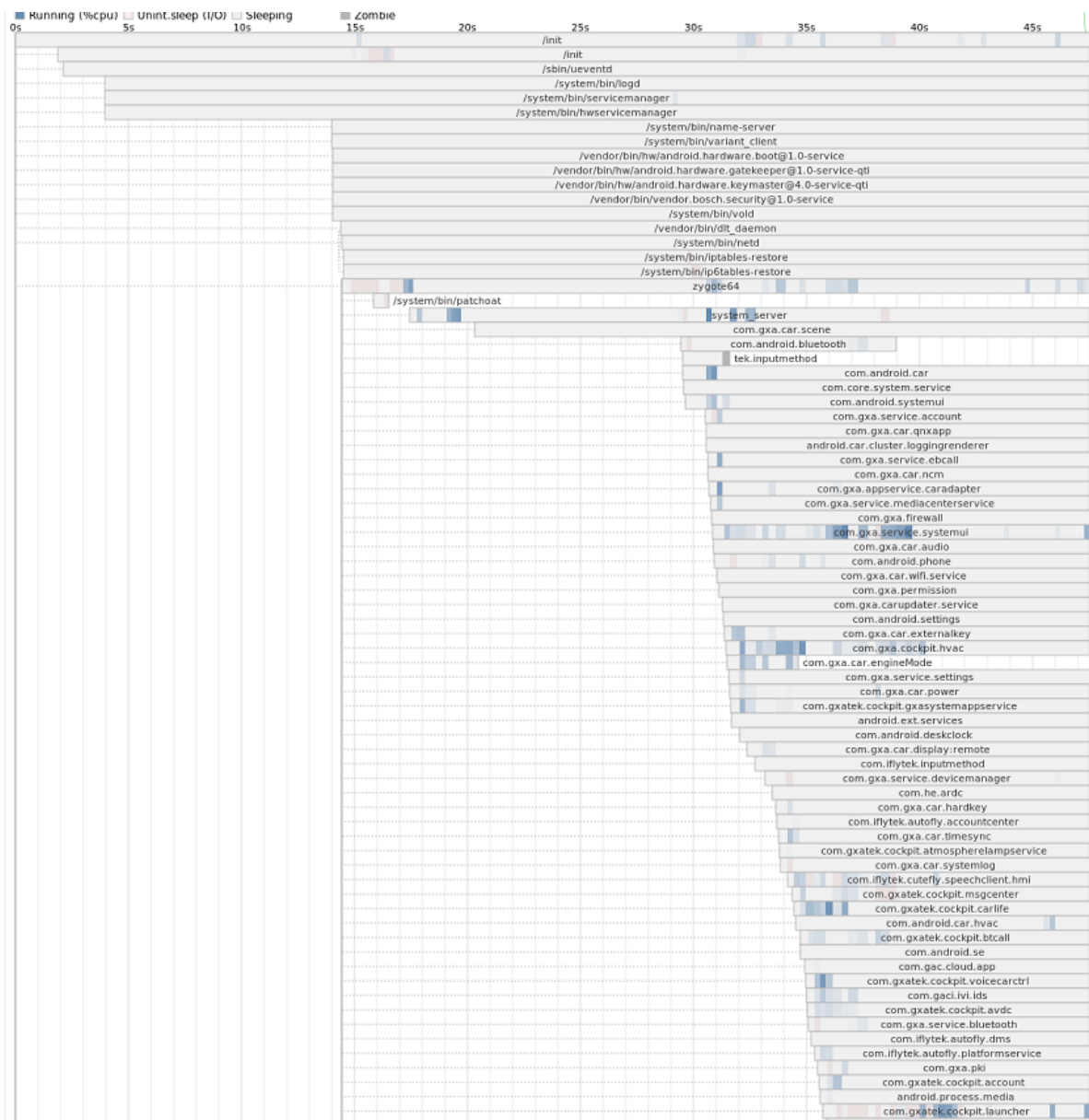
该报告中涉及到CPU, 内存, IO, View数据来自 GACXXX\_A88\_AVNT\_SE\_211202.R5.49\_R 周版本, 采集周期为573个周期, 采集时间为1.6小时的性能数据, 采集方式: monkey白名单测试。

系统启动时间基于没有优化合入的版本, 应用启动时间和apk包大小目前看有部分应用功能还没有全。

## 2.系统性能参数分析

该章节从系统层面分析当前系统各项性能指标。

### 2.1 启动耗时



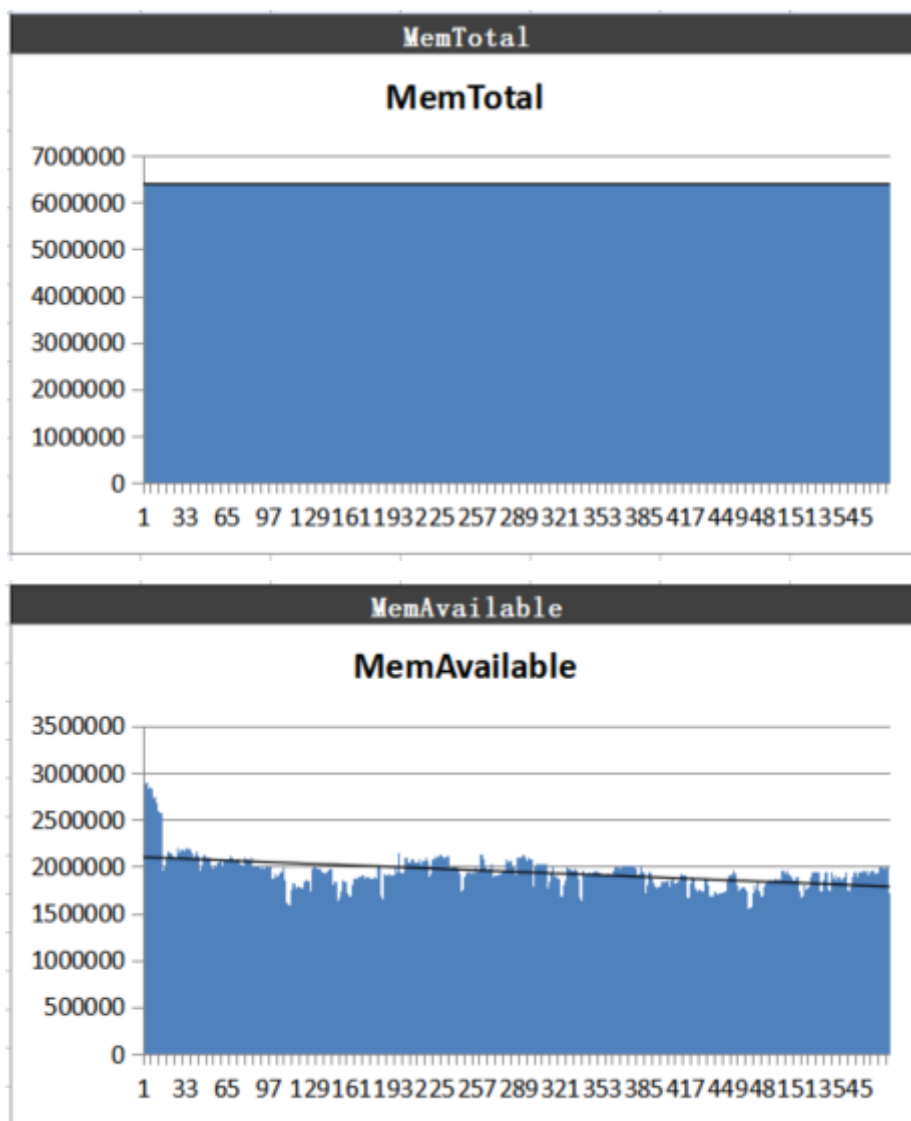
| 重要节点                    | timestamp       | 耗时ms  |
|-------------------------|-----------------|-------|
| init第一阶段                | 0               | 1719  |
| init第二阶段 (mount_all)    |                 | 1441  |
| zygote                  | 14.693          |       |
| PreloadClasses          | 17.249999       | 494   |
| PreloadResources        | 17.314211       | 64    |
| ZygotePreload           | 17.356779       | 677   |
| ZygoteInit              | 17.385883       | 706   |
| InitBeforeStartServices | 17.749919       | 171   |
| StartServices           | 34.535993(done) | 12756 |
| Starting phase 100      | 17.841818       |       |
| Starting phase 480      | 33.393858       | 15552 |
| Starting phase 500      | 33.398361       | 5     |
| Starting phase 520      | 33.464118       | 66    |
| Starting phase 550      | 33.468068       | 4     |
| Starting phase 600      | 34.415950       | 948   |
| Starting phase 1000     | 36.027039       | 1611  |
| boot_animation_done     | 36.023854(done) |       |
| FallbackHome            | 36.028943       | 1344  |
| LauncherActivity        | 37.697186       | 14499 |
| SystemUserUnlock        | 37.515143       | 1391  |

**当前Android系统启动时间为：52s！** 加上QNX系统启动应该会在60s以上。

从以上启动bootchart图、启动时间戳和耗时，可以看出有三个大耗时点需要调查。

- 需要调查为什么 **14.69s** 的时候才启动Zygote
- 需要调查哪个服务启动阻塞，导致启动延迟近**12s**
- 需要调查Launcher应用启动为什么耗费**14.5s**

## 2.2 内存

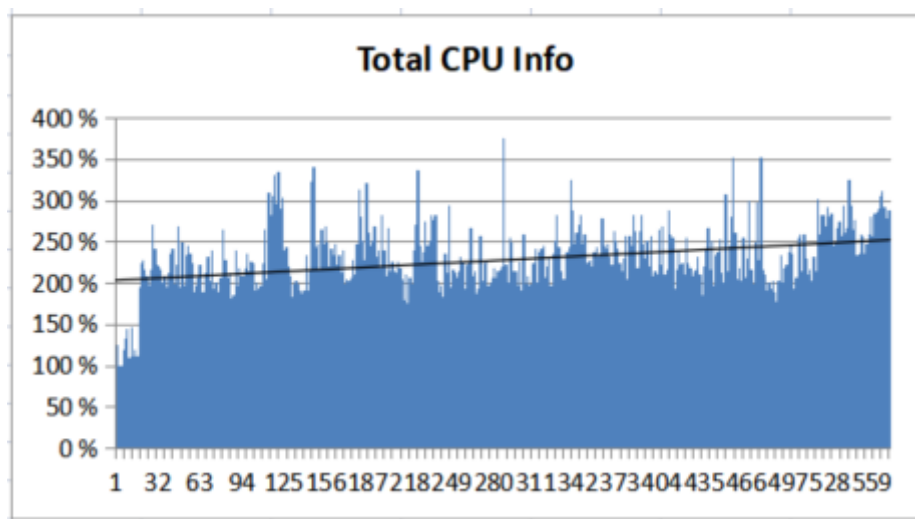


|        | 最大值   | 最小值   | 平均值   |
|--------|-------|-------|-------|
| 系统总内存  | 6241M | 6241M | 6241M |
| 系统可用内存 | 3093M | 1505M | 1897M |

#### 小结:

- 从当前系统整体来看，运行1.6小时，整体最低还剩1.5G左右，如果长时间运行，加上应用内存泄露，内存低于900M会触发系统LMK杀进程机制。**如果触发LMK机制，媒体等应用占大内存的应用如果在后台就有可能被系统杀掉。**

## 2.3 cpu



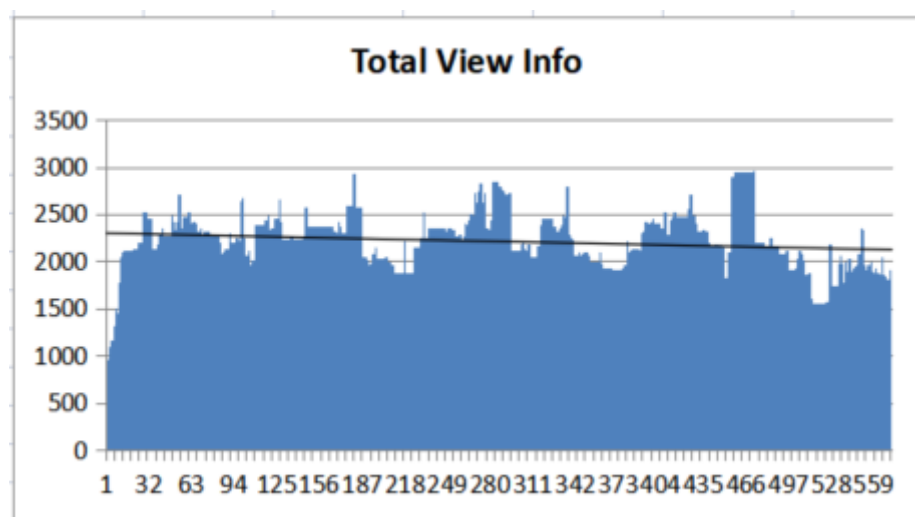
|         | 峰值   | 最小值 | 平均值  |
|---------|------|-----|------|
| 系统CPU占用 | 376% | 87% | 227% |

#### 小结:

- monkey模拟用户正常使用场景，CPU峰值为376%，49.35K DMIPS从图中看峰值出现的次数不多，说明当前系统CPU压力属于正常范围。后续需要观察压力测试CPU表现。
- 平均值为227%，占用Android分到算力的0.567，处于0.7健康值以下，需进一步看静态压力测试均值表现。

## 2.4 view

系统在某个时刻加载View的个数，可以侧面反应系统卡顿和应用的卡顿情况，如果一个界面view个数越多，那界面出现卡顿的概率会越高。官方推荐一个界面的view个数不超过80个对象。该值为参考值，需要根据需求以及应用本身布局、背景、RecyclerView过度加载view等很多因素有关，如果超过80个对象，需要应用先自查是否确有那么多View对象的必要。

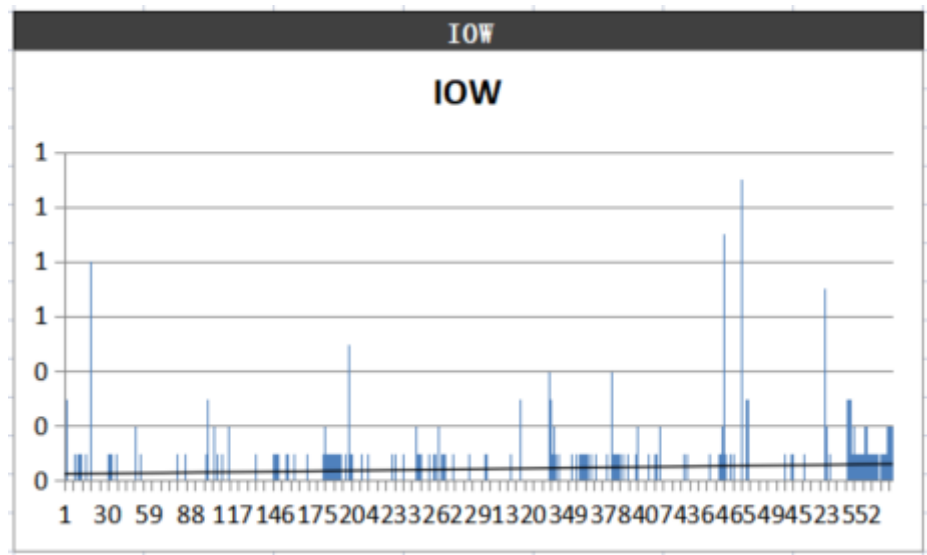


|           | 峰值   | 最低值 | 平均值  |
|-----------|------|-----|------|
| 系统总view个数 | 2970 | 583 | 2216 |

#### 小结:

- 从系统中总view均值来看，系统内存中缓存的view个数2216个，此值虽然不能反应系统是否卡顿，但是可以看出系统内存大量被占用的元凶就是View个数过多，因为通常一个应用最占内存的对象就是Bitmap对象。
- 每个应用FO需要根据后续小节对每个进程View个数超过80个的情况进程自查。

2.5 IO性能

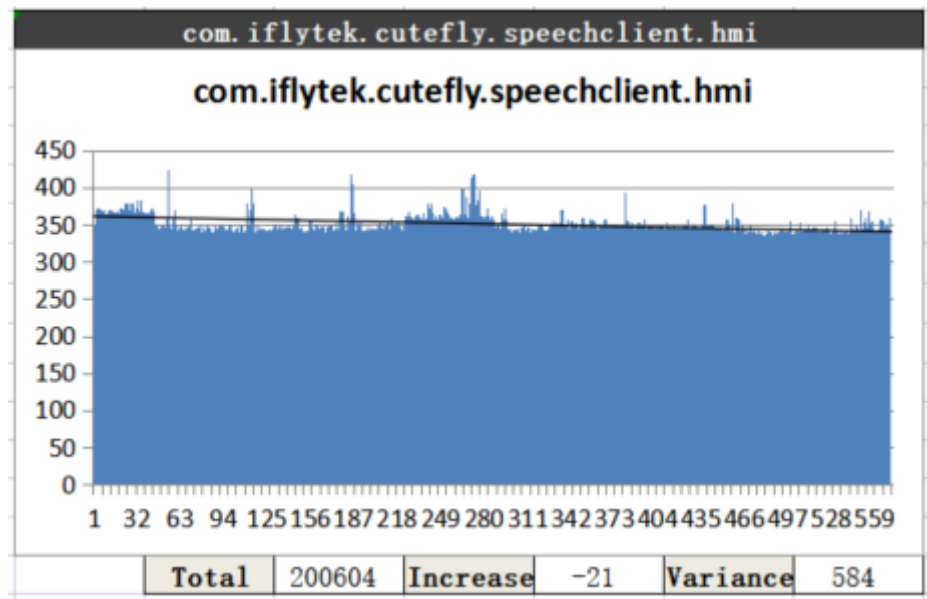


从上图中看，整个检测周期中IOW都比较低，说明当前CPU和IO都比较健康，暂时没有出现IO瓶颈现象。

3 应用内存问题暴露

该章节所有图横坐标单位为检测周期，纵坐标为应用内存大小，单位：M。

3.1 语音助理

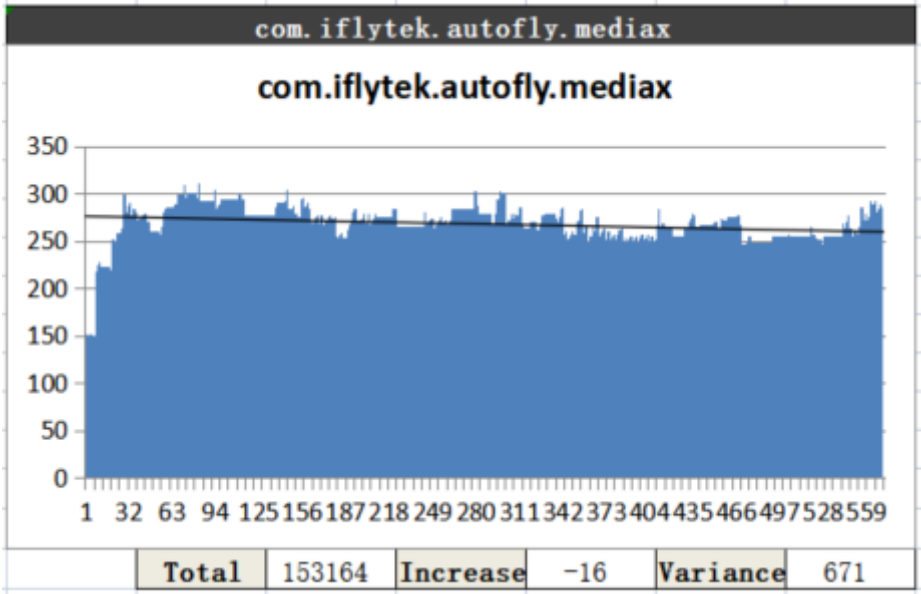


| 最大内存 | 最小内存 | 平均内存 | 内存抖动 | 内存泄露 | 大内存分配 |
|------|------|------|------|------|-------|
| 424  | 337  | 352  | 存在   | 无    | 存在    |

优化建议：

- 调查部分功能加载大量资源到内存的情况，看是否可以减少此部分资源加载，或者拆分资源加载，当前看有一次性加载80M左右资源到内存的情况。
- 大内存分配用对象池提高对象复用率。

3.2 媒体

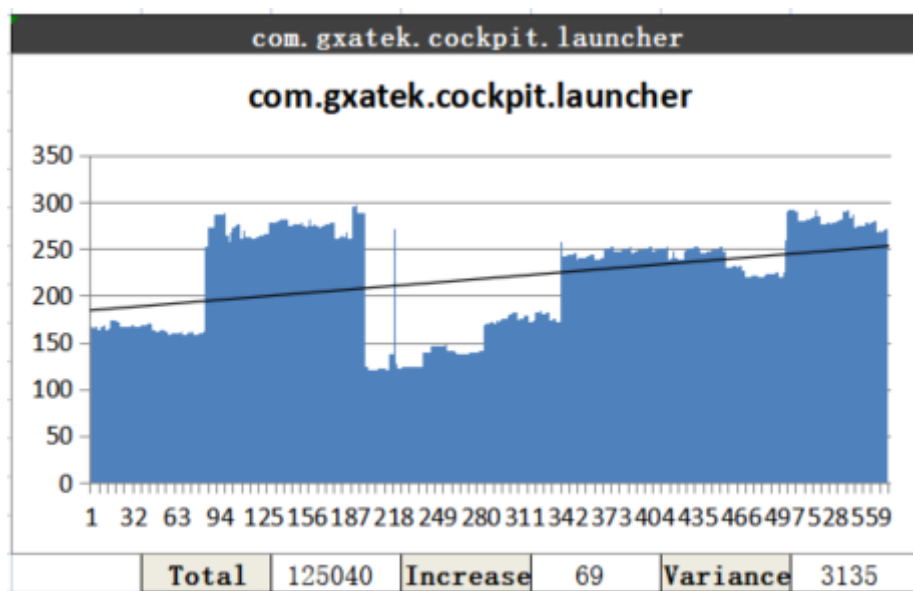


| 最大内存 | 最小内存 | 平均内存 | 内存抖动 | 内存泄露 | 大内存分配 |
|------|------|------|------|------|-------|
| 311  | 151  | 269  | 存在   | 无    | 存在    |

优化建议：

- RecyclerView按需加载bitmap，不要全部加载到内存中。
- 大内存分配用对象池提高对象复用率。
- 从151M跳到230M左右，再跳到290M左右一定是大量Bitmap加载，需要查看图片是否按需加载，布局是否合理，界面布局中是否有不必要背景，一个全局背景占用内存： $1920 * 1080 * 4 = 8.1M$ 。

3.2 Launcher

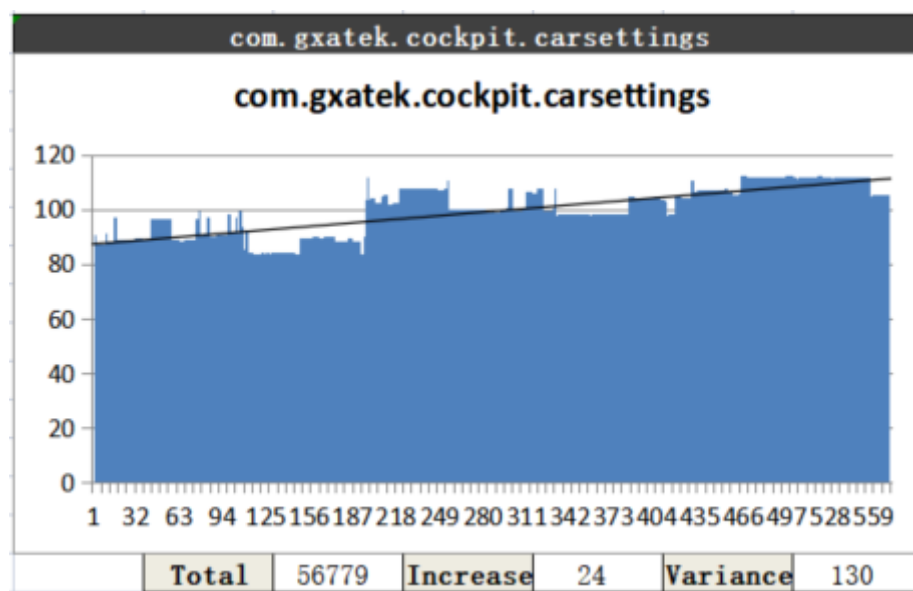


| 最大内存 | 最小内存 | 平均内存 | 内存抖动 | 内存泄露 | 大内存分配 |
|------|------|------|------|------|-------|
| 297  | 120  | 219  | 存在   | 存在   | 存在    |

优化建议：

- monkey跑单个应用保证稳定性。
- 确认大内存分配的场景，加载的文件是否可以拆分。
- 内存抖动使用对象池来提升对象复用。

### 3.3 车辆设置



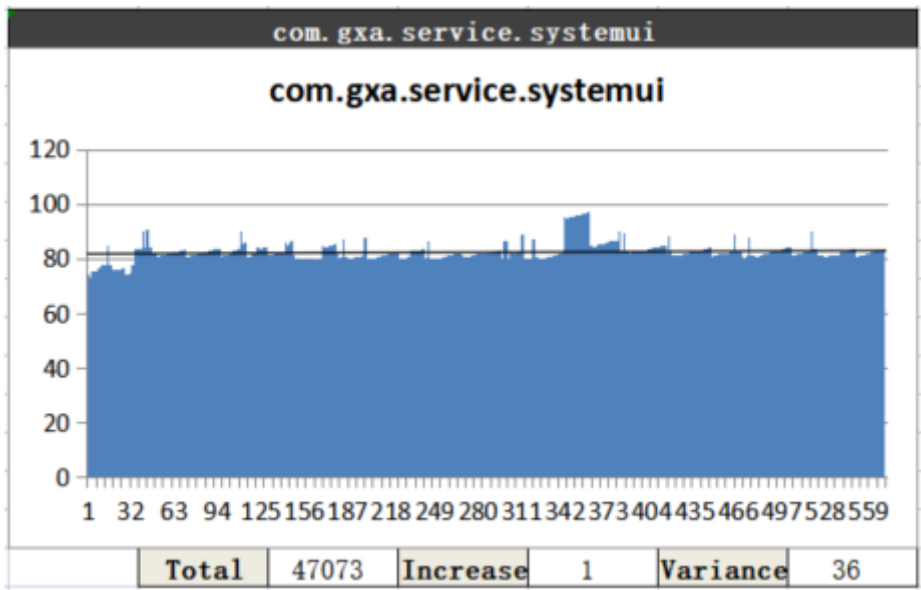
| 最大内存 | 最小内存 | 平均内存 | 内存抖动 | 内存泄露 | 大内存分配 |
|------|------|------|------|------|-------|
| 112  | 83   | 101  | 存在   | 存在   | 少量    |

优化建议：

- 使用leakcanary修复内存泄露问题。
- 调查阶梯内存情况，使用对象池增加对象复用。



3.4 systemui

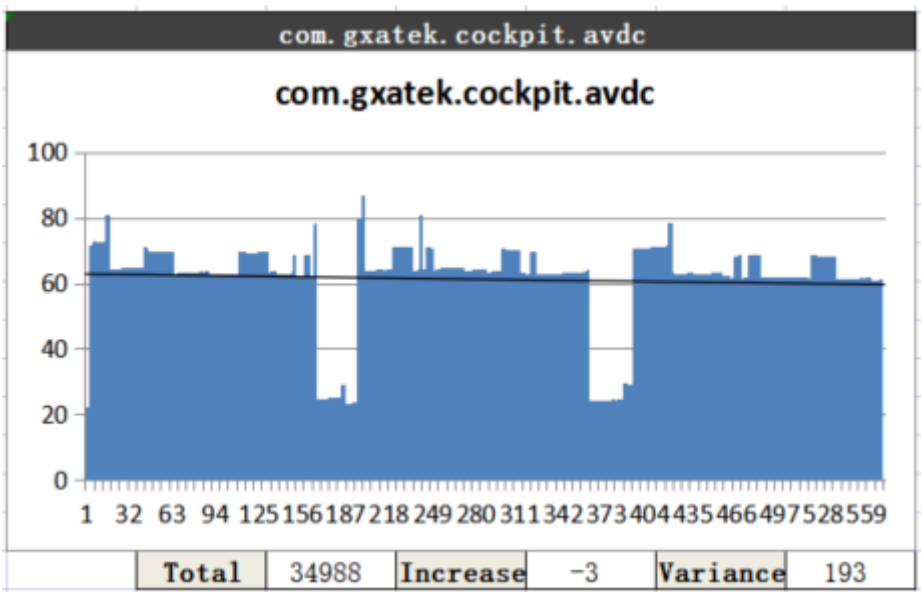


| 最大内存 | 最小内存 | 平均内存 | 内存抖动 | 内存泄露 | 大内存分配 |
|------|------|------|------|------|-------|
| 97   | 73   | 83   | 少量   | 无    | 少量    |

优化建议：

- 有一处出校阶梯内存，调查出现大对象分配的场景。

3.5 avdc

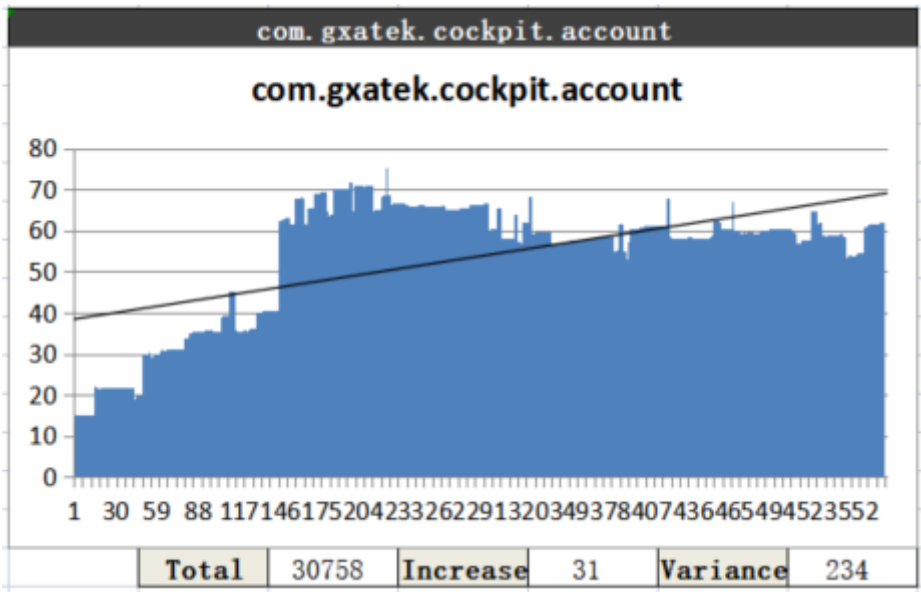


| 最大内存 | 最小内存 | 平均内存 | 内存抖动 | 内存泄露 | 大内存分配 |
|------|------|------|------|------|-------|
| 87   | 22   | 61   | 存在   | 无    | 存在    |

优化建议：

- 单应用monkey跑稳定性
- 调查大对象分配场景。

3.6 account

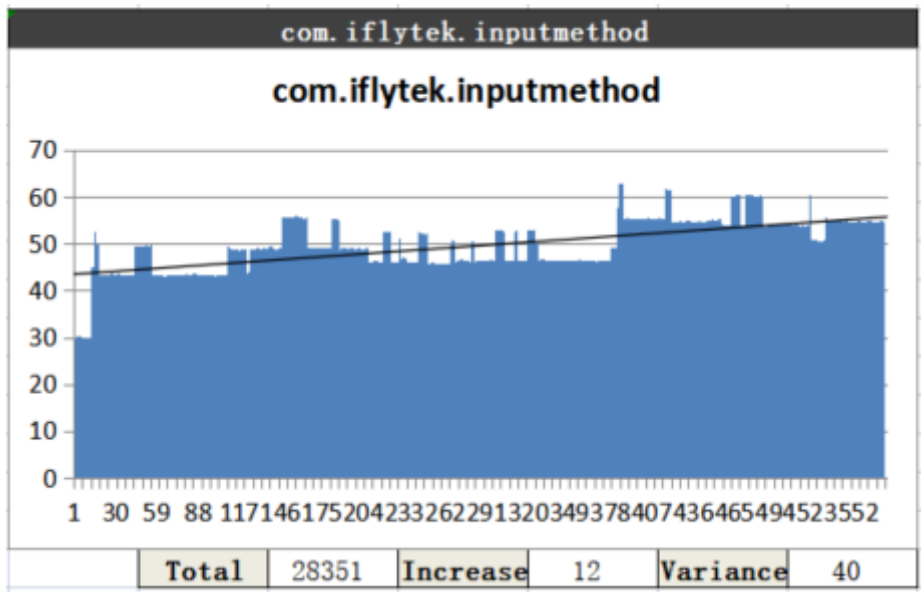


| 最大内存 | 最小内存 | 平均内存 | 内存抖动 | 内存泄露 | 大内存分配 |
|------|------|------|------|------|-------|
| 76   | 15   | 54   | 存在   | 存在   | 存在    |

优化建议：

- 1.使用leakcanary修复内存泄露问题。
- 2.调查大内存分配场景，使用对象池解决内存抖动。是否有频繁分配内存的情况。

3.7 输入法

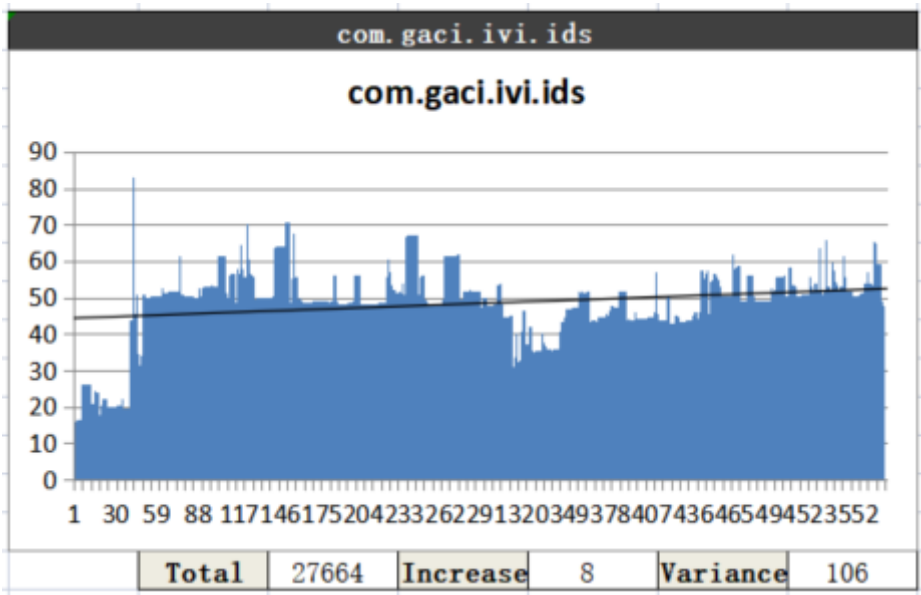


| 最大内存 | 最小内存 | 平均内存 | 内存抖动 | 内存泄露 | 大内存分配 |
|------|------|------|------|------|-------|
| 63   | 30   | 50   | 少量   | 少量   | 存在    |

优化建议：

- 1.使用leakcanary修复内存泄露问题。
- 2.调查出现阶梯内存是否是因为在后台界面被kill的情况。

3.8 安全助手

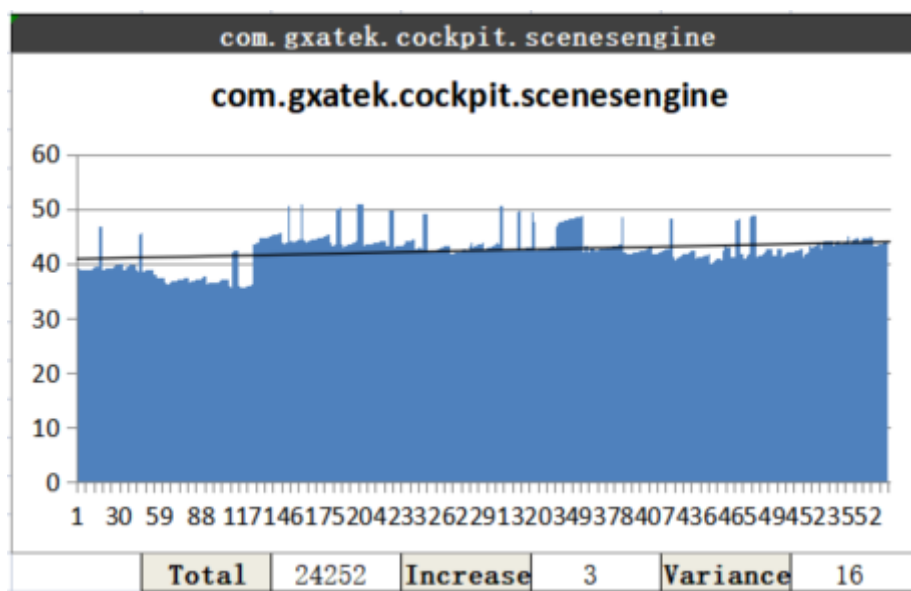


| 最大内存 | 最小内存 | 平均内存 | 内存抖动 | 内存泄露 | 大内存分配 |
|------|------|------|------|------|-------|
| 83   | 16   | 49   | 存在   | 少量   | 存在    |

优化建议：

- 1.使用leakcanary修复内存泄露问题。
- 2.调查大对象分配情况。
- 3.monkey单跑应用稳定性

3.9 场景引擎

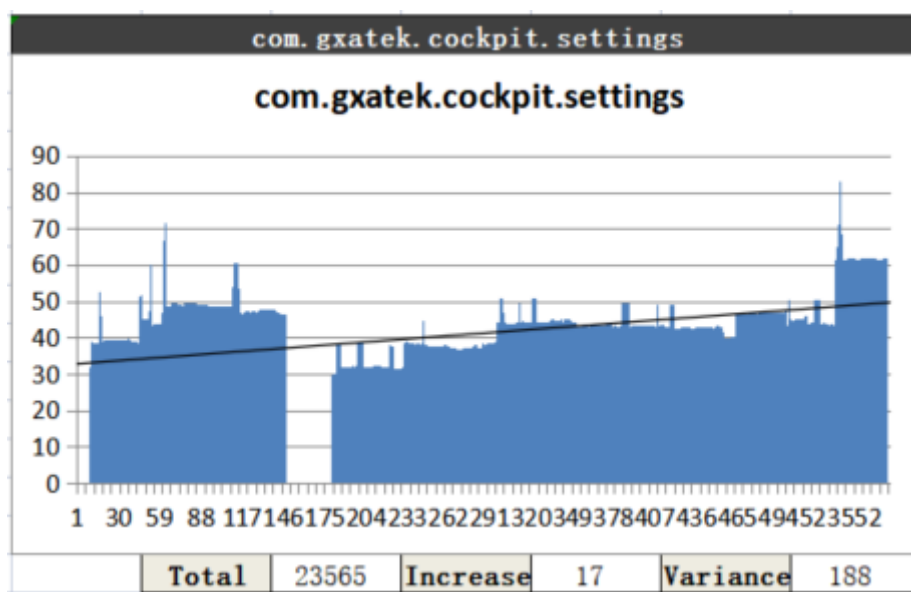


| 最大内存 | 最小内存 | 平均内存 | 内存抖动 | 内存泄露 | 大内存分配 |
|------|------|------|------|------|-------|
| 51   | 36   | 43   | 少量   | 无    | 少量    |

优化建议：

- 存在少量大对象分配的情况，调查是否可以复用。

### 3.10 设置

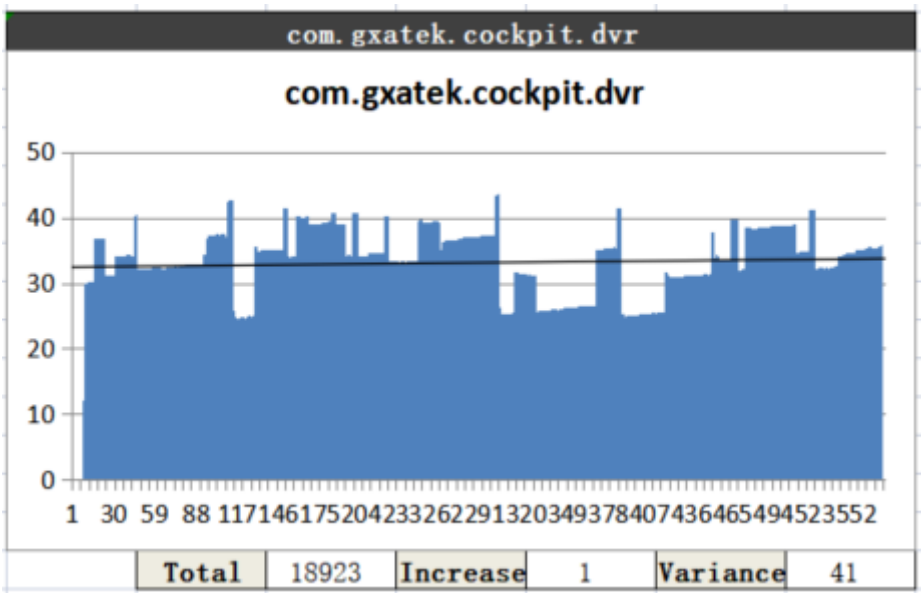


| 最大内存 | 最小内存 | 平均内存 | 内存抖动 | 内存泄露 | 大内存分配 |
|------|------|------|------|------|-------|
| 83   | 0    | 41   | 存在   | 少量   | 存在    |

优化建议：

- monkey跑稳定性
- 调查大内存分配场景，使用对象池解决内存抖动。

3.11 dvr

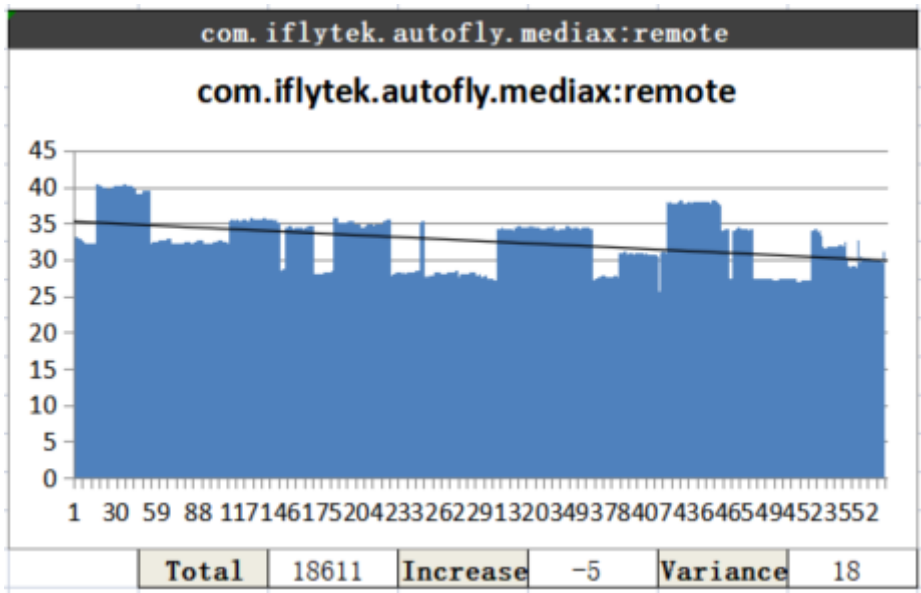


| 最大内存 | 最小内存 | 平均内存 | 内存抖动 | 内存泄露 | 大内存分配 |
|------|------|------|------|------|-------|
| 44   | 23   | 33   | 存在   | 无    | 存在    |

优化建议:

- 调查是否是界面到后台被回收。

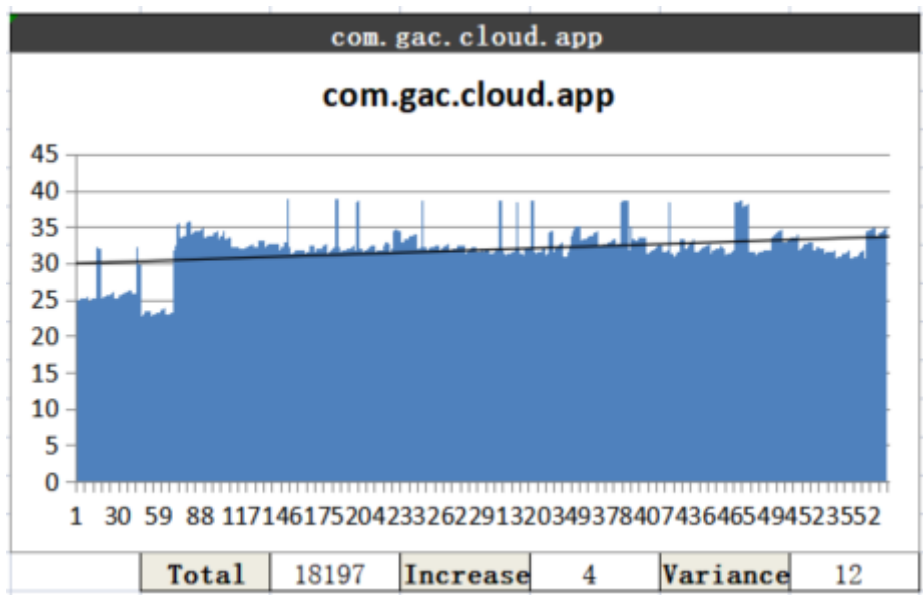
3.12 mediax:remote



| 最大内存 | 最小内存 | 平均内存 | 内存抖动 | 内存泄露 | 大内存分配 |
|------|------|------|------|------|-------|
| 40   | 26   | 33   | 存在   | 无    | 存在    |

优化建议:

3.13 GCS

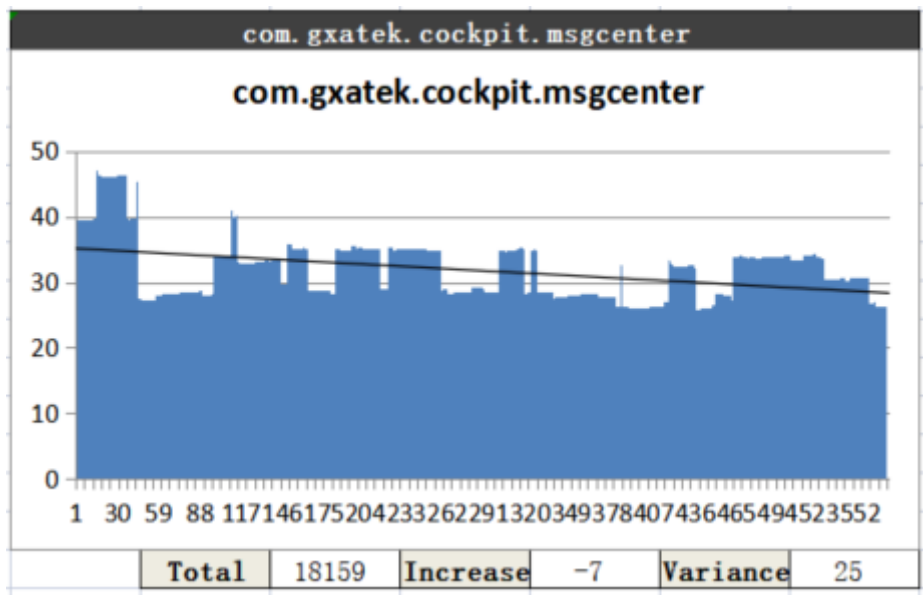


| 最大内存 | 最小内存 | 平均内存 | 内存抖动 | 内存泄露 | 大内存分配 |
|------|------|------|------|------|-------|
| 39   | 23   | 32   | 少量   | 少量   | 少量    |

优化建议：

- 调查针刺内存是否是因为隐藏界面被调起的情况。

3.14 msgcenter

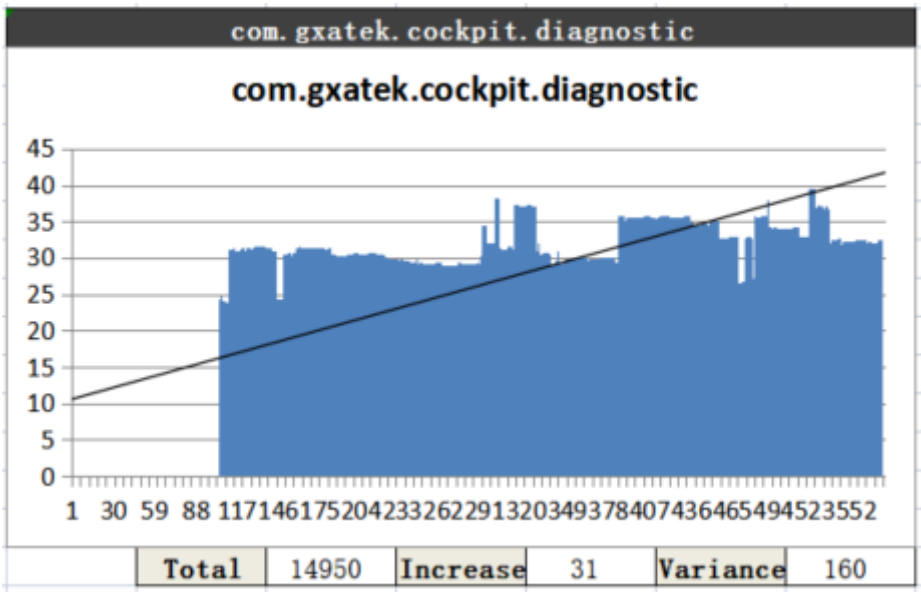


| 最大内存 | 最小内存 | 平均内存 | 内存抖动 | 内存泄露 | 大内存分配 |
|------|------|------|------|------|-------|
| 47   | 26   | 32   | 存在   | 无    | 存在    |

优化建议：

- 调查阶梯内存出现的原因。
- monkey跑应用稳定性。

3.15 诊断

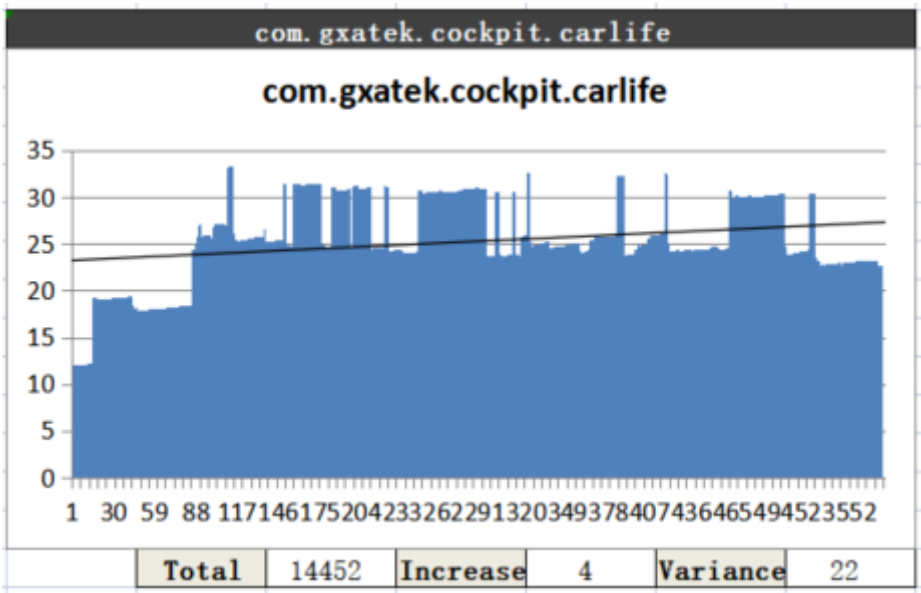


| 最大内存 | 最小内存 | 平均内存 | 内存抖动 | 内存泄露 | 大内存分配 |
|------|------|------|------|------|-------|
| 40   | 25   | 28   | 少量   | 少量   | 存在    |

优化建议：

- 调查大对象分配场景。

3.16 carlife

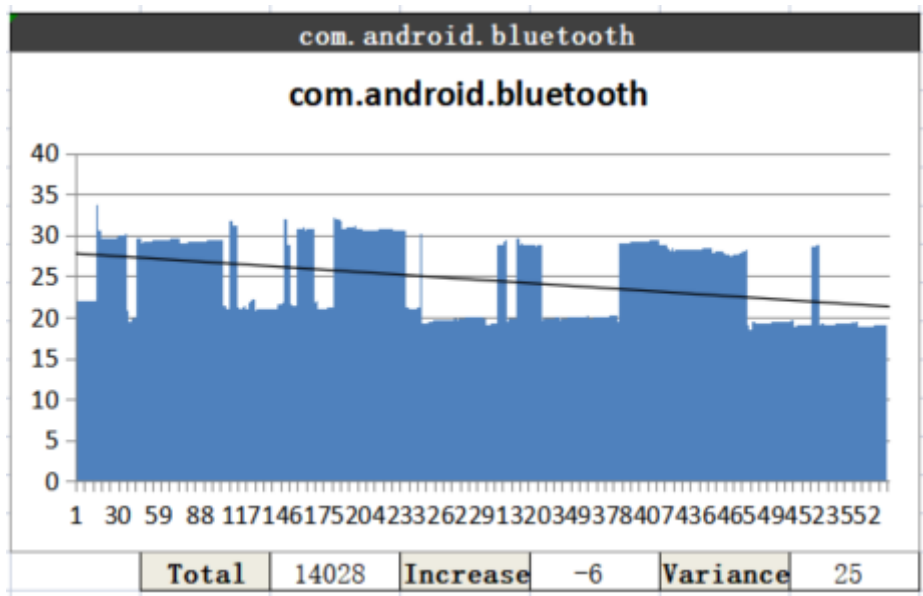


| 最大内存 | 最小内存 | 平均内存 | 内存抖动 | 内存泄露 | 大内存分配 |
|------|------|------|------|------|-------|
| 33   | 12   | 25   | 少量   | 少量   | 存在    |

优化建议：

- 调查阶梯内存出现的场景是否是界面被回收所致。

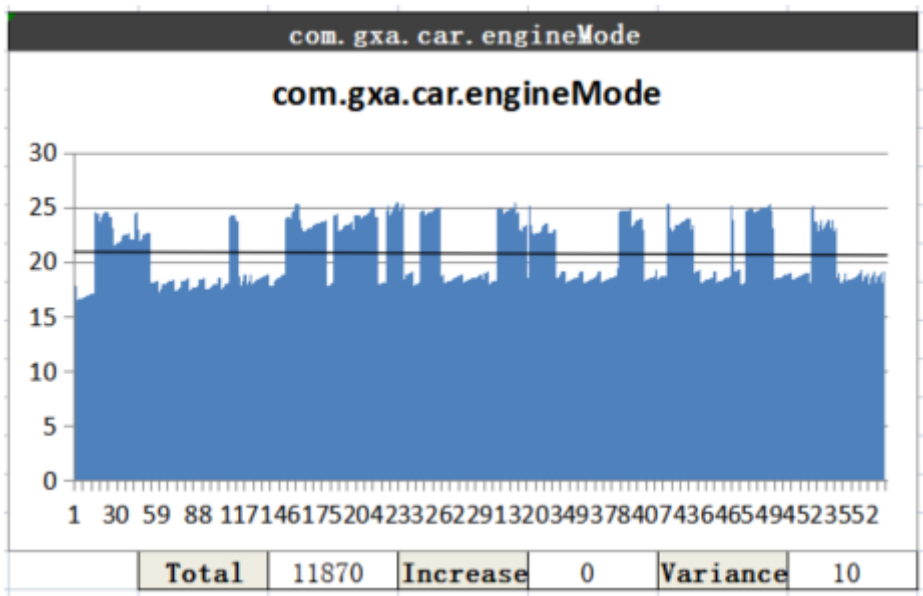
3.17 Bluetooth



| 最大内存 | 最小内存 | 平均内存 | 内存抖动 | 内存泄露 | 大内存分配 |
|------|------|------|------|------|-------|
| 34   | 19   | 25   | 少量   | 无    | 存在    |

- 优化建议：
- 调查大对象分配的场景。

3.18 engineMode



| 最大内存 | 最小内存 | 平均内存 | 内存抖动 | 内存泄露 | 大内存分配 |
|------|------|------|------|------|-------|
| 25   | 17   | 21   | 少量   | 无    | 存在    |

- 优化建议：
- 调查代码大对象分配的场景。



## 4 应用view

View过多会直接拉长绘制三大流程的时间，从而导致丢帧，最终给用户的感觉就是 滑动卡顿， 界面启动慢，滑动响应慢 等现象。因此此章节列出view对象创建比较多的应用，针对view多的问题，应用先自身进行一轮优化，优化方向可以参考：

- RecyclerView按需加载，不要将所有链接的图片都加载到内存中，按需求设置一个列队，更具用户滑动需求，加载队列的链接。
- 界面布局是否重叠，或者重叠过深。
- **设置了不必要的背景颜色或者背景图片**，此项请大家严格看，目前发现很多代码中都会有此问题，给layout设置背景，或者给控件设置不必要的背景。

一个应用界面不应该超过80个View对象，超过了，请务必确实是否有必要。**内存占用过大，LMK在adj值相等的情况下，会优先杀内存大的应用！！**

### 4.1 常见的场景

#### 4.1.1 阶梯view

如果出现阶梯view，说明当前界面退到后台就被回收了，下次启动需要重新加载界面，出现这种情况请**务必**检查view是否过多，是否必要，检测方法详见之前发过的一份卡顿分析文档《聚媒体应用卡顿分析.pdf》。

#### 4.1.2 针型View

出现针型view图，考虑是否有弹窗，针越长，弹窗是否view过于复杂。

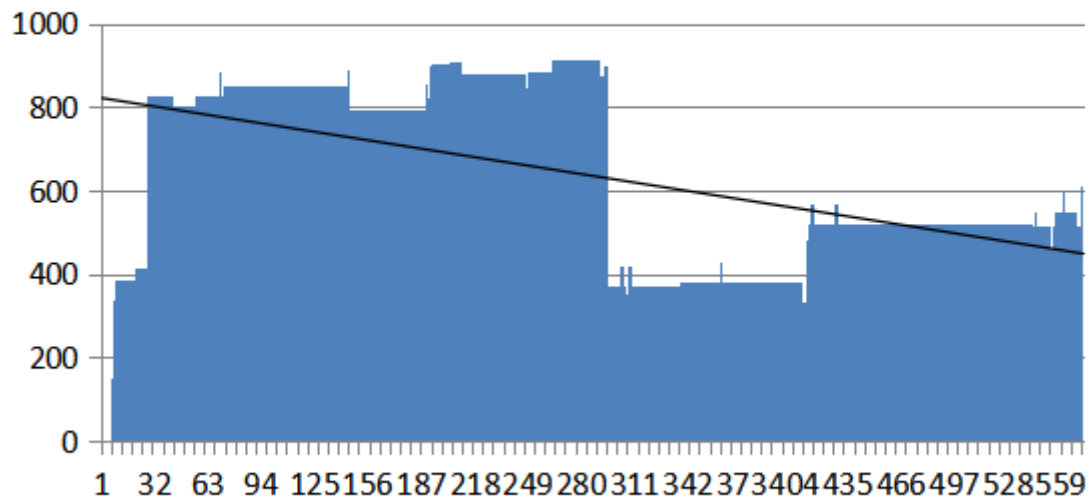
#### 4.1.3 杂乱阶梯View

出现杂乱阶梯View考虑有多个界面，每个界面的view都需要分析一下。一个layout或者一个控件就是一个view。杂乱的柱子越高，说明有界面view越复杂。

### 4.2 需要自查的应用

com.iflytek.autofly.mediax

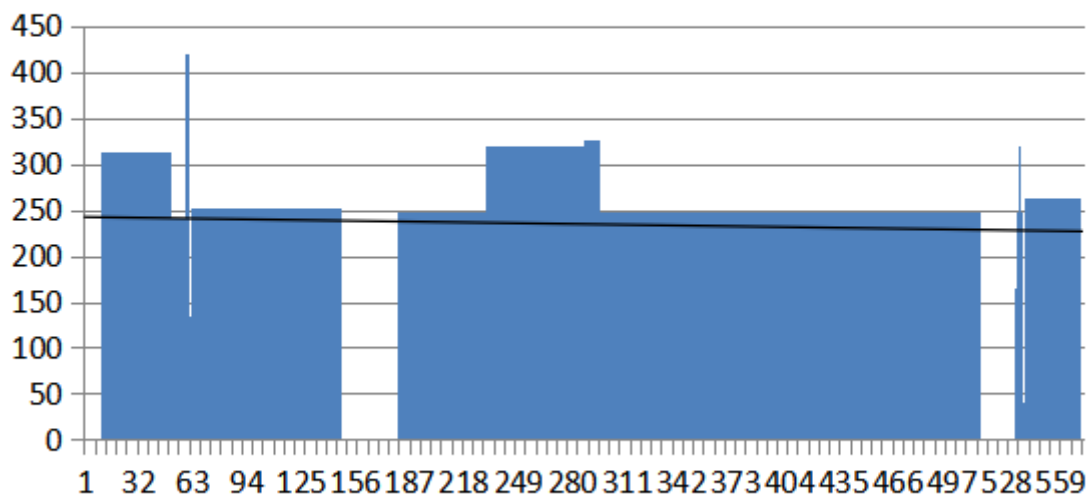
com.iflytek.autofly.mediax



|              |        |                 |      |                 |       |
|--------------|--------|-----------------|------|-----------------|-------|
| <b>Total</b> | 363698 | <b>Increase</b> | -374 | <b>Variance</b> | 48005 |
|--------------|--------|-----------------|------|-----------------|-------|

com.gxatek.cockpit.settings

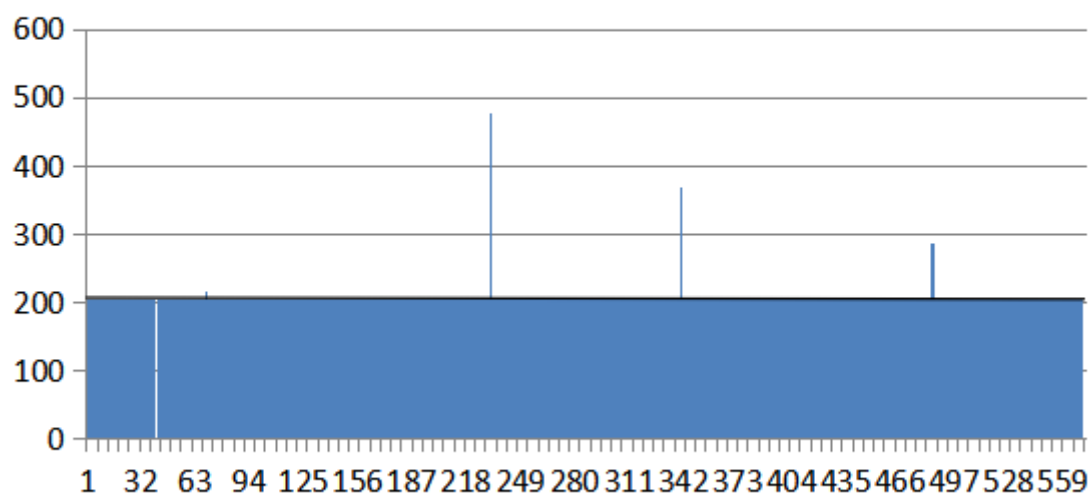
com.gxatek.cockpit.settings



|              |        |                 |     |                 |      |
|--------------|--------|-----------------|-----|-----------------|------|
| <b>Total</b> | 134073 | <b>Increase</b> | -16 | <b>Variance</b> | 7682 |
|--------------|--------|-----------------|-----|-----------------|------|

com.gxa.service.systemui

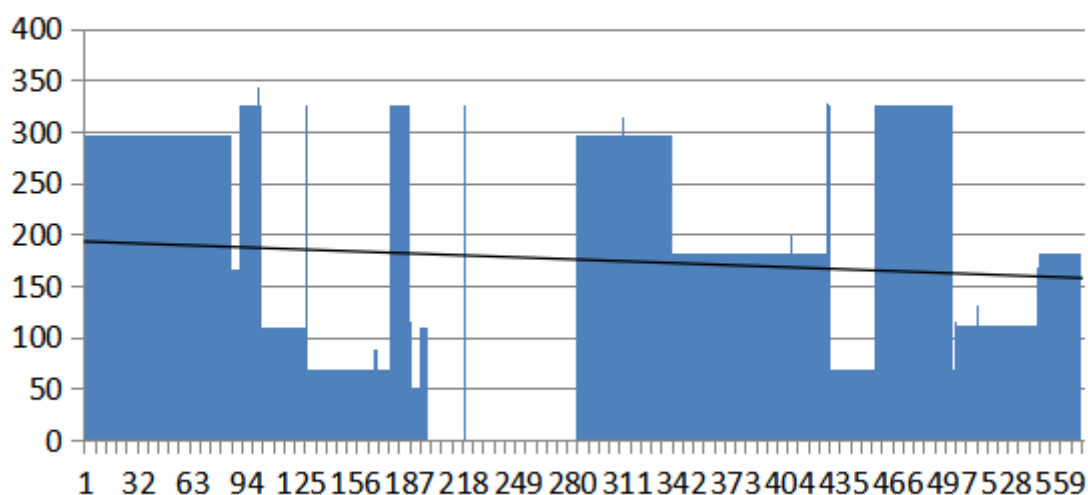
com.gxa.service.systemui



|       |        |          |    |          |     |
|-------|--------|----------|----|----------|-----|
| Total | 117174 | Increase | -2 | Variance | 409 |
|-------|--------|----------|----|----------|-----|

com.gxatek.cockpit.launcher

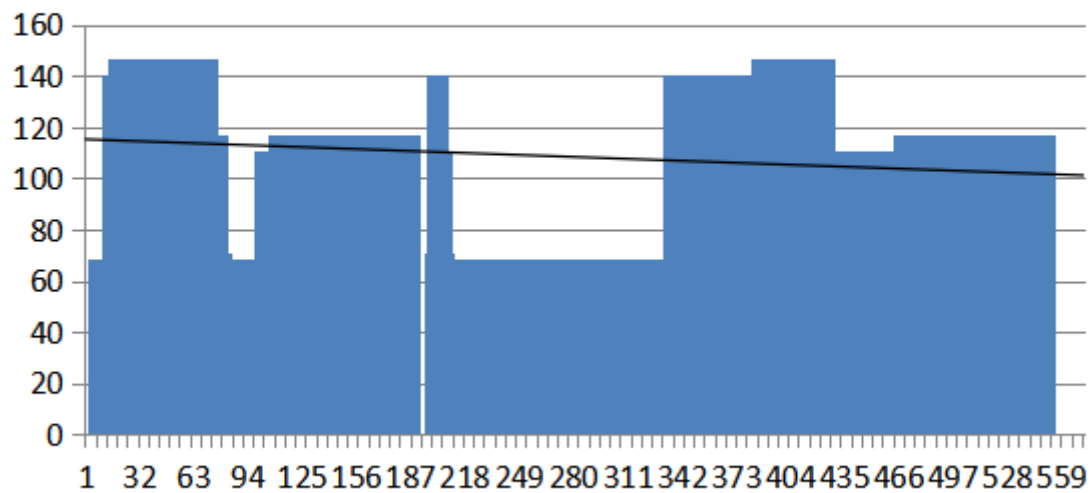
com.gxatek.cockpit.launcher



|       |        |          |     |          |       |
|-------|--------|----------|-----|----------|-------|
| Total | 100327 | Increase | -36 | Variance | 13242 |
|-------|--------|----------|-----|----------|-------|

com.gxatek.cockpit.carsettings

com.gxatek.cockpit.carsettings



Total

61905

Increase

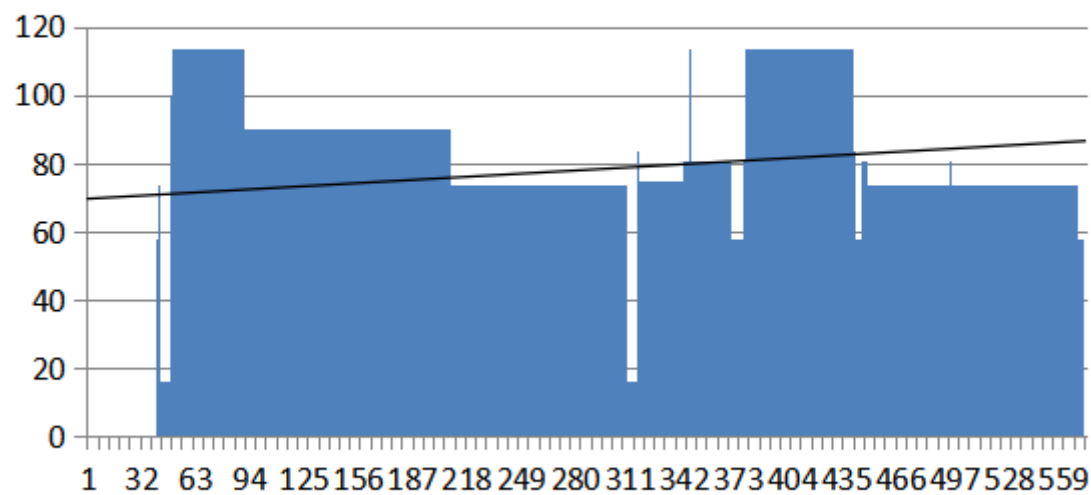
-14

Variance

1269

com.gaci.ivi.ids

com.gaci.ivi.ids



Total

44671

Increase

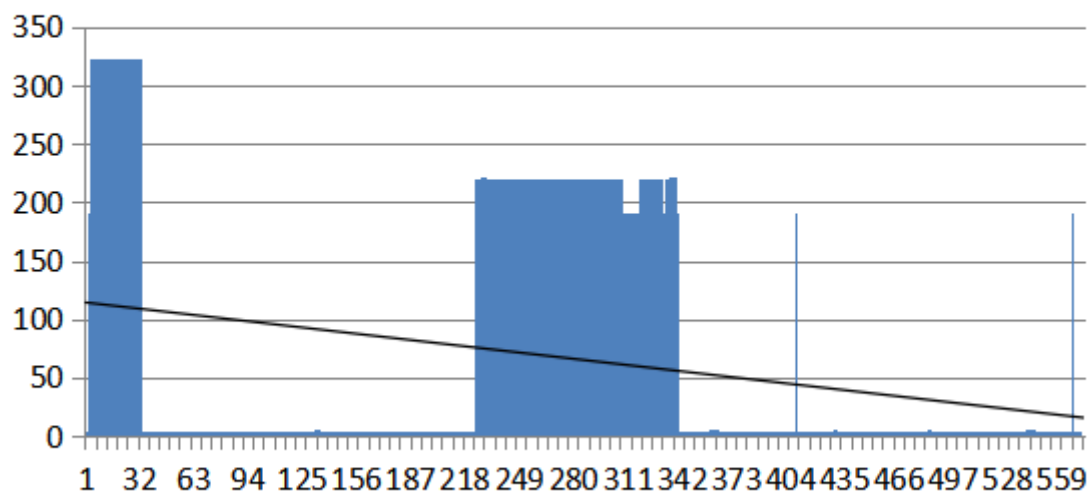
17

Variance

804

com.iflytek.cutefly.speechclient.hmi

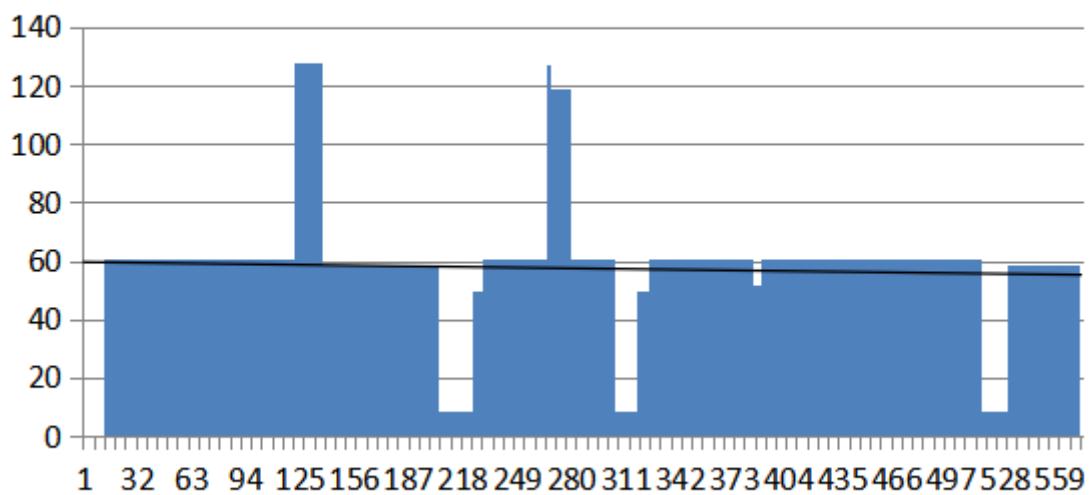
com.iflytek.cutefly.speechclient.hmi

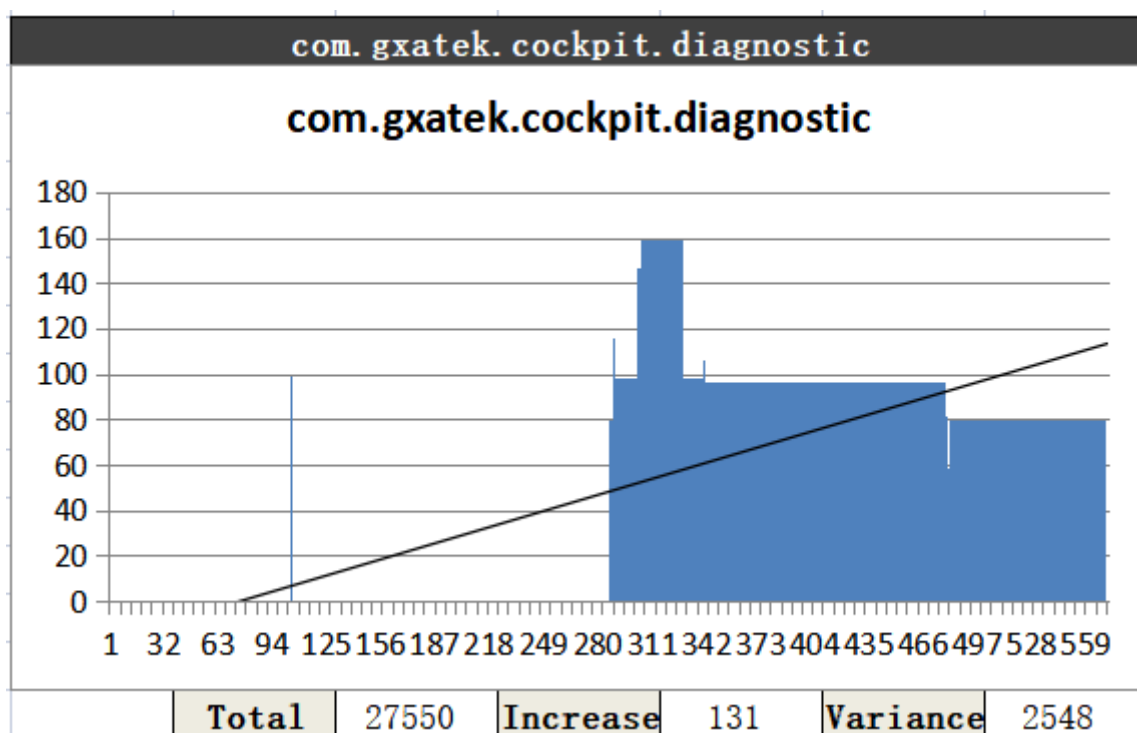
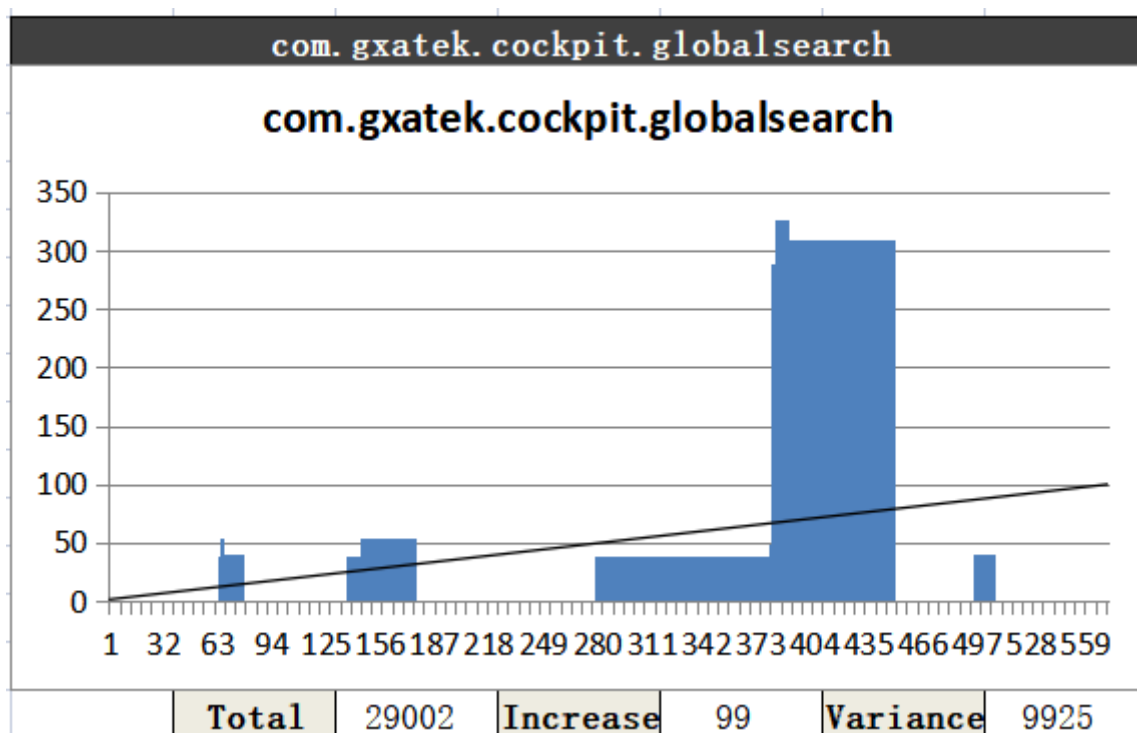


|       |       |          |     |          |       |
|-------|-------|----------|-----|----------|-------|
| Total | 37168 | Increase | -99 | Variance | 11042 |
|-------|-------|----------|-----|----------|-------|

com.iflytek.inputmethod

com.iflytek.inputmethod





## 5 启动时间

应用启动时间直接影响用户体验，针对轻量级严格要求在1秒之内，Google建议在500ms内，针对超过1秒的应用需要分析启动耗时原因，哪些资源是必须加载的。详细方法可以参考《GOS Launcher 启动耗时分析以及优化建议》。

### 5.1 需要自查的应用

| 应用名称               | 包名                            | 启动均值(ms) |
|--------------------|-------------------------------|----------|
| Launcher           | com.gxatek.cockpit.launcher   | 14499    |
| 聚合媒体（在线、本地音乐、本地电台） | com.iflytek.autofly.mediax    | 1888     |
| 爱奇艺                | com.qiyi.video.pad            | 1014     |
| 车服务                | com.gxatek.cockpit.carservice | 750      |
| Setting            | com.gxatek.cockpit.settings   | 632      |
| 天气                 | com.gxatek.cockpit.weather    | 621      |

目前很多功能还没有完全，是测试界面。

## 6 APK包大小

APK包大小会直接影响系统包大小，还会直接拉慢系统启动过程中PKMS扫描APK包的速度，导致提供启动耗时增加，更多的是增加了升级包的大小。

Google推荐的APK包大小最大不建议超过150M。因此需要应用先自查一轮哪些资源可以废弃，哪些图片可以转化为webp。详细优化方案见《应用APK瘦身调查与优化方案》。

### 6.1 需要自查的应用

| 应用名称               | 包名                                   | 包大小  |
|--------------------|--------------------------------------|------|
| 聚合媒体（在线、本地音乐、本地电台） | com.iflytek.autofly.mediax           | 131M |
| Launcher           | com.gxatek.cockpit.launcher          | 171M |
| 语音助理               | com.iflytek.cutefly.speechclient.hmi | 196M |
| 输入法                | com.iflytek.inputmethod              | 33M  |
| 天气                 | com.gxatek.cockpit.weather           | 34M  |
| 我的车                | com.gxatek.cockpit.carsettings       | 38M  |
| 消息中心服务             | com.gxa.service.messagecenter        | 52M  |
| 车服务                | com.gxatek.cockpit.carservice        | 56M  |
| 小程序容器              | com.gxatek.cockpit.miniprogram       | 56M  |