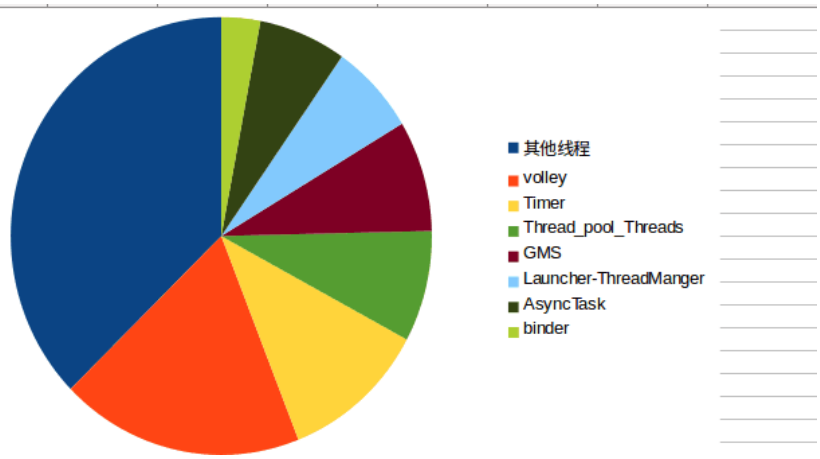
launcher线程优化方案

场景1:launcher启动后大约30s统计,不操作

总线程数:**134**

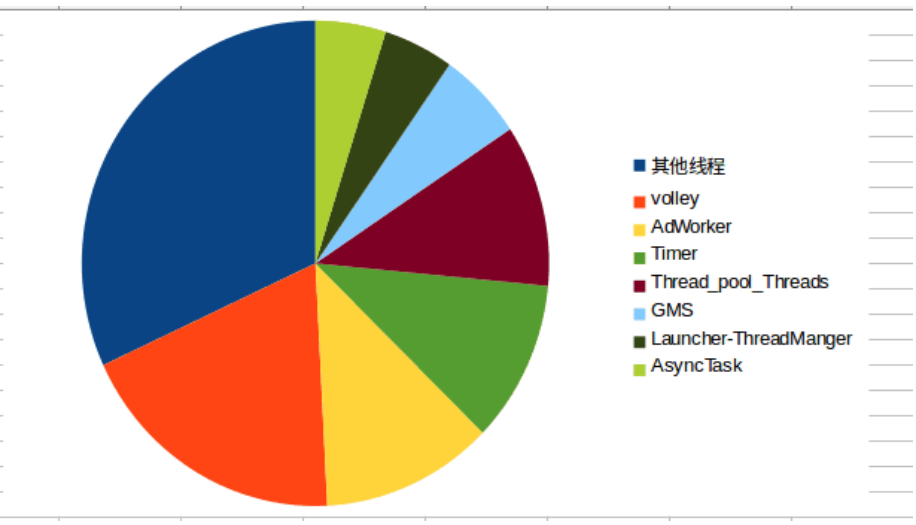
其他线程	50
volley	25
Timer	15
Thread_pool_Threads	11
GMS	11
Launcher-ThreadManger	9
AsyncTask	9
binder	4



场景2:launcher启动后进入各种操作,切换文件夹,换主题,应用商店等等

总线程数:**189**

其他线程	59
volley	35
AdWorker	22
Timer	20
Thread_pool_Threads	20
GMS	11
Launcher-ThreadManger	9
AsyncTask	9
binder	4



优化方案

1. Volley
   Volley线程数是25(场景1)/35个(场景2),过多,因为当前时间点有多个volley RequestQueue(每个quene包含4个NetworkDispatcher+1个cacheDispatcher), 主要包含两大功能非图片网络数据和图片网络数据, 现已优化为2个volley模块, 至少优化15个线程
2. Timer线程过多,占用16个(场景1)/20个(场景2),计划自定义Timer(handler+线程池)替代,动态优化
3. ThreadManager实现统一化,当前项目中充斥这各种Threadmanager,,大致分为以下几类

序号	路径	原理	备注
1	~~com.cleanmaster.net.ThreadManager~~	~~线程池~~	~~无用代码,可以删除~~
2	com.locker.net.ThreadManager	线程池	多任务并发
3	com.locker.cmnow.market	HandlerThread	不同种类并行,同一种类串行,同一种类的线程不存在线程安全问题,可以与**4**合并
4	com.ksmobile.business.sdk.utils.ThreadManager	HandlerThread	不同种类并行,同一种类串行,同一种类的线程不存在线程安全问题,可以与**3**合并
5	com.ksmobile.business.sdk.net.ThreadManager	线程池	主要是网络请求和图片下载(bitmapcache模块)

4. 线程池实例太多9个,线程数11个,这部分可以使用ThreadManager管理, 并且设置超时时间,一旦到达超时时间,如果当前池中没任务,就结束线程,避免无用站坑
5. 重写AsyncTask
   系统AsyncTask的机制导致AsyncTask的最多有cpu+1个线程永远不会结束,这样会操作线程的浪费,
6. AdWorker的优化,可能属于第三方库,