

[SCA] 质量优化框架 - 调研文档

一、背景

随着APP复杂度上升而带来的质量问题，需要调研是否有框架能支持对APP质量做优化。

二、方案调研

2.1 Booster功能模块调研

	功能	作用	测试结果	是否推荐	备注
性能优化	静态分析	检查的问题： 1. 发现潜在的性能问题，如：可能阻塞主线程/ UI 线程的 API 调用； 2. 发现风险 API 调用； 3. 分析依赖关系；	生成dot格式文件，每个类都会生成一个，可读性较差。		API 黑名单，可以借鉴这个名单，通过自定义 Lint 实现这个功能，可以方便集成到我们的代码检查中。
	线程优化	1. 线程优化 a. 将 corePoolSize 设置为 0 b. 为 maxPoolSize 设置上限 c. 允许核心线程在空闲时自动销毁 d. 尽可能将 HandlerThread 替换成 SingleThreadPoolExecutor 2. 应用的线程重命名，方便定位线程导致的奔溃问题。		待定	WMS和SPX线上未发现由于线程过多导致的OOM问题
	SharedPreferences 优化	通过自定义 SharedPreferences 避开 QueuedWork 在 onPause(), onDestroy() 等生命周期回调时在主线程中的同步操作。	\	X	使用内部的统一存储框架（MMKV）
	WebView 预加载	在 Application onCreate() 回调中注入预加载代码，当主线程 IDLE 时，启动 Chromium 引擎	首次加载速度提升 200ms左右	待定	提升效果不是特别明显，并且仅作用在应用首次打开WebView
包体积瘦身	PNG 图片压缩	有损压缩 PNG 图片，在Android 的构建流程 mergeRes 和 processRes 任务之间插入 PNG 压缩任务	WMS release包大小减少了 80KB，但是构建时间增加了30s左右	待定	项目中使用PNG图片不多，基本都是使用SVG
	WebP 图片压缩	与 PNG 图片压缩的思路一样，也是在 mergeRes 和 processRes 任务之间插入 WebP 压缩任务	\	X	WMS项目中使用 webp 图片很少，SPX项目没使用webp图片
	ZIP 文件压缩	在打包的过程对 AndroidManifest.xml、res/*、resources.arsc 压缩	对WMS项目进行测试几乎没有减少包大小	X	风险点：Google 官方并不推荐对 resources.arsc 和 so 进行压缩，这样会导致它们不能被直接 mmap 到内存
	资源索引内联	library 的 R 只是 application 的 R 的一个子集，所以，只要有了全集，子集是可以通通删掉的，而且，application 的 R 中的常量字段，一旦参与编译后，就再也没有利用价值（反射除外）	WMS release包减少 0.3MB	待定	ConstraintLayout 部分属性是不能删除的，框架有处理
	移除冗余资源	针对不同的地区删除一些冗余的像素密度 drawable	\	X	项目中基本都是使用SVG图片，一般不存在这个问题
系统 bug 修复	为系统奔溃兜底	通过Hook ActivityThread，给系统崩溃加兜底方案。	\	X	崩溃有专项在做
	Finalizer 导致的 TimeoutException	QC过程虚拟机内部Finalizer 守护线程超时异常，且超时参数在虚拟机内部设置，无法通过其他方式加大此时间戳，booster采用的方式是直接杀死该线程，并没有什么实质性的影响。	\	待定	目前WMS线上没有出现这中类型奔溃
	资源为 null 的问题	覆盖安装之后，App 由 Broadcast 唤起，由于系统 bug 导致加载了旧的 APK，这时可能会有静态资源或者System Service报空指针异常。	\	待定	目前WMS线上没有出现这中类型奔溃
	Android 7.1 Toast 崩溃	Android 7.1（API level 25）中Toast使用时可能会抛出异常，使用使用try-catch进行兜底处理	\	X	项目中已经使用统一的ToastUtils 工具类解决该问题

其它特性	Release 构建依赖检查	对 Release 构建的依赖库版本进行检查，避免依赖 SNAPSHOT 版本的库	\		可以在合入版本分支或者构建 Release包做检测
	Android 权限清单	用于列出工程中依赖的 AAR 中使用的所有 Android 权限，方便开发者排查权限相关的问题	\	X	对项目作用不大
	动态库清单	用于列出工程中依赖的AAR中使用的所有动态库，方便开发者排查动态库相关的问题	\	X	对项目作用不大
	构建中间产物清单	在针对 Android 工程开发 Gradle 插件的过程中，经常会用到一些内部 API，需要了解 Android 构建过程的细节和内部实现，以便于在合适的时机插入特定的 Task 来获取 AGP（Android Gradle Plugin）的中间产物，用于列出 Android 工程的构建中间产物。	\	X	对项目作用不大

三、总结

Booster 有17个功能模块，分了四个大类：

1. 性能优化：静态分析的思路可以借鉴，SharedPreferences优化我们使用MMKV了，线程优化和webview预加载，现阶段对我们收益不大。
2. 包体积瘦身：我们项目apk基本都在20MB以内，普遍不大，现阶段做这个优化对我收益不大。
3. 系统 bug 修复：Toast 优化和为系统奔溃兜底已经提供了统一解决方案，Finalizer 导致的 TimeoutException和资源为 null 的问题线上未发现这类型 crash。
4. 其它特性：Release 构建依赖检查可以考虑引入，其它三个作用不大。

四、参考资料

1. [深入理解Booster](#)