App动态加载调研

### 一、简介

主要思想：利用类加载器ClassLoader实现。

实现思路：将部分代码分离出来放在另外的APK中，做成插件APK的形式，在我们的应用程序启动后，在使用时动态加载该插件APK中的内容。该思路简单来说便是将部分代码放在了另外一个独立的APK中，而不是放在我们自己的dex中。这样一方面减少了我们自己dex中方法总数，另一方面也减小了dex文件的大小，因此可以解决如上两个方面的问题。对于这个插件APK包含的类，我们可以在使用到的时候再加载进来，这便是动态加载的思路。

### 二、相关文章

1.Android插件化认知：<http://blog.csdn.net/ccm_oliver/article/details/53928451>

2.Android组件化和插件化开发：<http://www.cnblogs.com/android-blogs/p/5703355.html>

3.Android插件化从入门到放弃-最强合集（基本上包括插件化知识）：<http://www.jianshu.com/p/353514d315a7>

4.Android中插件开发篇总结和概述：<http://blog.csdn.net/jiangwei0910410003/article/details/48104581>

### DroidPlugin

|  |  |
| --- | --- |
|  | DroidPlugin（star：5256） |
| 项 目 地 址 | github：<https://github.com/DroidPluginTeam/DroidPlugin> |
| Wiki | <https://github.com/DroidPluginTeam/DroidPlugin/blob/master/readme_cn.md> |
| 简 介 | DroidPlugin 是Andy Zhang在Android系统上实现了一种新的 插件机制 :它可以在无需安装、修改的情况下运行APK文件,此机制对改进大型APP的架构，实现多团队协作开发具有一定的好处。 |
| 分 析 文 章 | <http://www.jb51.net/article/102647.htm>  <http://www.jianshu.com/p/f1217cce93ef> |
| 优点 | 1.支持Androd 2.3以上系统  2. 插件APK完全不需做任何修改，可以独立安装运行、也可以做插件运行。要以插件模式运行某个APK，你无需重新编译、无需知道其源码。  3.插件的四大组件完全不需要在Host程序中注册，支持Service、Activity、BroadcastReceiver、ContentProvider四大组件  4. 插件之间、Host程序与插件之间会互相认为对方已经"安装"在系统上了。  5. API低侵入性：极少的API。HOST程序只是需要一行代码即可集成Droid Plugin  6.超强隔离：插件之间、插件与Host之间完全的代码级别的隔离：不能互相调用对方的代码。通讯只能使用Android系统级别的通讯方法。  7.支持所有系统API  8.资源完全隔离：插件之间、与Host之间实现了资源完全隔离，不会出现资源窜用的情况。  9. 实现了进程管理，插件的空进程会被及时回收，占用内存低。  10.插件的静态广播会被当作动态处理，如果插件没有运行（即没有插件进程运行），其静态广播也永远不会被触发 |
| 缺点 | 1.无法在插件中发送具有自定义资源的Notification，例如：a.带自定义RemoteLayout的Notification  b.图标通过R.drawable.XXX指定的通知（插件系统会自动将其转化为Bitmap）  2.无法在插件中注册一些具有特殊Intent Filter的Service、Activity、BroadcastReceiver、ContentProvider等组件以供Android系统、已经安装的其他APP调用。  3. 缺乏对Native层的Hook，对某些带native代码的apk支持不好，可能无法运行。比如一部分游戏无法当作插件运行（so库）。 |

### RePlugin

|  |  |
| --- | --- |
|  | RePlugin（360插件化方案）（star:3803） |
| 项 目 地 址 | github：<https://github.com/Qihoo360/RePlugin>  官网：  <https://replugin.org/> |
| Wiki | <https://github.com/Qihoo360/RePlugin/wiki> |
| 简 介 | RePlugin是一套完整的、稳定的、适合全面使用的，占坑类插件化方案，由360手机卫士的RePlugin Team研发， 也是业内首个提出”全面插件化“（全面特性、全面兼容、全面使用）的方案。 |
| 分 析 文 章 | <http://www.cnblogs.com/xinmengwuheng/p/7232395.html>  文章集合：  <https://github.com/Qihoo360/RePlugin/wiki/%E9%AB%98%E7%BA%A7%E8%AF%9D%E9%A2%98> |
| 优点 | 1.极其灵活：主程序无需升级（无需在Manifest中预埋组件），即可支持新增的四大组件，甚至全新的插件  2.非常稳定：Hook点仅有一处（ClassLoader），无任何Binder Hook！如此可做到其崩溃率仅为“万分之一”，并完美兼容市面上近乎所有的Android ROM  3.特性丰富：支持近乎所有在“单品”开发时的特性。包括静态Receiver、Task-Affinity坑位、自定义Theme、进程坑位、AppCompat、DataBinding等  4.易于集成：无论插件还是主程序，只需“数行”就能完成接入  5.管理成熟：拥有成熟稳定的“插件管理方案”，支持插件安装、升级、卸载、版本管理，甚至包括进程通讯、协议版本、安全校验等  7.数亿支撑：有360手机卫士庞大的数亿用户做支撑，三年多的残酷验证，确保App用到的方案是最稳定、最适合使用的 |
| 缺点 | 1.插件和宿主程序都要集成RePlugin，不是所有的apk都能作为插件（与现有需求不符） |

### VirtualAPK

|  |  |
| --- | --- |
|  | VirtualAPK（滴滴插件化方案）(star:4825) |
| 项 目 地 址 | github：<https://github.com/didi/VirtualAPK> |
| Wiki | <https://github.com/didi/VirtualAPK/wiki> |
| 简 介 | Wiki：  <https://github.com/didi/VirtualAPK/wiki> |
| 分 析 文 章 | <http://blog.csdn.net/lmj623565791/article/details/75000580>  <http://www.iteye.com/news/32006> |
| 优点 | 1.功能完善  2.兼容目前国内主流的手机平台  3.入侵性极低，插件开发不需要集成特定基类。 |
| 缺点 | 1.需要预先知道插件入口比如MainActivity全称  示例：  Intent intent = newIntent(); intent.setClassName("com.ryg.test", "com.ryg.test.MainActivity");  //插件入口  startActivity(intent);（与现有需求不符）   1. 插件工程需要经过处理，接入宿主工程   classpath 'com.didi.virtualapk:gradle:0.9.0'。  并且要用gradle生成apk  gradle clean assemblePlugin  或者  ./gradlew clean assemblePlugin  生成apk要取build->plugin->下的apk |

### Atlas

|  |  |
| --- | --- |
|  | Atlas（组件化，模块化开发）(star:5242) |
| 项 目 地 址 | Github：<https://github.com/alibaba/atlas> |
| Wiki | <https://alibaba.github.io/atlas/index.html> |
| 简 介 | Atlas是伴随着手机淘宝的不断发展而衍生出来的一个运行于Android系统上的一个容器化框架，我们也叫动态组件化(Dynamic Bundle)框架。它主要提供了解耦化、组件化、动态性的支持。覆盖了工程师的工程编码期、Apk运行期以及后续运维期的各种问题。 |
| 分 析 文 章 | <https://github.com/eatdefecat/AtlasDemo>  <http://www.cnblogs.com/wangcMove/p/6861163.html>  <http://blog.csdn.net/qq_34795285/article/details/77369071>  <https://mp.weixin.qq.com/s/OuHW3qE_InDfusRm1VEDew> |
| 优点 | 手机淘宝团队支持，数亿及应用基础。  工程期：实现host、bundle独立开发、调试的功能，bundle独立  运行期：实现完整的组件生命周期的映射，类隔离、资源共享等机制  运维期：增量更新修复能力，提供对class、so以及资源的增量更新修复能力，快速升级 |
| 缺点 | 1.AndroidManifest，已有Activity的manifest信息暂时不支持更改  2.window级别使用的资源保持不变 Activity切换的动画文件、通知栏使用的logo等如果修改了也不会生效，不过不会产生问题，使用的还是原来的样式  3.官方文档写的简略，集成有点困难。 |

### Small

|  |  |
| --- | --- |
|  | Small（倾向组件化，模块化开发）(star:3569) |
| 项 目 地 址 | Github：  <https://github.com/wequick/Small>  官网：  <http://code.wequick.net/Small/cn/home> |
| Wiki | <https://github.com/wequick/Small/wiki> |
| 简 介 | 世界那么大，组件那么小。Small，做最轻巧的跨平台插件化框架。  Small插件化方案适用于将一个APK拆分为多个公共库插件、业务模块插件的场景。 |
| 分 析 文 章 | <http://www.jianshu.com/p/cd152c75b0e0>  <http://blog.csdn.net/jiaweihaoku/article/details/52981813> |
| 优点 | 1.完美内置  所有插件支持内置于宿主包中  2.高度透明  插件编码、布局 编写方式与独立应用开发无异  插件代码 调试 与整包开发无异  3.极致剪裁  对插件分离所有 一切 能分离的 公共 代码、资源  4.无缝链接  通过设定 URI，宿主、本地化应用插件、本地化web插件、在线网页，以及任何自定义的插件之间能够 相互调起 与 传递参数。  5.跨平台  目前已支持 Android 、iOS 以及 HTML5 插件。并且三者之间可以通过同一套 JavaScript 接口进行通信。 |
| 缺点 |  |

### 八、dynamic-load-apk

|  |  |
| --- | --- |
|  | dynamic-load-apk(star:4916) |
| 项 目 地 址 | Github：  <https://github.com/singwhatiwanna/dynamic-load-apk> |
| Wiki | <https://github.com/singwhatiwanna/dynamic-load-apk/wiki> |
| 简 介 | DynamicLoadApk 是一个开源的 Android 插件化框架。由任玉刚 开发维护。 |
| 分 析 文 章 | <http://blog.csdn.net/chenzujie/article/details/52329739> |
| 优点 | 1.宿主程序与插件完全独立  2.宿主程序开放部分接口供插件与之通信  3.宿主程序耦合插件的部分业务逻辑 |
| 缺点 | 1、经过对DynamicLoadApk的分析我们知道它很大程度是通过代理的方式进行插件化，这也意味着一些静态注册由系统启动的BroadcastReceiver，ContentProvider无法支持  2、插件需要用that不能用this，这也的写法有点小别扭  3、 不支持自定义主题，不支持系统透明主题  4、插件和宿主资源 id 可能重复的问题没有解决，需要修改aapt中资源 id 的生成规则 |

### Android-Plugin-Framework

|  |  |
| --- | --- |
|  | Android-Plugin-Framework(star:1142) |
| 项 目 地 址 | <https://github.com/limpoxe/Android-Plugin-Framework> |
| Wiki | <https://github.com/limpoxe/Android-Plugin-Framework/wiki> |
| 简 介 | Android-Plugin-Framework是一个Android插件化框架，用于通过动态加载的方式免安装运行插件apk |
| 分 析 文 章 | <https://github.com/jdsjlzx/AtlasTutorial> |
| 优点 | 1.框架透明, 插件开发与普通apk开发无异，无约定约束  2.支持非独立插件和独立插件(非任意三方)  3.支持四大组件/Application/Fragment/Accessibility/LaunchMode/so  4.支持插件Theme/Style,宿主Theme/Style,轻松支持基于主题属性的皮肤切换  5.支持插件发送Notification/时在RemoteViews中携带插件中的资源（只支持5.x及以上, 且不支持miui8）  支持插件热更新：即在插件模块已经被唤起的情况先安装新版本插件，无需重启插件进程（前提是插件高度内敛，宿主不主动持有插件中的任何对象）  6.支持全局服务：即插件向容器注册一个服务，其他所有插件已经宿主都获取并调用此服务  7.支持DataBinding（仅限独立插件）  8.支持插件WebView加载插件本地HTML文件  9.支持插件Fragment/View内嵌宿主Activity中 |
| 缺点 | 1.不支持插件Activity转场动画使用插件中的动画资源  2.不支持插件Manifest中申请权限，所有权限必须预埋到宿主Manifest中  3.不支持第三方app试图唤起插件中的组件时直接使用插件组件的Intent。 第三方app要唤起插件中的静态组件必须由宿主程序进行桥接，即此组件需同时预埋到宿主和插件的Manifest中  4.不支持android.app.NativeActivity  5.不支持当一个插件依赖另一个插件时，被插件依赖的包含资源  6.不支持插件中的webview弹出原生Chrome组件 例如通过html的标签设置时间选择器。 说明：是否能支持原生组件取决于系统中使用WebView的实现。 如果是使用的Android System Webview，则可以支持。因为它packageId是以0x3f开头； 如果是使用的Chrome Webview，则不支持。因为它packageId是以0x7f开头，会和插件冲突。 这是采用Public.xml进行资源分组的缺陷。  6.不支持对插件或者宿主进行加壳加固处理。 |

### DynamicAPK

|  |  |
| --- | --- |
|  | DynamicAPK （star：2610） |
| 项 目 地 址 | <https://github.com/CtripMobile/DynamicAPK> |
| Wiki |  |
| 简 介 | DynamicAPK是一套用于实现多dex/apk加载的解决方案。它可以帮助你重新组织Android工程的配置和开发模式，实现多个子工程并行开发（以android studio module的形式），同时支持hot fix（在线修复有问题的功能）, 插件式载入不常用的功能（下载插件后再载入）。所有动态加载的插件不仅包含代码，也可以包含资源（资源的动态加载比代码要麻烦很多），因此是以APK形式实现的。 |
| 分 析 文 章 |  |
| 优点 | 1.更少的迁移成本（无需做任何activity/fragment/resource的proxy实现）  2.DynamicAPK不需要实现任何activity或fragment proxy来管理他们的生命周期。修改后的aapt会处理插件中的资源，因此R.java中的资源引用和普通Android工程没有区别。开发者可以保持原有的开发范式，无需做特殊的更改。  3.并发开发  4.提升编译速度  5.提升启动速度  6.Google提供的MultiDex方案，会在主线程中执行所有dex的解压、dexopt、加载操作，这是一个非常漫长的过程，用户会明显的看到长久的黑屏，更容易造成主线程的ANR，导致首次启动初始化失败。DynamicAPK可以在App启动时仅加载必须的模块，其他模块按需加载。  7.Hot fix (包含代码和资源)  8.按需下载和加载任意功能模块(包含代码和资源) |
| 缺点 | 缺点：该项目两年前项目，没有更新，对安卓后来版本不支持。 |

### ARouter

|  |  |
| --- | --- |
|  | ARouter（路由架构） |
| 项 目 地 址 | <https://github.com/alibaba/ARouter> |
| Wiki | <https://github.com/alibaba/ARouter> |
| 简 介 | Android平台中对页面、服务提供路由功能的中间件，我的目标是 —— 简单且够用。（大都应用于大型项目中界面跳转） |
| 分 析 文 章 | <http://www.jianshu.com/p/7cb2cc9b726a> |
| 优点 |  |
| 缺点 |  |

### 测试代码

Atlas--->Atlas测试代码

RePlugin--->RePlugin测试代码

Small---->Small测试代码

VirtualApk--->VirtualApk测试代码