

延迟加载 Dex 文件

2015-10-28 安卓应用频道

(点击上方公众号，可快速关注)

英文：Macarse

译文：伯乐在线 - lihenair

链接：<http://android.jobbole.com/81873/>

避免启动时加载多dex

如果你是一个Android开发者，你可能知道臭名昭著的dex方法个数限制的存在。如果你不知道我在说什么，请看看网上其他的博文。Sebastiano Gottardo 的《[DEX] 天空是极限？不，65K方法个数限制才是》或者 Matthias Käppler 的《“恭喜，你有许多代码了！”弥补Android的方法个数限制》都是很好的博文。

理解你的处境

首先要了解你的代码，什么增加了方法个数？

使用mihaip 的 dex-method-count 是一个检查库增加了多少方法的简单手段。警告：这个库会计算每个包。与有其他依赖或者不同包的库一起使用时要小心。

其次要问问自己：你维护的代码有多少？

首先我们注意到依赖关系遍布了所有的方法。支持库和Google Play服务的方法个数都是巨大的。经过调研，我们发现我们的方法个数是30k，剩下都是第三方库。

超过 65k 是否有意义？

去年Google IO 大会后，我有机会与在SoundCloud工作的 Fernando Cejas 讨论这个课题。他们的情况是，有一个本地组件，该组件有JNI接口，增加了很多方法。没有办法，他们的主用例有大量的方法，不得不使用MultiDex机制。

我们的情况不同，我们的应用不大，只是充斥了第三方库。

MultiDex有多糟？

Android版本低于棒棒糖（Lollipop）的用户，MultiDex会增加启动时间。我们决定用产品做一个小的AB测试，在应用启动时人为增加延迟。

在测试了不同值之后，我们得出了结论。如大家所料，延长启动对交互有显著的负面影响。我不能分享这些数字，但是请相信我。在晚上构建时，测试你的启动性能，并确保启动良好。

延迟加载Dex文件

如果你检查MultiDex库源码，它只修改了类加载器，以便在应用启动时加载额外的dex文件。问题是，为什么要在启动时加载？如果在启动后加载dex文件会怎么样？

浏览你应用中的依赖关系，是否这些依赖是最常见用例的一部分？在我们的应用中，答案是no。给你举些例子，比如支付类型的开发库：PayPal、Google Wallet、Bitcoin等等。你的用户会所有都用到吗？表单助手：Date pickers、credit card scanners。只有在用户注册时才会使用一次。

在启动时加载所有库有意义吗？

让我们来看看代码

最近，我有空写了一个例子应用。代码在这里。

该应用延迟加载Picasso库。启动后，当用户希望显示一个图片时，应用首先加载Picasso库。我们从不延迟加载Picasso库，因为他是我们最常用用例中的一部分，但是因为Picasso是开源的，很容易写出一个例子。

这是怎么工作的？

首先把要延迟加载的jar文件转换为dex文件。可以使用dx工具。在我的电脑里，它在/adt-bundle-mac-x86_64/sdk/build-tools/22.0.1/dx：

```
dx --dex --output=/path/to/the/library.dex /path/to/the/library.jar
```

生成dex文件后，将它放到assets文件夹

为了从代码中使用依赖，你需要将依赖标记为provided。

```
dependencies {  
    provided 'com.squareup.picasso:picasso:2.5.2'  
}
```

带provided的依赖库会在最终生成的dex文件中剔除。你可以测试下方法个数，带有compile标记的（1024个方法）和带provided标记的（214个方法）。

使用provided的依赖有点棘手。你需要保证知道模块加载完，不会使用依赖。代码里，我将所有Picasso调用封装到一个单独类。老实说，我没有花费很多心思去找到一种更优雅的方式。根据使用库的方式，在模块加载时，你可能想增加一个activity进行过渡。

代码的效果是显而易见的。当你希望加载predexed文件，需要把它从assets文件夹移动到应用指定的缓存目录。一旦移动完成，它调用一个MultiDex库中installSecondaryDexes()方法的修改版。我把MultiDex库源码增加到了工程里。你可以做一个diff，我只把方法的访问权限改成了public。只要predexed库加载完毕，你就可以使用Picasso库了。

我不记得做了其他事情。代码远远不够完善，请注意代码周围不同的TODO事项。我增加了一些注释来解释我走的那些捷径。

需要调查的事情：修复、测试等等

低版本测试

我在少量设备上进行了测试。很高兴知道能否在全部API级别上工作。

aar 的解决方案

aar可能含有资源，所以上面的方法不适用。

测试增加多个依赖

我从没有试着加载多个。AFAIK、MultiDex支持多个依赖，所以它应该可以工作。

在产品中测试

我们还没有在产品中测试过:(

混淆

就像bubbleguuum在这篇博文中提到的：

你不能混淆任何从附加dex中的类或接口继承来的类和接口（在你的例子中，继承了 Picasso中的类和接口）

结论

使用MultiDex不是一个轻松的决定。延迟加载Dex只是一种 hack，以延长方法个数少于65k的时间，直到更多的用户使用升级到比棒棒糖更新的版本。

安卓应用频道

微信号：AndroidPD



打造东半球最好的 安卓技术 微信号

商务合作QQ : 2302462408

投稿网址 : top.jobbole.com



微信扫一扫
关注该公众号