

21 | 除了 Cocoa, iOS还可以用哪些 GUI 框架开发?

18 | 怎么减少 App 电量消耗?

27 | 如何用 Flexbox 思路开发? 跟自动布局比, Flexbox 好在哪?

34 | iOS 黑魔法 Runtime Method Swizzling 背后的原理

43 | 剖析使 App 具有动态化和热更新能力的方案

28 | 怎么应对各种富文本表现需求?

24 | A-B 测试: 验证决策效果的利器

22 | 细说 iOS 响应式框架变迁, 哪些思想可以为我所用?

15 | 日志监控: 怎样获取 App 中的全量日志?

29 | 如何在 iOS 中进行面向测试驱动开发和面向行为驱动开发?

25 | 怎样构建底层的发布和订阅事件总线?

02 | App 启动速度怎么做优化与监控?

16 | 性能监控: 衡量 App 质量的那把尺

26 | 如何提高 JSON 解析的性能?

23 | 如何构造酷炫的物理效果和过场动画效果?

31 | iOS 开发学习资源和书单推荐

17 | 远超你想象的多线程的那些坑

01 | 建立你自己的 iOS 开发知识体系

04 | 项目大了人员多了, 架构怎么设计更合理?

05 | 链接器: 符号是怎么绑定到地址上的?

03 | Auto Layout 是怎么进行自动布局的, 性能如何?

09 | 无侵入的埋点方案如何实现?

10 | 包大小: 如何从资源和代码层面实现全方位瘦身?

150M 的限定  
App Store 规定了安装包大小超过 150MB 的 App 不能使用 OTA(over-the-air)环境下载,也就是只能在 WiFi 环境下下载。所以,150MB 就成了 App 的生死线,一旦超越了这条线就很有可能失去大量用户。

苹果的 App Thinning

App Slicing  
会在你向 iTunes Connect 上传 App 后,对 App 做切割,创建不同的变体,这样就可以适用到不同的设备。

On-Demand Resources  
主要是为游戏多关卡场景服务的。它会根据用户的关卡进度下载随后几个关卡的资源,并且已经过关的资源也会被删掉,这样就可以减少初装 App 的包大小。

Bitcode  
是针对特定设备进行包大小优化,优化不明显。

其他方面的优化

无用图片资源扫描  
图片资源的优化空间,主要体现在删除无用图片和图片资源压缩这两方面。而删除无用图片,又是最容易、最应该先做的。

图片资源压缩  
对于 App 来说,图片资源总会在安装包里占个大头儿。对它们最好的处理,就是在不损失图片质量的前提下尽可能地作压缩。目前比较好的压缩方案是,将图片转成 WebP

我的建议是,如果图片大小超过了 100KB,你可以考虑使用 WebP;而小于 100KB 时,你可以使用网页工具 TinyPng 或者 GUI 工具 ImageOptim 进行图片压缩。

通过 AppCode 找出无用代码  
但是,如果工程量不是很大的话,我还是建议你直接使用 AppCode 来做分析。毕竟代码量达到百万行的工程并不多。而,那些代码量达到百万行的团队,则会自己通过 Clang 静态分析来开发工具,去检查无用的方法和类。

运行时检查类是否真的被使用过  
通过 AppCode 的“运行时检查类是否真的被使用过”功能,可以检查出哪些类在运行时没有被使用过。这可以帮助你进一步精简代码,减少 App 的包大小。

通过 AppCode 找出无用代码  
但是,如果工程量不是很大的话,我还是建议你直接使用 AppCode 来做分析。毕竟代码量达到百万行的工程并不多。而,那些代码量达到百万行的团队,则会自己通过 Clang 静态分析来开发工具,去检查无用的方法和类。

运行时检查类是否真的被使用过  
通过 AppCode 的“运行时检查类是否真的被使用过”功能,可以检查出哪些类在运行时没有被使用过。这可以帮助你进一步精简代码,减少 App 的包大小。