|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 分类号 |  | |  | | | 密 级 | |  |
| U D C |  | |  | | | 编 号 | | 0003 |
|  | | | | | | | | |
| **读 书 笔 记** | | | | | | | | |
|  | | | | | | | | |
| Android应用性能优化最佳实践 | | | | | | | | |
|  | |  | |  |  | |  | |
|  | |  |  | |
|  | |  |  | |
|  | |  |  | |
|  | |  |  | |
|  | |  |  | |
|  | | | | | | | | |
| By key.guan  2017.5~ | | | | | | | | |

# 前言

## 为什么写这本书

一个好的应用，除了要有吸引人的功能和交互之外，在性能上也应该有高的要求，即使应用非常具有特色，或者功能和业务具有唯一性，在产品前期可能吸引了部分用户，但用户体验不好的话，也会给产品带来很差的口碑，如果有在体验上更好的竞品，用户也会很快转移。那么一个好的应用应该如何定义呢？主要有三方面：

* 业务/功能
* 符合逻辑的交互
* 优秀的性能

众所周知，Android系统作为以移动设备为主的一款操作系统，硬件配置有一定的限制，虽然配置现在越来越高级，但仍然无法和PC相比，在CPU和内存上的使用不合理或者耗费资源多时，就会碰到内存不足导致的稳定性问题、CPU消耗太多导致的卡顿问题等。例如，我们发布一款产品后会收到很多的反馈，这些反馈来自很多渠道，有用户反馈，有应用发布平台的反馈通道等。

面对这些问题时，大家想到的都是联系用户，然后看日志，特别是有关性能类问题的反馈，原因也非常难找，日志大多用处不大，为什么呢？因为性能问题大部分是非必现的问题，定位时很难复现，而又没有关键的日志，当然就无法找到原因了。这些问题非常影响用户的体验和功能的使用，所以解决这些问题是非常重要的。当前市场上讲解性能优化的书太少，即使有些书讲到，很多也是一笔带过，没有深入分析和寻找解决方案，所以有必要用一本书来从多个维度讲解在性能上我们面临了什么问题，如何解决这些问题，并在实际的项目中来优化我们的应用，以提高用户体验。

## 本书面向的读者

本书适合所有Android应用开发从业人员及在校学生，特别是有一定Android应用开发经验的开发人员，高级开发人员也可以通过本书了解更多的性能调优知识。

## 本书特色

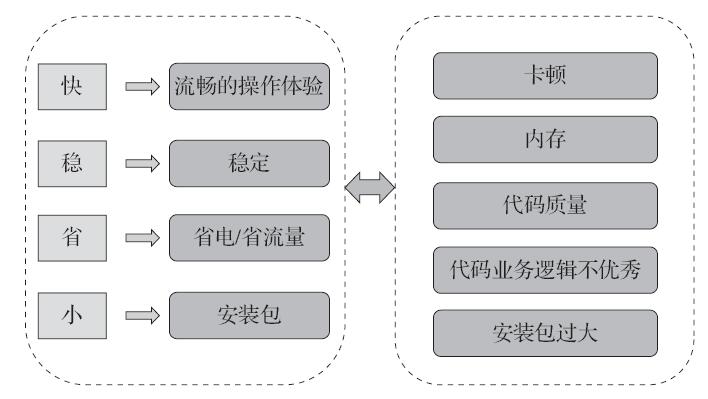
本书为进阶类图书，对于一些基础技术和基础理论知识不会做过多的阐述，特别是入门类的知识点，大家可以从其他书籍获取相关的知识。书中以性能优化为核心，深入剖析性能优化具体涉及的技术背景与优化方案，同时提供典型案例，帮助读者更深入地掌握Android应用开发技术，理解Android的运行机制和原理，掌握Android性能优化的思想，让开发者快速成长，打造高质量的Android应用。

## 本书的主要内容

可以把用户能体验到的性能问题主要总结为4个类别：

* 流畅
* 稳定
* 省电
* 省流量

性能问题的主要原因是什么，原因有相同的，也有不同的，但归根结底，不外乎内存使用、代码效率、合适的策略逻辑、代码质量这一类问题。本书讲解内容的目标和方向如下图所示。



从上图可以看到，打造一个高质量的应用应该以4个方向为目标：快、稳、省、小。

* 快：使用时避免出现卡顿，响应速度快，减少用户的等待时间，满足用户预期。
* 稳：降低crash率和ANR率，不要在用户使用过程中崩溃和无响应。
* 省：节省流量和耗电，减小用户使用成本，避免使用时导致手机发烫。
* 小：安装包小可以降低用户的安装成本。

这4类问题需要从根源上解决，也就是要解决图中第二个框里的问题：卡顿、内存使用不合理、代码质量差、代码逻辑不优秀、安装包过大。这些问题也是在开发过程中碰到最多的问题，在实现业务需求的同时，也需要考虑到这些点，多花时间去思考，避免功能完成后再来做优化和修复Bug，这个时候带来的成本会增加。如果是维护之前的代码，就需要使用一系列工具来发现问题点。

性能优化不是更新一两个版本可以解决的，是持续性的需求，结合到实际中，在一个新产品/项目开始时，由于人力和上线时间的限制，可以把优先级放低，但有些点是在写代码时就要考虑的，这就体现出程序员的技术功底。

本书强调性能调优的核心思想和方向如下：

发现问题→分析问题原因及背景→寻找最优解决方案→解决问题。

本书一共7章，在简单介绍了Android Studio的使用指南后，分别从绘制（UI）、内存、存储、稳定性、耗电以及安装包6个方面进行优化，从系统上深入分析绘制和内存的原理，一步步深入了解导致性能问题的本质原因，同时讲述了多种性能优化工具的使用，通过分析典型案例，得到有效的优化方案，从而实现更高质量的应用。书中所讲述的内容均基于Android 6.0系统。

## 勘误和资源下载

本书代码的下载地址：<https://github.com/lyc7898/AndroidTech>。

# 绘制优化

Android应用启动慢，使用时经常卡顿，是非常影响用户体验的，应该尽量避免出现。卡顿的场景有很多，按场景可以分成4类：UI绘制、应用启动、页面跳转、事件响应，如图2-1所示。在这四种场景下又有多个小分类，基本上覆盖了卡顿的各个场景。

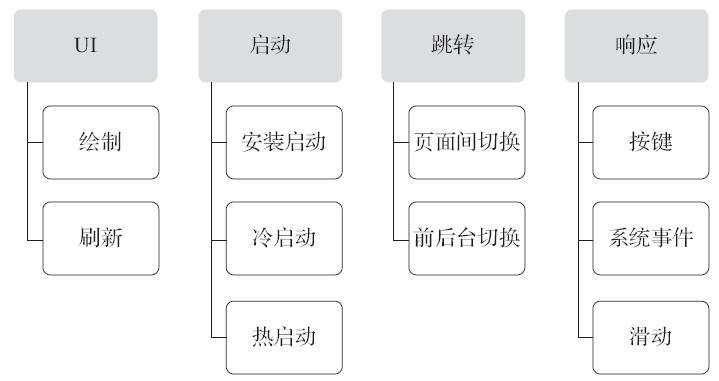


图2-1　卡顿主要场景

这4种卡顿场景的根本原因又可以分成两大类。

界面绘制：主要原因是绘制的层级深、页面复杂、刷新不合理，由于这些原因