

## <<Android应用性能优化>>

### 图书基本信息

书名：<<Android应用性能优化>>

13位ISBN编号：9787115272416

10位ISBN编号：7115272417

出版时间：2012-10

出版单位：人民邮电出版社

作者：[法] Hervé Guihot

页数：226

字数：355000

译者：白龙

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

## &lt;&lt;Android应用性能优化&gt;&gt;

## 前言

Android已经融入了寻常百姓的生活中。

当今世界，手机正从功能时代进化到智能时代，同时又诞生了令人爱不释手的平板电脑。

目前，应用程序开发者的可选择平台主要就是Android和iOS。

Android降低了甚至可以说是打破了移动开发的门槛，应用程序开发者编写Android应用程序只需要一台计算机就够了（当然还要有一些编程知识）。

工具都是免费的，几乎每个人都能写出数百万人会用的应用。

Android可以运行在各种设备上，从平板到电视。

开发者关键要做的就是保证应用可以顺利地在这些设备上运行，而且比竞争对手的还好。

对应用程序开发人员而言，Android开发的门槛已经很低了，你会发现，在许多情况下，自己不过是想在日益增长的Android应用程序市场上分一杯羹而已。

赖以谋生、实现明星梦，或者只是想使世界变得更美好……无论你编写程序所为何求，性能问题都是其中的关键。

要想阅读本书，最好能事先对Android应用程序开发基础有所了解，由此方能利用本书的妙诀良方让程序跑得更快。

尽管借助Android工具和在线文档可以很容易地创建应用程序，但性能优化（有时简直更像是一门艺术而不是科学）却无定法可循。

不管要优化的程序是已有的，还是从头编写的。

本书的目的就是要帮你找到简便的优化方法，以便使程序在几乎所有Android设备上都能取得不错的性能。

Android允许开发人员使用Java、C/C++，甚至汇编语言，所以，无论是更好地利用CPU特性，还是针对特定问题使用合适的编程语言，相信你可以用多种不同的方法来优化性能。

### 第1章 优化Java代码。

毫无疑问，你的第一个Android应用程序都是用Java开发的。

在这一章，你会了解到，选择算法要比实现算法更重要。

你还将学习如何利用简单的技术（如缓存和减少内存分配）来极大地优化应用程序。

此外，你还将学习让应用程序随时能够保持响应的方法，这是一个非常重要的性能指标。

此外还将介绍高效使用数据库的方法。

### 第2章 更进一步（或者说更底层，得看谈话对象）领略Android NDK。

尽管自从Android 2.2以后Java代码可以即时编译为机器码，但某些方法用C代码实现可以获得更棒的结果。

NDK还可以让你轻松地将现有代码移植到Android，而无需用Java重写一遍。

### 第3章 底层的汇编语言。

大多数应用程序开发很少用到汇编语言，但汇编语言能充分利用各个平台的专有指令集。

虽然这会增加复杂度和维护成本，但却是非常强大的优化秘诀。

汇编代码通常仅限于应用程序的某些特定部分，但不应忽略它的优点，仔细而有针对性的优化可以取得巨大成效。

### 第4章 探讨如何使用更少的内存来提高性能。

除了学习在代码中使用较少内存的简单方法，你还将了解到，由于CPU的设计方式，内存分配方式和内存访问也会对性能有直接影响。

第5章 如何在Android应用程序中使用多线程，以便保持随时响应，为越来越多可以同时运行多线程的Android设备提升性能。

### 第6章 测量应用程序性能的基础知识。

除了可以用API来测量时间外，一些Android工具还可以方便地查看应用程序执行时间耗费的具体情况。

### 第7章 确保应用程序合理使用电量的一些方法。

## <<Android应用性能优化>>

许多Android设备都由电池供电，因而节电非常重要，没人愿意使用过于耗电的应用。通过本章所述方法，可以不必牺牲Android程序的特性就能最大限度地减少功耗。

第8章 一些完善应用程序布局和优化OpenGL渲染的基本技术。

第9章 RenderScript。

它是Honeycomb引入的一个相对较新的Android组件。

RenderScript为性能而生，从首次发布以来已经有不少改进。

本章介绍如何在应用程序中使用RenderScript，顺便学习RenderScript定义的许多API。

我希望你喜欢上这本书，并在里面找到许多有用的技巧。

你会发现，很多技术不独适用于Android，还可以用在很多其他平台上，例如iOS。

就个人而言，我偏好汇编语言，希望能借着Android平台的快速发展以及其对NDK汇编语言的支持，能使Android吸引到更多的开发者。

至少，他们可以学到一门新技术。

但是，良好的设计和算法常常可以满足所有性能优化的需求，这才是关键。

祝你好运，我期待着你的Android应用程序！

## <<Android应用性能优化>>

### 内容概要

《Android应用性能优化》主要介绍如何调优Android应用，以使应用更健壮并提高其执行速度。内容包括用Java、NDK优化应用，充分利用内存以使性能最大化，尽最大可能节省电量，何时及如何使用多线程，如何使用基准问题测试代码，如何优化OpenGL代码和使用Renderscript等。

《Android应用性能优化》面向熟悉Java和Android SDK的想要进一步学习如何用本地代码优化应用性能的Android开发人员。

## <<Android应用性能优化>>

### 作者简介

Hervé Guihot

目前在联发科技公司（MTK，[www.mediatek.com](http://www.mediatek.com)）担任软件工程经理。他有十多年的嵌入式系统开发工作经验，主要与数字电视技术相关。目前正在研究如何将Android引入基于ARM的数字家庭平台。

## <<Android应用性能优化>>

### 书籍目录

#### 第1章 Java代码优化

##### 1.1 Android如何执行代码

##### 1.2 优化斐波纳契数列

###### 1.2.1 从递归到迭代

###### 1.2.2 BigInteger

##### 1.3 缓存结果

##### 1.4 API等级

##### 1.5 数据结构

##### 1.6 响应能力

###### 1.6.1 推迟初始化

###### 1.6.2 StrictMode

##### 1.7 SQLite

###### 1.7.1 SQLite语句

###### 1.7.2 事务

###### 1.7.3 查询

##### 1.8 总结

#### 第2章 NDK入门

##### 2.1 NDK里有什么

##### 2.2 混合使用Java和C/C++代码

###### 2.2.1 声明本地方法

###### 2.2.2 实现JNI粘合层

###### 2.2.3 创建Makefile

###### 2.2.4 实现本地函数

###### 2.2.5 编译本地库

###### 2.2.6 加载本地库

##### 2.3 Application.mk

###### 2.3.1 为(几乎)所有设备优化

###### 2.3.2 支持所有设备

##### 2.4 Android.mk

##### 2.5 使用C/C++改进性能

##### 2.6 本地Activity

###### 2.6.1 构建缺失的库

###### 2.6.2 替代方案

##### 2.7 总结

#### 第3章 NDK进阶

##### 3.1 汇编

###### 3.1.1 最大公约数

###### 3.1.2 色彩转换

###### 3.1.3 并行计算平均值

###### 3.1.4 ARM指令

###### 3.1.5 ARM NEON

###### 3.1.6 CPU特性

##### 3.2 C扩展

###### 3.2.1 内置函数

###### 3.2.2 向量指令

## <<Android应用性能优化>>

### 3.3 技巧

#### 3.3.1 内联函数

#### 3.3.2 循环展开

#### 3.3.3 内存预读取

#### 3.3.4 用LDM/STM替换LDR/STD

### 3.4 总结

## 第4章 高效使用内存

### 4.1 说说内存

### 4.2 数据类型

#### 4.2.1 值的比较

#### 4.2.2 其他算法

#### 4.2.3 数组排序

#### 4.2.4 定义自己的类

### 4.3 访问内存

### 4.4 排布数据

### 4.5 垃圾收集

#### 4.5.1 内存泄漏

#### 4.5.2 引用

### 4.6 API

### 4.7 内存少的时候

### 4.8 总结

## 第5章 多线程和同步

### 5.1 线程

### 5.2 AsyncTask

### 5.3 Handler和Looper

#### 5.3.1 Handler

#### 5.3.2 Looper

### 5.4 数据类型

### 5.5 并发

### 5.6 多核

#### 5.6.1 为多核修改算法

#### 5.6.2 使用并发缓存

### 5.7 Activity生命周期

#### 5.7.1 传递信息

#### 5.7.2 记住状态

### 5.8 总结

## 第6章 性能评测和剖析

### 6.1 时间测量

#### 6.1.1 System.nanoTime()

#### 6.1.2 Debug.threadCpuTimeNanos()

### 6.2 方法调用跟踪

#### 6.2.1 Debug.startMethodTracing()

#### 6.2.2 使用Traceview工具

#### 6.2.3 DDMS中的Traceview

#### 6.2.4 本地方法跟踪

### 6.3 日志

### 6.4 总结

## <<Android应用性能优化>>

### 第7章 延长电池续航时间

- 7.1 电池
- 7.2 禁用广播接收器
- 7.3 网络
  - 7.3.1 后台数据
  - 7.3.2 数据传输
- 7.4 位置
  - 7.4.1 注销监听器
  - 7.4.2 更新频率
  - 7.4.3 多种位置服务
  - 7.4.4 筛选定位服务
  - 7.4.5 最后已知位置
- 7.5 传感器
- 7.6 图形
- 7.7 提醒
- 7.8 WakeLock
- 7.9 总结

### 第8章 图形

- 8.1 布局优化
  - 8.1.1 相对布局
  - 8.1.2 合并布局
  - 8.1.3 重用布局
  - 8.1.4 ViewStub
- 8.2 布局工具
  - 8.2.1 层级视图
  - 8.2.2 layoutopt
- 8.3 OpenGL ES
  - 8.3.1 扩展
  - 8.3.2 纹理压缩
  - 8.3.3 Mipmap
  - 8.3.4 多APK
  - 8.3.5 着色
  - 8.3.6 场景复杂性
  - 8.3.7 消隐
  - 8.3.8 渲染模式
  - 8.3.9 功耗管理
- 8.4 总结

### 第9章 RenderScript

- 9.1 概览
- 9.2 Hello World
- 9.3 Hello Rendering
  - 9.3.1 创建渲染脚本
  - 9.3.2 创建RenderScriptGL Context
  - 9.3.3 展开RSSurfaceView
  - 9.3.4 设置内容视图
- 9.4 在脚本中添加变量
- 9.5 HelloCompute

## <<Android应用性能优化>>

- 9.5.1 Allocation
- 9.5.2 rsForEach
- 9.5.3 性能
- 9.6 自带的RenderScript API
  - 9.6.1 rs\_types.rsh
  - 9.6.2 rs\_core.rsh
  - 9.6.3 rs\_cl.rsh
  - 9.6.4 rs\_math.rsh
  - 9.6.5 rs\_graphics.rsh
  - 9.6.6 rs\_time.rsh
  - 9.6.7 rs\_atomic.rsh
- 9.7 RenderScript与NDK对比
- 9.8 总结

## &lt;&lt;Android应用性能优化&gt;&gt;

## 章节摘录

版权页：插图： Android Market可以让应用发布多个APK，每个针对不同的配置。

例如，可以让一个APK只使用ETC1纹理，另一个用PVRTC纹理——即为使用PowerVR的Android设备优化。

这些APK共享相同的AndroidMarket列表，AndroidMarket会为每个设备仔细选择适合的APK。

用户不必担心下载并安装的APK是否正确，这一切是自动和透明的。

注意并非所有的Android应用商店都支持此功能，所以如果你打算分发应用到多个商店，还是尽可能用适合所有设备的单一APK吧。

当然，纹理可能不是你需要发布多个APK的唯一原因。

例如，你可能针对旧设备发布小的APK，对较新的设备发布带有更多功能且更大的APK。

虽然可以使用多个APK，但它会使发布流程和维护变得复杂，因此，建议你尽可能尝试发布单一APK

8.3.5着色 OpenGL Es 2.0支持OpenGL Es着色语言（shadillg Language），以取代OpenGL ES 1.x中的固定变换函数和分散的管线（pipeline）。

这种语言基于C，可以让你编写自己的顶点（venex）和片段（fragment）着色来控制OpenGL管线。

像C程序一样，着色器可以非常简单，也可极其复杂。

虽然没有必须遵循的单一规则，但你应该尽可能减少着色器的复杂性，这会非常影响性能。

8.3.6场景复杂性 显然，渲染复杂的场景比简单的要花的时间长。

一个提高帧速率的方法是简化要渲染的场景，同时保持可接受的视觉质量。

例如，对于可看到的纹理，较远的物体可以减少细节，使用更少的三角形。

简单的对象使用的内存和带宽较少。

8.3.7 消隐 尽管GPU擅长几何运算，并能确定要渲染的物体，但应用应该尽力消隐（Culling）视线以外的物体，这样就不会向那些因为不可见而要丢弃的对象发送绘制命令。

消隐对象（甚至三角形）有许多方法，这些已超出本书的范围，帧速率低于预期可能是考虑不周的消隐方法造成的。

例如，可以快速消除摄像机（camera）后面的物体。

注意大多情况下你可以启用背面消隐（Backface Culling），这将会不会渲染对象背面的三角形。

## <<Android应用性能优化>>

### 媒体关注与评论

“本书详细介绍了优化Android代码的各种规则和技巧，揭开了Android和JAVA核心数据结构的神秘面纱。

最值得称道的是，作者展示了使用缓存、SQLite以及延长电池使用寿命的技术，这是每个严谨的开发人员都必须掌握的内容。

”——网友评论“市面上这种书并不多见！  
我想把这本书推荐给所有Android高级程序员。

”——网友评论

## <<Android应用性能优化>>

### 编辑推荐

Android应用开发者经常要想尽办法来提升程序性能。

Herve Guihot专著的《Android应用性能优化》主要介绍如何快速高效地优化应用，让应用变得稳定高效。

你将学会利用Android SDK和NDK来混合或单独使用Java、C/C++来开发应用。

把Herve Guihot专著的《Android应用性能优化》的内容学以致用，你的编程技术就会得到关键性的提升，写出的应用就会更为健壮高效，从而广受用户好评，并最终获得成功。

## <<Android应用性能优化>>

### 版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>