

移动设备应用程序的体系结构特征分析

黄永兵 陈明宇

(中国科学院大学 北京 100190)

(中国科学院计算技术研究所先进计算机系统研究中心 北京 100190)

摘 要 移动设备如智能手机、平板电脑已成为最主要的电子消费品,且呈现出快速增长的趋势.移动设备上运行的应用程序种类丰富多样,对硬件平台上不同资源的需求也大不相同;而移动设备的硬件平台在性能和功耗上有其自身的局限性.因而,分析移动设备应用程序体系结构层次的特征,对于硬件平台如处理器、内存等资源部件的设计,以及应用程序的优化,具有指导性的意义.文中选取了 Android 操作系统上的多类常用的应用程序,深入地分析了其在主流移动设备上的微体系结构相关的特征.结果表明,移动设备的应用程序普遍存在较高的指令缓存和指令转换后援缓冲器缺失率,并且分支预测失败率也较高.基于各程序的体系结构特征,文中抽取了部分最具代表性特征的应用程序,并提出了一个用于体系结构研究的移动设备基准测试程序 Moby. Moby 测试程序包括了浏览器、邮件客户端、音乐及视频播放器、文档阅读器及地图等应用.同时,文中还详细分析了 Moby 测试程序微体系结构无关的特征,如指令的组成、指令局部性特征、工作集大小及指令执行流等.

关键词 移动设备;应用程序;体系结构特征;移动互联网;移动设备基准测试程序

中图法分类号 TP302 **DOI号** 10.3724/SP.J.1016.2015.00386

Architecture Characteristics and Analysis of Mobile Device Applications

HUANG Yong-Bing CHEN Ming-Yu

(University of Chinese Academy of Sciences, Beijing 100190)

(Research Center for Advanced Computer Systems, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190)

Abstract Mobile devices such as smartphones and tablets have already become the primary consumer computing devices and still have rapid growth speed. The diverse requirements of mobile applications, coupled with the performance and power limitations of mobile platforms, has led to a growing need to understand their architectural behaviors, which are quite useful to provide architectural hints for the design of mobile platforms and guide applications' optimization. In this paper, we select several popular applications with diverse categories that executed on Android OS. Their micro-architectural dependent characteristics are deeply analyzed on mainstream mobile devices. Experimental results show that, most mobile applications suffer from high miss rate for instruction cache and instruction TLB, and high branch misprediction. Based on these architectural characteristics, we extract several typical applications with representative behaviors, and propose a mobile benchmark suite Moby for architectural researches. Moby contains diverse popular applications, including a web browser, an email client, music and video players, document processing applications and a map program. Moreover, we also study the micro-architectural independent behaviors for Moby applications, such as instruction features, working set size and instruction execution flow.

收稿日期:2013-12-18;最终修改稿收到日期:2014-11-26. 本课题得到国家“九七三”重点基础研究发展规划项目基金(2011CB302502)、国家自然科学基金(60925009,61221062,61272132)、中国科学院战略性先导科技专项(XDA06010401)和华为项目(YBCB2011030)资助. 黄永兵,男,1987年生,博士研究生,中国计算机学会(CCF)会员,主要研究方向为计算机体系结构、操作系统、高性能计算. E-mail: huangyongbing@ict.ac.cn. 陈明宇,男,1972年生,博士,研究员,中国计算机学会(CCF)会员,主要研究领域为高性能计算机体系结构、操作系统. Moby 下载地址: <http://asg.ict.ac.cn/projects/moby>.

Keywords mobile device; mobile application; architectural characteristic; mobile Internet; mobile device benchmark

1 引言

移动互联网的兴起正在逐渐地改变用户的生活方式,移动设备从简单的通信功能机转变为执行信息收集、集成、共享等功能复杂的智能终端.移动设备如智能手机、平板电脑已成为最主要的电子消费品,而且依旧呈现出迅速发展的趋势.至 2013 年 6 月,全球智能手机的用户数已达到 15 亿,相比 2012 年有 31% 的增长,但仅占全球人数的 21%,仍有较大的发展潜力.

移动互联网主要催生了两大产业:移动设备的应用程序以及硬件平台.移动设备上的应用程序种类及功能繁多、形式多样、实现方式灵活.数以亿计的移动应用被开发和使用,苹果的 AppStore 仅 2012 年单下载量就接近 200 亿次.不同的应用程序由于功能、编程语言、运行方式等的差异,对硬件资源的需求也大不相同.例如,在移动设备的操作系统中,Android 和 iOS 拥有最高的市场占有率.但由于 Android 和 iOS 分别利用 Java 和 Object-C 作为应用的编程语言,其移动应用的编程和运行方式不同,在相同硬件平台上的运行性能也可能存在较大的差异.

从硬件平台的角度来看,ARM 处理器^①占有最大的市场份额,具有代表性的有苹果的 Ax 处理器,高通的骁龙(Snapdragon)处理器.此外,Intel 的 Atom 系列处理器也是常用的移动设备处理器.移动处理器的性能增长迅猛,主频趋向 2 GHz,四核芯片也逐渐成为主流.同时,移动处理器的体系结构也在变革,指令执行方式从简单的顺序执行转变为多发射乱序执行,缓存(Cache)的容量和层次也在不断增加.但总体来说,核心芯片扩充核数的趋势将放缓,功耗及优化成为发展的重点.一方面现有操作系统技术及移动应用程序与多核硬件平台并不匹配,另一方面多核同时运行时功耗偏高,使得只在极少数应用场景下才能充分发挥多核性能.

优化移动设备硬件平台的性能及功耗,关键在于分析移动应用程序对硬件资源的利用率,以及其在运行过程中呈现出来的体系结构特征.针对这个需求,本文挑选了大量的移动应用程序,并在真实移动平台上详细地分析了这些应用程序的微体系结构相关的特征.根据应用程序的功能以及流行度等标

准,本文从 Google Play^② 和国内的一些应用市场中挑选了一批 Android OS 的应用程序作为实验目标,包括性能测试、社交、新闻、网络购物、视频播放、音频播放、文档处理、图像编辑、地图导航及电子邮件等应用.这些应用程序基本覆盖了各类常见的移动应用,具有代表性.本文在真实的 ARM 平台上测试了与应用性能密切相关的体系结构特征,包括指令和数据缓存(I/D Cache)、分支预测、转换后援缓冲器(TLB)、二级缓存(L2 Cache)和内存访问等指标.

微体系结构相关特征的实验结果表明,大部分应用程序有较高的指令周期数(CPI),这意味着应用程序不能充分有效地利用硬件平台提供的资源.指令相关的部件如指令 Cache、ITLB 的缺失率也比较严重,这体现了移动应用的指令执行相对复杂,而以往的处理器的优化主要着重于数据相关的部件.此外,交互式应用的指令 Cache 缺失率要明显高于自动化行为应用,而自动化行为的应用程序的数据 Cache 缺失率要明显高于交互式应用.

移动设备上的应用程序和移动处理器间存在不匹配的现象,这意味着,移动处理器需要根据应用程序的特征来改进其体系结构的设计.从处理器优化的角度来说,基准测试程序和体系结构模拟器是研究的基础.本文从上述被分析的大量应用程序中挑选出部分具有代表性特征的应用程序,形成一个移动设备的基准测试程序集(Moby)^[1]. Moby 基准测试程序集主要用于研究移动设备的体系结构,已被移植到全系统的体系结构模拟器 Gem5^[2]上,并实现开源^③.然而,移动端的应用程序经常需要用户的交互操作,而模拟器特别是全系统的体系结构模拟器不能有效地处理和重现特定的用户行为.尽管用户交互操作可以通过自动化工具如 AutoGUI^[3]来模拟,但本文提出利用应用程序的代表性操作来模拟应用的行为.代表性操作允许以命令行的方式来执行应用程序的主要活动(Activity),而无需用户的交互.这可以大大简化移动应用在模拟器上移植和执行的难度.

为了进一步地刻画 Moby 基准测试程序集的行

① ARM Processors. <http://arm.com/products/processors/index.php>

② Google Play Store. http://en.wikipedia.org/wiki/Google_Play

③ Moby benchmark suite. http://asg.ict.ac.cn/projects/moby_2013

为,本文还分析了其中各应用程序与硬件平台微体系结构无关的特征,包括调用库的数量、不同类型指令的特征、指令局部性、工作集大小以及应用运行过程中的指令执行轨迹等.实验结果表明,移动应用程序在运行时会创建数十个进程或线程,并调用数十个应用库函数.这无疑会增加硬件平台指令部件的处理复杂度,降低指令部件的执行性能.

总体而言,本文的主要贡献如下:(1)选取移动设备上大量不同类别的 Android 应用程序,并深入地分析了它们的微体系结构相关的特征;(2)发现移动设备的应用程序与硬件平台间存在不匹配的现象,而且处理器中与指令相关的部件执行效率差;(3)提出移动设备的基准测试程序 Moby,用于体系结构的研究,并支持在全系统体系结构模拟器上运行;(4)详细地分析了 Moby 各应用程序与微体系结构无关的特征.

本文第 2 节讨论本文评测的移动设备应用程序集;第 3 节介绍实验平台和相关工具;第 4 节分析移动应用程序与平台相关的微体系结构特征;第 5 节描述本文提出的移动设备基准测试程序 Moby;第 6 节进一步介绍 Moby 与微体系结构无关的特征;第 7 节简要介绍与本文相关的研究工作;第 8 节对全文进行总结.

2 测评应用程序集

应用测试集的选择直接决定了其呈现的特征是否具有代表性,然而数以亿计的应用程序不可能被完全代表.因而,本文从常用的 Google Play 应用市场入手,并根据其整理的应用分类,依次从各分类中挑选出部分应用程序.在各应用分类中,本文主要用新颖度、下载量、用户热度、交互性、发行版本号这 5 个指标作为衡量标准^[4].

(1)新颖度.每隔一段时间都会出现一些新颖的应用,它们属于应用商店分类中的一种,但是比同类的应用更受欢迎,因为它们满足了用户的新需求.

(2)下载量.下载量标志着这个应用的市场份额,同时标志着受用户的欢迎程度和流行程度.

(3)用户热度.用户热度标志了用户对这个应用的接受和使用程度.

(4)交互性.在移动应用中,交互式应用占据了 90% 的市场份额.根据交互设计的原则进行设计,只有能吸引用户的设计才能获得成功.

(5)发行版本号.发行版本号标志着一个应用的迭代发行次数,优秀的应用程序都有稳定的迭代

发行版本.

按照上述原则,本文挑选出的应用程序如表 1 所示.

表 1 应用程序分类

类型	应用名称	行为
性能测试应用	BBench, Antutu, SunSpider	自动化行为
社交类应用	QQ, Fanfou, SinaWeibo, Renren	交互式应用
新闻类应用	ZAKER, FeedGoal, NetEaseNews	交互式应用
网络购物类	eBay, Taobao, 360Buy	交互式应用
视频播放类	YouKu, VLC, MXPlayer	自动化行为
音频播放器	Duomiyinyue, SubSonic, Douban, TTPod	自动化行为
文档编辑类	Evernote, TomDroid, Documentstogo, KingsoftOffice	交互式应用
PDF 阅读工具	VuDroid, AdobeReader, FBReader	自动化行为
图像编辑类	Meituxiuxiu	交互式应用
文件查看器	ESFileExplorer	交互式应用
地图导航类	GoogleMap, BaiduMap	交互式应用
电子邮件类	K9Mail	自动化行为
游戏类	AngryBird, FrozenBubble	交互式应用

3 实验平台和工具

本文使用 Pandaboard-ES 作为移动设备的硬件平台,运行 Linaro Android 4.0.1^① 操作系统. Pandaboard-ES 是一个开放的 OMAP4 开发平台,由德州仪器的 OMAP4460 处理器和以此为核心的外围设备构成.OMAP4460 含有主频为 1.2 GHz 的双核 ARM Cortex A9 处理器,并集成了 1 GB 的 LPDDR2 内存单元. Cortex A9 处理器包括了大小为 32 KB 的 4 路组相联指令和数据 Cache,以及 512KB 大小的 16 路组相联的 L2 Cache.

Cortex A9 处理器的性能监控单元 (Performance Monitoring Unit)提供了一套性能计数器以及许多性能事件,用于监控处理器各资源部件的利用情况.其处理器的参考手册^②中列出了 Cortex A9 处理器支持的性能事件及其编号,每一个性能事件都对应一个微体系结构的特征.本文第 4 节所述的微体系结构相关的特征值,就是利用处理器提供的性能计数器得到的.然而,由于性能计数器的数量有限,本文所提到的微体系结构特征并不能一次测出,需要多次测量.此外,部分应用程序依赖于用户的交互,测量的结果存在一定的波动.在实验中,本文通过 10 次测量取平均值的方式来统计.所有的微体系结构特征由轻量级、低开销的性能计数器工具 TopMC^③ 来获得.

① Linaro. <http://www.linaro.org/>
② Cortex-A9 Technical Reference Manual
③ TopMC. <http://asg.ict.ac.cn/projects/topmc>, 2011

为了分析 Moby 基准测试程序集中各应用程序与微体系结构无关的特征,也就是应用程序自身的特征,本文使用全系统模拟器 Gem5^[2] 来获取相应的参数. Gem5 模拟器是一个常用的体系结构模拟器,能够在其上运行完整的操作系统和应用程序,并详细地模拟了处理器及内存等硬件部件,支持 Alpha、ARM 及 x86 等架构. 同时, Gem5 自带有几种常见 ARM SoC 的配置,如 ARM RealView 和 Versatile Express 平台的处理器、内存、I/O 外设等的配置参数. Pandaboard ES 开发板与 Versatile Express 平台类似,可以参考其配置.

4 微体系结构相关的特征分析

4.1 平均指令周期数和分支预测失败率

平均指令周期数(CPI)是反映应用程序在硬件平台上运行性能最直观的体现. CPI 越高,表示每条指令需要的处理器周期数越多. Cortex-A9 的理想

值是 0.5. 然而,如图 1 所示,这些应用的 CPI 值都在 1.5 以上,只有 Antutu 跑分测试在 1 附近. 交互式应用的 CPI 值基本都接近 3.5,自动化行为应用的 CPI 都在 2 附近波动,豆瓣客户端在 2 以上,其他的都在 2 以下. 可以看出,这些应用在 Cortex-A9 上运行得并不良好,部分原因是交互式行为的弊端:交互式行为提高了应用的鲜活性,较好的交互环境设计能提升用户体验,但是也带来了性能的损失.

在现代处理器设计中,分支预测技术确保了处理器的并行能力,并且作为高性能处理器的支持技术,还保证了指令级并行的可能性. 本文统计了测试程序集的分支预测失败率,图 2 描述了分支预测的情况:交互式应用的分支预测未命中率比自动化行为的应用要高. 从程序轨迹分析,现代编程模型是消息驱动的,主线程用消息泵将消息派发到对应的回调函数,本身在无限循环,因此,程序的行为变得无法估计,可预测性不强. 移动处理器需要扩充其分支预测部件,主要是分析程序调用栈的规律.

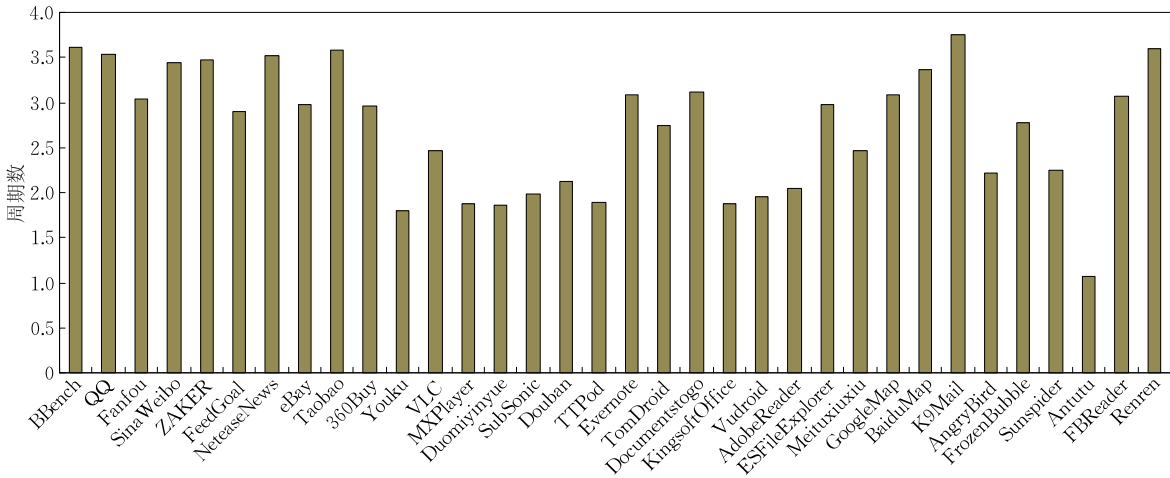


图 1 平均指令周期数

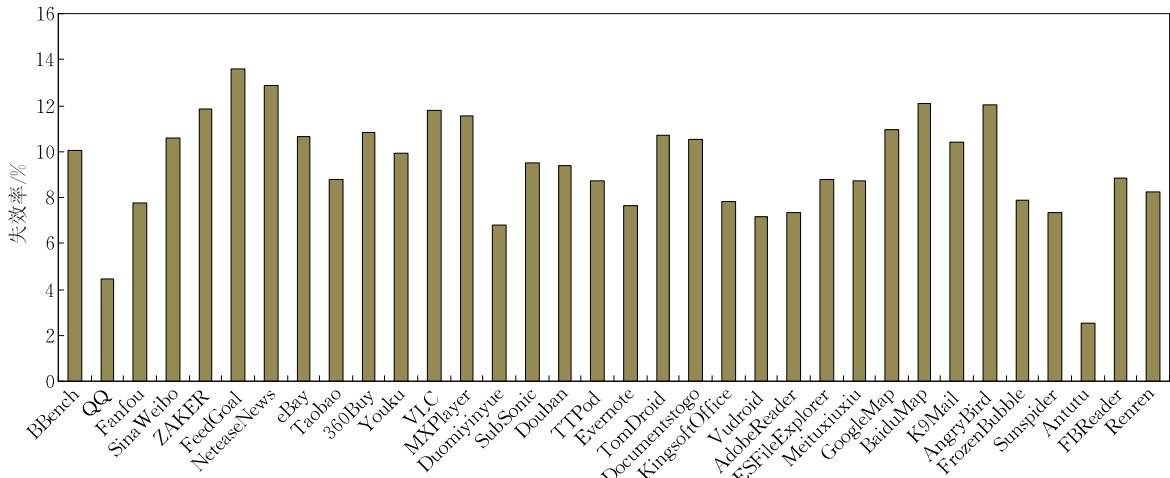


图 2 分支预测失败率

4.2 指令、数据缓存和 TLB 未命中率

本文统计了指令缓存和数据缓存的缺失率,统计的单位是每一千条指令发生缺失的次数.从图 3 中可以看出,交互式应用和非交互式应用在数据缓存缺失这一项上没有太大的差别,都在 5 上下波动.而指令缓存缺失不同.新闻阅读类软件的指令缺失率明显要高于播放类软件,两者的最大距离可以达到 10 以上.播放类软件的主要工作是对媒体文件重

复地进行编解码,执行的指令相对比较固定.总体而言,Android 上的应用程序会频繁地调用许多应用库,如 JVM 运行相关的库函数等.因而,移动设备上应用程序的指令执行会相对复杂,局部性会有所降低.以 KingsoftOffice 应用为例,本文在第 6 节分析了其指令执行流,可以看出移动应用程序复杂的指令特征.

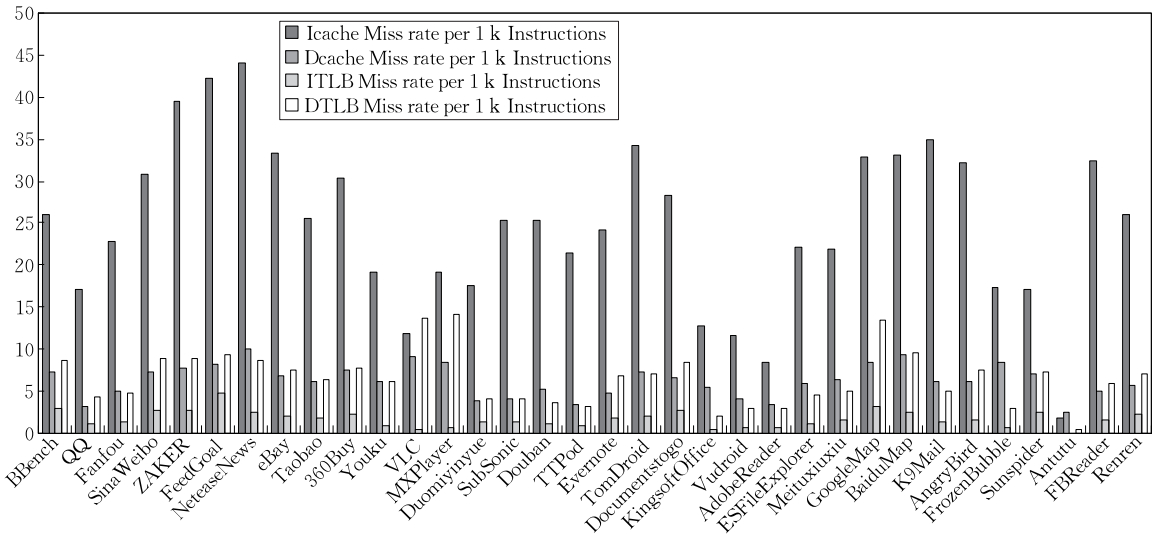


图 3 指令缓存、数据缓存和 TLB 缺失率

旁路后援缓冲器(Translation Lookaside Buffer, TLB),也可以看做是缓存的一种,它缓存的是虚拟地址到物理地址的映射关系.引起 ITLB 缺失的原因有很多,比如进程切换、调用未被访问过的动态库、以及由于 ITLB 的项数有限而发生的 TLB 替换.如图 3 所示的是各应用程序每一千条指令中 ITLB 缺失的数目.FeedGoal 的值为 4.9,播放器类应用的值在 1 左右,而其他大部分应用都在 2 附近.这说明播放器类的应用的指令访问局部性相对较好.而对于 DTLB 的缺失,VLC、MXPlayer 和 GoogleMap 每一千条指令的缺失数接近 14,高出其他应用近一倍.VLC 和 MXPlayer 主要是对视频文件进行编解码,需要频繁地读取新的数据,因此会造成 DTLB 的缺失.而其他大部分应用仅访问或操作少量的数据,现有 Cortex A9 处理器的 DTLB 项数以及分配替换策略能有效地处理这些数据的映射关系.

4.3 流水线阻塞

流水线技术的引入,大大提高了 CPU 的并行

处理性能.但是,指令之间的依赖性产生的冲突,包括资源分配、数据争夺及控制流的冲突,是不可避免的.流水线在发生上述情况的时候不得不停滞.当分支预测失效时,需要排空流水,并重新用正确的指令流填满;当出现数据冲突的时候,流水线需要停滞,等待资源.本文分别统计了流水线被阻塞的众多原因中常见的 3 项:TLB 缺失、数据缓存缺失及指令缓存缺失.

图 4 所示的是因不同部件缺失导致处理器停滞的周期数占整个程序运行周期数的比例.除了 3 个列出的处理器停滞原因,其他部件造成处理器停滞的比例相对较小,因而剩余的周期数可以当成是处理器正常运行的周期数.从图中可以看出,由指令缓存缺失导致的阻塞,交互式应用都在 40%左右波动,只有测试类的 SunSpider 略低.图 5 中描述了 TLB、指令 Cache 或数据 Cache 缺失各引入流水线停滞的平均周期数.可以看出,数据 Cache 缺失导致的阻塞比其他两种原因都要严重得多,而属文档处理类软件中特别明显.

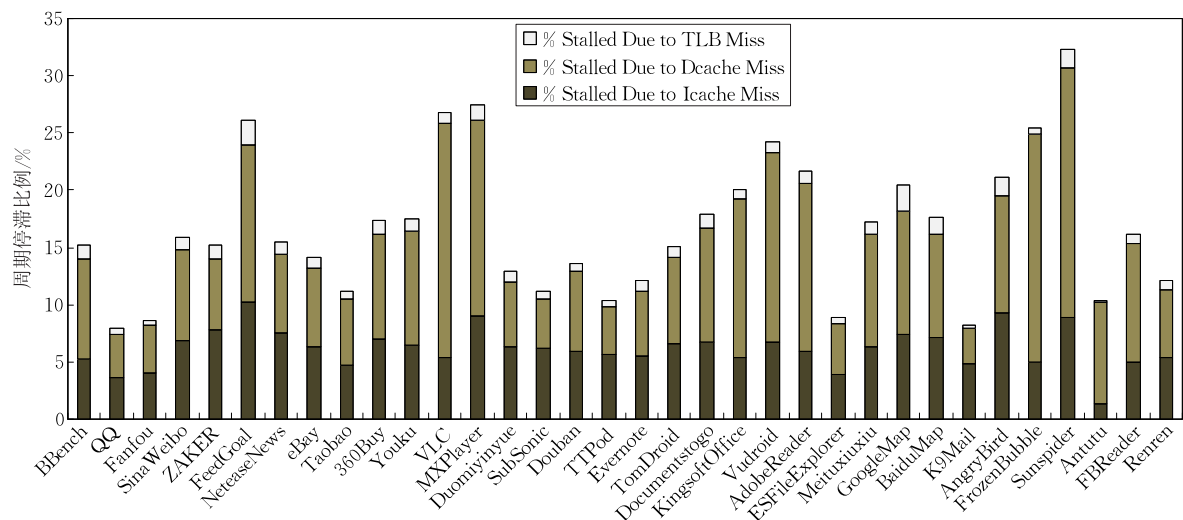


图 4 处理器的停滞周期比例

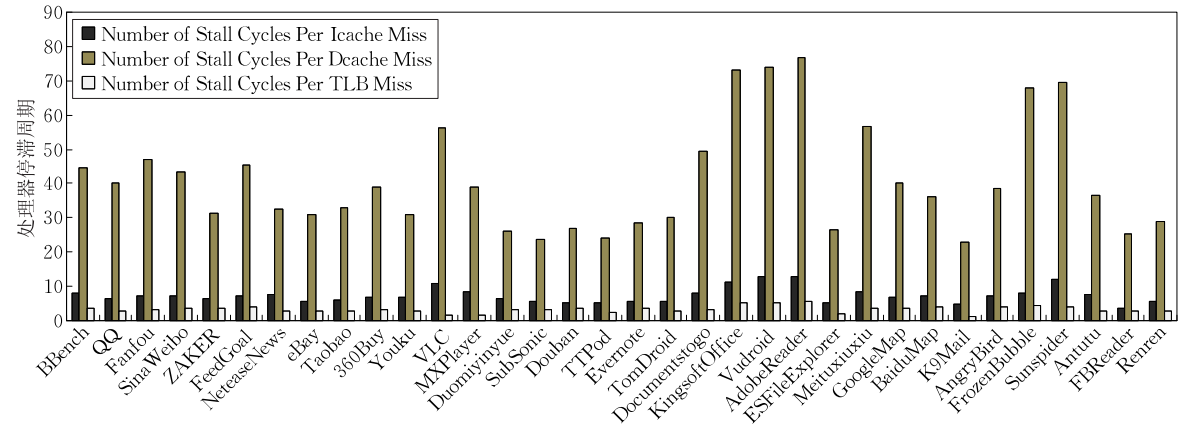


图 5 不同部件缺失引入的处理器停滞周期

4.4 二级缓存的缺失率

图 6 所示的是二级缓存(L2 Cache)的缺失率。大部分应用的 L2 Cache 缺失率都比较平缓,在 20%~30%之间。但是 KingsoftOffice、AdobeReader 和 VLC 的 L2 Cache 缺失率仍达到 50%。

为了进一步分析 L2 Cache 缺失的现象,本文分别统计了 L2 Cache 数据及指令请求各自的缺失率,如图 7 所示。整体而言,数据请求的缺失率偏高,基本都达到 40%,而大部分应用程序的指令请求缺失率也接近 20%,都相对偏高。这正说明移动应用的指

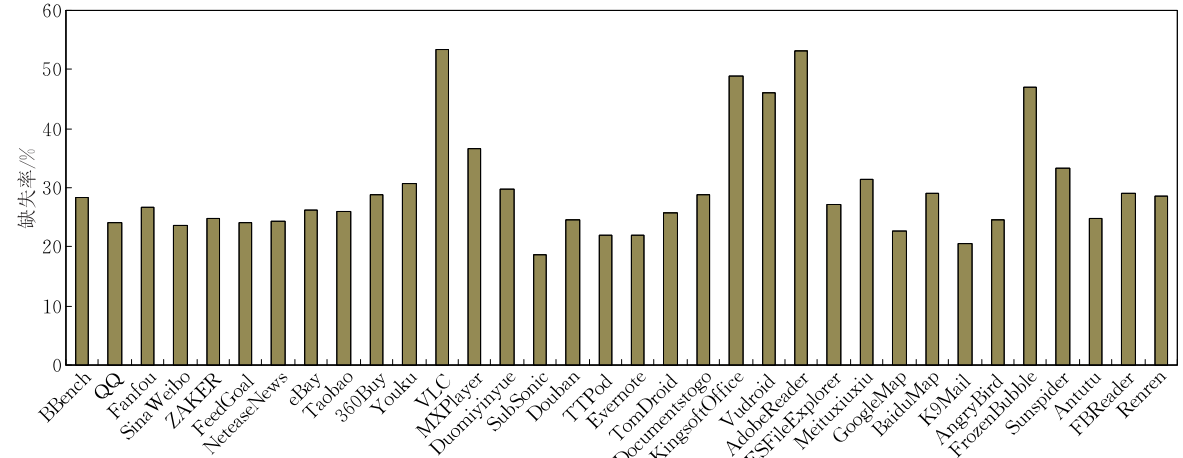


图 6 二级 Cache 缺失率

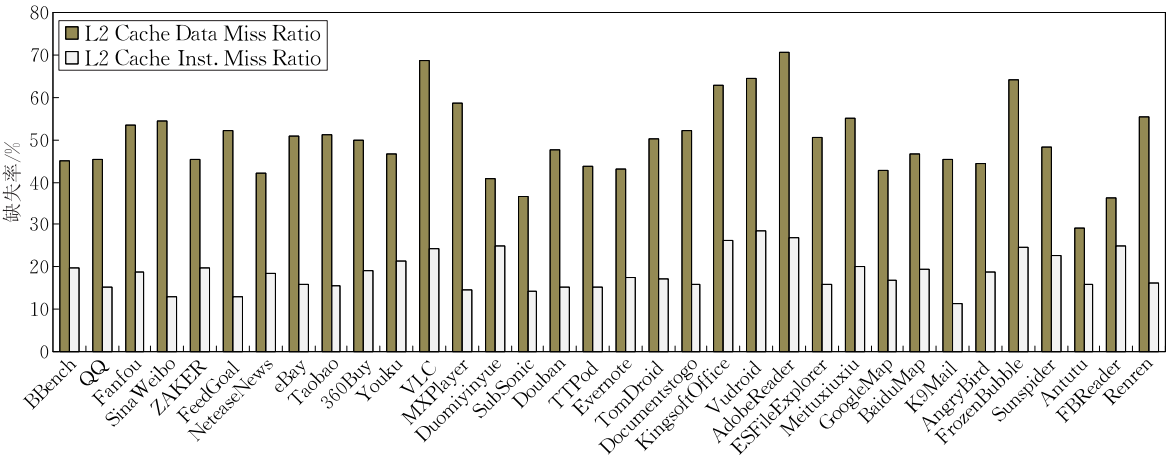


图 7 二级缓存中数据及指令请求的缺失率

令相比 SPEC CPU2006 等高性能应用,执行流比较复杂.同时,观察发现,KingsoftOffice、AdobeReader 及 VLC 的数据请求缺失率竟接近 70%.而这几个应用需要对文件进行处理,数据请求会占据较大的比例,因而导致整体的 L2 Cache 缺失率偏高.

5 基准测试程序 Moby

上一节对各移动应用程序在指定开发板上的微体系结构相关特征的分析,揭示了移动设备上的应用程序和移动处理器间存在不匹配的现象,这意味着,移动处理器需要根据应用程序的特征来改进其体系结构的设计.

在处理器体系结构研究中,基准测试程序和体系结构模拟器是基本的工具.然而,现有移动设备端的基准测试程序或者只关注某几种特定的应用程序^{①②},或者不能在模拟器上直接运行^[5].而高性能计算的基准测试程序如 SPEC CPU2006^[6]与交互式的移动应用存在较大的差异,不适合用于研究移动设备的体系架构.

为此,本文设计并提出了移动设备的基准测试程序 Moby.首先,Moby 集合了多类常用的应用程序,包括浏览器、邮件客户端、社交网络、新闻客户端、文档处理、地图、音视频播放器及游戏共 8 类应用.这 8 类应用程序基本覆盖了用户使用频度较高的应用.其次,Moby 的应用程序从上一节中描述的应用中选取,并且其微体系结构相关的特征尽量具有代表性.再次,对于模拟器,特别是体系结构全系统模拟器来说,用户的交互操作会阻碍应用程序的模拟以及可重现性.尽管用户操作的监控软件如

AutoGUI^[3]能够准确录制并回放用户的动作,但这类软件需要在低速的模拟器精心地制作录制用户的行为,效率相对较低下.通过分析认为,移动设备应用程序的主要功能一般都会以活动(Activity)的形式存在,而且能够以命令行的形式执行,只需手动指令命令的参数如输入输出文件等,而无需用户的交互.这种主要活动在本文中被称为典型操作.

表 2 介绍了 Moby 选取的 10 个应用程序、其使用类别,以及适用于模拟器的典型操作.目前,Moby 基准测试程序已经开源.

表 2 Moby 基准测试程序

应用程序	类别	典型操作
BBench ^[5]	浏览器	加载网页
K9Mail	邮件客户端	加载并显示邮件
SinaWeibo	社交网络	加载微博信息
NeteaseNews	新闻客户端	检查并加载新闻
KingsoftOffice	文档处理器	打开 doc/xls/ppt 格式文件
AdobeReader	文档处理器	打开 pdf 格式文件
BaiduMap	地图	加载指定区域的地图
MXPlayer	视频播放器	播放某个视频
TTPod	音频播放器	播放某个音乐
FrozenBubble	游戏	加载游戏

6 Moby 微体系结构无关的特征分析

微体系结构相关的特征,能够说明应用程序与硬件平台的匹配度.如果硬件平台改变,如处理器及内存资源的容量、分配策略等参数的变化,微体系结构相关的特征可能缺少参考意义.而微体系结构无关的特征能够真正了解应用程序自身内在的特性.

① SunSpider. <http://www.webkit.org/perf/sunspider/sunspider.html>
② Antutu. <http://www.antutu.com/index.shtml>

因而,本节将主要介绍 Moby 各应用程序与微体系结构无关的特征,包括了指令分解、工作集大小、应用的进程或线程数、调用库特征等.与执行长时间计算操作的高性能计算应用不同,大部分移动应用的典型操作只执行简单的操作,因而 Moby 各应用程序执行的指令数仅几十亿(Billion).不过,这个数量级的指令数更适合在全系统的模拟器上运行.本文利用 Gem5 模拟器执行了表 2 所示各应用的典型操作,并获得如下与微体系结构无关的特征.

6.1 指令特征及工作集大小

应用程序执行指令的组成,可以大致反映出程序运行过程中对各硬件资源的需求.例如,分支指令的数量直接决定着分支预测部件的处理压力,而 load/store 指令体现应用程序对 Cache 及内存容量及性能的敏感度.

表 3 描述了 Moby 各应用程序指令特征的总结.对于大部分应用来说,Load 指令和 Store 指令分别占据指令总量的 25% 和 12% 左右.与 SPEC CPU2006 的整数测试程序相比,Moby 中应用程序的 Load 指令和 Store 指令的特征都比较接近,差异不大.这说明,Android 的应用执行框架和运行环境在其应用程序的指令中占据着重要的地位.除了

MXPlayer 和 K9Mail 这两个应用,其他应用中 15% 的指令是分支预测指令.而且,在分支预测指令中,有 70% 的指令是条件分支指令.这意味着,大部分分支指令都有可能造成分支预测错误,进而导致流水线的回滚及性能的损失.

应用程序工作集的统计方式可以有多种,既可以是 Cache 行的粒度,也可以是物理页面的粒度.本文为了分析内存访问的特征,选取 4KB 大小的物理页面作为监控粒度.由于 Moby 各应用的典型操作仅执行几秒的时间,本文分析的工作集大小是整个操作执行过程中访问过的不同物理页面数.由表 3 可以看出,半数应用程序的工作集接近或超过 100MB.仅 K9Mail 和 FrozenBubble 的工作集小于 65MB,这与它们执行的指令数相对应.所有应用的工作集大小都远远超过了最后一级 Cache 的容量.然而,除了播放视频的 MXPlayer 应用,Moby 中其他应用程序的输入集都相对不大,仅为数 MB.这种现象显示,Moby 应用会访问很多物理页面,但是大部分物理页面仅有小量的数据被访问过.鉴于移动处理器的 TLB 项数有限,这暗示着应用程序运行过程中可能会发生频繁的 TLB 缺失.

6.2 指令局部性

从微体系结构相关的特征分析中可以得出,指令 Cache 的缺失率较大,也就是说局部性不好.本节从应用程序与微体系结构无关的特征中,对此现象进行阐述.本文利用重用距离这个指标来反映指令的局部性.重用距离是指连续两次访问请求期间,其他访问请求的数量.指令的重用距离,这里指连续访问同一 Cache 的时间间隔内,其他访问指令 Cache 的请求数.

图 8 所示的是 Moby 各应用程序对应指令 Cache 的重用距离.从图可知,所有应用中仅有 30% 指令

表 3 Moby 指令特征总结

应用程序	指令数量 (Billion)	分支 指令/%	Load 指令/%	Store 指令/%	工作集 大小/MB
BBench	2.48	14.4	23.1	12.2	80
K9Mail	1.18	11.0	20.0	9.3	64
SinaWeibo	2.23	16.9	27.2	14.7	114
NeteaseNews	2.65	16.6	25.9	12.2	104
KingsoftOffice	2.24	16.6	26.1	14.1	87
AdobeReader	2.09	15.2	23.7	12.2	83
BaiduMap	3.53	14.3	22.8	12.3	102
MXPlayer	3.84	18.2	23.8	12.8	97
TTPod	3.87	15.2	25.5	12.8	126
FrozenBubble	0.28	15.6	21.5	9.7	47

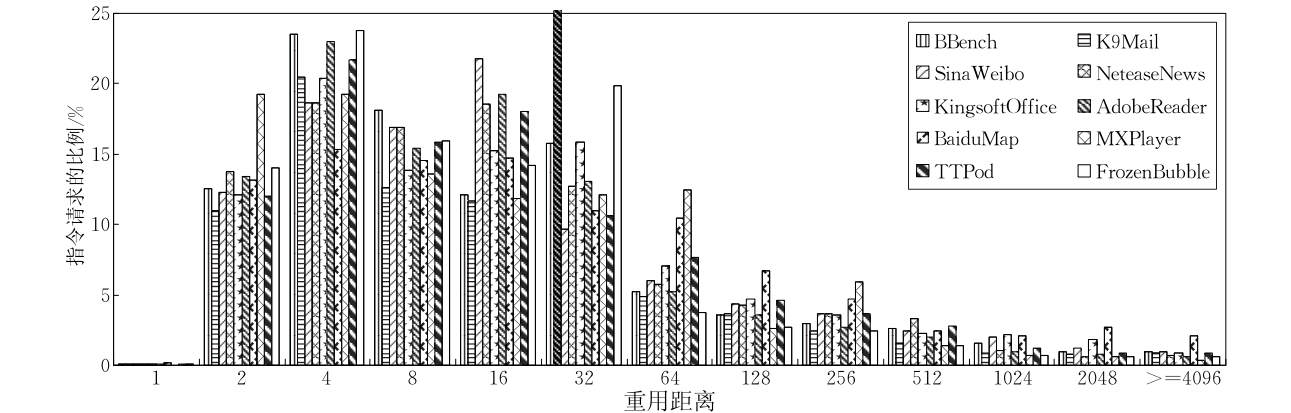


图 8 指令 Cache 的重用距离

请求的重用距离小于 4. 而本文实验中移动平台的指令 Cache 是 4 路组相联的. 以 Cache 中最常用的最近最少使用(LRU)替换策略而言,重用距离大于 4,意味着请求从第一次访问时的 MRU 位置,会慢慢地移动到 LRU 位置,直至被替换出去. 当下一次访问时,请求就会发生缺失. 这也就说明了 Moby 应用程序的指令 Cache 缺失率高的原因. 就算采用 L2 Cache 的 16 路组相联策略,也仅有 60%的请求会命中,仍然会有接近 40%的指令请求会发生缺失,而去访问开销较大的内存.

6.3 指令执行流

移动设备的应用程序一般都呈现出复杂的指令执行轨迹. 一方面,移动应用依赖于图像显示界面,需要频繁地调用图像库来描绘、合成新的图像;另一方面,Android 上的应用程序利用面向对象的 Java 语言编程,运行过程中也依赖 JVM 相关的库函数. 此外,移动设备中存在许多后台进程或线程,用于监控用户的操作、硬件平台的资源利用情况,这也可能改变应用程序运行过程中指令的执行轨迹.

图 9 显示了 Moby 各应用程序运行过程中创建的进程或线程数及调用库的数量. 大部分应用都创建了数十个进程或线程,调用库函数的数量基本都超过了 15 个. 其中,有 6 个应用甚至调用了 20 多个库函数. 这些库函数会显著地增加代码足迹,并给指令相关的部件如指令 Cache 和指令 TLB 的性能提出了更高的要求. 而且,多个进程或线程并发指令,也会相互干扰各自的执行流程,竞争硬件平台的资源.

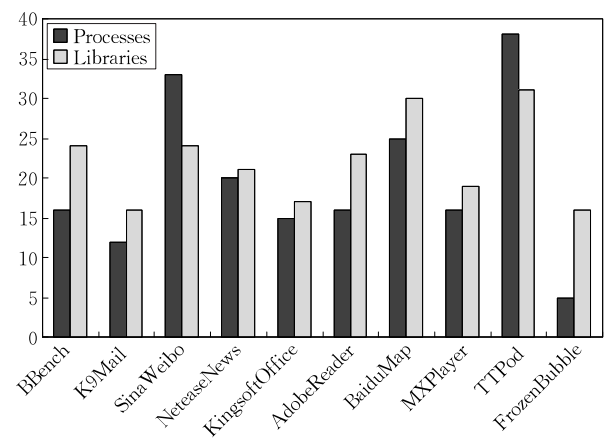


图 9 Moby 应用程序的进程或线程数以及调用库的数量

为了更好地了解指令的动态行为,本文收集了指令的访问序列,并将其对应到指令所在的可执行文件,例如源代码、库函数、操作系统内核等. 图 10

刻画了 KingsoftOffice 应用的指令执行流程图. 横坐标代表在时间序上应用已执行的指令数,而纵坐标显示各指令对应的二进制文件. 大体上,操作系统内核以及 5 个库函数主导了 KingsoftOffice 应用的指令. 这 5 个库函数分别是 dalvik-jit-code-cache、libdvm.so、libcutils.so、libnativehelper.so 以及 libc.so. 这几个库函数主要分为 3 类,Java 语言相关、C 语言相关,以及系统相关的函数. 同时,应用程序会在不同的文件中频繁地切换,如 libc.so 和内核基本是交叉执行的. 在这种情形下,指令的局部性以及分支预测的准确度都会受到影响.

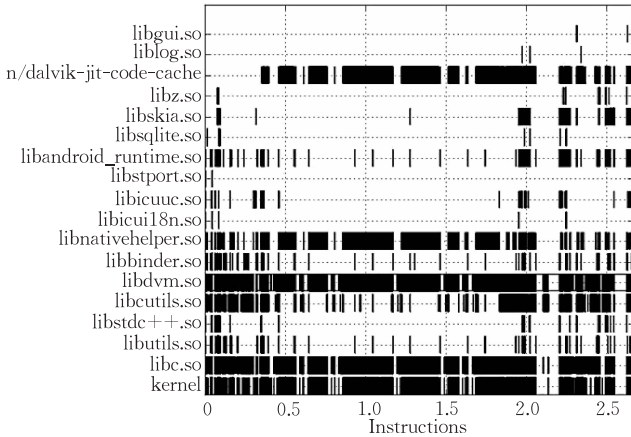


图 10 KingsoftOffice 程序的指令执行流

6.4 主成分特征分析

要设计一个基准测试程序集,不仅要详细地刻画各应用程序的特征,还需证明应用程序间存在较大的差异,能够代表具有不同特征的应用类别. 然而,不同的性能指标间都会存在一定的关联,有的是松耦合,有的是紧耦合. 因此,不能单纯地根据差异指标的数量来衡量. 本文采用主成分分析法^①区分指标间的关联度,以及不同指标的重要度. 主成分分析法可以将多个指标映射到形成多个关联度不大的主成员,而主成员的重要度通过概率权值来描述. 第一主成员,也就是概率权值最大的主成员,很大程度上反映了应用间的关系. 如果两个应用的主成员概率权值类似,则说明它们之间的相似度较大.

以上述微体系结构无关的特征指标为例,本文分析了 Moby 中应用程序主成分分析的结果,如图 11 所示. 横坐标显示各应用第一主成员(Dim 1)的值,而纵坐标刻画第二主成员(Dim 2)的权值. 第一主成员代表了 65%的差异度,第二主成员则代表

① Principal component analysis. https://en.wikipedia.org/wiki/Principal_component_analysis

19% 的差异度. 因而, 这两个主成员基本代表了应用程序间的差异度. 两个点距离越近, 意味着两个应用的差异度越低. 由图可以看出, Moby 的 10 个应用程序基本分散在不同的区域. 这也就是说, Moby 的应用程序具有各自独特的特征, 能够代表不同类型的应用程序.

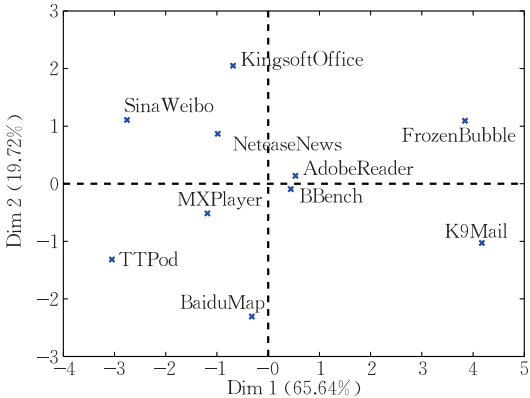


图 11 Moby 应用程序体系结构无关特征的主成分分析

7 相关工作

目前, 市场上有很多专业的评估软件, 用于对大型手机厂商新上市的机器做出测评, 测评的内容包括 CPU 计算性能、GPU 运算性能、内存读写性能等, 类似的有 EEMBC^①、Antutu、3D GLBenchMark^②、Geekbench^③ 以及测试 JavaScript 的 SunSpider, 测试 HTML5 的 Vellamo^④, 视频软解测试等软件. 然而, 大多数测试应用都是非开源的, 而且针对性很强, 不利于反映普遍情况, 因此, 我们只挑选了几个专业测试应用.

在研究领域, MiBench^[7] 是常用的嵌入式系统的基准测试程序, 包含了 6 类共 35 个应用程序. 不过这些应用程序与移动设备上的应用程序存在较大的差异, 包括编程语言及实现功能等. Gutierrez 等人在文献[5]中, 挑选并分析了几个应用程序作为测试程序例如浏览器应用 BBench、视频客户端, 不过其分析的应用程序种类不多, 没有包括地图、文档处理等类别的应用. MobileBench^[8] 仅选择浏览器、图像处理和视频播放应用, 并详细地分析了它们对不同体系结构部件如 Cache、TLB 等的敏感度. Sunwoo 等人^[3] 分析了几类应用的 Java 虚拟机、SQLite 和整体系统的性能. 总体来说, 已有工作仅分析了部分移动设备上的应用程序, 不能覆盖现有用户对应用程序的需求.

8 总 结

移动设备应用越来越广泛, 包括工业生产、日常办公、生活娱乐、学校教育等. 由于移动设备依靠电池续航, 因此在设计过程中对功耗和性能之间的权衡显得尤为重要. 并且, 在移动设备上的应用程序中, 不同类型的应用对于硬件资源的需求不同, 对于处理器资源的利用并不高, 很少应用能够得益于多核处理的设计. 因此, 需要对这些应用进行优化, 让它们能够更好地运行在移动设备上. 作为优化的指导, 需要分析不同应用程序的体系结构特征.

本文选取 12 类应用程序并对它们进行的微体系结构相关的特征进行了分析, 包括指令缓存、数据缓存、TLB、二级缓存相关的体系结构特征. 实验结果表明, 大部分移动设备上的应用程序都有较高的指令 Cache 和指令 TLB 缺失率, 存在较大的优化空间.

为了方便移动处理器的体系结构优化, 本文提出了一个移动设备的基准测试程序集 Moby. Moby 包含了浏览器、邮件客户端、音视频播放器、文档处理软件、地图及游戏等 10 个移动应用程序. 接着, 本文对这 10 个应用程序的自身特征, 也就是微体系结构无关的特征, 进行了详细的分析. 该基准测试程序集已被移植到常用的体系结构模拟器 Gem5 上, 并实现开源.

参 考 文 献

[1] Huang Y, Zha Z, Chen M, Zhang L. Moby: A mobile benchmark suite for architectural simulators//Proceedings of the IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS). Monterey, USA, 2014; 45-54

[2] Binkert N, Beckmann B, Black G, et al. The gem5 simulator. ACM SIGARCH Computer Architecture News, 2011, 39(2): 1-7

[3] Sunwoo D, Wang W, Ghosh M, et al. A structured approach to the simulation, analysis and characterization of smartphone applications//Proceedings of the IEEE International Symposium on Workload Characterization (IISWC). Portland, USA, 2013; 113-122

① EDN embedded microprocessor benchmark consortium. <http://www.eembc.org>

② Gfxbench. <https://gfxbench.com/result.jsp>

③ Geekbench. <http://www.primatelabs.com/geekbench>

④ Vellamo Mobile Benchmark. <http://www.quicinc.com/vellamo/>

- [4] Zhang Kun, Huang Yong-Bing, Chen Ming-Yu. Architecture characteristics and analysis of mobile device applications// Proceedings of the National Annual Conference on High Performance Computing. Guilin, China, 2013: 12-21(in Chinese) (张坤, 黄永兵, 陈明宇. 移动设备应用程序的体系结构特征分析//中国高性能计算学术年会. 桂林, 中国, 2013: 12-21)
- [5] Gutierrez A, Dreslinski R, Wenisch T, et al. Full-system analysis and characterization of interactive smartphone applications//Proceedings of the IEEE International Symposium on Workload Characterization (IISWC). Austin, USA, 2011: 81-90
- [6] Henning J. SPEC CPU2006 benchmark descriptions. ACM SIGARCH Computer Architecture News, 2006, 34(4): 1-17
- [7] Guthaus M, Ringenberg J, Ernst D, et al. Mibench: A free, commercially representative embedded benchmark suite// Proceedings of the IEEE International Workshop on Workload Characterization (IISWC). Austin, USA, 2001: 3-14
- [8] Pandiyan D, Lee S, Wu C. Performance, energy characterizations and architectural implications of an emerging mobile platform benchmark suite-mobilebench//Proceedings of the IEEE International Symposium on Workload Characterization (IISWC). Portland, USA, 2013: 133-142



HUANG Yong-Bing, born in 1987, Ph. D. candidate. His main research interests include computer architecture, operating systems and high performance computing.

CHEN Ming-Yu, born in 1972, Ph. D., professor. His main research interests include high performance computer architecture and operating systems.

Background

Mobile devices have already become the primary consumer computing devices with rapid growth. Thousands of diverse mobile applications are developed and widely used. Mobile applications always seek for high performance, but mobile platforms should consider both performance and power efficiency, especially for battery-driven devices. Due to the existence of mismatching between diverse mobile applications and various mobile platforms, there are requirements to study the architectural characteristics of mobile applications, in order to optimize the architectures of mobile platforms.

In recent two or three years, several researches were done on studying the architectural behaviors of mobile applications. However, most of them only focus on a small part of mobile applications such as web browser, video player, and game.

In this paper, we study 34 mobile applications from 13 different categories, which are carefully selected from the

commonly used Google Play application store. These applications almost cover all popular types' mobile applications and have high representative. Their micro-architectural dependent features are measured and analyzed on a mainstream ARM-based development platform. Moreover, based on above features, we pick out 10 applications and propose a mobile benchmark suite Moby. Moby has already been ported to full system architectural simulator Gem5 and used to optimize the micro-architectures of ARM-based mobile platforms.

This work was supported by the National Basic Research Program (973 Program) of China under Grant No. 2011CB302502, the National Natural Science Foundation of China under Grant Nos. 60925009, 61221062 and 61272132, the Strategic Priority Research Program of the Chinese Academy of Sciences under Grant No. XDA06010401, and the Huawei Research Program under Grant No. YBCB2011030.