

Android应用程序能耗分析与建模研究

段林涛^{1,2,3}, 郭兵¹, 沈艳⁴, 王毅¹, 张文丽¹, 熊伟¹

(1. 四川大学计算机学院 成都 610065; 2. 成都大学信息科学与技术学院 成都 610106;

3. 成都大学模式识别与智能信息处理四川省高校重点实验室 成都 610106; 4. 成都信息工程学院控制工程学院 成都 610225)

【摘要】应用程序能耗分析与建模是智能移动终端能耗优化的重要组成部分。针对智能移动终端丰富的应用程序,提出了一种基于应用程序运行时间的能耗模型。与精度高和复杂的应用程序组件能耗模型相比,该模型使用时间变量刻画和包含终端的功耗、性能等多种因素,而且运行时间容易精确测量和获取,能够快速估算应用程序运行时移动终端产生的能耗。实验结果表明,在GT-I9108、GT-I9308和GT-P3108实验平台下,该模型的能耗估算结果与Android操作系统应用程序框架提供的组件能耗模型测量值相比平均误差分别为0.89%、1.37%和0.29%,能够为移动终端用户便捷地预测应用程序消耗的电池电能提供帮助。

关键词 应用程序能耗模型; 电池电能; 能耗模型; 移动终端; 功耗

中图分类号 TP302

文献标志码 A

doi:10.3969/j.issn.1001-0548.2014.02.022

Analysis and Modeling of Android Application Energy Consumption

DUAN Lin-tao^{1,2,3}, GUO Bing¹, SHEN Yan⁴, WANG Yi¹, ZHANG Wen-li¹, and XIONG Wei¹

(1. School of Computer Science, Sichuan University Chengdu 610065;

2. School of Information Science and Technology, Chengdu University Chengdu 610106;

3. Key Laboratory of Pattern Recognition and Intelligent Information Processing, Chengdu University Chengdu 610106;

4. School of Control Engineering, Chengdu University of Information Technology Chengdu 610225)

Abstract Analysis and modeling of application energy consumption plays a vital role in the optimization of energy consumption of smart mobile devices. An application energy model based on application's running time is proposed. Compared with application's component energy models with high accuracy and complexity, the model is characterized by the time variable and contains a variety of mobile devices' properties, such as power consumption and performance, which can be used to rapidly estimate the mobile devices' energy consumption during application execution, and the execution time of the application is easy to measure and obtain. The experiment results show that the average error rate of proposed model is 0.89%, 1.37% and 0.29% compared with that measured by component energy model provided by Android application framework on GT-I9108, GT-I9308 and GT-P3108, respectively. The model can be used to help the end users to rapidly and conveniently predict the battery energy consumption of applications.

Key words application energy model; battery energy; energy model; mobile devices; power consumption

智能移动终端上丰富的应用程序为用户提供视听享受的同时也消耗着大量的电池电能。随着软件层、硬件层及软硬件协同节能技术的深入研究,应用程序级节能技术也越来越受到研究人员重视^[1]。2012年移动终端应用程序下载量超过456亿次,其中免费下载405.99亿次,付费下载50.18亿次^[2]。在这样繁荣的应用程序市场中却存在大量应用程序并未采用有效的节能技术为用户节能^[3]。因此为终端用

户提供测量与估算应用程序能耗的原型工具与方法就显得十分重要。

近年来智能移动终端能耗问题已经被广泛研究。在低功耗设计方面,文献[4-7]分别提出基于软件层和硬件层的系统低功耗设计方法。在功耗测量与建模方面,文献[8-9]使用功率计测量移动终端功耗,这种方法明显提升功耗测量精度,但对终端用户显得专业而不易于操作,同时也不便于在移动环

收稿日期: 2012-12-20; 2013-11-22

基金项目: 国家自然科学基金(61332001,61272104,61073045); 四川省杰出青年科技基金(2010JQ0011); 四川省产学研创新联盟合作项目(2012ZZ0010); 四川省科技支撑计划(2012GZ0088-1)

作者简介: 段林涛(1978-),男,博士生,主要从事嵌入式实时系统方面的研究。

境下提供应用程序实时能耗信息。文献[10-13]提出不使用辅助测量仪器的在线功耗模型。文献[14-15]提出基于组件利用率的功耗模型,根据组件各自的功耗特性,为每类组件的功耗单独进行建模。该方法提高了在线功耗模型估算的准确性,却增加了模型的复杂度和系统的估算成本。文献[16]使用微体系结构能耗模型估算单时钟周期指令能耗,提出基于函数、例程和服务的内核能耗估算模型。针对软件设计阶段对能耗进行评估的需求,文献[17]以进程代数语言(communicating sequential processes, CSP)的迹模型为基础,提出一种基于路径的体系结构级能耗分析评估模型。针对应用程序执行结束后部分组件,如GPS、Wi-Fi网络接口卡(network interface card, NIC)等仍处于较高功耗状态的尾部功耗(tail power)问题,文献[18]提出基于系统调用的能耗分析与建模方法。

本文利用Android操作系统应用程序框架提供的基于组件能耗模型的应用程序能耗测量方法,通过对三类移动终端上Temple Run应用程序能耗数据采集与分析,发现在不同终端上应用程序能耗与执行时间之间存在近似的线性关系,提出一种基于时间变量的应用程序能耗模型。与Android操作系统提供的基于组件能耗模型的测量值相比,时间能耗模型在保证一定误差范围(0.01%~7.82%)的基础上,方便终端用户利用独立于硬件功耗特性(power characteristics)的时间变量估算应用程序能耗。

1 应用程序能耗模型

移动终端是一种典型的软件驱动硬件的嵌入式设备。硬件的电路活动直接导致电池的放电行为,应用程序的执行会申请操作系统驱动底层硬件的电路活动,间接导致系统能耗的产生。现代移动终端包含丰富的硬件组件,如CPU、显示组件、图像视频组件、Wi-Fi NIC、2G/3G蜂窝网络接口、蓝牙、GPS和音频组件等。各硬件组件具有复杂的功耗特性,表现为软件使用到的硬件组件的工作状态在软件任务执行过程中会动态发生变化。

1.1 应用程序的组件能耗模型

设集合 $T=\{\tau_1, \tau_2, \dots, \tau_i, \dots, \tau_M\}$ 表示移动终端上运行的 M 个应用程序,集合 $D=\{d_1, d_2, \dots, d_i, \dots, d_N\}$ 表示移动终端包含的 N 类硬件组件,集合 $S=\{s_1, s_2, \dots, s_i, \dots, s_K\}$ 表示每类组件包含的 K 种工作状态,集合 $P=\{p_1, p_2, \dots, p_i, \dots, p_K\}$ 表示组件在 K 种工作状态下对应的非递增排列的 K 种功耗值,组件在不同功耗状态

间切换会产生额外能耗成本,图1显示程序 τ_i 在时间段 t_0 到 t_4 之间组件 d_i 的功耗状态转换模型。 $E_{s_2 \rightarrow s_1}$ 与 $E_{s_1 \rightarrow s_3}$ 分别表示组件 d_i 在 s_2 到 s_1 和 s_1 到 s_3 状态切换过程中产生的能耗成本。根据图1所示,程序 τ_i 在 t_0 至 t_4 时刻的运行导致组件 d_i 产生的能耗为:

$$E_i = p_2(t_1 - t_0) + p_1(t_3 - t_2) + E_{s_2 \rightarrow s_1} + E_{s_1 \rightarrow s_3} \quad (1)$$

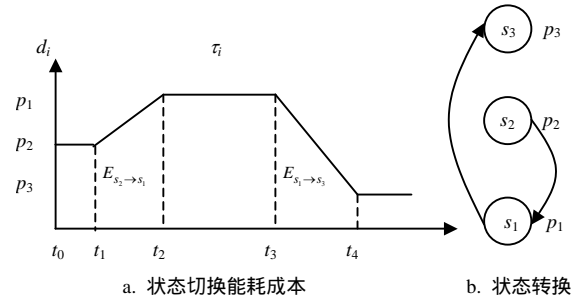


图1 程序 τ_i 在 t_0 到 t_4 时刻驱动组件 d_i 活动的功耗状态转换模型
则第 i 类组件的能耗模型为:

$$\begin{cases} E_i(t) = \sum_{j=1}^K P_i^j \cdot t_i^j + E_{sw_i} \\ E_{sw_i} = \sum_{i=1}^K \sum_{j=1}^K (E_{s_i \rightarrow s_j} C_{s_i \rightarrow s_j}) \\ t = \sum_{j=1}^K t_i^j + \sum_{i=1}^K \sum_{j=1}^K (t_{s_i \rightarrow s_j} C_{s_i \rightarrow s_j}) \end{cases} \quad (2)$$

式中, $E_i(t)$ 为第 i 类组件在时间 t 内的能耗值; P_i^j 为第 i 类组件处于第 j 种工作状态下的功耗值; E_{sw_i} 为第 i 类组件在状态转换过程中产生的能耗; $C_{s_i \rightarrow s_j}$ 表示状态转换系数,当 $i=j$ 或者组件在程序 τ_i 执行的时间 t 内不存在 s_i 到 s_j 的状态转换时 $C_{s_i \rightarrow s_j}=0$,否则 $C_{s_i \rightarrow s_j}=1$; $t_{s_i \rightarrow s_j}$ 表示组件从状态 s_i 切换到 s_j 的延迟时间。

应用程序通过操作系统驱动不同的硬件组件完成指令执行和数据存储。图2显示了一款下载量超5 000万次的互动游戏Temple Run在3种智能终端(GT-I9108、GT-I9308和GT-P3108)执行时能耗在组件一级的分布情况。Temple Run执行过程中会驱动SOC、显示组件、Wi-Fi NIC和传感器等组件。其中GT-I9108实验平台下处理器、显示组件和Wi-Fi NIC能耗分别占应用程序总能耗的28.69%、48.29%和22.75%。Temple Run执行导致Wi-Fi NIC电路活动的根本原因是游戏内置有第三方的广告模块,广告进程会申请操作系统驱动Wi-Fi组件访问广告服务器。

依据软件能耗由硬件电路活动产生的理论,应用程序能耗可以表示为应用程序申请操作系统驱动的组件能耗之和。设应用程序在执行时间 t 内,通过操作系统驱动了 N 类组件,每类组件有 K 种工作状

态, 应用程序的组件能耗模型可以表示为:

$$E(t) = \sum_{i=1}^N E_i(t) = \sum_{i=1}^N \left(\sum_{j=1}^K P_i^j \cdot t_i^j + E_{sw_i} \right) \quad (3)$$

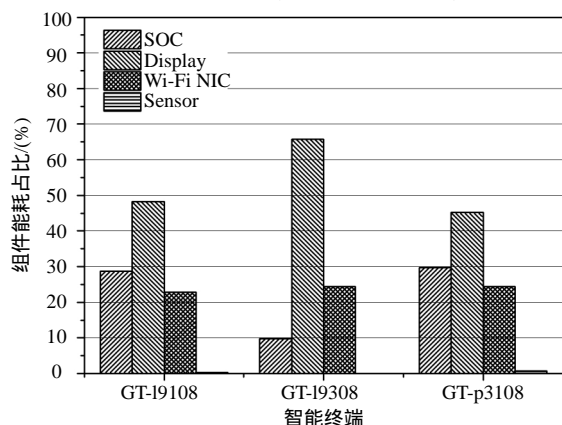
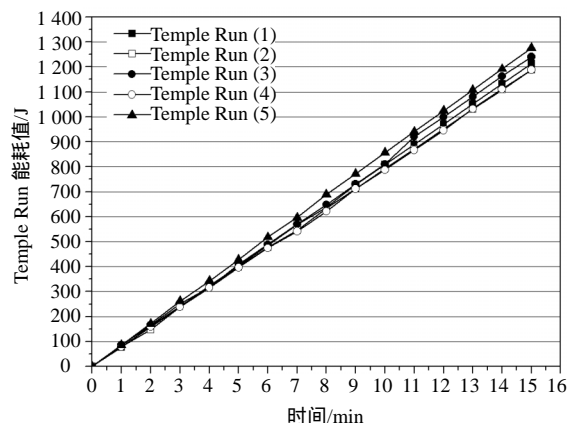


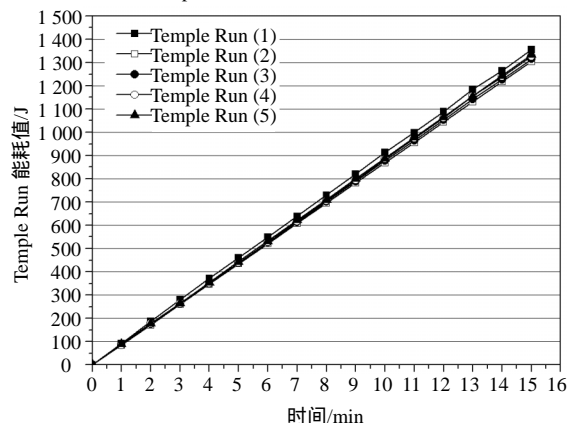
图2 3种终端上程序Temple Run的组件能耗分布

1.2 应用程序的时间能耗模型

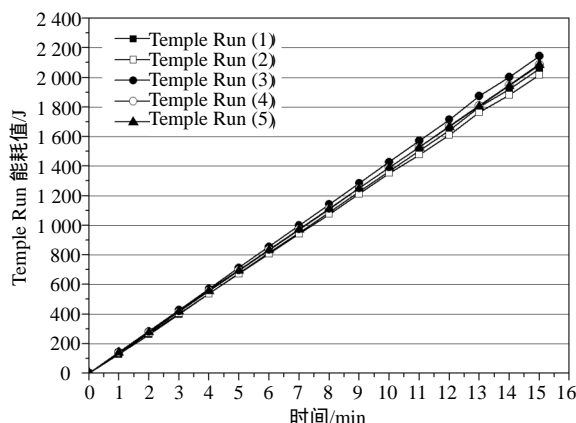
应用程序的组件能耗模型在描述应用程序能耗方面具有准确性高的特点, 但对普通终端用户该模型并不能快速估算软件能耗。本文通过组件能耗模型对Temple Run在3种终端上的能耗进行测试与分析, 提出一种简单、快速的应用程序时间能耗模型。



a. Temple Run在GT-I9108上的5次实验数据



b. Temple Run在GT-I9308上的5次实验数据



c. Temple Run在GT-P3108上的5次实验数据

图3 应用程序能耗与时间之间的关系比较

图3显示Temple Run在3种实验平台下能耗与时间之间存在近似的线性关系。本文基于简单线性回归方法得到时间能耗模型为:

$$E(t) = \beta_1 t + \beta_0 \quad (4)$$

式中, β_1 和 β_0 是独立于设备特性的回归系数, 根据最小二乘法可以计算出对应的值。基于该模型终端用户仅依赖时间变量即可估算应用程序能耗。

1.3 误差分析

应用程序的时间能耗模型利用时间变量刻画和包含了组件能耗模型中各组件的复杂功耗特性, 存在隐藏变量产生的误差值^[19]为:

$$\varepsilon = \frac{1}{N} \sum_{i=1}^N [E_{\text{actual}}(t) - E_{\text{model}}(t)]^2 \quad (5)$$

式中, N 表示实验迭代次数; $E_{\text{actual}}(t)$ 表示应用程序在 t 时间内的实际能耗; $E_{\text{model}}(t)$ 表示模型在 t 时间内的估算值。为最小化误差 ε , 使 $E_{\text{model}}(t)$ 尽可能接近 $E_{\text{actual}}(t)$, 需找到最优回归系数 β_1 和 β_0 。

2 能耗模型拟合

为了验证本文的时间能耗模型的有效性, 下面以Android智能手机为例具体确定能耗模型的参数, 实现函数拟合, 但模型和方法对于苹果IOS和Windows Phone等智能手机也是适用的。具体实验平台为Samsung Galaxy SII(GT-I9108)、Samsung Galaxy SIII(GT-I9308)与Samsung Galaxy Tab2 P3108(GT-P3108) Android智能终端。

2.1 组件功耗值的获取

移动终端各组件功耗值 P_i^j 可以根据Android应用程序框架提供的能耗分析器配置文件(PowerProfile.xml)获得。表1列出了GT-I9108、GT-I9308和GT-P3108每一类组件在不同工作状态下

的功耗值。如GT- I9108移动终端的处理器具有6种工作状态,当无应用程序执行时,处理器处于IDLE状态,具有最低功耗(7.4 mw)。当应用程序在处理器上执行时,处理器利用动态电压和频率调节技术(dynamic voltage and frequency scale, DVFS),根据预

定义的频率电压表动态调节工作电压和频率以节省电池电能。文件/sys/devices/system/cpu/cpu0/cpufreq中保存有处理器的5种工作频率:300、600、800、1 008、1 200 MHz,它们对应的功耗值分别是203.5、547.6、921.3、1 509.6、2 134.9 mw。

表1 硬件组件状态与对应功耗值

组件	实验平台			状态	功耗/mw		
	I9108	I9308	P3108		I9108	I9308	P3108
CPU	Dual-core 1.2GHz ARM Cortex-A9 Harvard Superscalar	Quad-core 1.4 GHz ARM Cortex-A9 Harvard Superscalar	Dual-core 1.0 GHz ARM Cortex-A9 Harvard Superscalar	IDLE	7.4	14.8	25.9
				STEP-0	2 134.9	2 134.9	444
				STEP-1	1 509.6	1 509.6	843.6
				STEP-2	921.3	921.3	1 106.3
				STEP-3	547.6	547.6	1 468.9
				STEP-4	203.5	203.5	-
Wi-Fi	802.11 a/b/g/n	802.11 a/b/g/n	802.11 a/b/g/n	OFF	0	0	0
				ON	1.11	1.11	3.7
				ACTIVE	307.1	355.2	555
GPS	aGPS, Geotagging	aGPS, QuickGPS, Geotagging	aGPS, QuickGPS, Geotagging	OFF	0	0	0
				ON	3.7	3.7	203.5
(Display)	Super AMOLED Plus (480 x 800)	Super AMOLED (1 280 x 720)	TFT (1 024 x 600)	OFF	0	0	0
				ON	277.5	262.7	592
				FULL	743.7	1406	740
				ON-0	9.99	125.8	11.1
(Cellular)	GSM, TD-SCDMA	GSM, TD-SCDMA	GSM, TD-SCDMA	ON-1	11.1	125.8	11.1
				SCAN	303.4	303.4	333
				ACTIVE	895.4	925	555
				OFF	0	0	0
(Bluetooth)	V3.0	V4.0	V3.0	ON	1.11	1.11	3.7
				ACTIVE	118.4	62.9	555

2.2 组件工作时间的获取

第*i*类组件在第*j*种工作状态下的工作时间 t_i^j 可以根据 Android 应用程序框架的 BatteryStats 和 BatteryStatsImpl提供的接口函数获取。基于应用程序的组件能耗模型,利用表2所示应用程序编程接口(API),Android为移动开发人员提供了对应用程序能耗进行分析和测量的能耗分析器,利用基于分析器设计的能耗监控器收集应用程序能耗是现有 Android平台提供软件能耗在线测量的方式。

表2 主要组件的工作时间获取函数

类	API	相关组件
BatteryStats	getTimeAtCpuSpeedStep	CPU
BatteryStats	getWakeTime	Wakelock
BatteryStats	getWifiRunningTime	Wi-Fi
BatteryStats	getSensorTime	Sensor
BatteryStatsImpl	getPhoneOnTime	Phone
BatteryStatsImpl	getScreenOnTime	Screen
BatteryStatsImpl	getScreenBrightnessTime	Screen
BatteryStatsImpl	getPhoneSignalStrengthTime	Phone
BatteryStatsImpl	getPhoneSignalScanningTime	Phone
BatteryStatsImpl	getWifiOnTime	Wi-Fi
BatteryStatsImpl	getGlobalWifiRunningTime	Wi-Fi
BatteryStatsImpl	getBluetoothOnTime	Bluetooth

2.3 组件能耗的获取

本文所有实验数据都是通过能耗监控器获取。能耗监控器作为守护进程工作在后台,间隔20 s收集一次应用程序、组件以及整个系统的能耗并记录在日志文件中。为提高应用程序能耗测量的准确性,应用程序执行过程中实际调用组件的总能耗 $E(t)$,需

排除仅运行能耗监控器条件下的系统基准能耗。实验过程设置移动终端为飞行模式,确保通信接口处于OFF状态,保证能耗监控器在后台运行,关闭显示屏,时间*t*后从日志文件获得一次系统基准能耗,实验迭代*N*次,对获取的系统基准能耗求均值计算出系统基准能耗均值 $E_{base}(t)$ 。在 Temple Run执行过程中,为确保实验环境一致,设置3类终端上Wi-Fi接口为ON,屏幕亮度为120并关闭音频设备。应用程序在能耗监控器开启的情况下反复执行*N*次,获得应用程序执行过程中整个系统的能耗均值 $E_{total}(t)$ 。

应用程序实际调用组件的能耗可以表示为:

$$E(t) = E_{total}(t) - E_{base}(t) \quad (6)$$

2.4 能耗模型参数的确定

实验中训练集数据的获取步骤是首先打开无线网络连接,然后运行应用程序。执行*t*时间后,收集一次实验数据,依据应用程序的组件能耗模型,按式(3)和式(6),计算应用程序能耗值,迭代*N*次,获得*N*组应用程序能耗与时间的变化关系。

对*N*组训练集数据采用线性回归方法拟合确定时间能耗模型回归系数。实验中分别对3类终端上运行的 Temple Run使用时间能耗模型建模,能耗监控器收集程序能耗及组件能耗,迭代执行*N*次,求均值后使用线性回归方法,最小化经验误差 ε ,求出时间模型的回归系数并记录在表3中。

表3 应用程序的时间能耗模型

应用程序	实验平台	时间能耗模型/ $J \cdot min^{-1}$
Temple Run	I9108	$E(t) = 81.506t - 1.481$
	I9308	$E(t) = 88.515t + 0.301$
	P3108	$E(t) = 138.786t - 2.748$

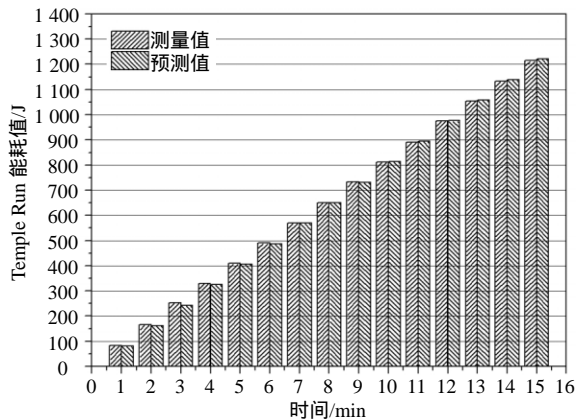
3 实验结果及分析

为评价时间能耗模型的准确性,本文利用时间能耗模型预测应用程序在执行时间 t 内的能耗值,并与测量值进行比较,发现时间能耗模型具有较高的准确性,3类实验平台下平均误差都在2%以内。

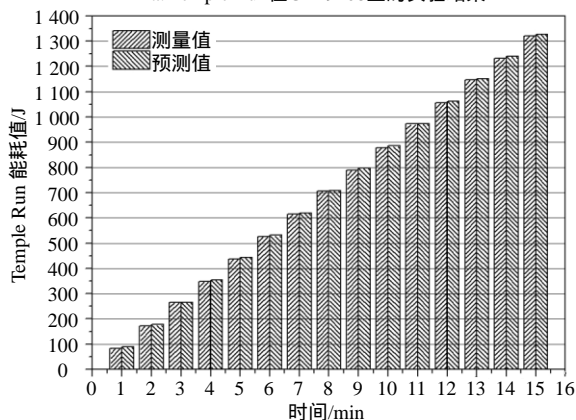
3.1 Temple Run

Temple Run是一款移动终端游戏。游戏软件从Android应用程序市场下载并安装在3台测试目标机上。为验证3类不同智能终端上应用程序的时间能耗模型的准确性,在每种终端上都全新执行一次Temple Run,获得一组测量数据。实验步骤是首先接入Wi-Fi无线网络,运行Temple Run,15 min后获取能耗监控器记录的日志文件。

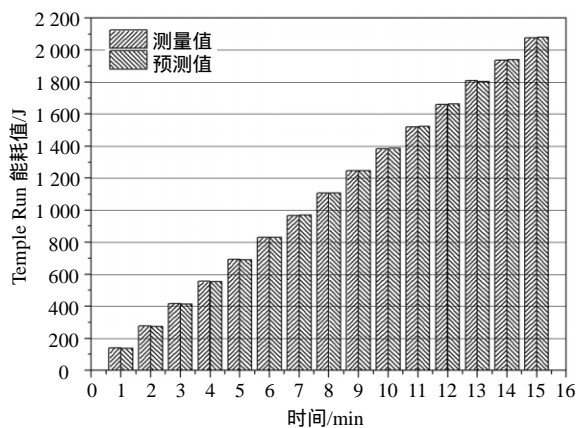
图4显示了基于组件能耗模型的测量值与使用应用程序的时间能耗模型得到的预测值的对比关系。从图4中可以看出时间能耗模型能很好反映Temple Run能耗真实值,在GT-I9108、GT-I9308和GT-P3108中平均误差分别接近0.89%、1.37%和0.29%。



a. Temple Run在GT-I9108上的实验结果



b. Temple Run在GT-I9308上的实验结果



c. Temple Run在GT-P3108上的实验结果

图4 Temple Run能耗测量值与预测值比较

3.2 能耗监控器

能耗监控器对每种应用程序的能耗进行采样并保持在一个比较低的能耗状态,如图5所示。能耗监控器15 min内在GT-I9108、GT-I9308和GT-P3108平台下的能耗值分别低于2.5、1.9、3.1 J。

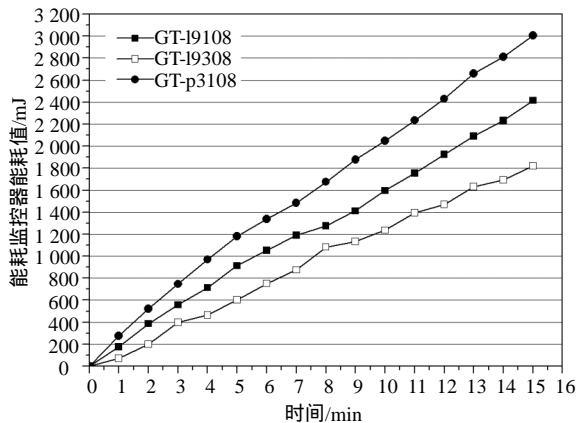


图5 能耗监控器的能耗

4 结 论

本文对移动终端上应用程序的组件能耗模型进行了分析与建模,针对现有能耗测量方法的优势与不足,提出了一种基于应用程序运行时间的时间能耗模型,利用时间变量刻画了组件的功耗、性能等多种因素,避免了对功率测量工具的依赖,简化了应用程序的组件能耗模型。利用Android应用程序框架提供的基于组件能耗模型的功率分析器和API接口对应用程序能耗进行测量,测量值验证了时间能耗模型的有效性。下一步工作主要包括将时间能耗模型与不同组件的复杂功耗特性相结合,综合能耗估算的速度与精度,进一步优化应用程序的时间能耗模型以及进一步降低能耗监控器自身能耗。

参 考 文 献

[1] 郭兵, 沈艳, 邵子立. 绿色计算的重定义与若干探讨[J].

- 计算机学报, 2009, 32(12): 2311-2319.
- GUO Bing, SHEN Yan, SHAO Zi-li. The redefinition and some discussion of green computing[J]. Chinese Journal of Computers, 2009, 32(12): 2311-2319.
- [2] GARTNER. Gartner says free apps will account for nearly 90 percent of total mobile app store downloads in 2012[EB/OL]. (2012-09-11). [2012-10-08]. <http://www.gartner.com/it/page.jsp?id=2153215>.
- [3] PATHAK A, HU Y C, ZHANG M. Bootstrapping energy debugging for smartphones: a first look at energy bugs in mobile devices[C]//Proceedings of the 10th ACM Workshop on Hot Topics in Networks. Cambridge, USA: ACM, 2011: 1-6.
- [4] DONG M, ZHONG L. Chameleon: a color-adaptive web browser for mobile OLED displays[J]. IEEE Transactions Mobile Computing, 2012, 11(5): 724-738.
- [5] LIU X, SHENOY P, CORNER M D. Chameleon: Application level power management[J]. IEEE Transactions on Mobile Computing, 2008, 7(8): 995-1010.
- [6] CHANG M, CHANG C, CHAO C, et al. Transistor-and circuit-design optimization for low-power CMOS[J]. IEEE Transactions Electron Devices, 2008, 55(1): 84-95.
- [7] 张戈. 片上多核处理器的结构级功耗建模与优化技术研究[J]. 自然科学进展, 2009, 19(12): 1398-1409.
- Zhang Ge. Research on architectural layer power modeling and optimizing technology on MPSoC[J]. Progress in Natural Science, 2009, 19(12): 1398-1409.
- [8] KENNEDY M, VENKATARAMAN H, MUNTEAN G M. Energy consumption analysis and adaptive energy saving solutions for mobile device applications in green IT: Technologies and applications[M]. Berlin: Springer-Verlag, 2011.
- [9] CARROLL A, HEISER G. An analysis of power consumption in a smartphone[C]//Proceedings of the 2010 USENIX Annual Technical Conference. Boston, USA: [s.n.], 2010: 21-21.
- [10] ZHANG L, TIWANA B, QIAN J, et al. Accurate online power estimation and automatic battery behavior based power model generation for smartphones[C]//Proceedings of the 8th IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis. Scottsdale, USA: IEEE, 2010: 105-114.
- [11] DONG M, ZHONG L. Self-constructive high-rate system energy modeling for battery-powered mobile systems[C]//Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services. Bethesda, USA: [s.n.], 2011: 335-348.
- [12] JUNG W, KANG C, YOON C, et al. DevScope: a nonintrusive and online power analysis tool for smartphone hardware components[C]//Proceedings of the 8th IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis. Tampere, Finland: IEEE, 2012: 353-362.
- [13] BALASUBRAMANIAN N, BALASUBRAMANIAN A, VENKATARAMANI A. Energy consumption in mobile phones: a measurement study and implications for network applications[C]//Proceedings of the 9th ACM SIGCOMM Conf. Internet Measurement Conference. Chicago, USA: ACM, 2009: 280-293.
- [14] CIGNETTI T, KOMAROV K, ELLIS C. Energy estimation tools for the palm[C]//Proceedings of the 3rd ACM international workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems. Boston, USA: ACM, 2000: 96-103.
- [15] SHYE A, SCHOLBROCK B, MEMIK G. Into the wild: Studing real user activity patterns to guide power optimizations for mobile architectures[C]//Proceedings of the 42nd Annual IEEE/ACM International Symposium on Micro-Architecture. New York, USA: IEEE, 2009: 168-178.
- [16] 赵霞, 郭耀, 雷志勇, 等. 基于模拟器的嵌入式操作系统能耗估算与分析[J]. 电子学报, 2008, 36(2): 209-215.
- ZHAO Xia, GUO Yao, LEI Zhi-yong, et al. Estimation and analysis of embedded operating system energy consumption[J]. Acta Electronica Sinica, 2008, 36(2): 209-215.
- [17] 张滕滕, 吴晓, 李长德, 等. 基于CSP的构件化嵌入式软件能耗分析与评估方法研究[J]. 计算机学报, 2009, 32(9): 1876-1883.
- ZHANG Teng-teng, WU Xiao, LI Chang-de, et al. On energy- consumption analysis and evaluation for component-based embedded system with CSP[J]. Chinese Journal of Computers, 2009, 32(9): 1876-1883.
- [18] PATHAK A, HU Y C, ZHANG M, et al. Fine-grained power modeling for smartphones using system call tracing [C]//Proceedings of the 6th Conference on Computer Systems. Salzburg, Austria: [s.n.], 2011: 153-168.
- [19] ALPAYDM E. Introduction to machine learning[M]. Massachusetts: The MIT Press, 2004.
- 编辑 漆 蓉