

软件能耗优化技术研究进展

赵霞^{1,2,3} 郭耀^{2,3} 陈向群^{2,3}

¹(北京工商大学计算机与信息工程学院 北京 100037)

²(高可信软件技术教育部重点实验室(北京大学) 北京 100871)

³(北京大学信息科学技术学院软件研究所 北京 100871)

(zhaoxia@sei.pku.edu.cn)

Research Progresses on Energy-Efficient Software Optimization Techniques

Zhao Xia^{1,2,3}, Guo Yao^{2,3}, and Chen Xiangqun^{2,3}

¹(College of Computer and Information Engineering, Beijing Technology and Business University, Beijing 100037)

²(Key Laboratory of High Confidence Software Technologies (Peking University), Ministry of Education, Beijing 100871)

³(Software Engineering Institute, School of Electronics Engineering and Computer Science, Peking University, Beijing 100871)

Abstract In battery-driven embedded systems and mobile devices design areas, energy consumption has become one of the most critical constraints. In order to design the embedded systems with high performance and low power, it is necessary to consider optimization techniques from both hardware and software perspectives and pursue the optimal tradeoff between performance and energy consumption. This paper presents a concept of software energy consumption and the characteristics of software-based low energy techniques, and focuses on the software-based low energy techniques during the system development stages. Software-based optimization techniques for reducing system energy consumption in development stage include instruction-level optimization, algorithm-level optimization, and software architectural-level optimization. The primal issues and recent research progresses on software energy consumption optimization are presented. As a key supporting technique for low energy software development, software energy consumption evaluation techniques are also analyzed in detail, which include how to estimate software energy consumed by the processor and by the whole system. The instruction-level energy consumption model, architectural-level energy consumption model, and the macro-based energy consumption model for processor energy consumption estimation are discussed in detail. Finally, several challenges and open issues in software energy consumption research are summarized.

Key words low energy; software energy consumption; optimization; estimation; energy-efficient computing

摘要 为了设计高性能低能耗的系统,需要从硬件设计和软件设计两个方面进行考虑,以取得性能和能耗的最佳权衡。研究利用软件技术降低系统能耗的问题,主要探讨系统开发阶段的低能耗软件优化与

收稿日期:2010-04-19;修回日期:2010-11-15

基金项目:国家“八六三”高技术研究发展计划基金项目(2008AA01Z133);“核高基”国家科技重大专项基金项目(2009ZX01036-001-002);国家“九七三”重点基础研究发展计划基金项目(2009CB320703);创新群体科学基金项目(60821003);中国博士后科学基金项目(20090450234)

评估技术. 优化技术包括指令级优化、算法级优化与软件体系结构优化 3 类, 阐述在各类优化技术研究中面临的问题和当前的研究工作进展; 深入讨论了低能耗软件优化的关键支撑技术——软件能耗估算, 指出并分析面向处理器和面向全系统的软件能耗估算面临的主要问题和研究进展; 最后展望进一步研究的主要问题和发展趋势.

关键词 低能耗; 软件能耗; 优化; 估算; 高能效计算

中图法分类号 TP316

随着微电子和移动计算技术的快速发展, 降低系统能耗已经成为一项重要的系统设计目标. 尤其在电池供电的嵌入式系统中, 能耗成为系统设计的重要约束, 在各个设计层次上得到广泛重视和深入研究.

在物理学中, 能耗是能量消耗 (energy consumption) 的简称, 单位时间里的能耗称为功耗 (power consumption). 在计算机系统中, 以 CMOS 数字集成电路为基础的微处理器、存储器等器件的能耗是器件在一段时间里消耗的电能, 按照 CMOS 电路的能耗公式计算^[1]. 系统能耗指整个计算机系统的能量消耗, 可以通过微处理器、内存、外设等部件能耗总和来计算.

如同计算机系统性能优化技术经历了从早期关注硬件性能, 到后来扩展到包括软件在内的整个系统的发展过程一样, 计算机系统的能耗优化技术也经历着从硬件到软件及整个系统的发展过程. 早期低能耗技术研究主要在硬件电路级、器件级、体系结构级展开, 通过改进电路结构和物理参数来降低硬件功耗^[2]. 随着硬件低功耗技术的发展, 人们的关注点逐渐扩展到由各种低功耗硬件和软件构成的系统. 为了设计高性能、低能耗的系统, 需要从硬件设计和软件设计两个方面进行全盘考虑, 以取得性能和能耗的最佳权衡. 近 10 年来, 在硬件给定的情况下, 通过软件技术降低系统能耗的问题得到广泛重视, 开发低能耗软件成为系统设计者的重要目标之一^[3].

从研究目标角度, 可以把现有的软件低能耗技术划分为能耗优化与能耗评估; 从面向的系统生命周期阶段的角度, 可以分为面向系统运行阶段和面向系统开发阶段, 如表 1 所示. 面向运行系统的软件低能耗技术把软件当作系统能耗管理者, 在系统运行时监测、调节硬件的功耗状态, 降低系统能耗. 面临的主要问题是满足系统功能和性能约束的前提下, 如何设置硬件单元在低功耗状态下工作, 达到降低系统能耗的目的.

面向开发阶段的软件低能耗技术把软件当作系统能耗的驱动者, 在系统开发阶段优化软件本身的指令序列、算法和结构, 降低系统能耗. 面临的主要问题首先是评估软件如何影响系统能耗, 其次是如何在满足系统功能和性能约束的前提下优化软件, 达到降低系统能耗的目的.

与硬件低功耗技术相比, 软件低能耗技术的特点如下:

1) 系统性. 软件低能耗技术从系统角度同时考虑硬件功耗状态和上层应用的需求, 以降低系统能耗为目的, 对系统能耗的影响范围更大.

2) 抽象性. 相对于从硬件电路和处理器体系结构层看功耗而言, 从软件层看系统能耗的抽象层次更高, 更易于从系统和软件开发者的角度理解和评估系统能耗, 同时也带来如何从软件角度正确评估系统能耗的问题.

3) 灵活性. 软件在开发、部署和应用方面比硬件具有更大的灵活性.

4) 互补性. 软件低能耗技术和硬件低功耗技术互为补充, 各有偏重. 只有软硬件协同, 发挥各自的优势, 才能高效地发挥硬件特性, 达到保证性能、降低系统能耗的目标.

基于以上特点, 软件低能耗技术在系统设计和软件研究领域受到广泛的关注, 特别是编译器、操作系统、中间件、应用软件等软件研究领域中都出现了软件低能耗技术的研究.

面向运行系统的软件低能耗技术主要关注运行系统部件的功耗状态和工作负载, 在满足系统性能要求的前提下, 通过软件来动态设置空闲部件到低功耗状态以降低系统能耗. 这方面的工作包括感知能耗的操作系统资源管理与任务调度^[4-5]、感知能耗的应用程序动态调节^[6]、感知能耗的多核处理器任务调度^[7]和动态功耗管理^[8]、感知能耗的网络协议^[9]等. 其中, 利用操作系统实现动态能耗管理方面的研究进行得比较深入, 在现有的 Windows, Linux, Symbian 等操作系统中已经得到应用, 相关进展见文献^[10].

Table 1 Overview of Software Low Energy Research

表 1 软件低能耗技术概览

Goal	Stages	Road Map	Key Issues	Sub-Issues
Optimization	Run Time	Treat software as managers of system energy, monitor and adapt the power state of the hardware.	Set the components to the lower power state, while satisfying constrain of the functionality and performance.	Energy-aware OS resource management and task scheduling, application-level adapting, multi-core task scheduling, network protocols.
	Development	Treat software as drivers of energy consumption, and optimize the software design.	Optimize instruction order, program structures, and algorithms, while satisfying constrain of the functionality and performance.	Compiler optimization, software algorithm optimization, and software architectural optimization.
Evaluation	Run Time	Physically measure and analyze system energy consumption.	Map the measurements to software components properly, and ensure the accuracy and the granularity of mapping.	Analysis on energy consumption of embedded systems and operating systems.
		Model the relation between energy consumption and system events, and estimate energy consumption online.	Ensure the accuracy of the models, and monitor the model parameters on real-time.	Evaluation on system energy consumption based on performance counters.
	Development	Estimate energy consumption based on system simulation.	Simulate the full system software, and build the effective and accurate energy consumption models.	Architectural-level energy consumption model, instruction-level model, full system energy simulation.
Estimate energy consumption based on software analysis and macro-model.		Model the relation between high-level software features and energy consumption.	Energy consumption estimation based on algorithm complexity, or features of application data.	

本文主要关注面向开发阶段的软件低能耗技术,这类技术主要研究软件的指令序列、算法以及软件体系结构的特征与系统能耗之间的关系,并围绕这些特征来优化软件,降低系统能耗。

为了研究软件特征对系统能耗的影响以及通过优化软件来降低系统能耗,需要考察在软件执行过程中涉及到的硬件部件的能耗。我们把软件执行过程中涉及到的硬件部件能耗之和称为软件能耗 (software energy consumption), 如式(1)所示:

$$E_{\text{software}} = \sum_{i \in \text{components}} E_i = \sum_{i \in \text{components}} (T_i \times P_i), \quad (1)$$

其中, T_i 是程序在硬件部件上的执行时间, P_i 是该硬件部件的平均功耗。根据该概念,把软件执行过程中涉及到的硬件部件能耗之和映射到软件组成部分(指令、例程、服务、构件等)上,称为软件能耗估算。以此为基础,研究软件特征对系统能耗的影响称为软件能耗评估。近年来,软件能耗估算成为快速评估软件优化效果的关键支撑技术,受到广泛关注^[11]。

本文首先分析面向开发阶段的低能耗软件优化技术,从指令级、算法级、软件体系结构级 3 个层次分析低能耗软件优化面临的主要问题及研究进展;

然后深入分析软件能耗估算技术,指出面向处理器和面向全系统的软件能耗估算面临的问题,讨论实现软件能耗估算的思路;最后展望软件低能耗技术领域有待研究的主要问题和发展趋势。

1 低能耗软件优化技术

面向开发阶段的低能耗软件优化技术在系统生命周期的开发阶段,把降低系统能耗作为关键设计目标之一,渗透到软件的指令生成、算法设计以及软件体系结构设计各个阶段。目前,利用编译器优化指令序列,通过改写源代码优化算法,以及通过高层软件设计优化软件体系结构等方法降低系统能耗的研究受到重视,并取得显著进展。

1.1 指令级优化

指令级优化主要在编译层面展开。随着计算机体系结构级低能耗技术的发展,研究者发现已有的许多性能优化编译技术,如代码优化、代码调度、寄存器选择等对处理器体系结构部件的功耗有显著影响,大大激发了软件研究者对利用软件技术降低系统能耗的兴趣^[12]。

Tiwari 等人首先提出软件能耗的概念^[2],将其定义为“软件执行过程中处理器及访存操作的能耗”,并探讨了低能耗编译优化的方法。

低能耗编译优化关注软件的基本构成元素——指令,在与机器无关的优化阶段或者代码生成阶段进行,优化指令序列,降低处理器和访存能耗,减少指令执行时间和降低硬件功耗都能降低系统能耗。多数情况下,通过性能优化可以减少指令执行时间,如传统的删除冗余操作指令、代码变换等方法都可作为低能耗编译优化方法。当性能优化使处理器功耗增加导致软件能耗增加时需要在性能和能耗之间进行权衡:不仅要满足一定的性能约束,还要使软件能耗最低,这就对现有编译优化技术带来新的挑战。

除性能优化方法以外,低能耗编译方法还包括降低硬件功耗,如降低处理器、存储系统的动态功耗和静态功耗。降低处理器动态功耗的编译优化方法包括采用低功耗指令代替高功耗指令、调节调度算法降低峰值功耗、调节指令执行占用的部件^[13]以及编译制导的动态电压/频率调节^[14-16]等。

降低存储系统动态功耗的基本方法是采用传统的存储访问优化方法,减少对 Cache 和存储器的访问^[17]。此外还包括改进寄存器文件分配算法,减少寄存器译码和指令总线信号翻转,优化数据和指令预取方法^[18]。

随着芯片尺寸变小和晶体管速度变快,在电路不翻转情况下,静态功耗成为电路的主要功耗来源。编译器对数据流和代码的分析非常重要,可以发现程序执行过程中流水线体系结构部件活动数目,利用编译转换代码缩减 Cache 访问的范围,控制细粒度的 Cache 替换,插入指令关闭不用的 Cache 单元来降低处理器和存储系统的静态功耗^[19]。

把编译优化软件能耗和能耗估算结合起来,是一项有吸引力的课题^[20]。把体系结构级参数、能耗估算模型作为输入,让编译器结合软件分析结果进行能耗估算,并根据能耗/性能/代码量的约束条件进行权衡,生成满足约束的代码,将会大大提高低能耗软件优化的效率。

随着多核处理器的发展和应用,能耗约束下的多核并行编译优化成为研究者们关注的新热点^[21]。并行编译器可以在发掘不同粒度任务并行性的同时,识别各部分任务执行时所需的处理器资源数目和频率,指导动态调节处理器能耗和频率,达到优化性能和降低能耗的目的。

1.2 软件算法优化

性能分析的研究表明,在实现同样功能的前提下,不同算法直接影响软件的代码量、内存访问量 and 外设访问量,进而显著影响处理器和系统的能耗。因而,在权衡性能与能耗的基础上,在源代码级优化软件算法和代码设计,也能达到降低软件能耗的目的。

在针对多任务并发嵌入式操作系统中,可以通过选用低能耗的任务间通信方式^[22]或改进虚拟内存管理策略^[23]来降低系统能耗。伴随着虚拟机在嵌入式移动设备上的广泛应用,Java 虚拟机能耗优化近年来受到重视。Java 执行环境中的两个重要组成部分 Just-in-time(JIT)和垃圾收集器对系统能耗有显著影响。通过减少指令数可以显著降低 JIT 能耗,需要特别关注的是垃圾收集器大量访存造成的能耗^[24]。垃圾收集器参数(收集频率、对象分配模式)及存储体系结构(内存槽数目和大小)不当会显著增加 Java 程序的能耗。高效率的虚拟机需要在垃圾收集算法、存储体系结构以及功耗管理策略之间协调^[25-26]。综合优化软件内存管理和存储体系结构设计来降低系统能耗是一个有待深入研究的问题。

应用程序的优化主要针对应用本身的特点。例如改进 MP3 音频解码器^[27],提高压缩率减少执行时间,优化数据压缩方法,改进传输决策等可以降低媒体播放器、无线传感节点和网络的能耗^[28]。

1.3 软件体系结构优化

软件体系结构为构造软件系统的结构、行为和关键属性提供了设计模型和指导,是满足软件质量需求的重要因素^[29]。低能耗也作为一种软件质量属性,成为软件体系结构设计者需要考虑的需求之一。

软件体系结构风格决定软件中各组成部分的相互关系,对分布式系统软件的系统能耗有显著影响^[30]。围绕软件体系结构风格优化软件能耗首要解决的问题是评估不同软件体系结构风格对系统能耗的影响,需要同时考虑硬件平台和应用的相关特性。如通过提炼关键特征参数,导出硬件平台能耗与软件体系结构特征间的映射关系,结合特定应用软件场景实例,得到用于评估和优化软件体系结构的软件能耗模型^[31]。在评估基础上,可以利用提取的软件体系结构图(SAG),通过替换、合并软件构件,合并任务、调整任务间通信模式来降低系统能耗^[32]。此外,通过扩展控制/数据流程图优化进程并发管理过程,如合并进程、传输数据批量化、计算迁移、选择高效通信方式等方法,也是降低系统能耗的有效方法^[33]。

2 软件能耗估算

作为低能耗软件优化的关键支撑技术之一,软件能耗估算不仅要把软件执行涉及到的硬件能耗映射到软件各组成部分上,更期望建立高层软件特征与系统能耗的关系,从而易于在软件层面评估和优化系统能耗。

现有的软件能耗评估方法主要有实测方法和基于能耗模型的方法。实测方法在实际运行的系统上利用专用测量分析仪器获得程序运行时系统或者某些部件的能耗^[34]。这种方法对测量仪器精度要求较高,分析的程序范围、分析精度和准确度有限,但对实际系统的动态功耗管理策略和软件能耗评估有直接的实用价值。

基于能耗模型的评估方法首要解决的问题是软件能耗的估算,即利用能耗模型算出所关注的软件组成部分的能耗。基于模拟器的软件能耗估算在模拟运行软件的基础上加入能耗模型来估算软件能耗。这种方法可以在嵌入式系统开发的早期研究软件能耗优化问题,不影响原有软件的执行过程,可分析软件的范围大、精度高,受到众多研究者的关注,其核心问题是建立高效准确的软件能耗模型。

软件能耗估算要计算软件执行期间涉及到的各硬件单元能耗的总和。一方面,硬件电路(ICs)的功耗模型已经得到了深入的研究^[2],这些模型为估算微处理器等部件的能耗奠定了基础。另一方面,把硬件单元的能耗通过累加的方式映射到构成软件的指令、例程、构件、服务等组件上,构成了软件能耗估算的基本运算。

根据抽象层次不同,面向处理器的软件能耗估算技术可以划分为基于指令级模型、基于体系结构级模型以及更高抽象层次、脱离硬件能耗模型的宏模型。抽象层次越低精度越高,但模拟运行开销越大;抽象层次越高估算速度越快,与软件特征关系越密切,但精度越低。

其次,软件能耗估算的关注范围也逐渐扩大,从只考虑处理器能耗,扩展到内存、外围部件以及全系统;被评估的软件从单个用户程序扩展到操作系统及完整的软件系统(应用程序、中间件和操作系统)。

下面首先分析面向处理器的软件能耗估算;然后分析面向全系统的软件能耗估算面临的问题和研究现状。

2.1 面向处理器的软件能耗估算

面向处理器的软件能耗估算早在 1994 年就得

到关注^[2]。其核心问题是建立程序在微处理器上执行及其访存操作的能耗模型。表 2 对 3 种处理器能耗估算模型的准确度、速度、复杂度、易于实现程度等特点作一定性对比。

Table 2 Comparisons of Three Estimation Models

表 2 3 种能耗模型对比

Key Point	Architectural Level	Instruction Level	Macro Model
Accuracy	Higher	Low	Lower
Abstraction	Lower	Low	High
Speed	Low	High	Higher
Complexity	High	Low	Low
Easy-to-Use	Harder	hard	Easy

2.1.1 基于指令级模型的能耗估算

指令级能耗模型把一个程序的能耗看作是該程序所有指令能耗之和^[2],具体而言是基本指令能耗 B_i 与指令数 N_i 乘积,加指令间能耗 $Q_{i,j}$ 与指令对数 $N_{i,j}$ 乘积,再加其他 k 类型的指令间影响造成的能耗 S_k ,表示为

$$E = \sum_i (B_i \times N_i) + \sum_{i,j} (Q_{i,j} \times N_{i,j}) + \sum_k S_k. \quad (2)$$

为了得到 B_i 和 $Q_{i,j}$,需要对整个指令集体系结构(instructions set architecture)的每一条指令,以及每一对指令序列都进行能耗测算,构造一个大小为 $O(N^2)$ 的指令能耗基础数据表。这个模型得到广泛重视,在近 10 年里得到不断改进,如把指令能耗基础数据表缩小到 $O(N)$,建立指令组间能耗模型,把指令能耗分解成处理器内部功能单元能耗的线性组合,简化了构造指令能耗基础数据表的工作量。

基于指令级能耗模型估算软件能耗的方法易于应用于指令级模拟器中,关键问题是建立相应体系结构指令集对应的指令能耗数据库。

2.1.2 基于处理器体系结构的能耗估算

在体系结构级处理器能耗模型中,处理器的单时钟周期能耗是在该周期中所有活动单元能耗的总和。一个程序的能耗则是在程序执行过程中所有时钟周期内处理器的单时钟周期能耗总和。

基于处理器体系结构的软件能耗估算精确到时钟周期,并细致地模拟指令执行过程中处理器体系结构部件的行为,与指令级能耗估算相比准确度更高,但模拟速度慢,模拟大型复杂程序时较为突出。

Brooks 等人提出的 Watch 是目前应用较为广泛的体系结构级处理器能耗模型^[35],把处理器中的

单元分为阵列结构、内容-地址全关联内存、组合逻辑和线路、时钟 4 类,对不同体系结构部件进行参数化建模来估算处理器能耗. Brooks 把 Wattch 能耗模型集成在一个现代超标量微处理器性能分析模拟器 SimpleScaler 里,形成可配置的性能和能耗分析工具. 经验证,基于 Wattch 的能耗估算工具的绝对误差在 10% 之内,相对误差在 10%~11% 之内. 由于其较高的精度以及在软硬件界面级别抽象的特点, Wattch 被处理器体系结构设计师、编译器设计者、以及系统软件开发者广泛用于处理器性能分析、处理器能耗估算和软件能耗研究中.

在体系结构级处理器能耗模拟基础上,可以获得执行指令过程中处理器在每个时钟周期的能耗,通过程序的指令地址流及符号信息分析,可以将处理器能耗映射到操作系统及应用程序的相应部分上,用于评估软件体系结构优化、编译优化以及软件算法优化的效果^[36].

2.1.3 基于宏模型的能耗估算

研究软件能耗宏模型的目的是建立易于获取的软件高层特征参数与软件能耗间的关系,用来评估软件能耗和软件能耗优化策略,而不再依赖硬件能耗模拟或者物理测量. 面临的主要问题是找到适当的特征参数和宏模型. 根据选择的参数不同,宏模型可以分为两类:基于软件特征的宏模型和基于系统运行参数的宏模型.

基于软件特征的宏模型方法完全脱离硬件功耗参数,直接建立高层软件特征和能耗的量化关系,把程序模块当作“黑盒”,根据感兴趣的软件特征构造相应的宏模型. 这种方法把函数能耗表示为一个与 n 维指标 $[c_1, c_2, \dots, c_n]$ 关联的线性公式:

$$P = \sum_{j=1}^n \omega_j \times c_j, \quad (3)$$

其中 $[\omega_1, \omega_2, \dots, \omega_n]$ 是需要确定的宏模型参数. 现有的宏模型参数包括算法复杂度、软件执行时间、函数性能分析结果、System C 的宏操作以及消息传送量等^[37]. 基于宏模型的软件能耗估算速度快,便于利用高层软件特征来评估和优化系统能耗^[38].

基于系统运行参数的宏模型主要利用处理器中的性能计数器和事件计数器提供的系统状态参数,如 Cache 缺失次数等,结合测量技术在线估算系统能耗^[39]. 这种方法需要利用实际运行系统中的硬件计数器,受限于处理器提供的硬件支持能力. 由于其为真实系统的提供运行时的能耗信息,因此对运行系统的动态能耗优化管理有很大的实用价值.

2.2 面向全系统的软件能耗估算

低能耗技术研究的目标是降低整个计算机系统的能耗. 处理器能耗只占整个系统能耗的一部分,而其他的 I/O 部件和控制器则对整个系统能耗有着显著影响. 面向全系统的软件能耗估算才能更全面地评估软件能耗优化的效果.

一种方法是基于系统模拟器的软件能耗估算,在系统功能模拟器上增加系统部件的能耗模型,估算包括 OS 在内的软件能耗. 实现系统模拟器是实现系统软件能耗估算分析的关键条件之一,需要模拟除处理器之外的系统部件以支持操作系统和应用程序模拟运行. 这类方法易于建立软件的指令、例程等细粒度组成部分与系统能耗之间的映射,得到广泛的重视. 已有研究者实现了基于 PalmOS 设备模拟器^[40]、基于 ARM 体系结构系统模拟器^[22]上的时钟精度的软件能耗估算. 这类方法解决了系统软件的模拟运行和处理器能耗的估算,但外围部件的能耗模型还比较缺乏. 如何对系统外围部件建立适当的能耗模型并与处理器能耗集成是需要解决的问题.

针对处理器的基于系统运行参数的宏模型,可以扩展成为面向系统的软件能耗宏模型. 例如,用 OS 统计性能计数的线性组合表达软件能耗^[41],建立电池和性能计数器的关联,根据测量数据反馈调节能耗估算模型等. 这种方法易于为 RAM 驱动、无线网卡、磁盘等部件建立能耗估算模型,但测量结果精度有限,如何将系统能耗与软件组成部分建立细粒度的映射是需要解决的问题.

3 总结与展望

如果没有软件的合理利用和高效优化,硬件的低功耗潜力和优势将难以得到充分发挥. 低能耗软件优化技术作为低能耗技术的重要组成部分,近年来受到前所未有的关注. 在指令级、算法级、软件体系结构级的优化以及软件能耗估算方面均取得了显著进展,并且还在继续发现和拓展新的研究话题. 一些尚待研究的问题包括以下方面.

1) 如何提高低能耗软件优化的系统化和自动化程度. 需要进一步系统化地分析各软件层次中影响系统能耗的关键因素,考虑软件优化方法对整个系统能耗的影响,利用自动化技术设计实现低能耗软件.

2) 如何提高运行时系统的“能耗感知”和“自适

应管理”系统能耗的能力. 根据电量供应和用户需求进行动态的系统能耗管理, 需要构建高效准确的运行时能耗管理框架, 进行感知能耗的系统管理决策及评估. 目前这方面还有许多难题需要解决.

3) 如何与硬件低能耗技术的快速发展同步, 更好地发挥硬件的特性. 随着硬件技术的发展, 新器件(如新型存储卡、超薄节能液晶显示屏)、新处理器(如多核处理器、低功耗处理器)层出不穷. 为了支持低能耗软件系统的开发, 需要研究基于新硬件平台的软件能耗估算、优化方法和工具.

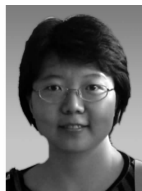
从移动计算设备到服务器领域, 低能耗技术在电路级、处理器体系结构级、系统软件、应用程序、网络等各个层次得到不断扩展和深入, 低能耗软件优化技术研究的软件粒度越来越大, 越来越受到研究者的重视. 随着互联网和云计算技术的发展, 能耗优化研究的范围不再局限于单个程序和分布式系统软件, 互联网能耗也成为研究对象. 如何在软件开发生命周期的各个阶段利用有效的优化方法和手段降低系统能耗的研究也已被提上日程^[42]. 为了能够最终打破计算机系统的能耗瓶颈, 需要在低能耗软件优化技术方面进行更多研究, 并与硬件低功耗技术紧密结合才能实现真正高性能低能耗的绿色计算目标.

参 考 文 献

- [1] Rabaey J M, Chandrakasan A, Nikolic B. Digital Integrated Circuits: A Design Perspective, Second Edition [M]. Beijing: Tsinghua Express, 2003: 213-217
- [2] Tiwari V, Malik S, Wolfe A. Power analysis of embedded software: A first step towards software power minimization [J]. IEEE Trans on Very Large Scale Integration, 1994, 2(4): 437-444
- [3] Fei Y, Zhong L, Jha N K. An energy-aware framework for dynamic software management in mobile computing systems [J]. ACM Trans on Embedded Computing Systems, 2004, 7(3): 1-31
- [4] Benini L, Bogliolo A, Micheli G D. A survey of design techniques for system-level dynamic power management [J]. IEEE Trans on VLSI Systems, 2000, 8(3): 299-316
- [5] Zhao Xia, Guo Yao, Chen Xiangqun. Transaction-based adaptive dynamic voltage scaling for interactive applications [C] //Proc of the 14th ACM/IEEE Int Symp on Low Power Electronics and Design. New York: ACM, 2009: 255-260
- [6] Fei Y, Zhong L, Jha N K. An energy-aware framework for dynamic software management in mobile computing systems [J]. ACM Trans on Embedded Computing System, 2008, 7(3): 1-31
- [7] Zhong Xiao, Qi Yong, Hou Di, et al. Tasks scheduling with dynamic voltage scaling on multi-core real-time systems [J]. Acta Electronica Sinica, 2006, 34(12): 2481-2484 (in Chinese)
(钟斌, 齐勇, 侯迪, 等. 基于DVS的多核实时系统节能调度[J]. 电子学报, 2006, 34(12): 2481-2484)
- [8] Bircher W L, John L K. Analysis of dynamic power management on multi-core processors [C] //Proc of ICS'08. New York: ACM, 2008: 327-338
- [9] Cheng Xiaoliang, Deng Zhidong, Dong Zhiran. A model of energy consumption based on characteristic analysis of wireless communication and computation [J]. Journal of Computer Research and Development, 2009, 46(12): 1985-1993 (in Chinese)
(成小良, 邓志东, 董志然. 基于无线通信和计算特征分析的能耗模型[J]. 计算机研究与发展, 2009, 46(12): 1985-1993)
- [10] Zhao Xia, Chen Xiangqun, Guo Yao, et al. A survey on operating system power management [J]. Journal of Computer Research and Development, 2008, 45(5): 817-824 (in Chinese)
(赵霞, 陈向群, 郭耀, 等. 操作系统电源管理研究进展[J]. 计算机研究与发展, 2008, 45(5): 817-824)
- [11] Bui V, Norris B, Huck K, et al. A component infrastructure for performance and power modeling of parallel scientific applications [C] //Proc of the 2008 Workshop on Component-Based High Performance. New York: ACM, 2008: 1-11
- [12] Hu Dinglei, Chen Shuming. Low power/energy compilation technology [J]. Acta Electronica Sinica, 2005, 33(4): 676-682 (in Chinese)
(胡定磊, 陈书明. 低功耗编译技术综述[J]. 电子学报, 2005, 33(4): 676-682)
- [13] Lorenz M, Marwedel P, Drager T, et al. Compiler based exploration of DSP energy savings by SIMD operations [C] //Proc of the 2004 Asia and South Pacific Design Automation Conf. Piscataway, NJ: IEEE, 2004: 838-841
- [14] Yi Huizhan, Chen Juan, Yang Xuejun, et al. A real-time dynamic voltage scaling algorithm based on syntax tree for low power [J]. Journal of Software, 2005, 16(10): 1726-1734 (in Chinese)
(易会战, 陈娟, 杨学军, 等. 基于语法树的实时动态电压调节低功耗算法[J]. 软件学报, 2005, 16(10): 1726-1734)
- [15] Zhao Rongcai, Tang Zhimin, Zhang Zhaoqing, et al. A study of the compiler-directed low power multithreading technology [J]. Journal of Computer Research and Development, 2002, 39(12): 1572-1578 (in Chinese)
(赵荣彩, 唐志敏, 张兆庆, 等. 编译指导的多线程低功耗技术研究[J]. 计算机研究与发展, 2002, 39(12): 1572-1578)
- [16] Hsu C H, Kremer U. The design, implementation, and evaluation of a compiler algorithm for CPU energy reduction [C] //Proc of the ACM SIGPLAN 2003 Conf on Programming Language Design and Implementation. New York: ACM, 2003: 38-48

- [17] Kandemir M, Vijaykrishnan N, Irwin M J, et al. Influence of compiler optimizations on system power [C] //Proc of the 37th Conf on Design Automation. New York: ACM, 2000: 304-307
- [18] Guo Y, Ben-Naser M, Moritz C A. PARE: A power-aware data prefetching engine [C] //Proc of Int Symp on Low Power Electronics and Design (ISLPED'05). New York: ACM Press, 2005: 339-344
- [19] You Y P, Lee C, Lee J K. Compilers for leakage power reduction [J]. ACM Trans on Design Automation of Electronic Systems, 2006, 11(1): 147-164
- [20] Kadayif I, Kandemir M, Chen G, et al. Compiler-directed high-level energy estimation and optimization [J]. ACM Trans on Embedded Computing Systems, 2005, 4(4): 819-850
- [21] Shirako J, Oshiyama N, Wada Y, et al. Compiler control power saving scheme for multi core processors [C] //Proc of the 18th Int Workshop on Languages and Compilers for Parallel Computing. Hawthorne, NY: SpringerLink, 2006: 362-376
- [22] Zhao Xia, Guo Yao, Wang Hua, et al. Fine-grained energy estimation and optimization of embedded operating systems [C] //Proc of 2008 Int Conf on Embedded Software and Systems Symposia. Los Alamitos, CA: IEEE Computer Society, 2008: 90-95
- [23] Joo Y, Choi Y, Park J, et al. Energy and performance optimization of demand paging with OneNAND flash [J]. IEEE Trans on Computer-Aided Design of Integrated Circuits and Systems, 2008, 27(11): 1969-1982
- [24] Hu S, John L K. Impact of virtual execution environments on processor energy consumption and hardware adaptation [C] //Proc of the 2nd Int Conf on Virtual Execution Environments. New York: ACM, 2006: 100-110
- [25] Velasco J M, Atienza D, Olcoz K. Exploration of memory hierarchy configurations for efficient garbage collection on high-performance embedded systems [C] //Proc of the 19th ACM Great Lakes Symp on VLSI. New York: ACM, 2009: 3-8
- [26] Chen G, Shetty R, Kandemir M, et al. Tuning garbage collection for reducing memory system energy in an embedded Java environment [J]. ACM Trans on Embedded Computing Systems, 2002, 1(1): 27-55
- [27] Simunie T, Benini L, et al. Source code optimization and profiling of energy consumption in embedded systems [C] //Proc of the 13th Int Symp on System Synthesis. Los Alamitos, CA: IEEE Computer Society, 2000: 193-198
- [28] Sharma A B, Golubchik L, Govindan R, et al. Dynamic data compression in multi-hop wireless networks [C] //Proc of the 11th Int Joint Conf on Measurement and Modeling of Computer Systems. New York: ACM, 2009: 145-156
- [29] Wolf A L, Fl C, Perry D, et al. Foundations for the study of software architecture [J]. ACM SIGSOFT Software Engineering Notes, 1992, 17: 40-52
- [30] Seo C, Edwards G, Popescu D, et al. A framework for estimating the energy consumption induced by a distributed system's architectural style [C] //Proc of the 8th Int Workshop on Specification and Verification of Component-Based Systems (SAVCBS'09). New York: ACM, 2009: 27-34
- [31] Seo C, Edwards G, Malek S, et al. A framework for estimating the impact of a distributed software system's architectural style on its energy consumption [C] //Proc of the 7th Working IEEE/IFIP Conf on Software Architecture (WICSA 2008), Los Alamitos, CA: IEEE Computer Society, 2008: 277-280
- [32] Tan T K, Raghunathan A, Jha N K. Software architectural transformations: A new approach to low energy embedded software [C] //Proc of DATA 2003. Los Alamitos, CA: IEEE Computer Society, 2003: 1046-1052
- [33] Fei Y, Ravi S, Raghunathan A, et al. Energy-optimizing source code transformations for operating system-driven embedded software [J]. ACM Trans on Embedded Computing Systems, 2007, 7(1): 1-26
- [34] Acquaviva A, Benini L, Ricco B. Energy characterization of embedded real-time operating systems [C] //Proc of Workshop on Compilers and Operating Systems for Low Power(COLP'01). New York: ACM, 2001: 13-18
- [35] Brooks D, Tiwari V, Martonosi M. Wattach: A framework for architectural-level power analysis and optimizations [C] //Proc of the 27th Annual Int Symp on Computer Architecture. New York: ACM, 2000: 83-94
- [36] Zhao Xia, Guo Yao, Lei Zhiyong, et al. Estimation and analysis of embedded operating system energy consumption [J]. Acta Electronica Sinica, 2008, 36(2): 209-215 (in Chinese)
(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)
- [37] Loghi M, Benini L, Poncino M. Power macromodeling of MPSoC message passing primitives [J]. ACM Trans on Embedded Computing Systems, 2007, 6(4): 31
- [38] Fei Y, Ravi S, Raghunathan A, et al. Energy-optimizing source code transformations for operating system-driven embedded software [J]. ACM Trans on Embedded Computing Systems, 2007, 7(1): 1-26
- [39] Chandra K, Selim G. A run-time, feedback-based energy estimation model for embedded devices [C] //Proc of the 4th Int Conf Hardware/Software Codesign and System Synthesis. New York: ACM, 2006: 28-33
- [40] Cignetti T L, Komarov K, Ellis C S. Energy Estimation Tools for the Palm [C] //Proc of the 3rd ACM Int Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems. New York: ACM, 2000: 96-103
- [41] Rivoire S, Ranganathan P, Kozyrakis C. A comparison of high-level full-system power models [C] //Proc of the 2008 Conf on Power Aware Computing and Systems (HotPower'08). New York: ACM, 2008: 3-8

- [42] Kaefer G, Green S E; Ideas for including energy efficiency into your software projects [EB/OL] //Proc of the 31st Int Conf on Software Engineering. [2009-06-10]. <http://www.cs.uoregon.edu/events/icse2009/specialSessions/#TB2>



Zhao Xia, born in 1972. Postdoctor and associate professor. Senior member of China Computer Federation. Her main research interests include low power design, system software and software engineering.



software engineering, etc.

Guo Yao, born in 1976. Associate professor. Member of China Computer Federation. His main research interests include system software, low power system design, compiler technology and



Chen Xiangqun, born in 1961. Professor. Senior member of China Computer Federation. Her main research interests include system software and software engineering, etc.

2012 年《计算机研究与发展》专题征文通知

——新型存储系统及其关键技术

近年来,随着国家和社会信息化发展的不断加速,信息存储的需求越来越广泛,数据存储量越来越大,目前存储系统的性能、功耗、容量、可靠性、安全性等各种问题严重阻碍了我国的信息化进程。因此,对云存储技术、SSD 技术、新兴存储芯片技术等新型存储系统及其关键技术的研究成了目前存储领域的研究热点。新型存储技术的发展将对现有存储系统结构产生巨大冲击和深远影响。近年来,国内外相关学者已在这些方面开展了大量的相关研究,也取得了一些重要研究成果。为此,《计算机研究与发展》将于 2012 年出版一个“新型存储系统及其关键技术”专题,报道该领域国内外科技工作者所取得的研究成果,提炼新型存储系统与应用中有待解决的关键问题,同时展望存储系统未来的研究方向,以推动我国存储系统与关键技术的科学研究和工程应用。

本期“新型存储系统及其关键技术”专题将面向国内外征集论文,欢迎广大学者、专家、工程技术人员积极投稿,现将专题论文征集的有关事项通知如下。

征文范围(但不限于):

①云存储技术与应用;②SSD (Solid State Disk)技术与应用;③新型的存储体系结构;④新兴存储技术及应用;⑤分布式及并行 I/O 技术;⑥存储安全问题;⑦存储系统可靠性、可用性 & 容灾问题;⑧归档存储系统;⑨存储虚拟化技术;⑩缓存技术及其一致性问题;⑪数据库存储技术;⑫移动存储技术;⑬存储能耗问题;⑭重复数据删除技术;⑮分级存储技术;⑯存储文件系统设计。

投稿要求

- ① 来稿应属于作者的科研成果,数据真实可靠,具有重要的学术价值与推广应用价值,未在国内公开发行的刊物或会议上发表或宣读过。
- ② 论文一律用 word 格式排版,论文格式体例参考近期出版的《计算机研究与发展》的要求(<http://crad.ict.ac.cn/>)。
- ③ 论文通过专辑投稿信箱(storage2012@126.com)发送电子稿,投稿时提供作者的联系方式(Excel 文档,按顺序包含:作者、论文标题、联系人、email、电话(手机)、通信地址、邮编)。

重要时间

截稿日期:2012 年 4 月 10 日

结果通知日期:2012 年 8 月 15 日

特约编辑

舒继武 教授 清华大学计算机科学与技术系 shujw@tsinghua.edu.cn

方 粮 研究员 国防科学技术大学计算机学院 Lfang@nudt.edu.cn

专辑投稿信箱: storage2012@126.com

通信地址:北京市海淀区清华大学计算机科学与技术系 邮编:100084

联系人:舒继武

电 话:010-62783505-5