

# 绿色软件技术研究进展

郭耀 赵霞  
北京大学

关键词：绿色软件

## 绿色软件的新含义

在2009年国际软件工程大会（International Conference on Software Engineering, ICSE'09）上，来自德国西门子公司的杰拉尔德·卡费尔（Gerald Kafer）博士在大会前沿技术主题报告中向与会者介绍了绿色软件工程（Green Software Engineering）的概念<sup>[1]</sup>。这是第一次在ICSE大会上把绿色和低功耗技术摆到了软件工程研究人员面前。卡费尔博士首先强调了绿色IT技术的重要性，然后着重介绍了如何把对于能耗和功耗的考虑加入到软件工程生命周期中，分析了在软件开发不同阶段需要考虑的与能耗相关的技术，并向软件工程领域的研究人员提出了“使软件项目更加绿色”的倡议。

在互联网上，绿色软件更广为人知的含义是指该软件的安装对系统几乎没有任何改变，除了软件的安装目录，不在注册表、系统文件夹等任何地方写入任何信息（或只在开始菜单中添加简单的快捷方式），而在卸载软件时只需要直接删除安装目录即可。

按照维基百科<sup>[2]</sup>的描述，绿色软件和便携软件（Portable Software）都需要满足三个条件：不需要安装、不需要建立或更改任何在该程序所在文件夹以外的文件和不修改注册表及系统设置档。

因此，现在网上经常使用的“绿色软件”其实是便携软件，并不包含环保的意思。虽然这样的“绿色软件”也能从一定程度上达到降低功耗的目标，但是与本文要讲的绿色计算环境下的绿色软件是截然不同的。

绿色计算的主要目标之一是降低计算机系统运行时的能耗，从而不仅能够直接降低计算机系统本身的能耗，同时因为计算机系统散热的减少而降低相应的冷却能耗，达到环保和绿色的目标。与此对应，本文中的绿色软件技术指的是通过软件优化技术来达到降低系统能耗的目的。为了表述上的方便，本文会用软件低能耗技术来指代绿色软件技术。

软件低能耗技术涉及的范围较广。首先，软件能耗优化技术会涉及到不同层次的软件系统，包括应用程序、编译器、中间件和操作系统等，在不同层次的软件系统中可以应用相应的优化技术来降低软件的能量消耗；其次，软件低能耗技术还要与系统硬件中的低功耗技术相结合，通过软件的方式来调节系统硬件中的不同功耗级别，以达到更好地降低系统总体能耗的目标。

本文将对低能耗软件相关技术进行简要综述和分析，包括软件能耗的估算方法、面向能耗的软件优化技术、软硬件协同的低能耗技术等；还将简要分析绿色软件的未来研究方向。

## 软件能耗的概念

软件能耗（Software Energy Consumption）是指对一个具有复杂软硬件的系统而言，完成一定计算工作量的功耗总和<sup>[3]</sup>。它可以定义为在软件执行期间，微处理器或者基于微处理器的计算机系统的能耗。软件在一段执行时间 $[t_1, t_2)$ 内的软件能耗近似表示为

$$E = \int_{t_1}^{t_2} P(t)dt$$

其中,  $P(t)$ 是相关硬件的功耗。一方面, 软件能耗与相关硬件的平均功耗成正比, 通过减少活动硬件单元的数目, 或设置硬件在低功耗状态下工作, 减少硬件的平均功耗可以降低系统能耗; 另一方面, 软件能耗与软件执行时间成正比, 在给定计算任务或程序功能的条件下, 通过优化程序结构和算法, 缩短程序执行时间也可以降低系统能耗。

对软件能耗进行优化的前提是能够对软件所产生的能耗进行快速准确的估算。软件能耗估算的目标是估算运行系统中各部分软件能耗的总和。通过建立与高层软件特征参数相关的软件能耗模型, 可以在软件层面对系统能耗评估和优化。

现有的软件能耗估算方法主要有实测方法和基于功耗模型的方法。实测方法在实际运行的系统上利用专用测量分析仪器来分析系统或者某些部件的能耗。这种方法对测量仪器精度要求较高, 分析的软件模块范围、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具有直接的实用价值。

基于功耗模型估算方法的基本思想是在软件模拟运行的基础上, 加入功耗估算模型来估算软件能耗。这种方法能够在软硬件系统开发早期支持软件能耗评估和分析, 其核心问题是如何建立软件能耗估算模型, 其中包括指令级的软件功耗模型<sup>[4]</sup>和基于体系结构的功耗模型<sup>[5]</sup>等。随着研究的深入, 还需要考虑如何建立基于全系统模拟的计算软件能耗方法。

## 软件能耗优化

前面已经提到, 软件能耗优化技术会涉及应用程序、编译器和操作系统等不同层次(表1)。由于能够全局地考虑系统硬件功耗状态, 又能够在满足应用系统需求、优化软件设计等高抽象层次视图中

考察降低系统能耗的方法, 因此, 与单纯的硬件低功耗技术相比, 软件低能耗技术对系统能耗的影响范围更大, 在开发和应用方面具有更大的灵活性。

表1 不同软件层次的低能耗技术

软件层次	主要技术
应用软件	算法优化、选择低功耗函数库、改进应用程序对操作系统的调用方式等。
编译器	功耗感知的代码变换、低功耗指令选择、减少对缓存(Cache)等存储器的访问、降低静态功耗、编译指导的动态电压调节等。
操作系统	动态电压/频率调节、动态调节硬件功耗状态、功耗感知的程序调度、低功耗驱动程序等。

## 应用程序低能耗

应用程序级低能耗技术是在软件的功能、结构和算法设计阶段, 在对性能与能耗进行权衡的基础上, 通过分析并优化软件的功能、结构和算法, 达到降低系统能耗的目的。

现有研究主要关注两点: (1) 程序的算法优化: 主要通过选择合适的算法降低程序运行时的系统能耗。例如, 利用源代码优化方法和分析工具, 可以对MP3<sup>1</sup>音频解码器进行算法、数据结构和指令流三种优化<sup>[6]</sup>; (2) 在多任务的嵌入式操作系统环境下, 针对操作系统和应用程序的优化: 如在分析软件能耗的基础上, 通过优化操作系统功能模块的实现方法、任务间同步、数据通信、任务切换以及应用程序对操作系统的调用方式等来降低运行软件的能耗。

## 编译器低能耗

狄瓦里(V. Tiwari)等人在软件能耗概念<sup>[4]</sup>的基础上研究了低功耗编译技术, 在代码编译过程中利用编译优化方法降低功耗, 优化可以在与机器无关的优化阶段, 也可以在代码生成的阶段进行<sup>[7]</sup>。

减少指令执行时间和降低硬件功耗都可以降低软件能耗。在多数情况下, 通过性能优化可以减少

<sup>1</sup> Moving Picture Experts Group Audio Layer III, 动态影像专家压缩标准音频层面3。MP3是当今较流行的一种数字音频编码和有损压缩格式, 用来大幅度地降低音频数据量。

指令执行时间,达到能耗优化的目的。如删除源代码中冗余操作、代码变换等传统方法都可作为低功耗编译优化方法。当性能优化导致处理器瞬时功耗增加,或者参与指令运行的部件单元增加,使软件能耗增加时,则需要在性能和能耗之间进行权衡。

从降低处理器动态功耗的角度分析,主要的编译优化方法包括采用低功耗指令代替高功耗指令的方法,通过调节调度算法降低级差功率和峰值功率或调节指令执行占用部件的方法,以及在程序中自动插入电压调节指令方法等。从降低存储系统功耗的角度分析,传统编译器进行的存储访问的优化减少了对缓存和存储器的访问,可以达到降低功耗的目的。此外,改进寄存器文件分配算法,减少寄存器译码和指令总线的信号翻转的方法也可以降低存储访问能耗。

## 操作系统低能耗

操作系统能够在系统运行期间根据系统运行状态,动态调节系统部件功耗状态,在系统范围内更有效地发挥低功耗硬件的特性,降低系统能耗。

从优化控制角度分析,操作系统能耗管理的主要问题是硬件特性基础上研究能耗管理系统模型以及能耗管理策略。从操作系统设计角度分析,操作系统管理能耗资源面临的主要问题是如何扩充现有操作系统资源管理框架来支持能耗管理,以及如何把能耗当作一种资源进行操作系统资源抽象,建立高效资源管理机制<sup>[10]</sup>。

根据硬件特性可以将操作系统中使用的主要低能耗策略划分为两类:动态功耗管理(Dynamic Power Management)和动态电压调节(Dynamic Voltage Scaling)策略。美国斯坦福大学的贝尼尼(L. Benini)等人<sup>[8]</sup>提出了动态功耗管理的概念:根据工作负载的变化,选择性地将系统部件设置到低功耗状态或者关闭空闲系统部件,以最小的活动部件数目或最小的部件功耗来提供系统所需的服务和性能级别。这一策略适用于对内存、硬盘和网卡等外围设备的功耗控制。而动态电压调节则主要是针对处理器经常会处于闲置状态的情况,尽量以较低

处理器频率来满足任务的性能要求,从而可以降低完成相同任务所需的能耗。

## 软硬件协同的能耗优化

除了上述软件低能耗技术之外,还可以把软件与硬件中的低功耗技术结合起来,通过软硬件协同的方式来调节系统硬件中的不同功耗级别,以达到更好地降低系统总体能耗的目标。

例如,利用软硬件协同的方法,在对缓存分区的基础上,编译器通过分析程序代码给硬件提供线索,利用增强的加载/存储(Load/Store)指令来控制细粒度的缓存替换,减少冗余的标签和数据存取,从而降低能耗。在这项技术中,编译器对数据流和代码的分析非常重要。通过分析可以发现程序执行过程中流水线体系结构中部件活动数目,以便编译器决定是否插入关闭部件单元的指令。

另外,数据和指令预取技术对系统功耗及能耗有显著的影响。数据预取对能耗带来的影响主要是由于预取算法中会出现较多的冗余,而且基于硬件的预取方式需要使用能耗较高的预取硬件(主要是保存历史和预测数据的表格)。因此,可以利用编译器产生的信息来减少冗余的预取操作,辅助降低预取硬件的功耗。实验表明,上述方式可以有效降低30%以上的数据预取能耗<sup>[9]</sup>。

## 总结和展望

要提高计算机系统的性能指标,除了需要研制速度更快的处理器、内存和硬盘等硬件设备和系统之外,还需要从软件的角度对系统性能进行优化,包括在算法、编译器和操作系统等方面的优化。虽然传统上功耗和能耗控制都是来源于硬件系统的概念,但是随着绿色计算技术的深入发展,在软件开发过程中软件功耗优化将会同性能优化一样成为必不可少的环节。

与面向性能的软件优化技术类似,面向功耗/能耗的软件优化技术也包括代码(算法)优化、编译优化、操作系统策略优化以及软硬件协同的能耗

优化技术等。为了能够充分发挥绿色软件技术的作用，需要在软件开发的整个生命周期过程中考虑可能对能耗产生影响的因素，在软件开发中把能耗当作与性能同等重要的指标，使之贯穿于需求分析、系统设计、编码实现、测试和维护等环节，真正达到绿色计算的目标。表2中给出了前述的卡费尔博士报告中所给出的在软件开发不同阶段需要考虑的绿色软件技术。

表2 绿色软件工程<sup>[1]</sup>

软件开发阶段	绿色软件技术
需求工程	检查动态电源管理的需求和平台需求；同时说明在测试中与能耗相关的需求。
系统设计	选择带有电源管理能力的平台；在系统构件中集成电源管理应用程序和测试接口；选择支持能耗分析（profiling）的开发环境；采用事件驱动的体系结构设计。
实现	实现电源管理接口和监控接口；使用函数分析（profiling）来查找函数库和源代码中能量消耗的热点；为每个函数定义“能耗预算”。
测试	在单元测试中对函数测试其“能耗预算”；使用测试工具来测试系统暂停/重启和应用程序的自动调节功能。
反馈和改进	集成功耗/能耗的管理、监控和报告功能；支持随着环境变化对系统能耗加以改进。

除此之外，绿色软件技术还面临更多的挑战，其中包括：

**如何更好地对软件能耗进行估算** 全系统软件能耗评估模型和工具的准确度和精细度有待提高。现有的全系统软件能耗评估模型和工具不能满足低能耗软件开发的需要，其中外围部件功耗估算模型与处理器能耗估算模型的合成要处理好时钟周期以及与程序指令对应的关系。

**如何更好地利用硬件系统的特性来优化软件能耗** 随着硬件技术的发展，新器件和新体系结构层出不穷。为了支持低能耗软件系统的开发，需要不断发展对应于新硬件平台的软件能耗估算模型和工具。

**研究面向多核系统的软件能耗优化技术** 在多核处理器已经成为主流的今天，如何从应用软

件、编译器、操作系统中更加有效地调节多核系统的能耗，依然是目前绿色计算研究中一个亟待解决的问题。

随着人们对绿色计算的关注，无论是移动计算设备还是大型数据中心都非常关注对功耗和能耗的控制。除了硬件的发展之外，绿色软件技术在其中无疑会发挥越来越重要的作用。我们相信，随着多方面新技术的不断涌现，在软硬件低功耗技术的共同促进下，IT产业一定会在不久的将来迈上“绿色”之路。■

### 致谢

本文研究工作得到了国家高技术研究发展计划（863项目）No. 2008AA01Z133的资助。



#### 郭 耀

CCF会员。北京大学信息科学技术学院、高可信软件技术教育部重点实验室副教授。主要研究方向为操作系统、低功耗设计和软件工程等。  
yaoguo@sei.pku.edu.cn



#### 赵 霞

CCF高级会员。北京大学信息科学技术学院、高可信软件技术教育部重点实验室博士后。主要研究方向为嵌入式系统软件、软件低能耗技术和软件工程。  
zhaoxia@sei.pku.edu.cn

### 参考文献

- [1] Gerald Kaefer, Green SE: Ideas for Including Energy Efficiency into your Software Projects, Technical Brief, 31st International Conference on Software Engineering, Vancouver, Canada, 2009, May 16-24
- [2] <http://zh.wikipedia.org/wiki/绿色软件>
- [3] T. K. Tan, A. Raghunathan, N. K. Jha. EMSIM: An energy simulation framework for an embedded operating system. In Proc. ISCAS 2002, 2002, pp. 464-467
- [4] V. Tiwari, S. Malik, A. Wolfe. Power analysis of embedded software: A first step towards software power minimization. IEEE Transactions on Very Large Scale Integration, vol. 2, no. 4, 1994, pp. 437-444

由于版面原因，查阅参考文献[5~10]请访问：[www.ccf.org.cn](http://www.ccf.org.cn)

- [5] D. Brooks, V. Tiwari, M. Martonosi. Wattch: A framework for architectural-level power analysis and optimizations. In 27th Annual International Symposium on Computer Architecture, 2000
- [6] Tajana, Imuni, L. Benini, et al. Source code optimization and profiling of energy consumption in embedded systems. In Proceedings of the 13th international symposium on System synthesis , 2000, pp. 193~198
- [7] 胡定磊, 陈书明. 低功耗编译技术综述. 电子学报, vol. 33, no. 4, 2005, pp. 672~676
- [8] L Benini, A Bogliolo, G D Micheli. A Survey of Design Techniques for System-Level Dynamic Power Management[C].In: IEEE Transactions On Very Large Scale Integration (VLSI) Systems. Washington, DC, USA: IEEE Computer Society, 2000. 299~315
- [9] Yao Guo, "Compiler-Driven Data Prefetching for Energy Efficiency", PhD Dissertation, University of Massachusetts, Amherst, 2007
- [10] 赵霞, 陈向群, 郭耀, 杨芙清. 操作系统电源管理研究进展. 计算机研究与发展. 2008,45(5): 817-824