

实时系统温度功耗管理的优化方法研究

李甜甜¹ 于 戈¹ 宋 杰²

¹(东北大学计算机科学与工程学院 沈阳 110819)

²(东北大学软件学院 沈阳 110819)

(litiantian_neu@163.com)

Optimization Research on Thermal and Power Management for Real-Time Systems

Li Tiantian¹, Yu Ge¹, and Song Jie²

¹(College of Computer Science and Engineering, Northeastern University, Shenyang 110819)

²(Software College, Northeastern University, Shenyang 110819)

Abstract Power consumption issue of real-time system has been paid much attention to by both academia and industry due to its constraints on energy, peak temperature and deadline of real-time task. Up to now, there have been many related researches. Temperature-unaware traditional researches usually adopt DVS to scale processor states for optimal power management. However, with the increasing power density of processors due to the continuous shrinking of chip size, the mutual effect between temperature and power has become unignorable. As a consequence, many new temperature-aware optimization approaches have derived based on the traditional methods. This paper firstly makes an overview of the three models (task, thermal and power) this research bases on; secondly, this paper divides existing researches into two categories: temperature-unaware traditional researches and temperature-aware optimization researches, and the latter one is further divided as single task optimization and multi-task scheduling; thirdly, this paper makes a comparison of the researches from mechanism, optimization goal and effect, and scheduling time etc., analyzing their advantages and disadvantages; finally, this paper points out the future research directions.

Key words real-time system; temperature aware; energy constrained; power consumption optimization; task scheduling

摘 要 实时系统的能量受限特性、峰值温度约束以及实时任务的时间约束使其能耗问题备受学术界和工业界的关注,目前已有很多相关功耗管理研究.不考虑温度因素的传统功耗管理大多仅通过动态电压调节技术(dynamic voltage scaling, DVS)方法调度处理器的状态实现,然而随着芯片尺寸的不断缩减,处理器的功耗密度越来越大,温度与功耗之间的相互影响已不容忽视,由此在传统管理研究的基础上又衍生出了很多温度感知的新方法.1)对实时系统温度功耗管理依托的3个模型(任务模型、热模型和功耗模型)进行总结整理;2)根据是否考虑温度因素将现有研究分为温度无关的和温度感知的2类进行综述,后者又按面向单任务/面向多任务进行分类;3)从具体机制、优化目标、优化效果以及调度时间等方面进行比较,分析现有研究的优缺点;4)指出未来研究方向.

收稿日期:2016-03-09;修回日期:2016-05-13

基金项目:国家自然科学基金项目(61433008,61502090)

This work was supported by the National Natural Science Foundation of China (61433008,61502090).

通信作者:宋杰(songjie@mail.neu.edu.cn)

关键词 实时系统;温度感知;能量受限;功耗优化;任务调度

中图法分类号 TP393

实时系统在嵌入式系统中很常见,它不仅要求系统计算结果的正确性,还要求结果的产生时间满足约束条件,现已被广泛用于各行各业.然而关于实时系统的性能、功耗以及温度的管理优化却随着芯片工艺的发展不断发生变化,面临前所未有的新挑战.

人类对高性能计算系统的追求驱动半导体技术逐渐从微米到亚微米再到深亚微米的发展,晶体管尺寸的不断缩减以及日益复杂的电路结构导致芯片的功耗密度显著增加.摩尔定律(Moore's Law)^[1]在集成电路(integrated circuit, IC)领域中广为所知,它指出在计算硬件的发展进程中,IC中晶体管数量将近每一年翻一番,后又在1975年基于处理器与存储器密度不同以及IC的高混合度这一事实将预测结果修正为每2年翻一番.此外,根据另一个定律Dennard Scaling^[2],随着晶体管尺寸的不断缩减,芯片的电压和电流将随之降低.然而,该定律已在2005年左右失效,这是因为芯片的电压和电流不可能无限降低,过低的电压和电流无法保证处理器的可靠性;该定律的失效直接导致的后果就是,能耗降低将代替性能提高成为电路设计最重要的挑战,如果忽略能耗问题,摩尔定律也将无法实现^①. Dennard Scaling定律失效后,摩尔定律虽然又持续了几年,但最终也因为无法持续降低保持处理器可靠运转的电压和电流而使晶体管无法保持以以往的高速度运转,还因为随晶体管尺寸变小而显著升高的静态能耗而在性能上并未获得提升^②. Dennard Scaling的失效还使得人们转向追求多核处理器,然而其受益却比Dennard Scaling持续有效情况下的预测受益低.综上所述,若要继续追求高性能系统,首先要解决系统的功耗问题,这就需要在系统运行时通过某些方法如合理的任务调度或芯片状态调节来对功耗进行管理,实时系统即使如此.

另外,晶体管尺寸缩减引起的量子反应以及为处理更小尺寸晶体管对芯片化学成分进行的改变最终导致静态功耗增加,从而使得芯片温度升高,升高的温度反过来再次使静态功耗增加,形成一个恶性循环^③.文献[3-5]也指出,晶体管尺寸的缩减导致功

耗密度不断升高,进而导致处理器温度快速升高,最终使得受温度影响的漏电功耗(静态功耗的主要成分)变得不可忽略,并逐渐成为处理器的主导功耗.温度除了对功耗造成影响外,过高的温度还会降低处理器的性能和可靠性,增大故障率.综上所述,温度和功耗密不可分,在对功耗进行管理时还要考虑温度因素.

基于上述分析,本文以实时系统为研究对象,探寻在满足其时间约束的前提下优化系统温度功耗管理的方法.1)实时系统的温度功耗管理方法均依托于3个基础模型:①用于刻画实时任务的任务模型;②用于描述温度与功耗之间相互影响关系的热模型;③用于度量实时系统功耗的功耗模型.本文首先总结这3个模型及其在各研究中的简化.2)实时系统的温度功耗管理研究随着芯片工艺的发展分为两大类:不考虑温度对功耗影响的优化研究和考虑二者相互影响的温度感知的优化研究.其中,后者又可分为2层:单任务优化和多任务调度.本文按照上述分析对实时系统的温度功耗管理方法进行分类描述,并分析其中存在的不足.3)本文对实时系统的温度功耗管理研究进行总结,并指出进一步研究方向.

1 实时系统温度功耗管理

1.1 实时系统简介

实时系统除了要求系统功能正确执行外,还对任务有一个额外的完成时间约束.时间约束是实时系统的一个重要特性,即使是微小的延时也可能造成严重的后果(如刹车的防抱死系统).时间约束在实时系统中通常被称作截止期限,且根据是否强制满足时间约束,系统又分为硬实时系统和软实时系统2种;前者要求必须满足时间约束;后者不作强制要求,但是延迟完成会降低运行结果的意义.

实时系统上的应用通常是一组简单的、高度重复的有限任务集合,也即任务是重复出现的.每个任务可以有很多实例,实例是最基本的工作单元,又称为作业.实时任务根据其作业发布频率特性可以分

① <http://www.radio-electronics.com/articles/electronics-components/moores-law-at-50-145>

② http://en.wikipedia.org/wiki/Dennard_scaling#cite_note-cartesian-2

③ <https://cartesianproduct.wordpress.com/2013/04/15/the-end-of-dennard-scaling>

为 3 类: 周期任务 (periodic tasks)、非周期任务 (aperiodic tasks) 和偶发任务 (sporadic tasks)^[6].

- 1) 周期任务. 任务每隔固定时间发布一次作业, 该时间间隔称为任务的周期.
- 2) 偶发任务. 任务随机发布作业, 但要求 2 个相邻作业之间的时间间隔大于等于一个预先给定的值, 该值称为最小间隔到达时间.
- 3) 非周期任务. 任务随机发布作业, 无约束.

周期任务和偶发任务附加的截止期限又可根据其与周期或最小间隔到达时间之间的大小关系分为隐式截止期限(等于)、约束截止期限(小于)和任意截止期限(无关). 此外, 一个任务生成的所有实例之间是语义相关的, 但其执行时间却是不同的, 主要依赖于系统的软硬件特性、输入数据以及与该实例交互的环境行为^[7]. 为确保满足任务的时间约束, 通常会指定任务的最坏执行时间 (worst-case execution time, WCET). WCET 的精确估计与系统的正确执行密切相关, 目前已有很多相关方法和技术^[8-9].

实时任务之间可能是独立的, 也可能存在某种依赖关系(一个任务的输入是另一个或多个任务的输出). 现有研究大多针对的是硬实时系统上的独立周期任务, 且假设截止期限与任务的周期相等.

1.2 温度功耗管理

实时系统的优化管理通常包括性能、功耗以及温度 3 个方面, 然而随着芯片工艺不断发展, 优化的侧重点也不断发生变化. 1) 不断缩减的晶体管尺寸使得芯片的功耗密度呈指数级增长^[10], 高性能芯片的功耗密度问题已经导致芯片的频率无法随着工艺水平的提高而提高, 也即实时系统的优化目标已经逐渐从性能提升转向能耗降低. 2) 功耗密度不断

上升的同时, 受功耗影响的芯片温度也随之剧增, 从而影响实时系统的性能和可靠性、降低系统的寿命^[11]. 与此同时, 受温度影响的芯片漏电流也不断增加, 从而使得先前可以忽略的漏电功耗的比重逐渐增加. 文献[12]中指出, 65 nm 工艺下, 芯片温度从 65℃ 上升到 110℃ 时, 漏电能耗和总能耗各自增加 38% 和 24%. 综上, 不断攀升的功耗密度, 使得芯片温度随之升高, 进而导致漏电功耗不断增加, 反过来再次提升芯片温度, 最终形成一个恶性循环. 要解决该恶性循环, 实时系统的优化管理需要综合考虑温度和功耗二者之间的相互影响, 对二者进行协同管理.

实时系统温度功耗管理的优化研究首先需要为实时系统建模, 包括用于刻画任务特征的任务模型、描述温度功耗关系的热模型和刻画系统功耗的功耗模型. 这 3 个模型在具体的研究中会有所变化、有所简化, 但本质上是相同的. 有了系统模型, 便可以在此基础上展开实时系统的温度功耗管理研究. 现有相关研究根据芯片制造工艺水平分为 2 个明显的阶段: 1) 工艺水平低时, 功耗密度较低, 不受温度因素影响的动态功耗占主导地位, 系统温度也较低, 这一时期的大部分优化研究不考虑温度因素或将其对功耗的影响看作常量, 优化目标通常为性能或能耗; 2) 工艺水平高时, 功耗密度较高, 芯片温度升高, 导致漏电功耗的比重不断增加, 最终超过动态功耗成为主导功耗, 这一时期的优化研究多是温度感知的功耗优化, 优化目标通常为性能(吞吐量)、能耗或能效. 阶段 1 中, 人们主要研究降低系统动态功耗的方法, 主要通过动态电压调节技术 (dynamic voltage scaling, DVS)^[13-15] 来实现. 考虑到动态功耗与处理

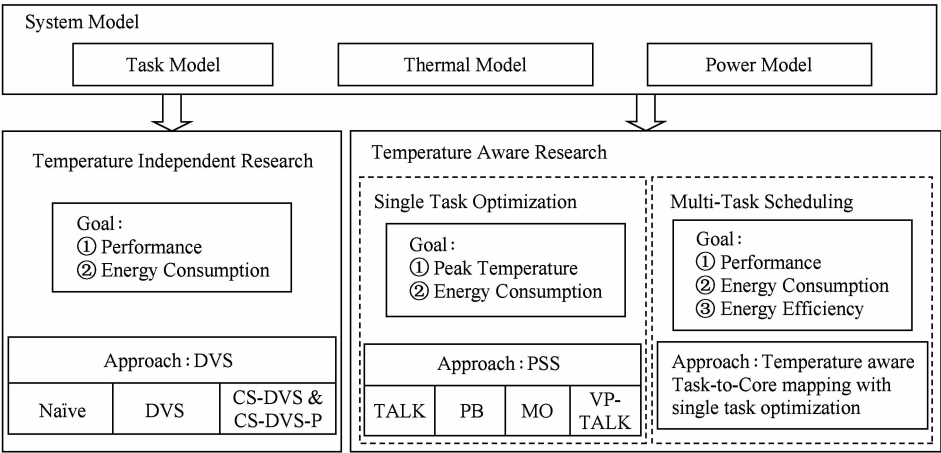


Fig. 1 Optimization research on thermal and power management for real-time systems.

图 1 实时系统温度功耗管理优化研究

器电压的 3 次方近似成正比,DVS 方法主要通过控制处理器的电压来提高性能或降低能耗. 阶段 2 中,人们不得不将温度与功耗之间相互影响考虑在内研究实时系统的功耗优化方法^[16-20],相关研究可分为 2 层:单任务优化和多任务调度. 对于前者,研究主要通过处理器状态调节(processor state scaling, PSS)方法来控制处理器的峰值温度,降低系统能耗. 任务的执行会产生动态功耗并导致系统温度上升,为了使系统的峰值温度不超过既定阈值,需要在适当时候调整处理器的状态使其有时间冷却,PSS 方法即是致力于寻找合适的处理器运行模式,使其既满足任务的截止期限约束,又满足系统的峰值温度约束. 对于后者,相关研究根据优化目标分为 3 类:性能优化、能耗优化和能效优化. 该类研究多是针对多核系统,在单任务优化之上添加 Task-to-Core 映射机制,用于确定最佳的实时任务到核的分配计划,以降低分配策略对系统温度的影响. 图 1 给出了实时系统温度功耗管理优化研究的整体框架图.

2 温度功耗模型

随着晶体管尺寸的不断缩减,使处理器的功耗密度越来越大,从而导致芯片温度不断升高,温度对功耗的影响变得不容忽视,而功耗反过来进一步影响温度,因此实时系统中对温度和功耗的共同管理势在必行. 温度功耗管理通常依托 3 个模型:任务模型、热模型和功耗模型. 下面对这 3 个模型分别进行描述.

2.1 任务模型

现有相关研究针对的实时任务通常具有周期性,且都有截止期限约束. 实时系统相关的能耗优化研究中,通常用一个二元组 $\langle P, C \rangle$ 来表示任务;其中, P 为任务的周期, C 为任务的最坏执行时间. 至于任务的截止期限 D ,很多研究^[5,10,13,16-19]都假设其与任务的周期相等,即 $D=P$. 另外,很多研究^[5,10,16-19]还认为实时任务的周期相同,因此只考虑实时系统下的单个周期任务的能耗优化问题.

2.2 热模型

实时系统通常会有一个峰值温度约束,这是因为处理器的温度超过某个阈值后会对设备造成损坏,影响处理器的性能以及正常工作,因此我们需要对处理器的温度进行控制. 考虑到温度受功耗的影响,下面给出描述二者关系的热模型.

鉴于热传导与电流现象之间的二象性(将处理

器内外看成是具有一定温差以及热阻的热导体体系,其中热量与电流对应、温差与电压对应、热阻与电阻对应、热容与电容对应),我们通常采用式(1)中所示的微分方程^[21]来描述处理器的温度变化行为(又称 RC 模型),并由此推导出温度的原函数. RC 模型不仅适用于描述实时系统处理器的温度变化,还适用于描述数据中心中节点的温度变化等,文献[22]对此作了类似描述,并给出了 RC 模型的示意图,如图 2 所示.

$$\frac{dT(t)}{dt} = \frac{P(t)}{C_{th}} - \frac{T(t) - T_{amb}}{R_{th}C_{th}}, \tag{1}$$

其中, C_{th} 为等效热容(J/K), R_{th} 为等效热抗(K/Watt), $T(t)$ 和 T_{amb} 分别为处理器温度和环境温度(K), $P(t)$ 为处理器在时刻 t 的功耗,单位为瓦特(Watt, W). 上述热模型中, C_{th} 和 R_{th} 的值与具体的工艺技术有关,现有研究中大多采用 65 nm 技术对应的参数值,分别为 340 J/K 和 0.8 K/W^[23]; T_{amb} 则通常选择常温值 25℃. 该模型现已被很多研究工作^[5,16-18,24-28]采用.

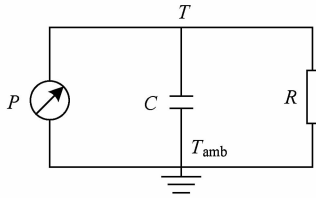


Fig. 2 RC model^[22].
图 2 RC 模型图^[22]

上述热模型是针对单核系统的,然而随着人们对性能的不断追求,单核已无法满足需求,由此出现了多核系统. 多核系统的热模型除了要考虑自身的温度变化规律外,还要考虑不同核之间温度的相互影响因素. 文献[29]给出多核系统的热模型:

$$\frac{dT_i(t)}{dt} = \frac{P_i(t)}{C_i} - \frac{T_i(t) - T_{amb}}{R_{ii}C_i} - \sum_{j \neq i} \frac{T_i(t) - T_j(t)}{R_{ij}C_i}, \tag{2}$$

其中, C_i 和 R_{ij} 分别表示处理器 i 的热容以及其与处理器 j 之间的热抗, R_{ij} 受实时系统中核的布局策略影响. 上述热模型与文献[30-31]给出的类似.

2.3 功耗模型

实时系统的功耗来源有很多,包括动态功耗、短路功耗、漏电功耗和固有功耗等. 其中,动态功耗是任务执行时产生的;漏电功耗是由漏电流产生的,它是静态功耗中的主导部分,部分研究直接将其称作静态功耗;固有功耗是处理器处于开启状态的固有代价. 特别地,短路功耗通常仅在信号转换时产生,

与动态功耗相比完全可以忽略^[7,32],因此相关研究采用的功耗模型通常只考虑其余三者.也即实时系统的总功耗 P_{total} 可以表示为动态功耗 P_{dyn} 、漏电功耗 P_{lk} 以及固有功耗 P_{on} 三者之和: $P_{\text{total}} = P_{\text{dyn}} + P_{\text{lk}} + P_{\text{on}}$.

2.3.1 动态功耗

动态功耗源于器件上电情况下的电路反转,受供电电压和时钟频率影响,而时钟频率又可以近似为供电电压的线性函数,因此动态功耗最终可表示为供电电压的 3 次方函数:

$$\begin{aligned} P_{\text{dyn}} &= C_1 V_{\text{dd}}^2 f, \\ f &= \frac{(V_{\text{dd}} - V_{\text{th}})^\mu}{V_{\text{dd}} T_{\text{max}}^\eta} \times 4.2824 \times 10^{14} \approx C_2 V_{\text{dd}}, \quad (3) \\ P_{\text{dyn}} &= C_3 V_{\text{dd}}^3, \end{aligned}$$

其中, C_1 为有效切换电容, V_{dd} 为供电电压, f 为时钟频率^[33-34], V_{th} 为阈值电压, T_{max} 为芯片可承受的最高温度(过高的温度会减慢处理器的计算速度), μ 和 η 为技术相关的经验参数. 65 nm 技术下 V_{th} , μ 和 η 的值分别为 0.3 V, 1.2 和 1.19^[12].

上述模型虽已被广泛应用,然而事实上动态功耗除了与处理器的频率 f 有关外,还与具体的任务有关,任务主要是通过影响处理器的使用率来影响能耗的.文献[35]考虑了这种特性,通过引进一个称作 *activity factor* 的参数 μ_i (取值区间为 (0, 1]) 提出一个更加合理的动态功耗模型(见式(4)),并指出不同任务中该参数的测量方法^[36].此外,文献[37]在已有研究的基础上给出了一个精确的动态功耗模型(参见式(5)),考虑了处理器的频率 f 和使用率 ω ,并通过实验证明了所提模型的正确性和准确性.动态功耗是系统功耗的一个重要组成部分,因此更准确的系统能耗优化必须要有一个更加准确合理的动态功耗模型.

$$P_{\text{dyn}} = \mu_i C_0 V_{\text{dd}}^3. \quad (4)$$

$$P_{\text{dyn}} = A f^3 + B f^3 \omega + C \omega + D. \quad (5)$$

2.3.2 漏电功耗

漏电功耗是由漏电流产生的,只要能量流过电路就会产生漏电流,因此漏电功耗是必然存在的.供电电压和温度是影响漏电流的 2 个主要因素,文献[12]利用 SPICE 软件模拟不同电路类型在多个温度值下的电流特性,并通过功耗曲线拟合得出漏电功耗和漏电流的表示:

$$\begin{aligned} P_{\text{lk}} &= N_g V_{\text{dd}} I_{\text{lk}}, \\ I_{\text{lk}}(V_{\text{dd}}, T) &= I(V_0, T_0) (A T^2 e^{\frac{\alpha V_{\text{dd}} + \beta}{T}} + B e^{\gamma V_{\text{dd}} + \delta}). \end{aligned} \quad (6)$$

其中, N_g 为系统中晶体管的数量; T 为处理器的温度, $I(V_0, T_0)$ 为特定的参照温度和电压下的漏电流, $A, B, \alpha, \beta, \gamma$ 和 δ 为调节方法、电路类型、技术以及设计相关的经验参数,参数值详见文献[12, 38].如式(6)所示,漏电流来源主要有 2 个, subthreshold leakage 和 gate leakage 分别对应于漏电流公式中的第 1 项和第 2 项;前者随温度升高而增高,后者与温度无关.

$$\begin{aligned} \hat{I}_{\text{lk}}(T) &= \hat{A} T^2 + \hat{B}, \\ \hat{A} &= \frac{I_{\text{lk}}(T_H, V_{\text{dd}}) - I_{\text{lk}}(T_L, V_{\text{dd}})}{T_H^2 - T_L^2}, \quad (7) \\ \hat{B} &= I_{\text{lk}}(T_L, V_{\text{dd}}) - \hat{A} T_L^2, \\ I_{\text{lk}}(i, k) &= C_1(k) + C_2(k) T. \end{aligned} \quad (8)$$

式(6)给出的漏电流模型虽然精确,但其较大的计算复杂度严重影响了在实际调度中的应用.对此,文献[16, 24]在计算复杂度和精确度之间进行权衡以简化漏电流模型,降低其计算复杂度.文献[16]认为处理器在其常规运行温度范围(303 K ~ 373 K)内,式(6)中 $e^{1/T}$ 的变化较小,因此将漏电流函数简化为温度的 2 次函数(见式(7)), T_H 和 T_L 分别为给定电压 V_{dd} 下的最高和最低参照温度;文献[24]根据泰勒公式将式(6)分为温度的线性函数部分和高阶函数部分,并将后者作为误差处理,随后设计大量实验通过曲线拟合获取线性部分中的参数值(与 v_k 有关),并验证线性模型的精确性.此外,文献[39]还对各种可能的线性简化模型进行推测分析,通过实验给出各种模型的精确度对比.上述所有模型中,式(8)给出的线性简化模型因其简洁性和准确性已被很多研究^[4, 17, 35, 40]采用.

2.3.3 固有功耗

固有功耗是处理器处于开启状态的固有代价,与漏电功耗一样,即使处理器处于空闲状态也会产生这部分功耗.与动态功耗和漏电功耗相比,固有功耗的值通常较低,但是当任务的执行时间较长时,固有功耗便不可忽略.固有功耗有 2 个主要来源: PLL 电路和 I/O 子系统; PLL 电路的电流高达 200 mA, I/O 子系统的供电电压在 2.5 V 和 3.3 V 之间,峰值电流高达 400 mA.另外,不同技术以及体系结构下的固有能耗值不同,文献[13]将固有功耗的值设为常量 0.1 W.

3 优化方法

实时系统的温度功耗优化管理是指在满足一定

约束条件(温度、时间)下,寻找最优解(单任务执行优化、多任务调度方案)的过程.不同应用场景下,优化目标不同,对应的优化问题定义也不同.根据温度约束条件的不同,实时系统优化研究可分为2部分:温度无关的优化研究和温度感知的优化研究.

3.1 温度无关的优化研究

温度无关的优化研究是指芯片技术还未发展到需要考虑温度因素对能耗影响时的实时系统优化研究.这类研究通常不考虑温度因素^[41],或是将其当作常量考虑^[42],优化目标也通常只是降低能耗.能耗优化虽然也分单任务和多任务,但是在这类研究中,二者的区分并不大:多任务通常也是按照 EDF (earliest deadline first), Priority-Based 等方法确定任务的执行顺序后,再来优化单个任务的能耗.考虑到这一时期,处理器的主导能耗是动态能耗,且影响动态能耗的主要因素是电压,大多数优化研究^[13-15]采用 DVS 技术来降低系统能耗;文献[43]针对实时系统给出了能量感知的动态功耗管理(dynamic power management, DPM)方法,该方法也可与 DVS 方法联合使用.

关于实时任务的时间约束,基于 DVS 的优化方法本质上均是通过控制处理器的运行速度(与电压呈正相关关系)来确保任务在最坏执行时间下也能够在此期限前完成.然而,该类优化方法的时间约束保障机制通常仅针对任务到达模式已知的实时系统,无法确保到达模式未知的任务约束.为了解决上述问题,文献[44]研究负载的不确定性,并进行特征描述;文献[45]则提出一种历史任务感知的在线 DVFS 方法为可能出现的负载高峰预留计算资源.

1) Naïve 方法

Naïve 方法将处理器的电压设置为可能的最大值,也即以最快的速度执行任务,任务完成后将处理器置于休眠状态以节省能耗.采用 Naïve 方法执行的实时任务通常都能够在此期限前完成,否则该实时任务在该处理器的能力之内是不可完成的,任何其他优化方法也都不可能完成,因为 Naïve 方法采取的是最快的速度执行任务.然而,也正因为执行速度最快,性能最高的 Naïve 方法的能耗也是最大的(因为动态功耗与电压的3次方成正比).Naïve 方法是后续能耗优化研究的基准方法,优化效果验证均以此为基准.

2) DVS 方法

DVS 方法将处理器电压设置为可能的最小值 $V_{\min}$ (由任务的最坏执行时间 C 和截止期限 D 确

定),也即正好在截止期限前完成任务,使得处理器全程均在工作.然而,电压的降低会导致任务执行时间的延长,从而增加系统的漏电能耗和固有能耗.事实上,电压降低导致的漏电能耗增加处于一个恶性循环中:电压降低,导致 V_{th} 降低(为了保证低压逻辑电路正常运转),进而 I_{lkg} 升高(见式(9)),同时任务的执行时间变长,导致漏电功耗增加;漏电功耗增加,导致芯片温度 T 升高,反过来又使得 I_{lkg} 增大,漏电功耗增加;这一恶性循环限制了 DVS 技术的应用深度^[34].

$$I_{lkg} \propto e^{-\frac{qV_{th}}{kT}}, \quad (9)$$

$$V_{th} = V_{th1} - K_1 V_{dd} - K_2 V_{bs},$$

其中, q 为单位电荷; k 为普朗克常量; V_{th} 为阈值电压; T 为绝对温度; V_{th1} , K_1 , K_2 和 V_{bs} 分别为技术参数,其值参见文献[33].

3) CS-DVS 和 CS-DVS-P 方法

鉴于采取极端电压的 Naïve 方法和 DVS 方法的一些缺陷,文献[13]提出了 CS-DVS 方法及其的一个改进算法 CS-DVS-P. CS-DVS 方法认为当电压降低到一定程度时,任务的执行时间将会大幅度延长,虽然动态能耗持续降低,但是执行时间的延长导致漏电能耗不可再被忽略.因此,CS-DVS 方法权衡动态能耗和漏电能耗,将电压设置为二者总能耗最低时的临界值 V_{cr} ,任务完成后将处理器置于休眠状态.电压高于该临界值时,动态能耗过高导致总能耗过高,电压低于该临界值时,漏电能耗过高导致总能耗过高. CS-DVS 方法首先给出一个时钟周期内的能耗模型(参见式(10)),而后通过对能耗关于电压进行求导来获得能耗最低时的临界电压值.该研究通过实验证明,CS-DVS 方法相较于 DVS 能够节省 5% 的能耗.

$$t_{cycle} = f^{-1}, \quad f \propto V_{dd},$$

$$E_{dyn} = P_{dyn} t_{cycle} = C_{eff} V_{dd}^2,$$

$$E_{lkg} = P_{lkg} t_{cycle} = f^{-1} \cdot L_g \cdot (I_{subn} V_{dd} + |V_{bs}| I_j),$$

$$E_{on} = P_{on} t_{cycle} = f^{-1} P_{on},$$

$$E_{cycle} = E_{dyn} + E_{lkg} + E_{on} = \alpha f^2 + \beta f^{-1}. \quad (10)$$

CS-DVS-P 方法则是在 CS-DVS 方法的基础上进一步考虑了唤醒代价和休眠代价之间的权衡,当休眠代价高于唤醒代价时,选择将处理器置于关闭状态以降低系统的整体能耗.为此,CS-DVS-P 方法还添加了一个任务延迟执行机制,尽可能将任务集中起来运行从而提高关闭处理器的概率.该研究通

过实验证明,CS-DVS-P 方法能够节省 18% 的能耗.

关于时间约束,这类方法对比所求的临界电压值 V_{cri} 和可能的最小电压值 V_{min} ,若 $V_{cri} < V_{min}$,则会牺牲部分能耗优化效果,优先满足任务的时间约束.

然而,这类方法依然存在 2 个问题:1)未考虑任务特征对功率的影响;2)采用简单的执行时间与频率成反比的关系模型.对于问题 1,处理器功率不仅受执行电压影响,还与其使用率有关,而该类方法普遍假设使用率不变,也即忽略了使用率对功率的影响.然而事实上,任务执行期间处理器的使用率实时变化,且不同特征的任务具有不同的使用率变化情况,从而导致:①同一任务相同电压下,不同时刻的功率也会因为使用率不同而不同;②同一任务不同电压下,使用率发生变化,功率也不同;③不同任务即使设置相同的执行电压,功率也会不同.对于问题 2,实际执行中时间并非简单地与频率成反比,道理同上.由此可见,能耗即为功率在时间上的积分,如若功率和时间的计算不准确,将会严重影响能耗的计算准确性,进而影响基于该模型的优化效果.文献[37]给出的功率模型虽然考虑了使用率的影响,但是一方面它针对的是 PC 机,未必适用于实时系统;另一方面该模型中的使用率只有在任务执行时才能通过系统接口实时获取使用率,并不满足这里的预先计算需求.文献[35]给出的功率模型也考虑了使用率的影响,但是粒度仅为较粗的任务级别,每个任务对应一个使用率影响因子(原文中称作 activity factor),并未考虑到该值在任务执行期间的动态变化.

3.2 温度感知的优化研究

温度感知的优化研究是随着半导体技术的不断发展而衍生出来的.越来越高的功率密度使得芯片的温度也越来越高,一方面使得受温度影响的漏电能耗无法再被忽略,另一方面过高的温度还会严重影响处理器的性能、稳定性和可靠性等.因此,在温度无关的优化研究的基础上,研究工作者将峰值温度约束考虑在内进一步提出了新的温度感知的优化研究.

通过详细调研,我们发现温度感知的相关研究可以分为单任务的优化研究和多任务的调度研究 2 个层面,二者相辅相成,可以综合应用.对于单任务的优化研究,现有工作^[5,10,16-20]大多致力于发现更高效的 PSS 方法,其研究目标也通常为降低峰值温度以及系统能耗.处理器状态包括工作、休眠以及关闭状态,其中处于工作状态时还要确定处理器的工作电压,对此部分 PSS 方法融合了 DVS 技术,通过寻找较优的系统电压以降低系统总能耗.对于多任务的调度研究,可按照优化目标将其分为 3 类:以提高性能为目标、以降低能耗为目标和以提高能效为目标.

3.2.1 单任务优化

关于单任务优化,本节主要介绍现有的典型的优化方法中的 4 种:TALK (temperature aware leakage minimization), PB (pattern based), MO (micro-oscillating), VP-TALK (voltage prediction TALK).

关于实时任务的时间约束,这 4 种方法本质上均通过以下方法来保障:将实时任务的最坏执行时间 C 作为时间条件来求解满足某种优化目标的最优解.如果最优解能够使得任务在最坏情况下也满足截止期限,那么就保证了实时任务的时间约束.

1) TALK 方法

TALK 及其改进方法^[19-20]将周期 P 等分 n 段,每段中根据当前的负载 C_i 以及温度 T_{cur} 来决定处理器是工作还是休眠,以此来控制处理器的峰值温度.

TALK 方法首先根据 RC(thermal resistances, thermal capacitances)建模^[28]给出芯片的温度升降函数,而后给出每个段中处理器状态的判定条件(成立时,判定结果为休眠,反之为工作),详见式(11).此处的温度升降函数与 2.2 节中给出的系统热模型本质上是一样的,其中芯片的当前温度可通过温度传感器或温度评估方法获取^[11]. TALK 方法的示意图如图 3 所示.该研究通过实验证明,该方法相对于 Naïve 以及 CS-DVS 方法分别能够平均节省 30% 和 11% 的能耗.

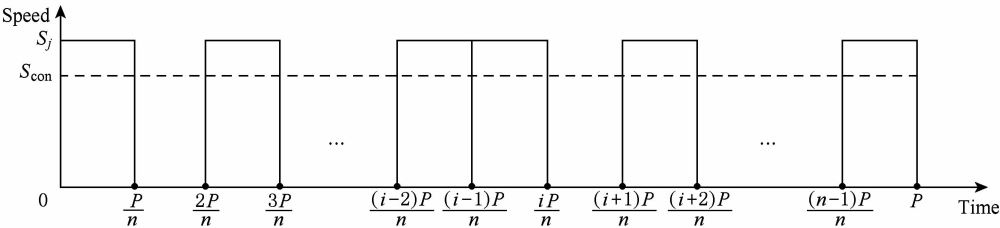


Fig. 3 TALK approach.
图 3 TALK 方法示意图

$$T_{\text{rise}}(t) = K_1 - (K_1 - T_{\text{cur}})e^{-\frac{t}{K_3}}, \quad (11)$$
$$T_{\text{fall}}(t) = K_2 + (T_{\text{cur}} - K_2)e^{-\frac{t}{K_3}},$$

判定条件:

$$\frac{C_i}{D_i - C_i} \leq \frac{T_{\text{cur}} - K_2}{K_1 - T_{\text{cur}}},$$

其中, K_1 和 K_2 为热平衡状态时的最高温度和最低温度; K_3 为热容(C_{th})和热抗(R_{th})的乘积; C_i 为当前还需执行的负载,其值为最坏执行时间 C 与已执行时间 t 的差; D_i 为截止期限 D 与已执行时间 t 的差.

TALK 认为漏电能耗为主导能耗,且主要考虑温度因素对能耗的影响,因此忽略了不受温度影响的动态能耗;该方法通过考虑了温度因素的判定条件来满足峰值温度约束,处理器状态的切换时间和能耗代价也可以通过简单的修改添加在内.然而,TALK 还存在以下缺点:①TALK 方法中处理器的工作电压为定值,但是文献中并未给出该值的确定方法,只是在实验部分声明采用 CS-DVS 方法中确定的临界电压值;②TALK 方法中系统能耗与 n 密切相关,但是文献中并未给出段数 n 的确定方法,只是通过实验结果得出了系统能耗随 n 的增加而降低的结论,且这个降低是有限的,受唤醒时间和能耗代价的影响(当 n 增到一定程度时,唤醒代价会抵消处理器置于休眠状态带来的能耗优化).换句话说,本文并未明确给出依据什么来确定 n 值以降低系统能耗,这是否意味着我们只能通过经验来设定 n 的值?

2) PB 方法

PB^[16]方法将任务周期 P 等分为 n 段,并在每段中控制处理器都以相同的模式运行(工作-休眠交叉进行),使得因执行任务而温度升高的处理器有时间来冷却,以此来控制处理器的峰值温度并降低漏电能耗,最后达到能耗优化的目的.

PB 采用式(7)所示的漏电流简化模型计算调度算法的温度效果以控制处理器的峰值温度;其次,该方法认为供电电压为固定值,因此将动态能耗看作常值,从而根据热模型推导出温度与时间之间的关系,并根据恒温定理(恒温状态下的段的起止温度相

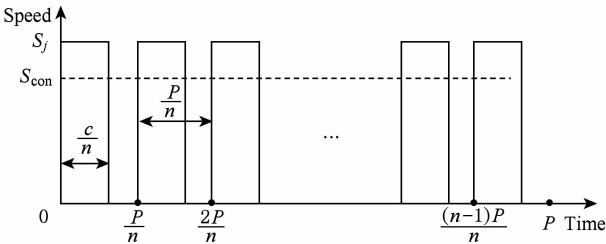


Fig. 4 PB approach.

图 4 PB 方法示意图

同)求解恒温值以控制处理器的峰值温度;最后将恒温状态下的系统能耗表示为段数 n 的函数,并通过求导获取能耗最小时的 n 值,参见式(12). PB 方法示意图参见图 4. 该研究设计实验验证该方法在 11 组不同 $\langle P, C \rangle$ 的测试基准下的优化效果,实验证明该方法与 TALK 方法相比能够节省 5%~8% 的能耗.

$$E(n) = n \left(\int_0^{\frac{C}{n}} P_{\text{act}}(t) dt + \int_0^{\frac{P-C}{n}} P_{\text{dor}} dt + E_{\text{sw}} \right) =$$
$$n \left(\int_0^{\frac{C}{n}} (P_{\text{dyn}} + v_{\text{dd}}(\hat{A}T^2 + \hat{B})) dt + E_{\text{sw}} \right) +$$
$$(P - C)P_{\text{dor}} = \Psi(n) + C(P_{\text{dyn}} + \hat{B}) +$$
$$(P - C)P_{\text{dor}}. \quad (12)$$

基于模式的优化方法 PB 能够让系统达到一个恒温状态,由此可以控制系统的峰值温度.与 TALK 相比,本文不是局部地调度每个段内的处理器状态,而是综合周期内的整体情况进行模式化调度,且明确给出能耗最小时段数 n 的确定方法.然而,PB 方法还存在 3 个缺点:1)PB 方法中处理器的工作电压为定值,但是对应文献中并未给出该值的确定方法;2)PB 方法中给出的系统能耗模型仅适用于恒温状态,而系统达到该状态还需一段时间,本文并未对此进行分析考虑;3)能耗模型只考虑了处理器状态切换的能耗代价,却忽略了相应的时间代价.事实上,PB 方法既然针对的是恒温状态,那么就可以将工作电压值设定为 V_0 , V_0 为某个满足峰值温度约束条件的温度下使得热模型中温度增长率为零时对应的电压值.

3) MO 方法

MO 及其改进方法^[5,10,17]在 PB 方法的基础上,将每个段中的处理器运行模式改为工作(高速)-工作(低速)模式,以此来更好地控制峰值温度并降低漏电能耗,最后达到能耗优化的目的.该运行模式已在其研究中得到证明,能够最大限度降低处理器的温度.

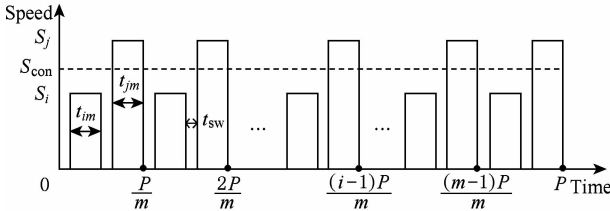


Fig. 5 MO approach.

图 5 MO 方法示意图

① MO 根据文献[24]中关于漏电流随温度变化的研究结果,将漏电流模型简化为线性模型^[4](参见式(8)),并根据热模型推导出温度随时间变化的

原函数(参见式(13)),进而求得给定段的结束温度;
②根据热模型利用简化函数以及每个段的结束温度,通过积分的方法推导出能耗的计算公式(参见式(14));③考虑电压切换时间和能耗代价以及时间和峰值温度约束,求解能耗最小时的段数 m (参见式(15)). MO 方法的示意图见图 5 所示. 该研究通过实验证明,该方法相对于 Naïve 方法以及 PB 方法能够分别节省 25.7% 和 11.2% 的能耗,峰值温度分别降低 23℃ 和 12℃.

$$P(k) = P_{\text{lg}}(k) + P_{\text{dyn}}(k) = v_k(C_1(k) + C_2(k)T) + C_3v_k^3,$$

$$\frac{dT(t)}{dt} = \frac{P(k)}{C_{\text{th}}} - \frac{T(t)}{R_{\text{th}}C} = A(k) - B(k)T(t),$$

$$A(k) = \frac{C_1(k)v_k + C_3v_k^3}{C_{\text{th}}},$$

$$B(k) = \frac{1}{R_{\text{th}}C_{\text{th}}} - \frac{C_2(k)v_k}{C_{\text{th}}},$$

$$G(k) = \frac{A(k)}{B(k)},$$

$$T(t) = G(k) + (T_0 - G(k))e^{-B(k)(t-t_0)}. \quad (13)$$

$$E(t_0, t_e) = \int_{t_0}^{t_e} P(t)dt = C_{\text{th}}(T_e - T_0 + \frac{1}{R_{\text{th}}C_{\text{th}}} \int_{t_0}^{t_e} T(t)dt) = C_{\text{th}}(G(k) + (T_0 - G(k))e^{-B(k)(t-t_0)} - T_0 + \frac{1}{R_{\text{th}}C_{\text{th}}}(G(k)(t_e - t_0) + \frac{1}{B(k)}(G(k) - T_0)(e^{-B(k)(t-t_0)} - 1))). \quad (14)$$

$$E(m) = m \cdot (E(t_a, t_b) + E(t_c, t_d)) + 2m \cdot E_{\text{sw}},$$

$$T_{\text{peak}}(m) \leq T_{\text{threshold}}. \quad (15)$$

MO 方法中给出的能耗评估方法相对于文献[46]中的方法,计算代价明显降低,其精确度受简化后的漏电流线性模型的精度影响,且实验证明其误差率在 4.8% 以内. 另外,MO 方法证明,相对于其他速度组合,采用 2 个最接近的速度组合完成分配给处理器的周期任务能够最大限度降低处理器的温度,这是因为电压切换跨度越大,切换的时间和能耗代价也越大. 因此,MO 方法中的 2 个工作速度分别为与最小速度 $S_{\text{con}} = C/P$ 邻近的 2 个速度值.

4) VP-TALK 方法

VP-TALK^[18] 在 TALK 的基础上给出了处理器运行电压的确定方法以及最优段数 n 的动态确定方法,弥补了 TALK 中不考虑任务负载以及未给出段数确定方法的缺陷. 对于前者,该方法通过负载量预测确定与其近似匹配的电压,弥补了 TALK 中不考虑任务负载的缺陷;首先将复杂任务按频谱长短

分为随机、周期和趋势 3 种成分,接着分别采用灰色模型、傅里叶模型和径向基函数(RBF)神经网络模型对这 3 种成分进行组合分析,最后获得负载的精确预测值并确定匹配的电压值. 对于后者,该方法根据 TALK^[19] 中获取的段数 n 的上下限,穷举所有值以确定能耗最小时对应的 n 值. 另外,该方法还以 TALK, PB, MO 以及 VP-TALK 四种源算法为候选算法,从中选取最佳的调度策略,并通过机器学习反馈指导调度策略的选择. 负载预测、调度策略选择以及调度策略评价 3 个模块共同构成的 DPTM 系统的工作流程图如图 6 所示:

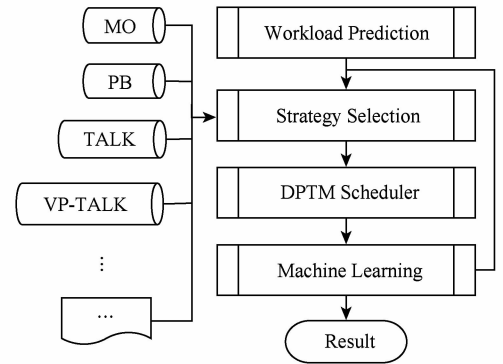


Fig. 6 Work flow of DPTM system^[18].

图 6 DPTM 系统工作流程^[18]

该研究通过实验证明,所提出的负载预测模型优于现有大多数相关模型及算法,平均误差仅为 2.89%;所提出的 VP-TALK 算法在较高的工作负载率和共同的峰值温度约束下,比 PB, MO, TALK 方法分别节能 20.54%, 11.04%, 11.42%;所提出的综合 4 种源算法基于机器学习的 DPTM 原型系统较为接近理想值,与其 $E_{\text{TOTAL}}/E_{\text{MAX}}/TP_{\text{AVG}}/TP_{\text{MAX}}$ 参数相比,分别只产生了 12.55%, 13.93%, 9.02℃, 7.91℃ 的差距.

然而,VP-TALK 中并未明确指出 DPTM 系统中负载预测的触发时机,也没有分析调度系统中负载预测、机器学习等占用的资源对任务执行的影响以及整个系统的调度代价与调度粒度之间的权衡关系,也没有给出预测偏差较大时的补救措施.

3.2.2 多任务调度

多任务调度致力于为一组实时任务寻找一个合适的执行计划,从而使所有任务在满足给定约束条件下达到某种优化目标.

关于约束条件,首先实时任务均有截止期限,执行计划必须保证所有任务在截止期限内完成;其次,实时系统还有一个温度上限,执行计划必须保证任务执行期间系统温度不超过该值. 针对上述约束条

件,衍生出了实时任务的可行性分析研究.文献[29, 40]首先分析了多任务可行性分析研究在时间和温度约束条件下存在的问题与挑战,接着将漏电流与温度之间的关联关系考虑在内提出了可行性调度的判定条件,并通过大量实验进行验证.关于优化目标,多任务调度优化研究主要分为性能、能耗以及能效3个优化目标.其中,性能是指给定时间内完成的任务数量;能耗是指完成给定任务所耗的能量;能效综合考虑前2个优化目标,衡量单位能耗完成的任务量.这3个优化目标是随着芯片工艺技术的不断发展而逐渐出现的.多任务调度确定执行计划后,在任务具体执行时有时也会采用3.2.1节中给出的单任务优化方法.

执行计划的确定是多任务调度的核心任务,具体方法根据系统类型分为单核和多核2种.对于单核系统,多任务通常按照EDF, Priority-Based(包括固定优先级^[47]以及考虑任务的剩余价值密度和执行紧迫性的动态优先级^[48]2种)等方法来确定任务执行计划.而对于多核系统,首先需要根据核的资源特征以及温度特性确定任务与核之间的映射关系,而后按照单核的方法来确定最终的执行计划.此外,考虑到即将要描述的3种不同优化目标的多任务调度研究均涉及到多核情况,本文将适当省略部分介绍.

1) 性能优化

多任务调度的性能优化需要解决的关键问题是任务执行计划的确定.文献[35]以表征性能的指标吞吐量为优化目标,将不同任务特性对能耗的影响考虑在内(引进一个activity factor,参见2.3.1节),首先通过热模型计算任务达到稳定状态时的温度 T_{ss} ,接着根据 T_{ss} 与 T_{max} (最高约束温度)之间的大小关系将任务分为Hot Task和Cool Task,最后根据任务场景采用PB或者“冷热任务组”(task pair of a hot task and a cool task)方法寻找满足温度约束的任务延迟最小的执行计划.前面提出的 T_{ss} 是指温度与能耗之间相互影响而达到的最终收敛温度,也即 $dT(t)/dt=0$ 时的温度.“冷热任务组”方法与MO方法类似,二者的区别是前者是在2个任务间切换以保证系统的峰值温度不超过既定阈值,后者则是通过一个任务在2个执行速度(与处理器频率相关)间切换来保证.然而,该文献中关于任务执行时间的计算,将其看作与处理器频率成反比,该方法得到的计算结果并不准确,执行时间还受任务特性的影响,不同任务即便是在相同频率设定下执行时间也不同.

文献[49]采用DVFS和任务迁移技术实时控制核的执行速度以及任务到核的分配策略来降低运

行在多核系统上的任务集合的整体完成时间,从而提高系统的性能.该文献为推导最优策略提供了理论基础,文中提出的Task-to-Core映射方法相较于基于功耗的线程迁移方法在性能上提高了20.2%.该文献提出的方法技术已被整合到一个温度感知的体系结构级模拟器MAGMA;MAGMA能够在合理的时间内模拟数百个核,为相关研究人员挖掘更优的温度感知的管理方法提供便利,现已被很多研究用来模拟多核实时系统.文献[50]针对多核处理器提出了一种动态温度管理方法,致力于在降低处理器峰值温度的同时尽可能减少对系统性能的影响.相比于已有工作,该文献考虑了负载以及核的产热行为,分析其对负载迁移决策的影响,并通过全局迁移特性来调控多核的工作频率.事实上,所有优化方法的挖掘均应考虑随之而来的代价以及对原有性能的影响,二者之间通常需要做个权衡.

2) 能耗优化

多任务调度的能耗优化研究又可以分为单核能耗优化和多核能耗优化.多任务单核能耗优化主要也是确定系统能耗最小的任务执行计划,这与前面讲的性能优化方法有异曲同工之处,只是将优化目标变为能耗,此处不再赘述.多任务多核能耗优化在单核能耗优化基础上还需考虑任务与核之间的映射关系.不同类型的任务在不同核上的执行能耗不同,因此在多任务调度中,还需为任务集选取合适的核映射方案.

对此,文献[51]提出一种粗粒度的宏观方法,将Hot Task分配给对其他核的温度影响较小的核,将Cool Task分配给对其他核的温度影响相对较大的核,以此来降低系统温度,进而降低能耗.文献[29]对其进行细化,针对不同的核分布情况设计不同的温度感知的Task-to-Core分配策略,所遵循的基本原则如下:按照热度对Hot Task进行排序,尽量将较热的任务分配给对周边核温度影响较小的核.具体步骤如下,对于一个有 N 个核、 N 个任务($T_1, T_2, \dots, T_N$)的实时系统,首先计算能够保证满足实时任务截止期限约束的最小执行速度,而后计算每个任务需要的功耗,并按照功耗对任务的热度进行排序,将热度较高的任务分配给位置较偏远的核.例如,对于一个 3×3 平面布局的实时系统,假设9个实时任务的热度排序为 $P_1 > P_2 > \dots > P_9$,那么9个任务到9个核的分配图如图7所示.

然而,相关文献[29, 51-52]只是定性研究了任务到核的静态映射方法,并未考虑系统运行过程中温度超出峰值时对实时任务的动态管理方法,比如

T_1	T_7	T_3
T_5	T_9	T_6
T_4	T_8	T_2

Fig. 7 Task allocation in a system with 3×3 layout.
图7 3×3布局下的任务分配示意图

可以通过任务迁移均衡各核的温度或通过 DVFS 等方法来达到在保障系统温度约束的同时,降低系统能耗.

3) 能效优化

“能效”一词在近些年来已被广泛关注,不仅涉及到个人计算机、数据中心服务器、超级计算机等,实时系统也不例外.文献[53]就分布式集群的能效进行分析,首先给出了能效定义,从理论上推导出能效最高的条件并通过实验进行验证,同时还指出了提高能效的几种途径.然而,实时系统的能效优化由于硬件等的差异与之有所相同.

文献[54]在给定平均吞吐量的约束下,提出一个多层框架,通过多核整合、DVFS 以及任务迁移等技术降低同构多核系统的能耗.该研究隶属于能效优化,然而却忽略了峰值温度约束,也未考虑温度与漏电功耗之间的关系.文献[55]研究了单核实时系统下温度感知的能效优化方法,通过控制时钟以及风扇速度来提高能效.然而,该研究采用了一个简化的热模型,未考虑温度对时延(过高的温度会降低处理器的计算速度,从而降低性能延长任务执行时间)的影响.文献[56]研究了多核实时系统下温度感知的能效优化方法,首先将温度对时延的影响考虑在内分析了系统的能效特性,接着提出通过在线 DVFS、任务迁移以及风扇速度控制实现系统的高能效,寻找能效最高时的 Task-to-Core 映射.实验证明该方法与单纯的性能优化相比,系统能效提高了 3.2 倍;与现有应用在 PC 上的能效优化技术相比提高了 37%.然而,该文献中 DVFS 以及 Task-to-Core 的调度频率过高,调度间隔分别为 10 ms 和 100 ms,因此应将调度代价(切换时间、能耗代价)考虑在内.文献[57]在采用运行时 DVFS 和 Task-to-Core 映射技术外,还对 DVFS 进行扩展,使其涵盖 Near-Threshold Computing, Nominal 和 Turbo-Boost 三种区间调度模式,以此解决深亚微米技术下受限的电压调节问题,从而进一步提高系统能效;此外,还将影响漏电能耗以及时延预测的过程变动信息融合在内,避免做出太过乐观的决策.

3.3 优化方法对比分析

根据前 2 节的描述可以看出实时系统温度功耗管理研究的发展历程:由传统的不考虑温度对功耗影响的性能/能耗优化到温度感知的单任务优化,再到温度感知的多核多任务优化.本节对前面给出的优化方法进行总结,主要从研究分类、具体机制、优化目标、电压是否恒定以及调度计划制定的时间这 5 个方面进行对比,如表 1 所示.通过对比,本文总结实时系统温度优化管理研究需要考虑的因素如下:

1) 优化目标.实时系统温度功耗管理的优化目标有很多,包括性能/吞吐量、能耗和能效等.优化目标不同,具体优化方法也不同.例如,性能优化通常是站在任务的角度出发的,吞吐量和能效优化则通常是站在系统的角度出发的,能耗则二者均可.从任务角度出发的最佳优化在系统角度看未必最优.

2) 优化对象.实时系统有单核也有多核,优化方法有针对单任务的也有针对多任务的.不同的优化对象,需要考虑的优化层面不同:多核优化通常还需要考虑 Task-to-Core 的映射优化,而多任务优化还需考虑任务组合运行的优化.任务组合运行时,还涉及到任务的中断与继续,由此衍生出任务切换代价.

3) 优化粒度.优化粒度越细,优化效果越好,然而代价也越高.这里,代价不仅包括优化算法的计算代价,还包括优化调度时的切换代价(如 DVFS 时处理器状态的切换代价和任务迁移时的迁移代价).因此,需要选取一个合适的优化粒度.

4) 模型简化.在优化过程中,往往会为了降低优化算法的计算复杂度而对一些模型或影响因素进行简化.漏电流函数是最常被简化的,简化方法有很多种,它们的适用条件、简化后的精度以及计算代价都不同.因此,有必要在精度与计算效率之间权衡,对模型做出恰当的简化.此外,优化目标函数中通常也会忽略一些诸如固有能耗、切换代价(包括能耗和时间)、不同任务对 CPU 使用率的影响以及温度对处理器频率的影响等因素,但这些因素分别在任务执行时间较长、切换较为频繁、任务类型(包括 CPU 密集型、IO 密集型、交互性等)相差较大以及温度变化较大时是不可忽略的,否则会严重影响优化效果.

5) 优化时间.优化方案的确定可以是静态的也可以动态的,二者各有优劣.前者无需考虑任务执行时的变动因素,比后者简单,但优化效果却不如后者;动态优化中,可以动态校准诸如工作电压、任务分配以及风扇速度等因素,从而使优化结果更准确.

Table 1 Optimization Approaches Comparison of Thermal and Power Management for Real-time Systems

表 1 实时系统温度功耗管理优化方法对比

Temperature Research	Category	Approach	Mechanism	Goal	Pros &. Cons	Fixed Voltage	Timing
Temperature Independent Research	Ignore P_{lkg}	Naïve	Set voltage to be the possible maximum value.	Performance	Highest performance with highest energy consumption.	Y	Offline
		DVS	Set voltage to be the possible minimum value.	Energy Consumption	Lower energy consumption with lower performance.	Y	Offline
	Consider P_{lkg}	CS-DVS &. CS-DVS-P	Set voltage to be the critical value.	Energy Consumption	Lowest energy consumption	Y	Offline
Temperature Aware Research	Single Task Optimization	TALK	TALK: Decide the state (active or sleep) of the processor by comparing current workload with temperature.	Energy Consumption	Minimize peak temperature and energy consumption, while resulting in possible deadline misses.	Y	Online
		PB	PB: Simplify the P_{lkg} model, and explore the optimal n and the best work-sleep oscillating mode.	Energy Consumption	Further reduce energy consumption compared with TALK, while energy consumption model is only for the steady state.	Y	Online
		MO	PSS: Equally divide period into n segments, then MO: Simplify the P_{lkg} model, and explore the optimal n and the best high-low oscillating speed mode.	Energy Consumption	Further reduce energy consumption on the basis of PB.	Y	Online
		VP-TALK	VP-TALK: Predict the workload to decide the voltage and optimal n , select the best one among the 4 methods, and improve DPTM through the feedback from machine learning.	Energy Consumption	Further reduce energy consumption compared with the above 3 methods through workload prediction.	N	Online
	Multi-task Scheduling	Execution Plan	① Divide tasks into Hot and Cold by comparing T_{ss} with T_{max} , then find the execution plan according to the specific scenarios.	Performance/ Throughput	Minimize the delay of task and increase the throughput.	N	Online
		DVFS &. Task-to-Core Mapping	② Dynamically decide the voltages of cores, and allocate tasks to cores according to the following principle: allocate hot tasks to cores with little temperature effect on the other ones.	Energy Consumption	Reduce energy consumption from different aspects.	Y	Online
			③ Analyze the system's energy efficiency features and schedule it with different granularities in regard to: fan-speed, Task-to-Core and DVFS.	Energy Efficiency	Improve EE from different levels and granularities.	Y	Online

4 总结与展望

本文综述了实时系统中多约束条件下的温度功耗管理优化方法:首先介绍并分析了温度功耗管理的相关知识及现有研究分类,而后从实时系统模型以及优化方法 2 个方面对研究现状进行总结. 实时系统模型是温度功耗管理优化的基础,包括任务模型、热模型、功耗模型等. 优化方法则随着半导体技术的发展根据是否考虑温度分为 2 部分:温度无关的优化研究以及温度感知的优化研究. 其中,后者又包

含单任务优化和多任务调度,这二者相辅相成,共同构成实时系统的优化机制. 目前,实时系统中多约束条件下的温度功耗管理研究已经日趋成熟,然而仍存在问题需要进一步研究完善,具体总结如下:

1) 考虑不同任务特性对动态功耗以及任务执行时间的影响,研究更精准的能耗(功率在时间上的积分)模型. 不同计算类型(可以是 CPU 密集型,也可以是 IO 密集型)的实时任务,即使采用完全相同的处理器调度方法,其执行功耗也差别很大,因为不同计算类型的任务对处理器的使用率影响很大,进而影响能耗;此外,实时任务执行过程中,不同处理

器状态(频率和电压)下执行时间不同,现有研究采用简单的执行时间与频率的反比关系远远不够精确,还需研究更准确的时间模型以确保能耗的计算准确性以及基于该模型的优化效果. 综上,将不同任务特性对功耗以及执行时间的影响考虑在内的能耗计算模型亟待研究与发现.

2) 考虑任务对资源的需求特性以及电压调节代价,探究合适的动态电压调节粒度. 现有的 DVS 及其衍生方法中所说的动态电压调节的粒度是任务,也即根据优化目标确定实时任务的最佳执行电压(整个执行期间均以该电压运行),而此处要强调的是应该将这一粒度细化,在任务执行期间可以设置固定的时间窗口定时选取最佳执行电压,这是因为任务执行的不同阶段对资源的需求不同,若是仅笼统地为任务选取单一的执行电压或者固定的电压模式势必会削弱 DVS 的优化效果. 当然,增大 DVS 的调节粒度是要增加额外代价的,这就需要根据实际情况探究合适的调节粒度,这也是未来研究的一个着眼点.

3) 研究实时任务执行期间处理器使用率的变化规律以及基于此的分类方法. 1.1 节中介绍的周期任务、偶发任务以及非周期任务是根据任务的实例发布规律来进行分类的,而实例发布规律对实时系统的功耗以及受功耗影响的温度并没有直接的关联关系. 相反,任务执行期间处理器使用率的变化情况却对系统功耗有较大影响,不同变化规律对应不同的优化方法. 因此,针对每种类型特性研究其对应的温度功耗优化方法,进而可以为每一种类型任务研发专用的实时系统,将会很大程度上降低优化的复杂度.

参 考 文 献

- [1] Moore G E. Cramming more components onto integrated circuits [J]. *Electronics*, 1965, 38(8): 114-117
- [2] Dennard R H, Gaensslen F H, Rideout V L, et al. Design of ion-implanted MOSFET's with very small physical dimensions [J]. *IEEE Journal of Solid-State Circuits*, 1974, 9(5): 256-268
- [3] Skadron K, Stan M R, Huang Wei, et al. Temperature-aware computer systems: Opportunities and challenges [J]. *IEEE Micro*, 2003, 23(6): 52-61
- [4] Quan Gang, Zhang Yan. Leakage aware feasibility analysis for temperature-constrained hard real-time periodic tasks [C] //Proc of the 21st Euromicro Conf on Real-Time Systems. Piscataway, NJ: IEEE, 2009: 207-216
- [5] Huang Huang, Quan Gang. Leakage aware energy minimization for real-time systems under the maximum temperature constraint [C] //Proc of the Conf on Design, Automation and Test in Europe. Piscataway, NJ: IEEE, 2011: 479-484
- [6] Isovici D, Fohler G. Efficient scheduling of sporadic, aperiodic, and periodic tasks with complex constraints [C] //Proc of the 21st IEEE Real-Time Systems Symp. Piscataway, NJ: IEEE, 2000: 207-216
- [7] Awan M A. Energy and temperature aware real-time systems [D]. Porto, Portugal: University of Porto, 2014
- [8] Wilhelm R, Engblom J, Ermedahl A, et al. The worst-case execution-time problem overview of methods and survey of tools [J]. *ACM Trans on Embedded Computing Systems*, 2008, 7(3): Article number 36
- [9] Puschner P, Burns A. A review of worst-case execution-time analysis [J]. *Real-Time Systems*, 2000, 18(2): 115-128
- [10] Huang Huang, Chaturvedi V, Liu Guanglei, et al. Leakage aware scheduling on maximum temperature minimization for periodic hard real-time systems [J]. *Journal of Low Power Electronics*, 2012, 8(4): 378-393
- [11] Skadron K, Stan M R, Sankaranarayanan K, et al. Temperature-aware microarchitecture: Modeling and implementation [J]. *ACM Trans on Architecture and Code Optimization*, 2004, 1(1): 94-125
- [12] Liao Weiping, He Lei, Lepak K M. Temperature and supply voltage aware performance and power modeling at microarchitecture level [J]. *IEEE Trans on Computer-Aided Design of Integrated Circuits and Systems*, 2005, 24(7): 1042-1053
- [13] Jejurikar R, Pereira C, Gupta R. Leakage aware dynamic voltage scaling for real-time embedded systems [C] //Proc of the 41st Annual Design Automation Conf. New York: ACM, 2004: 275-280
- [14] Zhai Bo, Blaauw D, Sylvester D, et al. Theoretical and practical limits of dynamic voltage scaling [C] //Proc of the 41st Annual Design Automation Conf. New York: ACM, 2004: 868-873
- [15] Aydin H, Melhem R, Mossé D, et al. Dynamic and aggressive scheduling techniques for power-aware real-time systems [C] //Proc of the 22nd IEEE Real-Time Systems Symp. Piscataway, NJ: IEEE, 2001: 95-105
- [16] Yang Chuanyue, Chen Jianjia, Thiele L, et al. Energy-efficient real-time task scheduling with temperature-dependent leakage [C] //Proc of the Design, Automation and Test in Europe Conference and Exhibition. Leuven, Belgium: European Design and Automation Association, 2010: 9-14
- [17] Chaturvedi V, Huang Huang, Quan Gang. Leakage aware scheduling on maximum temperature minimization for periodic hard real-time systems [C] //Proc of the 10th IEEE Int Conf on Computer and Information Technology. Piscataway, NJ: IEEE, 2010: 1802-1809

- [18] Zhao Guoxing, Yan Jiaqi, Luo Zuying. Dynamic power and temperature management based on accurate workload prediction [J]. Chinese Journal of Computers, 2014, 37(11): 2298-2308 (in Chinese)
(赵国兴, 闫佳琪, 骆祖堂. 基于任务精确预测的实时功耗温度管理[J]. 计算机学报, 2014, 37(11): 2298-2308)
- [19] Yuan Lin, Leventhal S, Qu Gang. Temperature-aware leakage minimization technique for real-time systems [C] // Proc of the IEEE/ACM Int Conf on Computer-Aided Design. New York: ACM, 2006: 761-764
- [20] Yuan Lin, Leventhal S R, Gu Junjun, et al. TALK: A temperature-aware leakage minimization technique for real-time systems [J]. IEEE Trans on Computer-Aided Design of Integrated Circuits and Systems, 2011, 30(10): 1564-1568
- [21] Sergeant J E, Krum A. Thermal Management Handbook: For Electronic Assemblies [M]. New York: McGraw-Hill Professional, 1998
- [22] Li Xiang, Jiang Xiaohong, Wu Zhaohui, et al. Research on thermal management methods for green data centers [J]. Chinese Journal of Computers, 2015, 38(10): 1976-1996 (in Chinese)
(李翔, 姜晓红, 吴朝晖, 等. 绿色数据中心的热量管理方法研究[J]. 计算机学报, 2015, 38(10): 1976-1996)
- [23] Skadron K, Stan M R, Huang W, et al. Temperature-aware microarchitecture [C] // Proc of the 30th Annual Int Symp on Computer Architecture. New York: ACM, 2003: 2-13
- [24] Liu Yongpan, Dick R P, Shang Li, et al. Accurate temperature-dependent integrated circuit leakage power estimation is easy [C] //Proc of the Design, Automation and Test in Europe Conference and Exhibition. San Jose, CA: EDA Consortium, 2007: 1526-1531
- [25] Cheng L, Huang K, Chen G, et al. Minimizing peak temperature for pipelined hard real-time systems [C] //Proc of the Design, Automation and Test in Europe Conference and Exhibition. Piscataway, NJ: IEEE, 2016: 1090-1095
- [26] Bao M, Andrei A, Eles P, et al. Temperature-aware voltage selection for energy optimization [C] //Proc of the Conf on Design, Automation and Test in Europe. New York: ACM, 2008: 1083-1086
- [27] Rai D, Yang H, Bacivarov I, et al. Worst-case temperature analysis for real-time systems [C] //Proc of the Design, Automation and Test in Europe Conference and Exhibition. Piscataway, NJ: IEEE, 2011: 631-636
- [28] Skadron K, Abdelzaher T, Stan M R. Control-theoretic techniques and thermal-RC modeling for accurate and localized dynamic thermal management [C] //Proc of the 8th Int Symp on High-Performance Computer Architecture. Piscataway, NJ: IEEE, 2002: 17-28
- [29] Chaturvedi V. Leakage temperature dependency aware real-time scheduling for power and thermal optimization [D]. Miami, Florida: Florida International University, 2013
- [30] Sharifi S, Ayoub R, Rosing T S. TempoMP: Integrated prediction and management of temperature in heterogeneous MPSoCs [C] //Proc of the Design, Automation and Test in Europe Conference and Exhibition. San Jose, CA: EDA Consortium, 2012: 593-598
- [31] Ukhov I, Bao M, Eles P, et al. Steady-state dynamic temperature analysis and reliability optimization for embedded multiprocessor systems [C] //Proc of the 49th Annual Design Automation Conf. New York: ACM, 2012: 197-204
- [32] Veendrick H J. Short-circuit dissipation of static CMOS circuitry and its impact on the design of buffer circuits [J]. IEEE Journal of Solid-State Circuits, 1984, 19(4): 468-473
- [33] Martin S M, Flautner K, Mudge T, et al. Combined dynamic voltage scaling and adaptive body biasing for lower power microprocessors under dynamic workloads [C] //Proc of the IEEE/ACM Int Conf on Computer-Aided Design. New York: ACM, 2002: 721-725
- [34] Mudge T. Power: A first-class architectural design constraint [J]. Computer, 2001, 34(4): 52-58
- [35] Huang Huang, Chaturvedi V, Quan Gang, et al. Throughput maximization for periodic real-time systems under the maximal temperature constraint [J]. ACM Trans on Embedded Computing Systems, 2014, 13(2): Article number 70
- [36] Brooks D, Tiwari V, Martonosi M. Wattch: 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
- [37] Song Jie, Li Tiantian, Wang Zhi, et al. Study on energy-consumption regularities of cloud computing systems by a novel evaluation model [J]. Computing, 2013, 95(4): 269-287
- [38] He Lei, Liao Weiping, Stan M R. System level leakage reduction considering the interdependence of temperature and leakage [C] //Proc of the 41st Annual Design Automation Conf. New York: ACM, 2004: 12-17
- [39] Huang Huang, Quan Gang, Fan J. Leakage temperature dependency modeling in system level analysis [C] //Proc of the 11th Int Symp on Quality Electronic Design. Piscataway, NJ: IEEE, 2010: 447-452
- [40] Quan Gang, Chaturvedi V. Feasibility analysis for temperature-constraint hard real-time periodic tasks [J]. IEEE Trans on Industrial Informatics, 2010, 6(3): 329-339
- [41] Lee C H, Shin K G. On-line dynamic voltage scaling for hard real-time systems using the EDF algorithm [C] //Proc of the 25th IEEE Int Real-Time Systems Symp. Piscataway, NJ: IEEE, 2004: 319-335
- [42] Jejurikar R, Gupta R. Dynamic slack reclamation with procrastination scheduling in real-time embedded systems [C] //Proc of the 42nd Annual Design Automation Conf. New York: ACM, 2005: 111-116

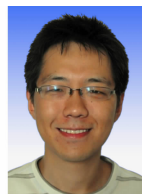
- [43] Bambagini M, Marinoni M, Aydin H, et al. Energy-aware scheduling for real-time systems: A survey [J]. *ACM Trans on Embedded Computing Systems*, 2016, 15(1): Article number 7
- [44] Das A, Kumar A, Veeravalli B, et al. Workload uncertainty characterization and adaptive frequency scaling for energy minimization of embedded systems [C] //Proc of the Design, Automation and Test in Europe Conference and Exhibition. San Jose, CA: EDA Consortium, 2015: 43-48
- [45] Lampka K, Forsberg B. Keep it slow and in time: Online DVFS with hard real-time workloads [C] //Proc of the Design, Automation & Test in Europe Conference & Exhibition. Piscataway, NJ: IEEE, 2016: 385-390
- [46] Liu Yongpan, Yang Huazhong, Dick R P, et al. Thermal Vs energy optimization for DVFS-enabled processors in embedded systems [C] //Proc of the 8th Int Symp on Quality Electronic Design. Piscataway, NJ: IEEE, 2007: 204-209
- [47] Abdeddaim Y, Chandarli Y, Davis R I, et al. Response time analysis for fixed priority real-time systems with energy-harvesting [J]. *Real-Time Systems*, 2016, 52(2): 125-160
- [48] Xia Jiali, Chen Hui, Yang Bing. A real-time task scheduling algorithm based on dynamic priority [J]. *Chinese Journal of Computers*, 2012, 35(12): 2685-2695 (in Chinese)
(夏家莉, 陈辉, 杨兵. 一种动态优先级实时任务调度算法 [J]. *计算机学报*, 2012, 35(12): 2685-2695)
- [49] Hanumaiah V, Vrudhula S, Chatha K S. Performance optimal online DVFS and task migration techniques for thermally constrained multi-core processors [J]. *IEEE Trans on Computer-Aided Design of Integrated Circuits and Systems*, 2011, 30(11): 1677-1690
- [50] Salami B, Noori H, Mehdipour F, et al. Physical-aware predictive dynamic thermal management of multi-core processors [J/OL]. *Journal of Parallel and Distributed Computing*, 2016 [2016-03-01]. <http://www.sciencedirect.com/science/article/pii/S74373151600037X>
- [51] Fan M, Chaturvedi V, Sha S, et al. Thermal-aware energy minimization for real-time scheduling on multi-core systems [J]. *ACM SIGBED Review*, 2013, 10(2): 27
- [52] Awan M A, Yomsi P M, Nelissen G, et al. Energy-aware task mapping onto heterogeneous platforms using DVFS and sleep states [J]. *Real-Time Systems*, 2016, 52(4): 450-485
- [53] Song Jie, Li Tiantian, Yan Zhenxing, et al. Energy-efficiency model and measuring approach for cloud computing [J]. *Journal of Software*, 2012, 23(2): 200-214 (in Chinese)
(宋杰, 李甜甜, 闫振兴, 等. 一种云计算环境下的能效模型和度量方法 [J]. *软件学报*, 2012, 23(2): 200-214)
- [54] Ghasemazar M, Pakbaznia E, Pedram M. Minimizing the power consumption of a chip multiprocessor under an average throughput constraint [C] //Proc of the 11th Int Symp on Quality Electronic Design. Piscataway, NJ: IEEE, 2010: 362-371
- [55] Shin D, Chung S W, Chung E Y, et al. Energy-optimal dynamic thermal management: Computation and cooling power co-optimization [J]. *IEEE Trans on Industrial Informatics*, 2010, 6(3): 340-351
- [56] Hanumaiah V, Vrudhula S. Energy-efficient operation of multicore processors by DVFS, task migration, and active cooling [J]. *IEEE Trans on Computers*, 2014, 63(2): 349-360
- [57] Maiti S, Kapadia N, Pasricha S. Process variation aware dynamic power management in multicore systems with extended range voltage/frequency scaling [C] //Proc of the 58th IEEE Int Midwest Symp on Circuits and Systems. Piscataway, NJ: IEEE, 2015: 1-4



Li Tiantian, born in 1989. PhD candidate. Student member of China Computer Federation. Her main research interests include energy efficient computing, and data intensive computing.



Yu Ge, born in 1962. Professor and PhD supervisor in Northeastern University. His main research interests include database theory and data flow.



Song Jie, born in 1980. PhD and associate professor in Northeastern University. His main research interests include cloud computing, data intensive computing and big data.