

# 基于多核处理器系统多维限制的节能实时调度<sup>①</sup>

张冬松<sup>1</sup>, 赵志峰<sup>1</sup>, 林志辉<sup>1</sup>, 王 珏<sup>1</sup>, 吴 飞<sup>2</sup>

<sup>1</sup>(镇江船艇学院 基础部, 镇江 212001)

<sup>2</sup>(上海工程技术大学 电子电气工程学院, 上海 201620)

**摘 要:** 介绍了多核处理器系统所面临的处理器实际限制、抢占调度实际限制和并行任务模型实际限制等多维限制挑战, 主要针对处理器开销模型、有限抢占模型和复杂并行任务模型等方面, 深入探讨了基于系统实际多维模型的多核节能实时调度研究, 为促进多核处理器系统在实时嵌入式领域的应用提供理论和技术参考。

**关键词:** 多核; 实时系统; 节能调度; 开销; 限制

## Energy-efficient Real-time Scheduling Based on Multidimensional Limitation in Multi-core Systems

ZHANG Dong-Song<sup>1</sup>, ZHAO Zhi-Feng<sup>1</sup>, LIN Zhi-Hui<sup>1</sup>, WANG Jue<sup>1</sup>, WU Fei<sup>2</sup>

<sup>1</sup>(Zhenjiang Watercraft College of PLA, Zhenjiang 212001, China)

<sup>2</sup>(College of Electronic and Electrical Engineering, Shanghai University of Engineering Science, Shanghai 201620, China)

**Abstract:** This paper introduces the multidimensional-limitation-based challenges for multi-core systems such as practical limitations of processor, preemptive scheduling and parallel task model. To solve this problem, it discusses the energy-efficient real-time scheduling strategies based on the practical multidimensional models including the processor overhead model and limited-preemptive model and complex parallel task model, and provides an theoretical and technical reference for the application of multi-core systems to real-time embedded field.

**Key words:** multi-core systems; real-time systems; energy-aware scheduling; overhead; limitation

实时系统应用广泛, 大量存在于汽车电子、电信通讯、互动电视、数字医疗、航空航天和便携式电子设备等领域。这些领域的发展迫切需要提供性能更强大、操作更灵活、效能比更高的微处理器系统, 以满足实时应用越来越复杂化、智能化的需求。当前, 多核处理器(On-chip Multiprocessors, CMP 或 Multi-core 或 Many-core, 以及片上多处理器、多处理器片上系统)已经成为市场的主流<sup>[1]</sup>, 嵌入式实时应用正朝着多核处理器平台的方向不断发展<sup>[2]</sup>。但是, 多核处理器的高性能会带来高能耗。例如 Intel Core2 Duo 处理器的功耗在 3GHz 的工作频率下达到 130W, Intel Core i7 处理器的最大功耗也超过 130W。而能耗是很多实时嵌入式系统首要考虑的因素, 特别是无人操控装备、无线移动和便携式计算设备<sup>[3]</sup>。这些设备往往都是依

靠电池供电, 但电池受重量、体积和尺寸的限制, 储能能力有限, 尤其在军事、航空等特殊领域不可能允许频繁更换电池。高能耗不仅能够缩短电池驱动的嵌入式设备的使用, 增加体积和重量, 而且会因为散热的增加而提高相应的冷却成本, 影响性能, 减少设备寿命, 降低系统可靠性。此外, 这些嵌入式系统为了满足实时性能约束, 往往采用保守设计策略, 进一步导致更高的系统能耗。

由于处理器能耗往往占整个计算机系统能耗的 50%, 所以降低处理器能耗已经成为实时多核系统中节能设计的关注焦点。例如, 不论是高性能服务器系统, 还是嵌入式实时系统, 处理器的功耗总是总功耗重要的组成部分<sup>[4]</sup>。事实上, 电容切换活动产生的动态功耗和泄露电流产生的静态功耗是 CMOS 处理器能

① 基金项目:国家自然科学基金(61402527,61272097);上海市科委项目(13510501400);上海市教委创新项目(12ZZ182)

收稿时间:2015-01-28;收到修改稿时间:2015-03-25

耗的两个主要来源<sup>[5]</sup>。动态电压频率调节(Dynamic Voltage Frequency Scaling, DVFS)技术<sup>[6]</sup>和动态功耗管理(Dynamic Power Management, DPM)技术<sup>[7]</sup>已经被应用于现代多核处理器系统中, DVFS 可以在运行时降低供应电压和执行频率以减少处理器的动态功耗,而 DPM 可以在运行时关闭处理器以减少处理器的静态功耗。由于动态功耗通常是处理器速度(或称频率)的凸函数和递增函数,所以一般来说速度越低,动态能耗越少。但是,较低的执行速度会延长实时任务的执行时间,又因泄露电流的影响而消耗更多的静态能耗。因此,节能实时调度研究就是需要兼顾动态能耗和静态能耗,在满足实时约束的条件下最小化系统能耗<sup>[8]</sup>。

从上世纪 90 年代末开始,节能实时调度技术研究已经成为当前学术和工业界共同关注的前沿领域和热门课题之一,并且随着“绿色计算”需求的提出,变得越来越重要。许多研究者已经针对多核多处理器系统节能实时调度提出了大量行之有效的解决方法,但是其现有技术还无法很好地应对多核处理器系统的处理器实际限制、抢占调度实际限制和并行任务模型实际限制等一些共性问题。为此,本文介绍了当前多核处理器系统所面对的多维限制挑战,深入探讨了基于系统实际多维模型的多核节能实时调度算法,将进一步促进多核处理器系统在实时嵌入式领域的大规模应用。

## 1 多核处理器系统中多维限制挑战

虽然多核系统节能实时调度算法研究已经取得了一些进展,但是,在大多数情况下,之前研究使用的系统模型都做了许多简化假设,这使得现有技术仍无法应对多核处理器实际限制、抢占调度实际限制和并行任务实际限制所带来的挑战。本文认为,这些限制既是挑战又是机遇,他们的成功解决,将最终决定多核节能实时调度研究未来的发展方向。

### 1.1 多核处理器实际限制

理想多核处理器模型假设处理器具有连续调节电压和频率、没有转换开销,但是由于物理限制,具备 DVFS 和 DPM 技术的实际多核处理器总是存在状态切换开销、频率更改开销、离散性的电压/频率调节级别以及多种可用睡眠状态<sup>[9]</sup>。对于许多硬实时应用来说,是否考虑开销、考虑哪些开销将对节能实时调度产生很大的影响,忽视这些细节可能会得到次最优甚至是

无效的结果。例如,在 70 纳米技术中 Transmeta 处理器具有 483 微焦的能耗开销和至少 2 毫秒的时间开销<sup>[10]</sup>。如果状态切换的能耗开销大于能耗节余,显然,把处理器转入睡眠状态可能不会节能<sup>[10]</sup>。此外,不同类型的真实多核处理器具备各自不同的实际物理限制,如频率更改开销和状态切换开销等多维限制,所以针对同构和异构多核处理器平台下不同开销模型的节能调节策略显然是不同的。

### 1.2 抢占调度实际限制

在实时调度算法中,通常认为,抢占调度是一个先决条件,可以提高任务集的可调度性,以满足实时系统的时间约束。但是,在许多情况下,完全抢占式调度会产生许多不必要的抢占;另外,虽然一些任务可以在执行过程中任意时刻被抢占,但其他任务却由于事务性操作如中断而不应该被抢占;最后,任意抢占会因为高速缓存未命中(Cache miss)等原因而引起显著的运行时代价,可能会导致任务执行时间的波动,以至于降低系统的可预测性和影响能耗。特别是,在每次抢占发生时需要考虑至少四种不同类型的代价:调度代价、流水线代价、Cache 相关代价和总线相关代价<sup>[11]</sup>。综上,由于抢占策略决定了对抢占限制的程度,不同抢占策略则会产生不同的调度算法。因此,分别基于不同抢占调度的多核节能实时调度算法会具有不同的节能效果。

### 1.3 并行任务模型实际限制

多核系统通常由多个不同的运算程序相互协同来完成复杂的功能,对多种不同信息的处理不再是相互独立的,而是在一定程度上可以并行执行同时又具有某种顺序依赖关系的运算任务。这使得以往关注串行任务或简单并行任务模型的节能实时调度技术遇到挑战,需要研究更多新的并行任务模型。此外,具备并行和顺序依赖关系的不同并行任务模型具有不同的属性,则相应的调度算法也不同,这会影响到实时调度下的节能方法,还会扩展了节能实时调度算法的应用范围。

## 2 基于多维限制的多核节能实时调度研究

实时多任务节能调度以实时任务集作为调度对象,在实时调度中根据任务集的实际执行过程进行合理的动态电压频率或功耗模式调节。为了确保时限约束,实时多任务节能调度算法通常不改变实时任务的优先

级, 只在已有实时调度算法中, 加入松弛时间分析和频率调节的功能. 和传统非节能实时调度方法相比, 早期的实时多任务节能调度实质上是在 DVFS 硬件节能技术下的针对实时多任务集的单核处理器资源分配问题<sup>[3]</sup>.

在多核处理器实时系统中, 根据任务分配策略不同, 节能实时调度研究可以分为两种基本方法: 全局法和划分法. 这里, 划分法是指每个任务只分给一个处理器, 不允许迁移发生; 而全局法是指任务或任务实例可以迁移和执行在不同处理器上. 另外还可以根据系统运行平台的不同, 分为同构多核系统和异构多核系统节能实时调度. 进一步根据实时任务模型不同, 可分为周期任务、偶发任务、基于帧任务、基于 DAG 任务等节能实时调度算法. 此外根据任务调度顺序产生的时机和方式不同, 可分为各种离线或在线节能调度算法. 最后根据节能策略产生的时机和方式不同, 还可分为静态或动态节能实时调度算法. 静态方法是指处理器在调度执行实时任务过程中, 处理器电压和频率值事先确定, 在该任务运行时不会改变. 而动态方法是指处理器电压和频率值可以在该任务运行中改变.

已有的多核系统节能实时调度技术研究成果至今还缺乏系统的分类和总结. 下面从应对系统的多核处理器实际限制、抢占调度实际限制和并行任务实际限制三个方面分析目前的研究现状, 进行分类总结, 给出未来研究方向的展望.

## 2.1 基于处理器实际开销模型的同构/异构多核节能实时调度

主要由切换活动产生的动态功耗和主要由泄露电流产生的静态功耗是 CMOS 处理器中两个主要的功耗来源. 通过使用 DVFS 技术执行在低频率下, 是绝大多数时候静态功耗忽略不计最好的节能方法. 过去十几年, 同构多核处理器系统中节能实时调度已经被广泛地研究, Chen 等<sup>[8]</sup>对此进行过全面的回顾. 当考虑具有每个核 DVFS 的实时嵌入式系统时, 许多研究成果<sup>[12]</sup>已经被提出. 文献[12]针对周期任务模型, 提出了一种最优的多处理器节能实时调度算法, 这种最优性未考虑转换开销. 当考虑只有一个全局 DVFS 的多核处理器系统时, Yang 等人<sup>[13]</sup>提出了一种基于帧实时任务的节能调度算法, 而对静态功耗忽略不计. 该算法高度严格, 它很难扩展到处理具有不可忽视静态

功耗的系统中或周期性实时任务. 文献[14]放松了文献[13]中假设, 针对周期性实时任务集, 考虑处理器转入低功耗空闲模式具有不可忽视的能耗和时间开销情况. Seo 等人<sup>[14]</sup>提出动态平衡多核的任务负载, 调整了活跃核的数量, 以最优化执行的功耗且减少低负载时泄露功耗. 对于周期性实时任务集, 文献[15]根据能耗最坏情况下近似因子分析而提出了一种近似方法. 综上所述, 为最小化能耗而得到最优的任务划分问题是一个 NP 难问题<sup>[13]</sup>.

相比同构多核处理器系统, 当前关于异构多核处理器系统节能实时调度方面的研究仍在发展中. Yu 等人<sup>[16]</sup>提出了一种实时系统离线任务分配方法, 该方法考虑了具有动态电压频率调节功能的异构处理单元. Luo 等人<sup>[17]</sup>考虑了有向无环图任务模型, 针对异构分布式系统提出了列表调度策略. Chen 等人<sup>[18]</sup>考虑了一种只有两种类型异构处理器情况, 并提出了一个基于任务执行时间在不同的处理器类型上比率的多项式时间近似方法. 虽然 DVFS 技术具有优势, 但是以上研究成果均忽略了静态功耗.

对于静态功耗不可忽视的系统, 以低于关键频率的频率执行任务可能消耗更多的能耗. 这是因为当运行在较低的频率时静态能耗会随着执行时间的延长而变大, 它可能占能耗的主要部分. 这推动了 DVFS 和 DPM 之间的结合, 以至于处理器核在完成他们的工作负载后会降低频率然后关闭. Yang 等人<sup>[19]</sup>提出了一种基于动态规划的近似算法, 当处理器类型的数量是一个小常数时, 该方法提供了多项式时间解. 然而, 在一般情况下, 当多处理器类型的数量限制放宽时, 该方法具有指数时间和空间复杂度. Chen 等人<sup>[20]</sup>针对异构平台上周期性实时任务提出了一种任务分配算法. 在前面的步骤中分配的任务, 还可以通过已知的启发式方法被重新分配, 如首次适应 FF、最好适应 BF、最坏适应 WF 或最后适应 LF 方法. 以上方法<sup>[19][20]</sup>均假设该系统的静态功耗是一个常数, 且因较大的睡眠状态切换开销而不能被减少. 但是对于现代多核处理器而言, 这个假设并不成立, 因为现代多核处理器可以包含多个睡眠状态, 且进入和离开睡眠状态需要多种状态切换开销, 据此可以降低系统的静态功耗. 文献[9]针对这样的异构多核平台, 提出了一种节能任务映射方法以减少整体系统能耗. 该方法基于划分调度的思想, 在每个 CPU 上采用 EARTH<sup>[21]</sup>节能调度方法, 综合

考虑了任务属性对所分配的处理器上活跃功耗和静态功耗的影响。

无论何时系统更改处理器电压和频率设置,以及关闭和唤醒处理器,引起一些时间和能耗开销是很自然的。显然,未考虑处理器开销的系统模型会误导节能调度策略以至于阻止获得更大的能耗节余。综上,是否考虑开销、考虑哪些开销将对节能实时调度产生很大的影响。当不考虑处理器具有的频率更改开销和状态切换开销等物理限制时,多核节能实时调度问题通常可以归结为三个问题的求解:

(1)分配问题,即实时任务应该在哪一个处理器核上执行?

(2)优先级问题,即每个任务应在什么时刻、以相对其他任务实例的何种顺序来执行?

(3)速度调节问题,即调度到给定处理器上执行的就绪任务应该以何种速度来执行?

但当多核处理器系统具有不可忽视的频率更改开销和状态切换开销时,多核节能实时调度问题还可以归结出一个问题的求解:

(4)处理器开销问题,即基于频率更改开销来设定何时调节活跃状态的处理器速度,基于状态切换开销来设定何时将处理器转入睡眠状态?

对于越来越不容忽视的开销问题,许多研究者已经开始关注具有处理器实际开销限制的节能实时调度问题。Xu 等<sup>[22]</sup>根据工作负载确定活跃处理器的个数,并结合实际处理器具有的状态切换开销提出节能实时调度算法。Langen 等<sup>[23]</sup>则进一步考虑具有离散速度的系统,针对有向无环图任务模型,提出启发式节能实时调度算法。而 Chen 等<sup>[10]</sup>通过说明关键速度的非最优性以及负载均衡的非最优性,针对基于帧的任务模型和周期任务模型,提出了一种考虑状态切换开销的节能实时调度算法。Huang 等<sup>[24]</sup>基于划分调度方法,提出一种考虑实际处理器状态切换开销的在线节能实时调度算法。

表 1 列出不同处理器实际开销模型下主要节能实时调度算法比较结果。由此可见,我们发现没有一种算法能够适合所有的情况和处理器实际限制约束,各种限制约束的节能调度算法都有其不同的适用条件,例如有时没有考虑实际开销的节能实时调度甚至比没有节能调度的能耗更大。

表 1 不同实际开销模型下主要节能实时调度算法比较

处理器实际开销模型	文献	系统运行平台	任务模型	任务调度策略
未考虑	[12]	同构多核	周期任务	全局法
	[13][15]	同构多核	基于帧任务或周期任务	划分法
	[16][18][19][20]	异构多处理器	周期任务	
	[17]	异构多处理器	有向无环图任务	
	[9]	异构多核	周期任务	
考虑	[10][22]	同构多处理器	基于帧任务或周期任务	划分法
	[23]	同构多处理器	有向无环图任务	
	[14][24]	同构多核	周期任务	

## 2.2 基于有限抢占调度模型的开销敏感多核节能实时调度

随着多核平台在嵌入式实时应用中越来越多的使用,绝大多数多核实时调度算法都是假设没有抢占开销的理想情况,通常采用完全抢占式调度。当多核处理器系统存在大量非抢占任务和不可忽视的运行时报占开销时,多核节能实时调度问题在上一小节的四个问题基础上还可以归结出第五个问题的求解:

(5)抢占问题,即每个任务应在什么时刻、以运行时能否被其他高优先级任务所抢占的何种方式来执行?

为了减少抢占带来的运行时开销和保持任务集的可调度性,针对实时单处理器系统,一些基本的有限抢占调度算法已经被提出:抢占阈值调度<sup>[25]</sup>、延迟抢占调度<sup>[26]</sup>和固定抢占点调度<sup>[27]</sup>。这里,每种有限抢占调度算法都有其优缺点。首先,抢占阈值调度机制有简单直观的接口,实现开销小。但是,抢占代价不容易评测,因为每次抢占位置和每个任务的总抢占次数无法离线确定。其次,使用延迟抢占调度,每个任务的抢占次数更好评测,但每次抢占的位置仍不能离线确定。最后,固定抢占点算法是一种产生较少抢占和较高可调度性的方法,可以从任务代码中固定和确定抢占次数以及抢占点位置,但是固定抢占点算法需要在程序中添加显式的抢占点,未来获取原程序的可移植性仍是一个挑战。由于实时任务的事务性操作,虽然任务可分为抢占或非抢占任务,但是时限保证通常

只针对由所有任务是抢占的或非抢占的构成的实时任务集。综上可知,对于由抢占或者非抢占任务混合构成的一般性实时任务集,虽然有一些工作<sup>[28]</sup>已经提出一种多核实时可调度性分析框架,可以保证给定任务集的时间需求,但迄今为止还未见相关的多核节能实时调度。文献<sup>[29]</sup>面向单处理器平台提出了基于有限抢占框架的节能实时调度方法,但在每次任务到达时都要分析低优先级任务来“阻止”高优先级任务,使得该方法会带来较大的时间和能耗开销。

为了将有限抢占调度方法引入到多核系统的节能实时调度算法中,本文提出可以考虑抢占阈值调度、延迟抢占调度和固定抢占点调度策略,并关注多核节能实时调度中下列问题:

问题 1: 如何保证由抢占任务和非抢占任务构成实时任务集的时间要求?

问题 2: 既然可以通过限制一些可抢占任务以非抢占方式执行来提升可调度性,那么在如何才能为每个可抢占任务发现能耗最优的不可抢占“分配”? 如果它是 NP 问题,那么如何发现次最优的解?

本文提出可以针对每种有限抢占调度策略,通过分析给定抢占或非抢占任务的执行如何在给定的动态优先级或静态优先级策略下影响其他抢占或非抢占任务,将为抢占任务和非抢占任务开发一个响应时间分析框架。虽然一个抢占任务的响应时间计算需要知道何时它的执行过程中最后一个单元才能完成,但是一个非抢占任务的响应时间计算只需要知道它的执行过程中第一个单元何时开始。一旦开始,非抢占任务的剩余执行过程将没有中断地运行到完成状态。通过使用这些属性和指定的优先级策略,研究者可以解决了问题 1,为使用有限抢占调度策略的任意调度算法开发了一个可调度性分析框架。

对于问题 2,本文提出可以首先分析如果一个给定可抢占任务不允许被抢占时,一个抢占或非抢占任务的执行对其他抢占或非抢占任务的影响将如何变化。根据结果,虽然可以得到一种“最优”地控制每个可抢占任务发生抢占的算法,但如果再结合能耗最小化目标,这将是 NP 问题。因此,研究者可以针对有限抢占调度策略,提出多种启发式的开销敏感多核节能实时调度算法。

### 2.3 基于复杂并行任务模型的开销敏感多核节能实时调度

多核处理器的高性能正是通过多线程并行化来获

得的,而并行处理的高性能又极大地依赖于并行任务的调度。因此多核系统中实时调度应关注并行任务模型的实时调度,重点考虑如何将多个线程映射到多个处理器核或者功能单元,目标是既满足任务的实时性要求又获得好的整体性能。但是多核系统中实时调度研究大多假设单个线程的串行任务模型或简化的并行任务模型<sup>[30]</sup>,已有的研究成果或是基于启发式和仿真测试<sup>[31]</sup>,或是将并行调度思想与经典实时调度算法简单结合起来<sup>[30]</sup>,还没有考虑更复杂的并行处理情况,也没有提出成熟的可调度性测试方法。而有向无环图 DAG 并行任务模型,与周期任务模型相比更加实用,能直接表达任务间的优先约束及通信开销等,在多核处理器并行任务调度研究中广为使用<sup>[32]</sup>。针对不同实时应用领域,研究人员陆续提出新的任务模型,增强表达实际应用的能力,不断提高系统的可调度性和调度成功率。例如,多帧任务模型用于对执行时间有变化的任务进行建模;循环实时任务模型将 DAG 图任务模型扩展为具备条件分支表达能力;基于流的任务模型是一种面向流数据的任务模型;偶发 DAG 模型<sup>[33]</sup>中每个循环任务被建模为一个有向无环图,它可以表示在同构多处理器平台上执行的循环且有顺序约束任务;场景敏感数据流图 SADF 模型<sup>[34]</sup>是由多个应用场景(又称为模式)和一个有限状态机 FSM 构成,用于表示运行在嵌入式多处理器平台上流应用。

近年来,基于 DAG 任务模型的多核节能实时调度相关研究,则着重针对实时嵌入式系统的各种流应用。这类应用中大量的单个循环任务将被允许独占执行在指定的处理器核或处理器核组上。这样的执行环境需要更多的任务模型来表示。为了表示在多核处理器平台上执行的循环且有顺序约束任务,偶发 DAG 并行任务模型<sup>[33]</sup>已经被提出。在该模型中,一个循环的偶发任务被表示为一个有向无环图 DAG、周期和相对时限,它可以表示在同构多处理器平台上执行的循环且有顺序约束任务。DAG 的每个顶点表示一个串行作业(又称为子任务),同时 DAG 的边表示这些子任务之间的顺序关系。DAG 的所有子任务同时释放,且需要在指定的相对时限内完成执行。如果确定这样一个循环任务被调度是否总是可以在所指定的多个处理器核上满足所有时限且能耗最小化,那么该节能调度问题显然是 NP 难题,但可以考虑有效的近似求解。

虽然文献<sup>[35]</sup>也提出了一种贪婪方法来解决基于

SADF 图模型的多处理器节能实时调度问题, 但该方法易于遭受状态空间爆炸, 增加分析时间, 降低效能. 进而, 文献[34]基于 SADF 图模型提出了一种考虑电压频率更改开销的多处理器频率分配算法, 该算法能够在减少能耗的同时满足吞吐率约束. 相比文献[35], 该算法还提高选择合适多处理器电压和频率调节点的分析效率, 但是该算法并没有考虑多核处理器平台特点以及其他多维限制. 此外, 场景敏感数据流图实时任务在多核系统中最优节能实时调度仍是 NP 难题. 因此, 研究者可以引入多种启发式思想, 结合多核处理器实际开销模型, 对基于偶发 DAG 并行任务模型和基于 SADF 图并行任务模型的节能实时调度方法进行研究.

最后, 对于实时流应用程序的节能调度, 通常需要根据任务的最坏执行时间进行分析. 当前程序的最坏时间分析方法一般有两种: 统计执行和静态估计. 统计执行方法给程序提供各种输入, 测试程序的执行时间, 大致找到最坏执行时间; 静态估计方法采用程序分析技术分析程序的执行时间, 可以正确给出程序的最坏执行时间, 但是对于多核处理器系统, 这种方法往往给出的是很悲观的最坏执行时间. 因此, 研究者可以利用多个并行任务执行之间所产生的动态松弛时间, 在任务调度之前回收松弛时间并做出电压频率调节策略, 提出开销敏感的多核节能调度设计.

#### 2.4 未来发展

对多核系统进行节能实时调度时面临的处理器实际限制、抢占调度实际限制和并行任务模型实际限制等问题, 可以通过任务集分解、实时调度以及节能策略调节等方式综合运用来解决, 但是由于多核硬件平台的差异性以及任务负载与开销间的耦合性较强等原因导致解决问题的重点在于如何实现多个维度约束的合理折中.

对多核系统中单一维度限制进行节能实时调度的研究已有较多成果, 分别在提高可调度性、降低能耗以及减少开销等单个方面取得了较好的效果. 现阶段研究比较多的是针对多核处理器实际开销模型进行节能实时调度的技术, 未来更多的研究可能会采用有限抢占调度模型已经作为任务调度策略来更好地减少抢占带来的运行时开销, 以实现更好的节能效果. 面向偶发有向无环图模型和场景敏感数据流图等复杂并行任务模型的多核节能实时调度技术最近才引起人们的

关注, 但是随着多核嵌入式流媒体应用越来越多, 会产生许多特殊的节能需求, 对于这些节能需求, 沿用传统的单维方法可能效果不理想, 而采用融合多维的方法则会产生较好的效果, 未来对该技术的研究也会更加受到人们的重视.

### 3 结语

本文由同构多核到异构多核, 由处理器频率更改开销模型到处理器状态切换开销模型, 由完全抢占调度模型到有限抢占调度模型, 由基于偶发有向无环图到基于场景敏感数据流图并行任务模型, 开展了多方面的研究和探索. 综上, 本文研究解决如何基于多核系统多维限制实现节能实时调度的问题, 为今后多核系统在实时嵌入式领域的推广应用打下坚实的理论基础.

#### 参考文献

- 1 Saidani T, Piskorski S, Lacassagne L, Bouaziz S. Parallelization schemes for memory optimization on the cell processor: a case study of image processing algorithm. In: Pierfrancesco F, ed. The Workshop on Memory Performance: Dealing with Applications. New York. ACM. 2007. 9–16.
- 2 Hirata K, Goodacre J. Arm mpcore: the streamlined and scalable arm11 processor core. In: Onodera H, ed. Asia and South Pacific Design Automation Conference 2007, Yokohama. IEEE. 2007. 747–748.
- 3 雷霆. 基于动态电压调整的实时节能调度方法研究[博士学位论文]. 合肥: 中国科学技术大学, 2005.
- 4 易会战. 低功耗技术研究—体系结构和编译优化[博士学位论文]. 长沙: 国防科学技术大学, 2006.
- 5 Jejurikar R, Pereira C, Gupta R. Leakage aware dynamic voltage scaling for real-time embedded systems. Malik S, eds. Proc. of the 41st Annual Design Automation Conference. San Diego. ACM. 2004. 275–280.
- 6 Chandrakasan A, Sheng S, Brodersen RW. Low-power CMOS digital design. IEEE Journal of Solid-State Circuit, 1992, 27(4): 473–484.
- 7 Rele S, Pande S, Onder S, Gupta R. Optimizing static power dissipation by functional units in superscalar processors. Horspool RN, Lecture Notes in Computer Science 2304. Berlin. Springer Heidelberg. 2002. 85–100.

- 8 Chen JJ, Kuo CF. Energy-efficient scheduling for real-time systems on dynamic voltage scaling (DVS) platforms. In: Ceballos S, ed. 13th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications. Daegu. IEEE. 2007.28–38.
- 9 Awan MA, Petters SM. Energy-aware partitioning of tasks onto a heterogeneous multi-core platform. In: Goddard S, ed. Real-Time and Embedded Technology and Applications Symposium. Philadelphia. IEEE. 2013. 205–214.
- 10 Chen JJ, Thiele L. Energy-efficient scheduling on homogeneous multiprocessor platforms. In: Shin SY, ed. ACM Symposium on Applied Computing. New York. ACM. 2010. 542–549.
- 11 Buttazzo G, Bertogna M, Yao G. Limited preemptive scheduling for real-time systems: a survey. IEEE Transactions on Industrial Informatics, 2013, 9(1): 3–15.
- 12 Funaoka K, Kato S, Yamasaki N. Energy-efficient optimal real-time scheduling on multiprocessors. In: Lisa OC, ed. The 11th IEEE Symposium on Object Oriented Real-Time Distributed Computing. Orlando. IEEE. 2008. 23–30.
- 13 Yang CY, Chen JJ, Kuo TW. An approximation algorithm for energy-efficient scheduling on a chip multiprocessor. Conference on Design, Automation, and Test in Europe (DATE). Munich. IEEE. 2005. 468–473.
- 14 Seo E, Jeong J, Park S, Lee J. Energy efficient scheduling of real-time tasks on multicore processors. IEEE Trans. on Parallel and Distributed Systems, 2008, 19(11): 1540–1552.
- 15 Pagani S, Chen JJ. Energy efficiency analysis for the single frequency approximation (SFA) scheme. ACM Trans. on Embedded Computing Systems, 2014, 13(5s): 158:1–25.
- 16 Yu Y, Prasanna V. Power-aware resource allocation for independent tasks in heterogeneous real-time systems. In: Martin DC, ed. Ninth International Conference on Parallel and Distributed Systems. IEEE. 2002. 341–348.
- 17 Luo J, Jha NK. Static and dynamic variable voltage scheduling algorithms for real-time heterogeneous distributed embedded systems. In: Sherlekar SD, ed. Proc. of the 2002 Asia and South Pacific Design Automation Conference. Bangalore. IEEE. 2002. 719–728.
- 18 Chen JJ, Thiele L. Energy-efficient task partition for periodic real-time tasks on platforms with dual processing elements. In: Werner B, ed. 14th IEEE International Conference on Parallel and Distributed Systems. Melbourne. IEEE. 2008. 161–168.
- 19 Yang CY, Chen JJ, Kuo TW, Thiele L. An approximation scheme for energy-efficient scheduling of real-time tasks in heterogeneous multiprocessor systems. In: Benini L, ed. Proc. of the Conference on Design, Automation and Test in Europe. Belgium. IEEE. 2009. 694–699.
- 20 Chen JJ, Schranzhofer A, Thiele L. Energy minimization for periodic real-time tasks on heterogeneous processing units. In: Alessandro M, ed. IEEE International Symposium on Parallel & Distributed Processing. Rome. IEEE. 2009. 1–12.
- 21 Awan MA, Petters SM. Enhanced race-to-halt: A leakage-aware energy management approach for dynamic priority systems. In: Werner B, ed. 23rd Euromicro Conference on Real-Time Systems. Porto. IEEE. 2011. 92–101.
- 22 Xu R, Zhu D, Melhem R, Mossé D. Energy-efficient policies for embedded clusters. In: Frenger P, ed. ACM SIGPLAN/SIGBED Conference on Languages, Compilers, and Tools for Embedded Systems, New York. ACM. 2005. 1–10.
- 23 Langen PJD, Juurlink BHH. Leakage-aware multiprocessor scheduling for low power. In: Prasanna VK, ed. Parallel and Distributed Processing Symposium. Rhodes Island. IEEE, 2006. 80–87.
- 24 Huang H, Xia F, Wang J, Lei S, Wu G. Leakage-aware reallocation for periodic real-time tasks on multicore processors. In: Stojmenovic I, ed. Fifth International Conference on Frontier of Computer Science and Technology. Changchun. IEEE. 2010. 85–91.
- 25 Wang Y, Saksena M. Scheduling fixed-priority tasks with preemption threshold. In: Titsworth FM, ed. The 6th IEEE Int. Conference on Real-Time Computing Systems and Applications, Hong Kong. IEEE. 1999. 328–335.
- 26 Baruah S. The limited-preemption uniprocessor scheduling of sporadic task systems. In: Tovar E, ed. The 17th Euromicro Conf. on Real-Time Systems (ECRTS'05), Balearic Islands. IEEE. 2005. 137–144.
- 27 Burns A. Preemptive priority based scheduling: An appropriate engineering approach. Advances in Real-Time Systems, London. Prentice Hall. 1994. 225–248.
- 28 Lee J, Shin KG. Controlling preemption for better schedulability in multi-core systems. In: Kato S, ed.

- Real-Time Systems Symposium, San Juan. IEEE, 2012. 29–38.
- 29 Bambagini M, Bertogna M, Marinoni M, Buttazzo GC. An energy-aware algorithm exploiting limited preemptive scheduling under fixed priorities. In: Eduardo T, ed. The 8th IEEE International Symposium on Industrial Embedded Systems. Porto. IEEE. 2013. 3–12.
- 30 Kato S, Ishikawa Y. Gang EDF scheduling of parallel task systems. In: Baker TP, ed. The 30th IEEE Real-Time Systems Symposium. Washington. IEEE. 2009. 459–468.
- 31 Collette S, Cucu L, Goossens J. Integrating job parallelism in real-time scheduling theory. *Information Processing Letters*, 2008, 106: 180–187.
- 32 Becehi M, Crowley P. Dynamic thread assignment on heterogeneous multiprocessor architectures. In: Alderighi M, ed, eds. The 3rd Conf on Computing Frontiers. New York. ACM. 2006. 29–40.
- 33 Baruah S, Bonifaci V, Marchetti-Spaccamela A, Stougie L, In: Wiese A. A generalized parallel task model for recurrent real-time processes. In: Kato S, ed. Real-Time Systems Symposium. San Juan. IEEE. 2012. 63–72.
- 34 Damavandpeyma M, Stuijk S, Basten T, Geilen M, Corporaal H. Throughput-constrained DVFS for scenario-Aware dataflow graphs. In: Goddard S, ed. Real-Time and Embedded Technology and Applications Symposium. Philadelphia. IEEE. 2013. 175–184.
- 35 Zimmermann J, Bringmann O, Rosenstiel W. Analysis of multidomain scenarios for optimized dynamic power management strategies. In: Preas K, ed. Design, Automation & Test in Europe Conference & Exhibition. Dresden. IEEE. 2012. 862–865.