

doi:10.3969/j.issn.1673-9833.2012.06.013

基于粒子群优化的多核处理器系统节能调度算法

赵 阳, 满君丰, 龙永新, 郭德彪, 易 旺

(湖南工业大学 计算机与通信学院, 湖南 株洲 412007)

摘 要: 针对嵌入式多核系统中的实时性和能耗问题, 提出了基于粒子群优化的多核处理器系统节能调度算法。通过对多核处理器系统任务调度和能量消耗的分析, 建立了新的编码策略和相应的目标函数, 将任务划分、任务调度及电压选择3个过程整合到粒子群算法的一个单迭代寻优循环中, 并兼顾系统的实时性, 将问题转化为有约束的粒子群优化问题, 利用可行性规则的约束处理技术, 形成基于约束粒子群优化的整体节能调度算法。实验结果表明: 本文算法能以较快的速度收敛于使系统能耗更少的调度解。

关键词: 嵌入式多核处理器系统; 实时性; 能耗; 粒子群优化; 约束优化

中图分类号: TP303

文献标志码: A

文章编号: 1673-9833(2012)06-0051-04

Energy-Saving Scheduling Algorithm of Multi-Core Processor System Based on Particle Swarm Optimization

Zhao Yang, Man Junfeng, Long Yongxin, Guo Debiao, Yi Wang

(School of Computer and Communication, Hunan University of Technology, Zhuzhou Hunan 412007, China)

Abstract: In order to solve the problems of real-time and energy consumption for embedded multi-core system, an energy-saving scheduling algorithm based on particle swarm optimization is proposed. According to the analysis of task scheduling and energy consumption for multi-core processor system, a new coding strategy and the corresponding target function are established. Three processes of task partition, task scheduling and voltage selection of energy-efficient scheduling for multi-core processor system are integrated into a simple iterative optimization cycle of the particle swarm algorithm, and compatible with the system real-time, the original issues are transformed into new ones of constrained particle swarm optimization, and a constraint processing technology based on feasibility rules is introduced. Thus a new overall energy-saving scheduling algorithm is gained based on constrained particle swarm optimization. The simulated results show that the proposed algorithm can rapidly converge to the optimal scheduling solution of less energy consumption.

Keywords: embedded multi-core processor; real-time; energy consumption; particle swarm optimization; constrained optimization

0 引言

近年来, 日益突出的能耗问题促使处理器的体系结构由单核向多核转变。多核处理器比单核处理器具有性能和效率优势, 其是将任务调度策略与低

功耗技术相结合来有效控制其运行速率及功耗, 是降低系统能耗的有效途径。然而, 目前的低功耗任务调度技术大多针对单核系统, 难以充分达到对多核处理器能耗优化的效果。

多核处理器系统的低功耗调度问题是一个典型

收稿日期: 2012-10-13

作者简介: 赵 阳 (1987-), 男, 河南周口人, 湖南工业大学硕士生, 主要研究方向为嵌入式实时系统,

E-mail: yang2010314159@163.com

的 NP 完全问题。国内外学者对此展开了大量的研究,着力寻找一种次优的启发式任务执行方法。目前,针对该问题的算法大多采用两阶段^[1]或三阶段^[2]的启发式策略。Vida Kianzad 等人^[3]提出了一种基于遗传算法的整体节能调度框架 CASPER (combined assignment, scheduling, and power-management), 其利用遗传算法将任务的分配、调度和 DVS (dynamic voltage scaling) 策略整合到单一的循环中进行迭代寻优, 得到最优的调度, 实验结果表明: 整体的节能调度框架平均节能效果优于“分开-结合”的算法。Chang Pochun 等人^[4]针对异构多核处理器系统提出了基于蚁群算法的整体寻优算法, 其节能效果优于 CASPER, 但算法执行时时间较长。因此, 本文将粒子群优化算法引入多核处理器系统的低功耗调度研究中, 采用一种新的编码策略, 将任务集的划分、调度及电压选择一起考虑, 同时为了保证任务集在不违背截止期的前提下达到节能的效果, 将问题转化为有约束的粒子群寻优过程, 利用基于可行性规则的粒子群优化 (particle swarm optimization, PSO) 算法以较快的速度寻得最优解。

1 问题描述

1.1 系统模型

针对基于同构的片上多核处理器系统, 假设多核处理器系统由同构的 n 个处理单元 $P = \{P_1, P_2, \dots, P_n\}$ 组成, 某一实时的嵌入式应用由 m 个子任务 $T = \{t_1, t_2, \dots, t_m\}$ 组成, 且这 m 个子任务满足一定的约束关系。该约束关系可用有向无环图 $G(T, E, W, D)$ 表示, 其中, $E = \{e_{ij} = (t_i, t_j) | t_i, t_j \in T\}$ 是有向边的集合, (t_i, t_j) 表示任务 t_i 是 t_j 的直接前驱, 则 t_j 是 t_i 的直接后继, t_i 必须在 t_i 执行完毕后才能开始执行, 元素 e_{ij} 的值为子任务 t_i 和 t_j 之间的通信代价; $W = \{w_1, w_2, \dots, w_m\}$ 表示任务节点的权重, w_i 是任务 t_i 的估计运行时间; D 表示应用的截止时间, 任何一个给定的有向无环图的截止期都能根据文献[5]中的方法计算得到。集合 $S = \{s_{ij} \in (1, 0) | i \in [1, n], j \in [1, m]\}$ 表示处理器核与任务之间的对应关系, 当 $s_{ij} = 1$ 时, 表示任务 t_j 在处理器核 P_i 上运行。任务 t_j 的开始时间 $B(t_j)$ 和结束时间 $F(t_j)$ 可由以下公式求得:

$$B(t_j) = \text{Max}_{t_k \in \text{Pre}(t_j)} \{F(t_k) + (s_{ik} - s_{ij})e_{kj}\}, \quad (1)$$

$$F(t_j) = \eta_j w_j + B(t_j). \quad (2)$$

式中: $\text{Pre}(t_j)$ 表示任务 t_j 的所有直接前驱的集合;

$\eta_j \in [1, k]$ 是执行 t_j 时的电压级别。

系统完成整个应用的总时间可表示为

$$F_t = \text{Max}_{j=1}^m [F(t_j)]. \quad (3)$$

1.2 能耗模型

大部分多核处理器系统采用 CMOS 集成电路, 该类电路主要考虑电容充放电过程所引起的动态功耗。系统的动态功耗 P_{dyn} 与供电电压 V_{dd} 的关系是

$$P_{\text{dyn}} = C_L V_{\text{dd}}^2 f, \quad (4)$$

式中: C_L 表示系统的负载电容;

f 为处理单元的工作频率, 即

$$f = k(V_{\text{dd}} - V_t)^2 / V_{\text{dd}}, \quad (5)$$

其中, k 为常数, V_t 是电路的门限电压, 一般情况下有 $V_{\text{dd}} \gg V_t$, 因此, 处理单元的频率和供电电压间基本上呈线性关系。

某个任务执行所需时间 t 可表示为 N_c/f , 其中, N_c 为执行某一任务所需的时钟周期数, 其不能被 DVS 改变, 因此, 可得到每个任务在不同电压下的执行时间和功耗^[6]。执行任务所消耗的能量为

$$E = P_{\text{dyn}} t = C_L V_{\text{dd}}^2 N_c. \quad (6)$$

从式(6)可以看出, 降低处理单元的供电电压可平方级地降低系统的动态功耗, 但又会线性地增加任务的执行时间。

基于动态电压调整的节能调度算法的基本思想是, 通过在运行时刻动态地调节处理器核的执行电压, 避免处理器核总是运行在最高电压下, 且保证任务执行时间不超过规定时限的前提下, 有效地减少任务能耗。本系统经电压调节后, 其能耗可表示为

$$E' = \sum_{i=1}^m E'_i = C_L \sum_{i=1}^m V_{\text{dd},i}^2 N_{c,i}, \quad (7)$$

式中: $V_{\text{dd},i}$ 是任务 t_i 经调整后的电压;

$N_{c,i}$ 是执行任务 t_i 所需要的时钟周期数。

2 粒子群优化算法

2.1 算法思想

粒子群优化算法是由美国心理学家 Kennedy 等人^[7]于 1995 年提出的一种基于群体智能的随机优化算法。该算法的基本思想是, 先在问题的可行解空间和速度空间初始化粒子群, 即确定每个粒子的初始位置和速度, 其中位置用于表征问题解, 例如, d 维搜索空间中的第 i 个粒子的位置矢量和速度矢量分别表示为 $X_i = [x_{i,1}, x_{i,2}, \dots, x_{i,d}]$ 和 $V_i = [v_{i,1}, v_{i,2}, \dots, v_{i,d}]$; 通过评价各粒子的目标函数, 确定迭代至 t 代每个粒子所经过的最佳位置矢量 $P_i = [p_{i,1}, p_{i,2}, \dots, p_{i,d}]$ 以及整个群体所发现的最佳位置矢量 $G = [g_1, g_2, \dots, g_d]$; 找到

粒子和群体最佳位置矢量时,粒子更新自己的速度矢量和位置矢量^[8],即:

$$V_{i,j}(t+1) = \omega V_{i,j}(t) + c_1 r_1 [P_{i,j}(t) - X_{i,j}(t)] + c_2 r_2 [G_j - X_{i,j}(t)], \quad (8)$$

$$X_{i,j}(t+1) = X_{i,j}(t) + V_{i,j}(t+1), i=1,2,\dots,m, j=1,2,\dots,d. \quad (9)$$

式中: ω 为惯性权因子; c_1 和 c_2 为学习因子,一般 $c_1=c_2=2$ ^[8]; r_1 和 r_2 为0~1之间的随机数; m 为种群大小。

2.2 问题编码

粒子群算法解决特定问题的首要任务是确定决策解的编码方式,这也是构造粒子运动规则的前提。本文通过引入一种新的粒子群编码机制使得该问题的决策解同时包含任务的执行单元划分和电压级别选择。假设任务集中有 m 个任务,多核处理器系统有 n 个处理单元,且处理器的运行电压可离散化为 k 个级别,将第 i 个粒子位置矢量和速度矢量分别描述为 $X_i=(x_{i,1}, x_{i,2}, \dots, x_{i,2m})$, $V_i=(v_{i,1}, v_{i,2}, \dots, v_{i,2m})$,前 m 维向量表征任务至处理单元的划分策略,后 m 维向量表征每个任务在其对应的处理单元上的电压级别。例如,当 $m=5$ 时, $x_{i,2}=3$ 表示任务 t_2 被分配至处理单元 P_3 上执行; $x_{i,7}=2$ 表示任务 t_2 以电压级别2运行。粒子的速度矢量必须满足 $V_{i,j} \in \{-V_{\max}, V_{\max}\}$,位置矢量必须满足:

$$\begin{cases} x_{i,j} \in [1, n], 1 \leq j \leq m; \\ x_{i,j} \in [1, k], m+1 \leq j \leq 2m. \end{cases} \quad (10)$$

本编码方案要求位置矢量的每一维都是自然数,且将式(9)中的 $X_{i,2}(t+1)$ 做向上取整运算,即

$$X_{i,j}(t+1) = \lceil X_{i,j}(t) + V_{i,j}(t+1) \rceil. \quad (11)$$

由于子任务之间存在约束关系,其必须按照一定的执行顺序执行,因此,本文引入高度值来保证其在给定的处理器核上的执行顺序。对任意任务 $t_i \in T$,其高度值 $h(t_i)$ 为:

$$h(t_i) = \begin{cases} 0, & Pre(t_i) = \emptyset; \\ 1 + \max(h(t_j)), & t_j \in Pre(t_i). \end{cases} \quad (12)$$

2.3 目标函数

多核处理器系统的优化目标是当执行某一给定实时应用时,降低系统能耗的同时不能违反应用的执行时限,因此,可将该问题转化为有约束的粒子群算法迭代寻优过程。根据粒子群中的任一粒子 $pa(i)$ 所代表的调度策略,可计算得到其相应的执行完成时间和能耗,则本文目标函数可表示为:

$$\begin{aligned} F_{\text{obj}}(i) &= E'(i), \\ \text{s.t. } F_i(i) &\leq D. \end{aligned} \quad (13)$$

2.4 种群初始化

1) 初始化粒子 $pa(i)$ 的位置矢量 X_i 。先生成位置矢量 X_i 的前 m 维分量,其具体实现步骤是:①计算每个子任务的高度值;②随机从 $T(h)$ (即任务集 T 中拥有高度值 h 的任务集合)选择一个任务 t_i ;③从处理单元随机选择一个处理器核 P_j ;④将任务 t_i 分配给 P_j ;⑤重复执行步骤②~④,直至集合 $T(h)$ 的任务完全被分配。生成位置矢量的后 m 维分量代码如下:

For $j=m$ to $2m$

$X_{i,j} = \text{randomN}(1, k) // \text{randmN}$ 表示随机自然数函数。

2) 初始化粒子 $pa(i)$ 的速度矢量 V_i 。生成粒子 $pa(i)$ 的速度矢量 V_i 的 $2m$ 维分量代码如下:

For $j=1$ to $2m$

$V_{i,j} = \text{randomF}(-V_{\max}, V_{\max}) // \text{randomF}$ 表示随机浮点数函数。

3) 将粒子 $pa(i)$ 加入粒子群 M 中,判断粒子数目是否达到既定要求,若不满足,则返回步骤1)继续执行。

3 CASPSO 算法

基于可行性规则的粒子群优化算法^[9]处理有约束的优化问题时,其原理较简单,采用可行性规则更新粒子的最佳位置和种群的最佳位置,且无须增加额外参数。假设 P_i^t 为种群中第 i 个粒子在第 t 代时的最优位置矢量, X_i^{t+1} 为该粒子在第 $t+1$ 代时的所在位置。标准PSO算法中,只有当 $f(X_i^{t+1}) < f(P_i^t)$ 时,才有 $P_i^{t+1} = X_i^{t+1}$,而在基于可行性规则的PSO中,若以下3种情况中的任意一种发生,就执行 $P_i^{t+1} = X_i^{t+1}$,否则 $P_i^{t+1} = P_i^t$ 。3种情况如下:

1) P_i^t 不可行,而 X_i^{t+1} 可行;

2) P_i^t 和 X_i^{t+1} 均不可行;

3) P_i^t 和 X_i^{t+1} 均不可行,且 $V_{\text{viol}}(X_i^{t+1}) < V_{\text{viol}}(P_i^t)$,

其中, $V_{\text{viol}}(x) = \sum_{i=1}^N \max[g_i(x), 0]$ 是不可行解约束违反量的函数。

结合本文实际问题的编码、目标函数及种群初始化方案,节能调度算法CASPSO的描述如下。

输入:任务图 G ,处理器核数目为 n ,执行时限为 D 。

输出:在不违背应用的执行时限的前提下,产生一个节能的调度,具体步骤如下。

1) 按初始化 M 个粒子,令 $k=0$

2) 确定各粒子的最佳位置,根据可行性规则确定种群的最佳位置;

3) 检查 k 是否小于迭代终止数, 如果 k 小于迭代终止数, 则转至步骤 4); 如果 k 大于等于迭代终止数, 则输出群体最佳位置;

4) 利用公式 (8) 和 (11) 分别对粒子的速度和位置矢量进行更新。

5) 每个粒子的位置矢量的前 m 维分量的更新可能破坏任务间执行的先序关系, 因此须对每个处理器核上的任务按其高度值进行排序。

6) 利用目标函数对各粒子进行评价, 并基于可行性规则更新粒子的最佳位置和种群的最佳位置, 并转至步骤 3)。

4 实验结果

为了验证算法的有效性, 本文采用与文献[3]相同的仿真环境, 即多核处理器系统包含 6 个同构的处理器核, 每个处理器核都能独立地调整执行电压, 且有 4 种离散电压模式: (1.75 V, 1 000 MHz), (1.40 V, 800 MHz), (1.20 V, 600 MHz) 和 (1.00 V, 466 MHz)。实验将任务集分为 2 部分: 第一部分包括 10 个表示实际应用的任任务图 (RG1~RG10); 第二部分包括 5 个由 TGFF^[5] 随机产生的 DAG (tgff1~tgff5), 且每个图中包含 50~100 个节点。

将 CASPER 算法与本文算法进行比较, 其能耗节省情况见表 1。表中能耗节省率 $\eta = (E_1 - E_2) / E_2$, 式中, E_1 为应用 DVS 的系统能耗, E_2 为未考虑 DVS 的系统能耗。

表 1 CASPER 与 CASPSO 能耗节省对比

Table 1 Energy-saving contrast of CASPER and CASPSO

任务图	T / E	D	能耗节省率 /%	
			CASPER 算法	本文算法
RG1	16/24	65	60.7	67.8
RG2	17/28	50	54.3	60.9
RG3	14/15	130	44.0	40.2
RG4	20/39	2 120	52.3	60.8
RG5	28/32	225	61.5	75.2
RG6	28/32	460	65.5	73.8
RG7	41/69	925	62.2	60.9
RG8	18/29	665	65.6	73.6
RG9	95/158	151	50.5	61.2
RG10	361/684	17 154	62.2	73.5
tgff1	43/74	1 400	50.5	58.8
tgff2	68/119	2 000	59.3	71.2
tgff3	93/170	3 300	53.5	66.5
tgff4	90/158	3 300	67.3	68.4
tgff5	100/208	5 400	50.0	62.3

从 1 表中可知, CASPER 算法的平均能耗节省率为 53.8%, 而本文算法为 64.9%。实验结果表明, 本

文算法的节能效果比 CASPER 算法好。

CASPER 和本文算法对任务图 RG9 进行调度时的收敛曲线见图 1。从图中可以看出, 本文算法的收敛速度比 CASPER 算法快, 且可收敛到更优的解。

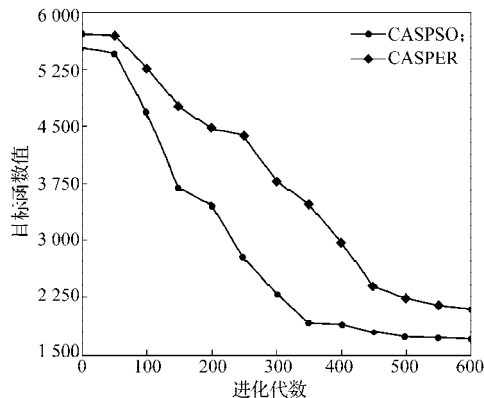


图 1 CASPER 和 CASPSO 收敛曲线

Fig. 1 Convergence curves for CASPSO and CASPER

5 结语

目前, 多核处理器系统的节能调度研究多采用分段式的调度策略, CASPER 算法是基于遗传算法实现多核处理器系统中的整体节能调度框架, 但存在着难以收敛于全局最优解和收敛速度较慢等问题。本文通过将多核处理器系统中的 3 个过程整合到粒子群优化迭代寻优的过程中, 并基于可行性规则更新粒子和种群的最佳位置, 实验结果表明, 本文所提算法能较快地收敛于全局最优解, 且优于 CASPER 算法。

参考文献:

- [1] Gruian F, Kuchcinski K. LEneS: Task Scheduling for Low-Energy Systems Using Variable Supply Voltage Processors [C]//Proceedings of the ASPDAC 2001 Asia and South Pacific Design Automation Conference. Yokohama: IEEE Press, 2001: 449-455.
- [2] Schmitz M T, Al Hashimi B M, Eles P. Energy-Efficient Mapping and Scheduling for DVS Enabled Distributed Embedded Systems[C]//Proceedings of Design, Automation and Test in Europe Conference and Exhibition (DATE '02). Paris: IEEE Press, 2002: 514-521.
- [3] Kianzad V, Bhattacharyya S S, Gang Qu. CASPER: An Integrated Energy-Driven Approach for Task Graph Scheduling on Distributed Embedded Systems[C]//16th IEEE International Conference on Application-Specific Systems, Architecture Processors(ASAP 2005). [S. l.]: IEEE Press, 2005: 191-197.

(下转第 82 页)