

一种面向多处理器系统的在线低功耗调度算法

曹仰杰¹, 钱德沛^{1,2}, 伍卫国¹, 董小社¹

(1. 西安交通大学电子与信息工程学院, 710049, 西安; 2. 北京航空航天大学计算机学院, 100191, 北京)

摘要: 针对当前多处理器系统中的散热瓶颈问题, 基于处理器动态速度调节技术, 提出了一种在线低功耗调度算法(PEQUI). PEQUI以动态均衡算法(EQUI)为基础, 公平地分配处理器资源, 依据处理器功耗与运行速度间存在非线性关系, 以正比于系统任务数的方式调节处理器运行速度. 与传统低功耗调度算法相比, PEQUI仅基于当前待调度任务的信息进行决策, 决策参数少. 以能量消耗与任务执行流时间为评价算法性能的指标, 利用在线竞争分析方法证明了PEQUI算法与最优离线算法相比可达到常数竞争比(< 10). 模拟结果表明, PEQUI比最近到达处理器共享算法(LAPS)和恒速EQUI算法能更好地优化系统整体性能和能量消耗. 在相同负载情况下, 与LAPS相比, PEQUI在降低功耗的同时系统平均运行时间也降低了近7%.

关键词: 多处理器系统; 在线低功耗调度; 动态速度调节; 竞争分析

中图分类号: TP301 **文献标志码:** A **文章编号:** 0253-987X(2010)08-0015-05

An Online Power-Efficient Scheduling Algorithm for Multiprocessor Systems

CAO Yangjie¹, QIAN Depei^{1,2}, WU Weiguo¹, DONG Xiaoshe¹

(1. School of Electronics and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China;

2. Department of Computer Science and Technology, Beihang University, Beijing 100191, China)

Abstract: Aiming at addressing the bottleneck problem of heat dissipation in multiprocessor systems, an online power-efficient scheduling algorithm PEQUI is proposed based on the dynamic speed scaling technique. PEQUI is capable of fairly allocating processor resources by applying the dynamic partitioning strategy (EQUI). Moreover, PEQUI adjusts processor speeds in proportion to the number of active jobs by taking the non-linear relationship between processor's power consumption and its execution speed into account. Compared with traditional power-efficient algorithms, PEQUI is able to make irrevocable decisions by using only information of the current active jobs, and needs fewer decision-making parameters. Online competitive analysis and a comparison with the optimal offline algorithm show that PEQUI achieves a constant competitive ratio with respect to the total execution time and energy. Simulation results show that PEQUI achieves better performance and lower power consumption than algorithms such as the latest arrival processor sharing (LAPS) and EQUI-based strategies with constant speed. A comparison with LAPS under the same workloads shows that PEQUI effectively reduces power consumption while the execution time reduces near 7%.

Keywords: multiprocessor system; online power-efficient scheduling; dynamic speed scaling; competitive analysis

随着处理器性能不断提升, 散热瓶颈促使降低 功耗成为研究热点. 依据任务已知信息的多少, 低功

耗调度算法可分为离线式、半在线式及在线式。离线式调度算法主要适用于任务信息(如释放时间,完成时间等)事先都已知的情况,其优化目标是在满足任务限定条件下最小化系统功耗。离线式调度算法通常归结为某类组合优化问题^[1-3],求解相对简单。然而,由于任务信息难以获取,此类算法很难在实际中应用。半在线式调度算法通常假定任务的最迟完成时间已知,在确保任务如期完成前提下最大限度降低处理器运行速度,即减小系统功耗。由于半在线式算法仍依赖于任务特定信息,因此只能适用于某些特殊环境,如实时系统,目前常见的低功耗调度算法多属于半在线式^[4-6]。与前者相比,在线式调度算法减少了对任务信息的依赖,因此更具有通用性。然而,在线式低功耗调度算法常采用启发式算法,在定性分析上难度较大,目前仅有少数算法被证明具有理论下界^[7]。

本文面向多处理器系统,基于动态速度调节技术,提出了一种在线式低功耗调度算法(PEQUI)。利用分摊局部竞争分析法证明该算法与最优离线算法相比可达到常数竞争比(<10)。模拟实验结果表明在同样系统负载情况下,与最近到达处理器共享算法(LAPS)相比,PEQUI不仅降低了系统功耗,而且使系统平均运行时间降低近7%。

1 在线低功耗调度算法

本文面向数据可任意划分的并行计算任务,着重研究 non-clairvoyant 在线自适应调度算法,即调度器仅基于任务当前及历史信息做出决策,而对任务未来情况,如并行度、剩余工作量和运行完成时间等一无所知。利用扩展的在线竞争比分析方法——分摊局部竞争分析法寻求调度算法的理论性能下界,并通过模拟实验进行对比验证。

1.1 假设与定义

定义1 并行任务片 $J_{i,j} = \langle w_{i,j}, \Gamma_{i,j}, h_{i,j} \rangle$ 是一个三元组,其中 $w_{i,j}$ 为任务 J_i 的第 j 个任务片的工作量, $\Gamma_{i,j}$ 为第 j 片的加速度函数, $h_{i,j}$ 为并行度。特殊情况下,如果 $h_{i,j} \geq P$ (P 为处理器总数目),则该任务片被称为可完全并行化;如果 $h_{i,j} = 1$,则该任务片是串行的。因此,任意并行任务 J_i 可表示为 $J_i = \langle J_{i,1}, J_{i,2}, \dots, J_{i,q} \rangle$ 。任务 J_i 的总工作量为 $w_i = \sum w_{i,j}$ 。假定在任意时刻 t 任务 J_i 分配的处理器资源个数为 $a_i(t)$,则有 $\sum a_i(t) \leq P$ 。

定义2 任务 J_i 在时刻 t 被称为活动的,当且仅当

在时刻 t 任务 J_i 已释放且未执行完成,即 $r_i < t < c_i$ 。 c_i 、 r_i 为任务完成时间和释放时间。 t 时刻系统内所有活动的任务数表示为 $N(t)$ 。

定义3 $\text{prefix}[J_i(t)]$ 表示任务 J_i 在 t 时刻前完成的工作量, $\text{subfix}[J_i(t)]$ 表示任务 J_i 在 t 时刻的剩余工作量。

定义4 流时间 f_i 表示任务 J_i 在系统中的总逗留时间,即 $f_i = c_i - r_i$ 。系统总流时间 $F(J)$ 是所有任务流时间之和,即 $F(J) = \sum f_i = \int N(t) dt$ 。

定义5 系统代价函数 $G(J) = F(J) + E(J) = \int N(t) dt + \int u(t) dt$, 其中 $E(J)$ 表示能量消耗, 功耗函数 $u(t) = s(t)^\alpha$, $s(t)$ 为运行速度, α 为常数。

问题1 任意有限任务集 $J = \{J_1, J_2, \dots, J_n\}$ 和 P 台速度可调节处理器集 $M = \{M_1, M_2, \dots, M_P\}$, 任务 $J_i (1 \leq i \leq n)$ 在任意时刻 t 到达, 任务 J_i 工作量 w_i 、并行度 h_i 及完成时间 c_i 未知, 寻求在线调度策略 A 的动态调整任务集与处理器资源映射关系 $\langle J, M \rangle$ 及相应处理器运行速度集 S , 并使得调度策略 A 相对于任务集 J 的代价函数 $G(J)$ 存在在线竞争比。

针对以上问题,在不考虑功耗限制的情况下,最少剩余处理时间优先(SRPT)算法^[9]是目前已知的最优离线算法(OPT)。

1.2 算法描述

与传统意义的调度算法相比,基于动态速度调节技术的在线低功耗调度算法(PEQUI)的主要思想:①用改进动态均衡算法 EQU^[8] 作为资源分配策略,并引入资源调节参数 λ , 改善传统 EQU 算法资源浪费较大的弊端;②依据处理器功耗与运行速度间存在非线性关系,以正比于系统任务的数方式调节处理器运行速度。算法描述如图1所示。

根据系统负载的变化情况,PEQUI 基于公平原则在活动的任务间进行资源分配。从资源利用率来讲,算法有很好的自适应性,当系统负载相对较小时通过动态调整参数 λ 的值,PEQUI 将比传统 EQU 算法可以更好地节省资源。

2 PEQUI 算法理论分析与证明

利用分摊局部竞争分析方法对所提出的 PEQUI 算法进行性能分析。分析过程中假定处理器速度 s 在 $(0, 1]$ 区间内连续可调,在实际应用中可通过变换函数还原成实际对应速度值,每个处理器在执行过程中可中断且中断开销忽略不记。

输入: 任务集 $J=\{J_1, J_2, \dots, J_n\}$, 处理器资源集 $M=\{M_1, M_2, \dots, M_P\}$ 及参数 λ, α ;
 输出: 任务集与处理器资源映射表 $\langle J, M \rangle = \{a_1, a_2, \dots, a_n\}$ 及处理器相应运行速度集 $S=\{s_1, s_2, \dots, s_P\}$.
 执行步骤:
 初始化 $\langle J, M \rangle$ 及 S ;
 While (新任务到达或任务完成)
 Q = 系统内当前活动任务集;
 While ($J_i \in Q$)
 $a_i = P^\lambda / \|Q\|$
 While ($M_i \in M$)
 $s_i = (\|Q\|/P)^{1/\alpha}$
 输出 $\langle J, M \rangle$ 及 S .

α 为功耗与速度关系因子, 通常值为 3; λ 为资源

调节参数; Q 为活动任务集临时变量

图 1 PEQUI 算法描述

引理 1^[4] 设 $f(x)$ 为任意一个连续且单调增的实函数, $f^{-1}(x)$ 是 $f(x)$ 的反函数, 如果 $f(0)=0$, 则对于任意 $g, h \geq 0$, 不等式 $\int_0^h f(x) dx + \int_0^g f^{-1}(x) dx \geq gh$ 成立, 式中等号当且仅当 $f(g)=h$ 时适用.

定理 1 针对前述问题 1, 本文提出的 PEQUI 算法相对于优化目标函数 $G(J)$ 的竞争比为 $O((1/\alpha)P^{1/\alpha})$.

采用构造势能函数和利用分摊局部竞争分析方法来证明定理 1.

(1) 构造势能函数. 为简化表达, 把 t 时刻 PEQUI 算法的任务数量函数表示为 N_t^p , OPT 算法的任务数量函数表示为 N_t^o . 在任意时刻 t , 令 $n_t^p(z) = \sum_{i=1}^{N_t^p} \{\text{subfix}[J_i(t)] \geq z\}$, 即 $n_t^p(z)$ 表示 PEQUI 算法在 t 时刻系统中所有任务剩余工作量大于或等于 z 的任务个数, 其中 z 是正实数变量. 相应地,

$n_t^o(z) = \sum_{i=1}^{N_t^o} \{\text{subfix}[J_i(t)] \geq z\}$ 表示 t 时刻 OPT 算法系统中所有任务剩余工作量大于或等于 z 的任务个数. 图 2、图 3 分别为 PEQUI 和 OPT 算法引发势能函数变化示意图. 从图 2、图 3 可知, $n_t^p(z)$ 、 $n_t^o(z)$ 是单调递减梯形函数. 由于最优离线算法优先完成剩余工作量最小的任务^[9], 因此在很小的 Δt 时间内只有最短任务的工作量发生变化 (如图 3b 所示), 且减少量为 $s_i^o P^\lambda \Delta t$, 其中 s_i^o 是离线算法对应的处理器速度. 由于 PEQUI 对每个任务公平地分配处理器资源, 因此在 Δt 时间内 PEQUI 将使每个处于活动任务工作量发生变化, 且减少量最大为 $s_i^p P^\lambda \Delta t / N_t^p$ (如图 2b 所示). 由以上分析定义势能函数

$$\Phi(t) = \eta \int_0^\infty \left[\sum_{i=1}^{N_t^p} i^{1-1/\alpha} - (n_t^p(z))^{1-1/\alpha} n_t^o(z) \right] dz$$

式中: η 为常数.

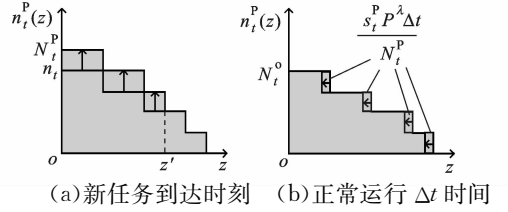


图 2 PEQUI 引发势能函数变化示意图

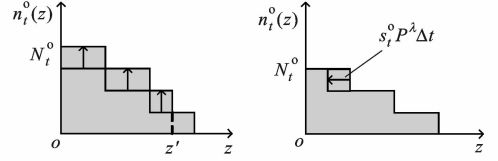


图 3 OPT 引发势能函数变化示意图

(2) 分摊局部竞争分析^[9]. 证明 $\Phi(t)$ 满足下述 4 个条件. ①有界性: 对于所有 t , $\Phi(t)$ 是非负的, 即 $\Phi(t) \geq 0$, 由于 0 时刻和 ∞ 时刻系统内不存在活动的任务, 因此 $\Phi(0) = \Phi(\infty) = 0$, 即有界性成立. ②到达条件: 假定工作量为 z' 的任务到达, 势能函数 Φ 的变化如图 2a、3a 所示, 显然当 $z \geq z'$ 时, $n_t^p(z)$ 和 $n_t^o(z)$ 保持不变, 即 Φ 保持不变; 当 $z < z'$ 时, Φ 被积函数增量为 $(n_t^p(z) + 1)^{1-1/\alpha} - (n_t^p(z))^{1-1/\alpha}$. $(n_t^p(z) + 1) \leq 0$, 即 Φ 不减. ③完成条件: 当任务完成时 $n_t^p(z)$ 和 $n_t^o(z)$ 减少 1 的时刻恰为 $z=0$, 因此势能函数值保持不变. ④运行条件: 在很小的 Δt 时间内, 当系统没有任务到达或完成时, 有

$$\frac{d\Phi(t)}{dt} =$$

$$\frac{\eta}{\Delta t} \int_0^\infty \left[\sum_{i=1}^{N_{t+\Delta t}^p} i^{1-1/\alpha} - (n_{t+\Delta t}^p(z))^{1-1/\alpha} n_{t+\Delta t}^o(z) \right] dz -$$

$$\frac{\eta}{\Delta t} \int_0^\infty \left[\sum_{i=1}^{N_t^p} i^{1-1/\alpha} - (n_t^p(z))^{1-1/\alpha} n_t^o(z) \right] dz$$

由 $n_t^p(z)$ 和 $n_t^o(z)$ 的单调性和引理 1, 可证得

$$\frac{d\Phi(t)}{dt} \leq \eta \left(-N_t^p P^{1-1/\alpha} + \left(1 - \frac{1}{\alpha} \right) P N_t^p \right) + \eta \left(\frac{1}{\alpha} P (s_i^o)^\alpha + N_t^o P^{1-1/\alpha} \right)$$

令 $\eta = 2/(P^{1-1/\alpha} - (1-1/\alpha)P)$, 易知

$$\frac{dG(t)}{dt} + \frac{d\Phi(t)}{dt} \leq P^{1/\alpha} \left(\frac{1/\alpha}{1-1/P^{1/\alpha}} \right) \frac{dO(t)}{dt}$$

综上, 由分摊局部竞争理论可知, 当系统中处理器数目 P 较大时, 在线低功耗调度算法 PEQUI 的竞争比为 $O((1/\alpha)P^{1/\alpha})$.

在线竞争比分析方法虽然不能全面分析算法的实际性能,但它给出了在最坏情况下在线算法与最优离线算法间的差距. 竞争比越小,表明在线算法在实际中性能也可能越好. 对于 PEQUI 算法,由于系统内处理器资源数目通常为常数,因此在最坏情况下,PEQUI 也可以达到常数级竞争比.

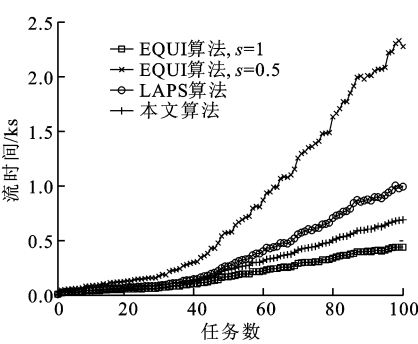
3 模拟实验

采用改进的在线调度算法模拟器(Malleable-Lab)^[10]对本文 PEQUI 算法进行了实际性能验证. Malleable-Lab 是一款支持并行负载可任意划分的多处理机模拟器,以 Intel XScale 处理器作为基准. 本文对 Malleable-Lab 扩展了功耗统计支持.

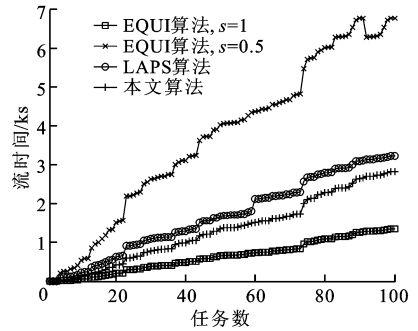
利用改进后的 Malleable-Lab 模拟器构建了有 64 个处理核心的多机系统,其中每个处理核心都具有动态电压频率调节功能. 实验采用并行负载模型 Downey^[11]及 Cirne^[12]为基准工作负载. 为确保实验结果的有效性,每组实验任务数从低到高变化,并且针对同样数量的任务分别进行 10 次实验,最后取实验结果的均值. 在相同的工作负载输入的情况下,算法的性能评价标准分别采用系统流时间、能量消耗以及总性能代价比,其中总性能代价比=(在线算法流时间代价+在线算法能量消耗代价)/(离线算法流时间代价+离线算法能量消耗代价). 总性能代价比越小,说明算法性能提升或降低功耗效果越好.

图 4、图 5 分别对比了恒定速度 EQUI、LAPS 及本文算法在不同并行负载时的总流时间和能量消耗. 实验结果表明,不同并行负载对算法性能产生了一定影响. 由于 Cirne 负载模型考虑更多因素,如用户提交模式等,使任务提交时间分布更加不均匀,因此对算法性能影响较大,算法间差距也更为突出. 两类不同负载的实验结果表明,相比其他 2 种算法,本文算法在系统性能和能量消耗之间取得了较好的均衡. 由图 4、图 5 可以看出,当 $s=1$ 时 EQUI 算法获得较好的性能,即较小的流时间,但同时消耗的能量也最多. 当 $s=0.5$ 时 EQUI 算法消耗的能量较小,但此时系统性能受到严重影响. 与 LAPS 算法相比,在相同负载情况下,PEQUI 算法在降低系统整体功耗的同时,系统的平均运行时间也降低了近 7%左右.

图 6 的实验结果反映了以 Downey 并行负载为输入时,参数 λ 对本文 PEQUI 算法的影响. λ 取值情况决定了系统内当前可使用处理器资源的数目, λ

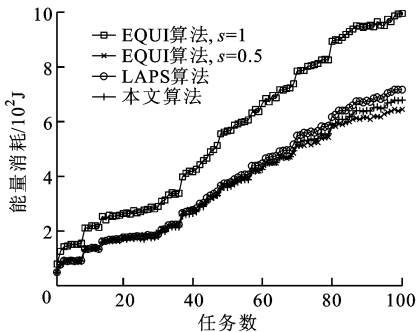


(a)Downey 并行负载

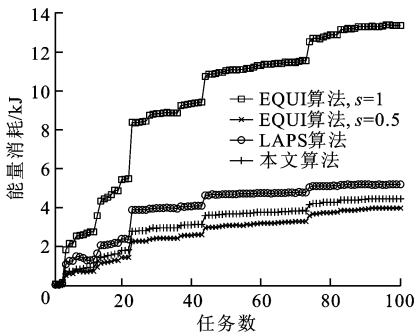


(b)Cirne 并行负载

图 4 3 种算法总流时间对比



(a)Downey 并行负载



(b)Cirne 并行负载

图 5 3 种算法能量消耗对比

越小系统内处理器资源数目也越少. 从图 6 可知,当系统负载较小时,减少 λ 值有利于降低整体系统的性能代价比. 然而,随着系统负载的增加,减少处理器的数目虽然可以降低系统功耗,但会严重影响系

统性能并导致任务执行时间的延长,最终将导致整个系统性能代价比上升.实验数据也表明,当系统内活动的任务数超过系统内总处理器数的50%时,如果继续减少 λ 的值将会严重影响系统的性能代价比,即在此状况下算法很难在确保性能与降低功耗这2个矛盾的目标之间达到均衡.

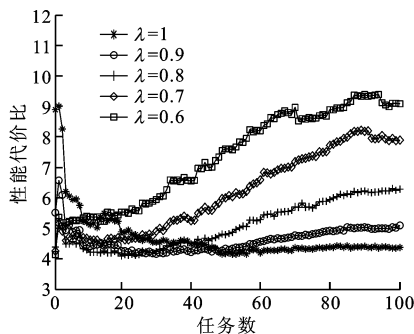


图6 λ 取不同值时本文PEQUI算法的性能比较

4 结束语

芯片集成度的飞速发展以及功耗墙的限制推进了多核处理器技术的发展与广泛应用,灵活多样的并行负载为提高系统整体性能及降低功耗提供了更大的优化空间.本文对多处理器系统的在线低功耗调度策略进行了研究,基于动态速度调节技术所提出的在线低功耗PEQUI算法不仅从理论上保证了算法的有效性,而且模拟实验结果也表明,与传统的低功耗算法相比,PEQUI算法更好地权衡了系统整体性能与功耗的平衡.

我们进一步的研究内容是增强算法的适用度以适应更广泛的平台,如一些异构系统,根据工作负载运行时的特点,寻求合适的速度调节函数使不同处理器工作在不同速度水平,在确保系统性能的前提下进一步降低系统功耗.

参考文献:

- [1] YAO F, DEMERS A, SHENKER S. A scheduling model for reduced CPU energy [C] // Proceedings of FOCS. Piscataway, NJ, USA: IEEE, 1995:374-382.
- [2] KWON W C, KIM T. Optimal voltage allocation techniques for dynamically variable voltage processors [J]. ACM Transactions on Embedded Computing Systems, 2005, 4(1):211-230.

- [3] IRANI S, SHUKLA S, GUPTA R. Algorithms for power savings [J]. ACM Transactions on Algorithms, 2007, 3(4):41-64.
- [4] ALBERS S, FUJIWARA H. Energy-efficient algorithms for flow time minimization [J]. ACM Transactions on Algorithms, 2007, 3(4):49-66.
- [5] BANSAL N, PRUHS K, STEIN C. Speed scaling for weighted flow time [C] // Proceedings of SODA. New York, USA: ACM, 2007:805-813.
- [6] LAM T W, LEE L K, TO I, et al. Speed scaling functions for flow time scheduling based on active job count [C] // Proceedings of ESA. Berlin, Germany: Springer, 2008: 647-659.
- [7] CHAN Ho-Leung, EDMONDS J, LAM T W, et al. Nonclairvoyant speed scaling for flow and energy [C] // Proceedings of STACS, Berlin, Germany: Springer, 2009: 409-420.
- [8] EDMONDS J. Scheduling in the dark [C] // Proceedings of STOC. New York, USA: ACM, 1999:179-188.
- [9] PRUHS K. Competitive online scheduling for server systems [J]. ACM SIGMETRICS Performance Evaluation Review, 2007, 34(4):52-58.
- [10] CAO Yangjie, SUN Hongyang, SHU Wenjing, et al. Malleable-lab: a tool for evaluating adaptive online schedulers on malleable jobs [C] // Proceedings of PDP. Piscataway, NJ, USA: IEEE, 2010:11-18.
- [11] DOWNEY A B. A parallel workload model and its implications for processor allocation [C] // Proceedings of HPDC. New York, USA: ACM, 1997:112-124.
- [12] CIRNE W, BERMAN F. A comprehensive model of the supercomputer workload [C] // Proceedings of 4th Workshop on Workload Characterization. New York, USA: ACM, 2001: 140-148.

[本刊相关文献链接]

利用投影时序逻辑的多内核进程调度建模与验证. 西安交通大学学报, 2010, 44(3):52-57.

基于反馈的片上多处理器系统层次负载平衡算法. 西安交通大学学报, 2008, 42(2):179-183.

曙光5000A天体大规模数值模拟软件性能测试. 西安交通大学学报, 2009, 43(10):71-75.

自适应大规模服务器集群监控系统的构建. 西安交通大学学报, 2008, 42(4):399-403.

(编辑 刘杨 赵大良)