

实时多处理器系统的双优先级调度算法

张益嘉, 马洪连, 丁 男

(大连理工大学计算机学院, 辽宁 大连 116023)

摘要: 针对同构多处理器系统提出一种基于双优先级的实时任务调度算法。对偶发任务进行接受测试, 进一步提高了系统对偶发任务调度的成功率。模拟结果表明, 当多核处理器系统利用率达到极限时, 该算法依然能够在完成强实时周期任务的成功调度前提下, 保证软实时周期任务和偶发任务具有较高的调度成功率。

关键词: 同构多处理器系统; 双优先级; 偶发任务

Dual Priority Schedule Algorithm for Real-time Multi-processor System

ZHANG Yi-jia, MA Hong-lian, DING Nan

(School of Computer, Dalian University of Technology, Dalian 116023, China)

【Abstract】 This paper presents a real-time task schedule algorithm based on dual priority for the homogeneous multi-processor system. Aiming at the character of sporadic tasks, the algorithm pretests the sporadic tasks and further improves the success rate of sporadic tasks schedule. The simulation results show that when the multi-core processor system utilization is at the limit, the algorithm is still able to schedule the hard real-time periodic tasks successfully and ensure the high success schedule rate of soft real-time periodic tasks and sporadic tasks.

【Key words】 homogeneous multi-processor system; dual priority; sporadic tasks

DOI: 10.3969/j.issn.1000-3428.2011.01.092

1 概述

近年来随着实时系统广泛地应用于航空航天、军工、多媒体等领域, 其复杂程度有了明显的提高^[1]。复杂实时系统通常包括多种不同类型的实时任务, 从任务的实时性来看, 可分为强实时任务和软实时任务; 从任务的产生规律来看, 可分为周期性任务和非周期性任务。

在实时系统控制中, 对于强实时周期任务要求必须在硬截止期内完成, 对于软实时周期任务允许一定比例的丢弃, 对于偶发任务(强实时非周期性任务)要求在固截止期内完成。

另一方面由于复杂度和精度的提高, 实时系统对于处理器性能提出了更高的要求, 多处理器系统越来越多地应用在复杂实时系统的研究中^[2]。如何在多处理器系统中有效地调度实时系统中的强实时周期性任务、软实时周期性任务和偶发任务是多处理器系统能否在复杂实时系统中得以广泛应用的关键性问题。

双优先级调度算法最早由文献[3]提出并应用于实时系统的任务调度, 其突出的优点是算法实现简单而且计算量小。文献[4]将双优先级算法进行改进, 使之可以在单核上调度强实时周期性任务、软实时周期性任务和非周期性任务。文献[5]将双优先级算法应用在多核处理器系统中, 实现了基于多核处理器系统的强实时周期任务和非周期任务的混合调度, 但没有考虑实时系统中具有固截止期的偶发任务。本文提出了一种针对同构多核处理器系统的双优先级调度算法, 该算法可以在多核架构下对强实时周期性任务、软实时周期性任务和偶发任务进行有效的调度, 并对偶发任务进行接受测试, 进一步提高了系统对偶发任务调度的成功率。

2 系统模型和任务模型

设一个实时同构多处理器系统有 $m(m>1)$ 个处理器,

$P_1, P_2, \dots, P_m$ 分别表示 m 个处理器, 它们在系统中所处地位相同, 具有相同的处理能力, $P_1, P_2, \dots, P_m$ 之间通过共享内存的方式进行数据交换。本文对强实时周期任务、软实时周期任务和偶发任务做如下形式化描述:

(1) 强实时周期任务集表示为 $HT = \{\tau_{h1}, \tau_{h2}, \dots, \tau_{hn}\} (n \geq 1)$, 每个强实时周期任务 τ_{hi} 描述为:

$$\tau_{hi} = \langle C_{hi}, T_{hi}, D_{hi}, P_{hi} \rangle (i=1, 2, \dots, n)$$

其中, C_{hi} 、 T_{hi} 、 D_{hi} 、 P_{hi} 分别表示任务 τ_{hi} 的最坏执行时间、任务周期、任务时限和优先级提升时间。

(2) 软实时周期任务集表示为 $ST = \{\tau_{s1}, \tau_{s2}, \dots, \tau_{sn}\} (n \geq 1)$, 每个软实时周期任务 τ_{si} 描述为 $\tau_{si} = \langle C_{si}, T_{si}, D_{si} \rangle (i=1, 2, \dots, n)$ 。其中, C_{si} 、 T_{si} 、 D_{si} 分别表示任务 τ_{si} 最差执行时间、任务周期、任务时限。

(3) 偶发任务集表示为 $SP = \{\tau_{sp1}, \tau_{sp2}, \dots, \tau_{spn}\} (n > 1)$, 第 i 个偶发任务实例 τ_{spi} 描述为 $\tau_{spi} = \langle T_{spi}, C_{spi}, D_{spi}, \lambda \rangle$ 。其中, T_{spi} 、 C_{spi} 、 D_{spi} 、 λ 分别表示偶发任务实例 τ_{spi} 的到达时刻、最坏执行时间、任务实例的时限和平均到达率。偶发任务实例符合平均到达率为 λ 的泊松过程。

假设所有任务之间相互独立, 不存在任务同步和互斥; 忽略任务切换、抢占和迁移的开销以及任务释放的抖动。

3 多核处理器的双优先级调度算法

3.1 相关定义

定义 1 强实时周期任务优先级提升时间是指强实时周

作者简介: 张益嘉(1979—), 男, 工程师、硕士, 主研方向: 实时系统控制, 嵌入式系统设计; 马洪连, 教授; 丁 男, 讲师、博士研究生

收稿日期: 2010-07-10 **E-mail:** zhyj@dlut.edu.cn

期任务实例从释放到优先级从低带提升到高带的时间。

定义 2 软实时周期任务平均丢失率是指在软实时周期任务的执行过程中, 由于错过任务时限而被丢弃的软实时周期任务实例个数与到达总个数之比。

定义 3 偶发任务平均丢失率是指在偶发任务的执行过程中, 偶发任务实例丢弃个数与到达总个数之比。

3.2 任务分配策略

多核处理器的任务分配问题属于 NP 完全问题, 本文针对实时系统不同任务的特点结合静态分配和动态分配的优点, 使用混合分配的策略。首先将强实时周期任务和用来处理偶发任务的周期服务器预先静态分配到多核处理器上, 并离线计算各自的提升时间。每个强实时周期任务和周期服务器在到达提升时间前, 和软实时周期任务一起按照各自优先级进行全局的动态调度, 一旦强实时周期任务或周期服务器到达提升时间, 则立刻提升到预先分配好的处理器上优先执行。所以只要准确计算出强实时周期任务和周期服务器的提升时间就能够保证强实时周期任务和周期服务器在其硬截止期内完成。

3.3 带有周期性服务器的双优先级调度策略

双优先级调度算法的核心思想是将任务优先级的带宽分为 3 个部分: 高带优先级(HP)、中带优先级(MP)和低带优先级(LP)。HP 中的任务优先级总是比 MP 和 LP 中的优先级高, 而 MP 中的优先级总是高于 LP 中的优先级。

如图 1 所示, $P_1, P_2, \dots, P_m$ 表示 m 个处理器, LPQ 表示处于 LP 中的任务队列, MPQ 表示处于 MP 中的任务队列, $HPQ_i (1 \leq i \leq m)$ 表示 P_i 处理器相应的 HP 任务队列。每个强实时周期任务实例(包括周期服务器)释放时处于 LPQ , LPQ 中任务的优先级根据 RMS 算法进行设置。每个软实时周期任务实例释放时处于 MPQ , MPQ 任务的优先级根据 EDF 算法进行设置。每个强实时周期任务实例(包括周期服务器)有相应的提升时间, 一旦到达提升时间, 并且强实时周期任务实例没有被完成, 则立即从 LPQ 中提升至其预置处理器对应的 HPQ 中。每个处理器优先响应各自 HPQ 队列中的任务请求, 如果相应 HPQ 队列为空, 则从 MPQ 和 LPQ 中选择优先级最高的任务执行。

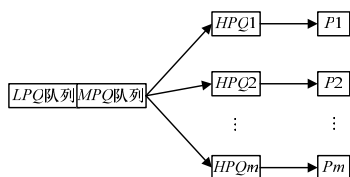


图 1 双优先级算法调度

双优先级算法的关键在于强实时周期任务提升时间的计算, 其计算公式如下式所示:

$$W_i^{n+1} = \sum_{j \in hp(i)} \left\lceil \frac{W_j^n}{T_j} \right\rceil C_j + C_i \quad (1)$$

这个递归公式用来预先计算强实时周期任务 τ_{hi} 的最坏响应时间 W_i ; $hp(i)$ 表示优先级高于 τ_{hi} 的所有强实时周期任务集合; C 和 T 分别是强实时周期任务的最差执行时间和周期; 递归起始条件 $W_i^0=0$; 结束条件 $W_i^{n+1}=W_i^n$; τ_{hi} 的提升时间用 P_{hi} 表示, 则 $P_{hi}=D_i-W_i$ 。

为了保证强实时周期任务调度不受影响, 本文使用周期服务器处理偶发任务。一个周期性服务器等同于一个强实时周期任务, 同样被预先静态分配到一个指定处理器, 系统开始运行时周期服务器处于 LPQ , 如果到达其提升时间并且没

有被执行完, 则被提升到相应的 HPQ 中。当周期服务器执行时, 首先检查偶发队列里是否有尚未执行的偶发任务, 如果偶发队列非空, 周期服务器就执行位于队列头的偶发任务; 如果偶发队列为空, 周期服务器自动挂起, 等待下一个周期的运行。

3.4 偶发任务的接受测试

实时系统中的偶发任务由于产生具有不确定性, 通常实时系统无法保证对偶发任务的完全调度。为了更为合理地调度偶发任务, 本算法首先对到达的偶发任务进行接受测试。如果偶发任务通过接受测试, 则将偶发任务加入偶发任务队列; 否则将偶发任务丢弃, 加入偶发队列中的偶发任务按照 EDF 算法进行调度。对于偶发任务进行接受测试不但可以更合理地利用系统的处理资源, 提高偶发任务的调度成功率, 而且可以使系统在偶发任务释放的初始时刻就能判断任务能否被执行, 使系统有时间对于重要的偶发任务进行容错响应。

本文使用基于 RMS 调度的周期服务器处理偶发任务, 所以在每个周期 T_s 内周期服务器有 C_s 个单位处理时间。接受的偶发任务之间基于 EDF 算法进行排序, 当第一个偶发任务 $\tau_{spi} < T_{spi}, C_{spi}, D_{spi}$ 到达时, 设服务器的空闲时间用 δ_{spi} 表示, 则如式(2)所示。如果 $\delta_{spi} \geq 0$, 系统接受任务 τ_{spi} , 并将 τ_{spi} 插入偶发队列中, 否则丢弃任务 τ_{spi} 。

$$\delta_{spi} = \left\lfloor (D_{spi} - T_{spi}) / T_s \right\rfloor C_s - C_{spi} \quad (2)$$

当偶发队列里已有 n 个偶发任务时, 偶发任务 $\tau_{spi} < T_{spi}, C_{spi}, D_{spi}$ 到达, 首先按式(3)计算 σ_{spi} :

$$\sigma_{spi} = \left\lfloor (D_{spi} - T_{spi}) / T_s \right\rfloor C_s - C_{spi} - \sum_{D_{spk} < D_{spi}} (C_{spk} - \theta_{spk}) \quad (3)$$

其中, θ_{spk} 表示目前偶发队列中偶发任务 τ_{spk} 已经完成部分的执行时间。如果 $\sigma_{spi} > 0$, 则下一步检查偶发队列中时限在 D_{spi} 之后的所有偶发任务相应的 σ_{sp} 。如果 $\sigma_{sp} > 0$, 则接受偶发任务 τ_{spi} , 并将 τ_{spi} 按照 EDF 算法设置的优先级加入偶发队列中, 否则丢弃偶发任务 τ_{spi} 。

偶发作业接受测试的具体过程如图 2 所示。

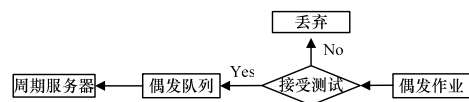


图 2 接受测试流程

4 模拟结果及分析

为了评估新算法的性能, 进行了一系列的模拟研究。重点关注当多核处理器系统总利用率接近极限情况下的软实时周期任务丢失率和周期服务器大小对偶发任务丢失率的影响。

在模拟测试中, 多核处理器的个数 m 取值范围为 2~8。模拟测试中的任务集分为强实时周期任务、软实时周期任务和偶发任务。每个处理器预先分配 10 个强实时周期任务, 并使每个处理器预置的负载利用率相同, 其变化范围为 63%~75%, 每个强实时周期任务的周期范围为 100~3 000, 其最小公周期大于 300 000。软实时周期任务数固定为 10, 并保证整个系统的软实时周期任务利用率为 25%, 其软实时周期任务的周期范围为 100~3 000, 其最小公周期大于 300 000。偶发任务测试实例服从平均到达率 $\lambda=0.002$ 的泊松过程, 并使其系统利用率为 2%, 将周期服务器的系统利用率固定设置为 3%。系统实际总负载利用率变化范围为 90%~102%。

在模拟中, 按上述规则随机产生测试任务集, 并离线计

算强实时周期任务的提升时间, 将模拟时间固定设置为 500 000, 同时将系统中的处理器个数分别设为 $m=2$, $m=4$ 和 $m=8$ 。

经过大量的模拟测试, 软实时周期任务丢失率的结果如图 3 所示, 图中每个点都是对 50 个不同的测试集模拟测试 100 次以后的平均结果。

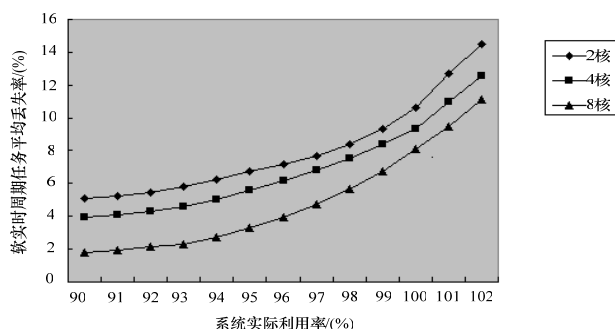


图 3 系统利用率对软实时周期任务平均丢失率的影响

由图 3 可知:

(1)随着系统实际利用率的增加, 无论是 2 核、4 核还是 8 核的多核处理器系统的软实时周期任务的丢失率都明显增加, 并且其曲线斜率随着系统利用率的增加而增大。这说明基于双优先级的多核调度算法在保证强实时周期任务完成的前提下当系统利用率达到极限时, 算法对软实时周期任务的调度能力下降明显。

(2)在系统利用率相同的情况下, 多核处理器系统的处理器个数越多其软实时周期任务的丢失率越小。这是因为不考虑并行开销的理想情况下, 基于双优先级的多核调度算法对软实时周期任务采用的是全局的动态调度, 在核数越多时越容易找到空闲的处理器处理软实时周期任务。

为了测试周期服务器大小对偶发任务调度成功率的影响, 固定强实时周期任务的系统利用率为 67%, 软实时周期任务的系统利用率为 25%, 偶发任务实例服从 $\lambda=0.01$ 的泊松过程, 并且固定其系统利用率为 2%, 系统总利用率为 94%。调整周期服务器利用率为 2%~4%, 测试结果如图 4 所示。图中每个点都是对 50 个不同的测试集模拟测试 100 次以后的平均结果。

由图 4 可知, 增大周期服务器的大小可以降低偶发任务的丢失率, 但随着周期服务器利用率的不断增加其效果越来越不明显。当周期服务器从 2.00%增大到 3.25%时, 偶发任务丢失率从 34.82%降低到 2.21%, 而当周期服务器从 3.25%增大到 4%时, 偶发任务丢失率仅从 2.21%降低到 1.77%。

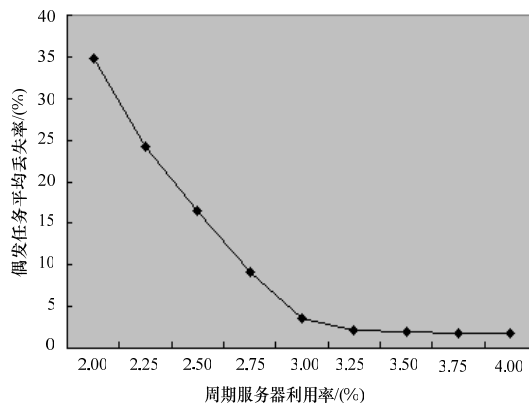


图 4 周期服务器大小对偶发任务平均丢失率的影响

5 结束语

实时多处理器系统中的任务调度问题一直是实时系统中的重要研究课题。本文在深入研究实时系统中不同类型任务特点的基础上, 提出了一种针对同构多核系统的基于双优先级的实时任务调度算法。该算法能够对在满足强实时周期任务时限要求的前提下高效地调度软实时周期任务和偶发任务。针对偶发任务随机的特点, 该算法对偶发任务进行接受测试处理, 进一步提高了偶发任务调度成功率。

参考文献

- [1] 乔 颖, 邹 冰, 方 亭, 等. 一种实时异构系统的集成动态调度算法[J]. 软件学报, 2002, 13(12): 2251-2258.
- [2] 李仁发, 刘 彦, 徐 成. 多处理器片上系统任务调度研究进展评述[J]. 计算机研究与发展, 2008, 45(9): 1620-1629.
- [3] Davis R, Wellings A. Dual Priority Scheduling[C]//Proc. of the 16th IEEE Real-time Systems Symposium. Pisa, Italy: IEEE Computer Society Press, 1995: 100-109.
- [4] 刘 怀, 费树岷. 基于双优先级的实时多任务动态调度[J]. 计算机工程, 2005, 31(18): 16-18.
- [5] Banus J M, Arenas A. Dual Priority Algorithm to Schedule Real-time Tasks in a Shared Memory Multiprocessor[C]//Proc. of the 17th International Parallel and Distributed Processing Symposium. Nice, France: [s. n.], 2003.

编辑 顾逸斐

(上接第 264 页)

光源充足柔和、色彩协调, 注重娱乐系统及客服系统的完善, 努力提升旅客个性化服务水平。

参考文献

- [1] 张曙光. 京沪城际高速铁路系统优化研究[M]. 北京: 中国铁道出版社, 2009.
- [2] 俞展猷. 铁道车辆舒适性评价方法的发展与研究现状[J]. 铁道车辆, 2004, 42(3): 1-6.
- [3] 王国胤. Rough 集理论与知识获取[M]. 西安: 西安交通大学出版社, 2001.

- [4] 曾晓辉, 文 展. 不完备信息系统的属性约简算法[J]. 计算机工程, 2009, 35(24): 185-187.
- [5] 贺仲雄. 模糊数学及应用[M]. 天津: 天津科学技术出版社, 1983.
- [6] 曾黄麟. 粗糙集理论及其应用[M]. 重庆: 重庆大学出版社, 1998.
- [7] 郑丽英, 王海涌, 刘丽艳. 基于粗糙集和模糊聚类理论的文本分类系统的研究与实现[J]. 铁道学报, 2007, 29(1): 45-49.
- [8] 陈 嵘, 王 平, 陈小平. 客运专线舒适性评价问题探析[J]. 铁道工程学报, 2005, (6): 6-9.

编辑 金胡考