第33卷 第11期 计 算 机 工 程 2007年6月
Vol.33 No.11 Computer Engineering June 2007
·博士论文· 文章编号：1000—3428(2007)11—0021—02 文献标识码：A 中图分类号：TP316

改进的抢占阈值调度任务响应时间分析方法

王 涛，刘大昕，张健沛
(哈尔滨工程大学计算机科学与技术学院，哈尔滨 150001)

摘 要：现有的基于抢占阈值调度的任务响应时间分析方法对实时任务系统进行可调度性判定时，对任务响应时间估计过低，造成任务错过期限的现象。针对上述缺点不足，该文提出改进的基于抢占阈值调度的任务响应时间分析方法，考虑了任务释放抖动和时钟嘀嗒调度的影响，使用改进的任务参数计算系统任务时间需求函数。仿真对比结果表明，改进后的方法较单纯固定优先级抢占阈值调度下的任务响应时间分析方法得到更加精确可调度性分析结果。

关键词：抢占阈值；嘀嗒调度；响应时间；可调度性；释放抖动

Improved Response Time Analysis Method for Scheduling Tasks with Preemption Threshold

WANG Tao, LIU Daxin, ZHANG Jianpei
(Department of Computer Science and Technology, Harbin Engineering University, Harbin 150001)

【Abstract】 The existing response time analysis method for scheduling tasks with preemption threshold is lower in task response time, resulting in the tasks missing deadline. For the flaws above, this paper presents an improved response time analysis method for scheduling tasks with preemption threshold. Taking into account of the effect and time demand of release jitter and tick scheduling by using the modified task parameters in the computation of the time-demand function of task, simulation result shows that the improved method can obtain more exact analysis result using preemption threshold scheduling.

【Key words】 Preemption threshold; Tick scheduling; Response time; Schedulability; Release jitter

任务响应时间分析是经典实时调度理论研究的核心问题之一。文献[1]全面讨论了速率单调调度可调度条件及相关算法，扩展了RM算法的适用范围。目前，抢占阈值[2]调度技术是一种比较新的实时系统任务调度技术，与固定优先级抢占调度非常类似，但其克服了抢占调度对系统的响应和吞吐量的负面影响，弥补了固定优先级抢占与非抢占调度技术存在的缺陷。因此，抢占阈值调度技术得到实时系统研究人员的极大关注。文献[2]详细描述了抢占阈值调度方法，与抢占和非抢占调度模型相比，任务集可调度性有了显著改进。同时利用π_i级繁忙区间概念，给出最坏情况响应时间计算公式以及相关优先级和抢占阈值最优分配算法。然而，文献[3]没有考虑到抢占阈值调度下任务释放抖动和嘀嗒(Tick)调度对计算任务响应时间的影响，任务时间需求考虑不够全面完善可能会导致任务低估任务响应时间而错过期限。

本文针对上述研究的不足，提出考虑释放抖动和嘀嗒调度情况下固定优先级任务响应时间计算方法，修正了文献[3]中关于响应时间计算的不足并且通过提出改进的任务参数，考虑因嘀嗒调度导致的额外时间需求。仿真试验结构表明，改进的方法与单纯计算基于抢占阈值调度的固定优先级任务响应时间的方法相比，有效提高了任务集可调度利用率。

1 基于抢占阈值调度的响应时间分析

为了确定一个任务是否满足它的期限，首先，以从该任务的临界时刻开始的时间函数为参数，计算该任务在临界时刻释放的作业以及所有较高优先级任务的总处理器时间需求函数。接着，在每个作业的期限到达之前，检测这个时间需求能否被满足。称这个测试为时间需求分析。

定义 1 称系统中每个任务τ_i的第 1 个作业$j_{i,1}$的释放时间$r_{i,1}$为任务τ_i的相位，记作Φ_i, $\Phi_i = r_{i,1}$。一般，不同任务有不同相位。某些任务同相是指它们有相同的相位。

定义 2 临界时刻是指：(1)如果任务τ_i的每个作业的响应时间都小于或者等于τ_i 的相对期限D_i，那么在该临界时刻释放的τ_i 中的作业具有所有作业的最大响应时间；(2)如果τ_i中有些作业的响应时间超过D_i，则在该临界时刻释放的作业响应时间大于D_i。

定义 3 一个π_i级繁忙区间$(t_0,t]$的开始时间是t_0，当：(1)在该时间以前，τ_i 中被释放的所有作业都已完成；(2) τ_i中有一个作业在该时间释放。该区间的结束时间是从t_0开始，τ_i中被释放的所有作业第一次全部完成的时刻t。即在区间$(t_0,t]$，处理器一直忙于执行优先级等于或高于π_i的作业，并且在繁忙区间执行的作业都是在该区间内释放的，且当该区间结束时，没有积压需要后续执行的作业。

1.1 π_i级繁忙区分析

繁忙区分析研究对象是一个任务实例到达到完成的多线程调度细节。在传统可抢占固定优先级模型中，设u为任务τ_i的一个实例，任务τ_i的繁忙区计算公式如下：

基金项目：黑龙江省自然科学基金资助项目(F2005-02)
作者简介：王 涛(1973－)，男，博士生，主研方向：实时系统调度；刘大昕、张健沛，教授、博导
收稿日期：2006-06-13 **E-mail：**wt730620@126.com

$$w_i(u)=u\cdot C_i+\sum_{\forall j,\pi_j>\pi_i}\left\lceil\frac{w_i(u)}{T_j}\right\rceil\cdot C_j \tag{1}$$

一个任务最坏情况阻塞时间为

$$B_i=\max_{\forall j:P_j\ge P_i>P_j} C_i \tag{2}$$

改进的响应时间计算方法与文献[3]中方法主要有两点不同：

(1)在计算一个任务响应时间时，将其以前的请求考虑进来，使用一个不同的循环终止条件；

(2)加入了对任务释放时间抖动的计算。

S_i为任务τ_i最坏情况开始时间，加入任务释放抖动时间的计算公式：

$$S_i(u)=B_i+u\cdot C_i+\sum_{\forall j:P_j>P_i}(1+\left\lfloor\frac{S_i(u)+J_j}{T_j}\right\rfloor)\cdot C_j \tag{3}$$

式中，F_i为任务τ_i最坏情况完成时间，加入任务释放抖动时间的计算公式：

$$F_i(u)=S_i(u)+C_i+\sum_{\forall j:P_j>\overline{P_i}}(\left\lceil\frac{F_i(u)+J_j}{T_j}\right\rceil-(1+\left\lfloor\frac{S_i(u)+J_j}{T_j}\right\rfloor))\cdot C_j \tag{4}$$

基于以上改进得到：

定理 改进的任务τ_i最坏情况响应时间为

$$r_i=\max_{\forall u:0\le u\le Q}(F_i(u)+J_i-uT_i) \tag{5}$$

其中，$Q=\lfloor L_i/T_i\rfloor$[5]，L_i为抢占阈值调度下最长的π_i级繁忙区，并且

$$L_i=B_i+\sum_{\forall j:P_j\ge P_i}\left\lceil\frac{L_i+J_j}{T_j}\right\rceil\cdot C_j \tag{6}$$

1.2 嘀嗒调度对响应时间的影响

计算任务τ_i的响应时间时，可以使用以下改进的任务参数，来考虑因嘀嗒调度引入的额外时间[3]：

假设p_0表示嘀嗒的大小，即连续两次时钟中断的时间长度，用周期为p_0的周期任务模拟调度程序，该任务具有系统中最高的优先级，其执行时间e_0为调度程序处理时钟中断的时间；用CS_0表示调度程序把一个作业从未决队列转移到就绪队列的最大时间开销。

(1)在较高优先级任务集合中加入任务$\tau_0=(p_0,e_0)$;

(2)每个较高优先级任务$\tau_k(k=1,2,\ldots,i)$的执行时间e_k增加$(K_k+1)\ CS_0$。其中K_k为τ_k可能自挂起的次数；

(3)对于每个较低优先级的任务$\tau_k(k=i+1,\ldots,n)$,在较高优先级任务集中增加一个任务(p_k,CS_0);

(4)τ_i因低优先级任务的不可抢占而产生的阻塞时间为

$$b_i(np)=(\left|\max_{i+1\le k\le n}\theta_k/p_0\right|+1)p_0 \tag{7}$$

其中，θ_k为较低优先级任务τ_k的不可抢占部分的最大执行时间。

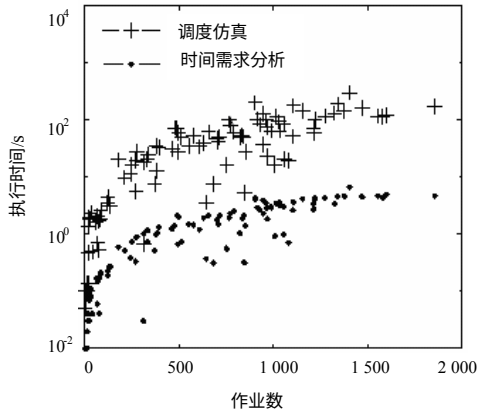
2 仿真试验结果

为测试改进的抢占阈值调度任务响应时间分析方法及验证使用时间需求分析方法对任务集可调度性测试的有效性，在 P4 1.8Ghz, 512MB RAM, Matlab7.0 中实现本文基于时间需求分析的方法，并与传统模拟调度进行比较。传统的模拟方法是执行一个调度仿真，任务集具有给定的先验知识，如释放时间、执行时间、优先级等。

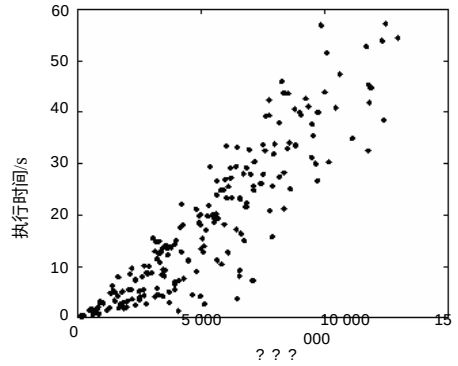
两种方法的对比严格按照试验进行，并没有进行理论复杂性分析，并且两种方法使用相同的随机产生的任务集，对方法所需执行时间进行分析比较。

试验分两组，第 1 组针对两种方法执行时间随机产生 100 个任务集。每个任务集所含任务数为在区间[2,15]中给定的随机数，任务周期从集合{1,5,15,30,60,100}中随机挑选，最大超周期为 300，任务执行时间和最大抖动及嘀嗒大小均随机产生，执行时间按比例使任务集利用率达到 90%，抖动控制在[0,3)之间，与相对较大的任务周期相比，这样随意选择的抖动范围不会对分析结果产生过大的影响。图 1(a)描述 100 个任务集两种方法下的执行时间表现，方法的分析时间与作业数量成反比，表现出分析时间对作业数的强烈依赖。图 1(a)显示出基于时间需求分析的方法具有更快的执行时间。

第 2 组试验针对更大周期范围的大任务集考察方法执行时间与作业数之间的依赖程度。任务周期在{1,5,10,15,20,30,50,60,80,100}中随机选择，最大超周期为 1 200。任务集所含任务数在区间[10,50]内。由于调度仿真方法表现过差，本组试验仅考察了基于时间需求分析的方法，该方法在本组试验中最大执行时间是 58s，并且执行时间与作业数表现出接近线性依赖关系，如图 1(b)所示。



(a) 执行时间比较



(b) 50 任务的作业时间需求

图 1 仿真结果

上述两组试验结果表明基于响应时间需求分析的任务可调度性检测方法性能优于传统调度仿真方法，同时反映出这种方法可以应用于分布式大任务集的可调度性分析中。

3 结论

任务响应时间分析是实时系统任务可调度性判定的核心技术，本文在新型抢占阈值调度模型下考虑任务释放抖动和时钟嘀嗒调度对任务响应时间计算的影响，提出一种考虑较完整的周期任务响应时间计算方法，并通过实例验证了方法的有效性，在以前工作的基础上为实时系统任务可调度性分析增加了一种有效的计算方法。

（下转第 36 页）