

一种任务优先级的综合设计方法*

金宏⁺, 王宏安, 王强, 戴国忠

(中国科学院 软件研究所 人机交互技术与智能信息处理实验室, 北京 100080)

An Integrated Design Method of Task Priority

JIN Hong⁺, WANG Hong-An, WANG Qiang, DAI Guo-Zhong

(Human Computer Interaction and Intelligent Information Processing Laboratory, Institute of Software, The Chinese Academy of Sciences, Beijing 100080, China)

+ Corresponding author: Phn: 86-10-62559307 ext 8807, E-mail: hjin@iel.iscas.ac.cn; hjin@ihw.com.cn

<http://iel.iscas.ac.cn>

Received 2002-06-11; Accepted 2002-08-23

Jin H, Wang HA, Wang Q, Dai GZ. An integrated design method of task priority. *Journal of Software*, 2003,14(3):376-382.

Abstract: A scheduling algorithm based on priority table design is presented in this paper. Any two characteristic parameters (e.g., relative deadline and slack) of a task are combined to design its priority table so that the deadline is nearer or the slack is shorter, the priority is higher. The priority of a task is uniquely determined by its relative deadline and slack. For any task, its unique priority can be obtained by using Lagrange interposing algorithm on the designed priority table. Compared with the classical EDF and LSF policies, simulated results show that the proposed algorithm improves the efficiency of task scheduling, i.e., to designate the priority of a task, increases the succeed ratio of task scheduling, and decreases the missed deadline percentage. The proposed algorithm can be applied to dynamically schedule real-time tasks in real-time systems.

Key words: real-time system; dynamic scheduling; scheduling policy; task priority; scheduling success ratio

摘要: 提出了一种基于优先级表设计的调度算法. 将任务的相对截止期和空闲时间这两个特征参数结合起来, 综合设计任务的优先级表, 使得截止期越早或空闲时间越短, 任务的优先级越高, 而且任务的优先级由相对截止期和空闲时间惟一确定. 对于任意一个任务, 可通过对设计的优先级表进行二元多点插值获得相应任务的惟一优先级. 与传统的 EDF 和 LSF 算法进行仿真比较, 仿真结果表明, 通过优先级表设计方法来确定任务的优先级, 提高了任务调度的成功率, 降低了任务截止期的错失率. 该方法可应用于实时系统中实时任务的动态调度中.

关键词: 实时系统; 动态调度; 调度策略; 任务优先级; 调度成功率

中图法分类号: TP311 文献标识码: A

实时系统是一个用来处理具有定时限制的工作负载的任务处理系统. 任务处理的主要目标是要满足任务

* Supported by the National Natural Science Foundation of China under Grant No.69896250 (国家自然科学基金); the National High-Tech Research and Development Plan of China under Grant No.2001AA413020 (国家高技术研究发展计划)

第一作者简介: 金宏(1962-), 男, 江苏如皋人, 博士, 副研究员, 主要研究领域为实时系统, 故障诊断, 神经网络应用.

的截止时间.如对实时数据库系统来说,这些工作主要包括事务对各种资源的调度^[1],即对多个事务占用 CPU 的调度,多个并发运行事务之间的数据、I/O 和内存等资源的调度,调度各种资源的使用对实时数据库的功能和性能都有极大的影响.

在实时任务调度中,大量研究都集中于对调度策略的应用,调度策略中使用最多的是优先级调度法.当实时任务到达时,指派一个优先级,进入任务队列.在任务队列中,优先级最高的任务排在第一位,由高到低顺序排列.若考虑影响调度的两个主要因素是任务到达时间和任务截止时间,那么在指派任务优先级时,要综合考虑这两个因素,使超过截止时间和任务夭折的情况尽可能少.

基于优先级的调度方法主要有单调速率算法^[2]、截止期最早最优先(earliest deadline first,简称 EDF)^[2,3]、空闲时间最短最优先(least slack first,简称 LSF)^[1]、最早放行最优先^[1]、可达截止期最早最优先^[1]、价值最高最优先^[4]、价值密度最大最优先^[4,5]等策略,其中,EDF 和 LSF 是讨论最为广泛的两种策略,本文将以这两种策略作为讨论的比较对象.

任务的优先级通常由任务的一个或多个特征参数来确定,如截止期、空闲时间或关键性(任务的重要程度)等.如果优先级仅仅是由某个特征参数来确定是不够的^[6],如 EDF 策略将最高优先级指派给具有截止期最早的任务,LSF 策略将最高优先级指派给具有空闲时间最短的任务.因为截止期短的任务不一定是非常关键的,其空闲时间也不一定短,反之亦然,所以需要研究至少利用两个特征作为任务的独立特征参数,建立任务优先级与这些特征参数之间的、一对一的对应关系^[7,8].目前的许多研究比较集中在综合关键性与截止期这两个任务属性上,如关键性-截止期优先(基于相对截止期与关键性之比值来分配优先级)^[7]和加权的最早截止期价值密度优先^[9].这些方法的主要不足在于一个任务与其所确定的优先级值可能不是一一对应的,即不同的任务可能有相同的优先级.

本文提出一种基于优先级表设计的调度方法,简记为 PTD(priority table design),根据这种设计,对于具有不完全相同属性的两个任务,它们所分配得到的优先级值不同.不失一般性,本文在讨论优先级表设计时考虑了任务的相对截止期和空闲时间两个特征参数,这种设计思想可以推广到对其他任意两种或多种特征参数进行的综合设计中.在后面的仿真举例中,将本文提出的 PTD 算法与传统的 EDF 和 LSF 算法相比较,任务截止期的错失率作为衡量不同调度算法优劣的性能指标.

1 优先级表

优先级分派策略可以看成是一个函数,适用于单个任务/事务或一个任务/事务组.当用于单个任务时,函数的结果就是对应于分派策略所确定的、该任务的优先级;当用于任务组时,函数的结果是这些任务的一个排序表,在实施调度时,优先级最高者排第一,最低者排最后.

1.1 空闲时间与相对截止期

记一个任务为 $T(a, de, c)$,其中,属性 a, de 和 c 分别表示任务的到达时间、截止期和剩余执行时间.对于抢占式动态调度来说, c 是随时间变化的,并可以理解为任务 T 在区间 $[t, de)$ 上执行 c 个单位时间, t 表示当前时刻且 $a \leq t < de$.所谓任务 T 的空闲时间定义为 $s = de - (t + c)$,因此,任务的空闲时间是动态变化的.

任务的相对截止期是指任务到达时间和截止期之差,在抢占式动态调度过程中,这种相对截止期可定义成任务的截止期与当前时刻之差(即, $d = de - t$),也是随时间变化的.根据空闲时间的定义,由于任务的剩余执行时间大于零,因此,对于一个硬实时任务来讲,只有当它的空闲时间小于其相对截止期时才有调度意义,或有讨论其优先级的必要性.需要说明的是,相对截止期的早晚与截止期的早晚是一致的.

1.2 优先级表的定义

将任务的相对截止期(d)和空闲时间(s)这两个参数的取值范围划分成若干个不同的取值区间,每个区间选择一个典型的值来代表这个区间.不失一般性,考虑相对截止期的 m 个典型的值 d_1, d_2, \dots, d_m (满足 $d_1 < d_2 < \dots < d_m$),和空闲时间的 n 个典型值 s_1, s_2, \dots, s_n (满足 $s_1 < s_2 < \dots < s_n$).对于具有不同的典型相对截止期 d_i 和典型空闲时间 s_j 的任务 T_{ij} ,赋予其特定的优先级值 $P(T_{ij}, d_i, s_j)$, ($i=1, \dots, m, j=1, \dots, n$),简记为 P_{ij} ,从而得到事先能够确定的优先级设

计表 $P=(P_{ij})$. 对于具有任一相对截止期 d 和空闲时间 s 的任务, 当 d 和 s 是典型相对截止期之一和典型空闲时间值之一时, 则查表可获得该任务的优先级; 当 d 和 s 中至少有一个不是典型值时, 则通过对优先级设计表进行插值得到该任务的优先级, 插值算法将在第 4 节给出. 而如何对典型的任务赋予其特定的优先级值是本文讨论的关键.

1.3 EDF和LSF策略的优先级表

根据 EDF 定义, 任务 T_{ij} 的优先级可以定义为 $P(T_{ij}, d_i, s_j)=d_i$; 根据 LSF 定义, T_{ij} 的优先级可以定义为 $P(T_{ij}, d_i, s_j)=s_j$. 为了分析简便起见, 这里将任务的优先级值重新定义为相对截止期的下标或空闲时间的下标, 也即, 若按 EDF 策略, $P(T_{ij}, d_i, s_j)=i$, 或按 LSF 策略, $P(T_{ij}, d_i, s_j)=j$.

不失一般性, 取 $m=n=8$, 假设任务 T_{ij} 在 $j \leq i$ 时才有优先级, 图 1 和图 2 分别给出基于 EDF 和 LSF 策略的优先级表. 如图 1 所示, 对于具有不同相对截止期的任务, 截止期越早, 任务的优先级就越高; 对于具有同一相对截止期的任务, 无论它们的空闲时间有多长, 它们的优先级都是相同的. 如图 2 所示, 对于具有不同空闲时间的任务, 空闲时间越短, 任务的优先级就越高; 对于具有同一空闲时间的任务, 无论它们的截止期的早晚, 它们的优先级都是相同的. 由此可见, EDF 和 LSF 只考虑任务的一个特征参数, 要么是截止期, 要么是空闲时间, 这使得对于具有不完全相同截止期或空闲时间的任务, 它们所分配得到的优先级不是惟一的. 在后面的讨论中, 我们将讨论对相对截止期和空闲时间进行线性加权平均方法以及基于优先级设计的综合设计方法, 在对优先级表设计的过程中, 也同样考虑用整数作为一个任务的优先级值来讨论.

	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s
d_1	1								
d_2	2	2							
d_3	3	3	3						
d_4	4	4	4	4					
d_5	5	5	5	5	5				
d_6	6	6	6	6	6	6			
d_7	7	7	7	7	7	7	7		
d_8	8	8	8	8	8	8	8	8	
d									

Fig.1 EDF Priority Table

Fig.1 EDF priority table
图 1 EDF 策略优先级表

	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s
d_1	1								
d_2	1	2							
d_3	1	2	3						
d_4	1	2	3	4					
d_5	1	2	3	4	5				
d_6	1	2	3	4	5	6			
d_7	1	2	3	4	5	6	7		
d_8	1	2	3	4	5	6	7	8	
d									

Fig.2 LSF Priority Table

Fig.2 LSF priority table
图 2 LSF 策略优先级表

2 线性加权综合

考虑任务 T 的优先级是其相对截止期 d 和空闲时间 s 的线性加权平均:

$$P(T, d, s) = \alpha * d + (1 - \alpha) * s, \tag{1}$$

其中 $0 \leq \alpha \leq 1$ 为加权系数. 显然当 $\alpha=1$ 时, 变成 EDF 策略; 当 $\alpha=0$ 时, 变成 LSF 策略. 此外, 若取 α 为任务已完成工作的部分, 称为混合 (hybrid) 方法, 此时, 任务已完成的工作越多, 其相对截止期的系数就越大. 当任务已完成工作不到一半 ($\leq 50\%$) 时, 取 $\alpha=0$, 即此时只考虑空闲时间参数, 否则取 $\alpha=1$, 此方法称为一半对一半 (half-half) 方法. 后两种方法只有在任务的工作量事先知道的前提下才能使用^[10].

这种线性加权方案的不足: ① 加权系数的选择问题; ② 任务的优先级不是惟一的. 事实上, 考虑两个任务 T_1 和 T_2 , 其中, 任务 T_1 的相对截止期和空闲时间分别是 d_1 和 s_1 , 任务 T_2 的相对截止期和空闲时间分别是 d_2 和 s_2 . 假设 $\alpha \neq 0$, 则当这两个任务的相对截止期差和空闲时间差之间存在一定的线性关系时, 即

$$d_1 - d_2 = -(s_1 - s_2)(1 - \alpha) / \alpha, \tag{2}$$

由式(1)确定的优先级完全相同, 这也是我们提出通过优先级表设计来惟一确定任务优先级的原因.

3 优先级表设计方法 PTD

在综合考虑任务相对截止期和空闲时间两个特征参数时,需要考虑下面两个方面的设计原则:

(1) 截止期越早和空闲时间越短,任务的优先级就越高;

(2) 对于任意两个任务,在它们的相对截止期和空闲时间两个参数中,只要至少有一个参数不同,那么它们的优先级值也应该不同,即任务的优先级分配是惟一的。

3.1 优先级表的设计

3.1.1 优先级等级

任务的优先级等级与相对截止期、空闲时间之间的层次关系可以用一种线性关系来描述,比如,

$$d_i + ks_j = g, \tag{3}$$

其中, k 为线性关系的斜率, g 可以看成是优先级等级. 对于确定的斜率 k , g 越小, 优先级等级越高. 比如, 对于 $g_1 < g_2$, 满足 $d_i + ks_j = g_1$ 的任务优先级要高于满足 $d_i + ks_j = g_2$ 的任务优先级. 图 3 中带箭头的斜向上直线表示一种不同等级的排序, 越往左上, 等级越高.

3.1.2 优先级设计表

不失一般性, 在第 1.3 节的假设条件下, 式(3)可以简单地用下式来代替:

$$i + j = g, \tag{4}$$

其中 i 和 j 分别为截止期 d_i 和空闲时间 s_j 的下标. 这时, 按式(4)的等级来设计任务的优先级, 图 4 给出其中一种优先级设计方案表. 图 3 和图 4 给出的优先级是在满足式(4)的条件下, 随空闲时间递增; 此外, 我们也可以在满足式(4)的条件下, 选择优先级随相对截止期递增, 或在优先级等级高的部分随空闲时间递增, 在优先级等级低的部分随相对截止期递增等.

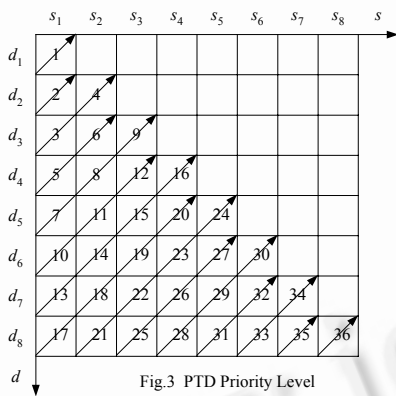


Fig.3 PTD Priority Level

Fig.3 PTD priority level

图 3 PTD 策略优先级等级图

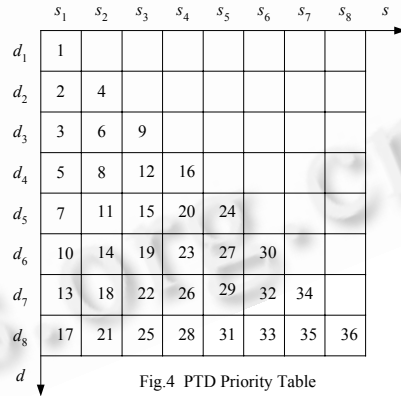


Fig.4 PTD Priority Table

Fig.4 PTD priority table

图 4 PTD 策略优先级表

3.2 优先级分配公式

$\forall (d_i, s_j)$, 任务 T_{ij} 的优先级可以定义成

$$P(T_{ij}, d_i, s_j) = j + \sum_{k=0}^{i+j-2} a_k, \quad \text{当 } i+j-1 \leq m \text{ 时}, \tag{5}$$

$$P(T_{ij}, d_i, s_j) = m + 1 - i + \sum_{k=0}^{i+j-2} a_k, \quad \text{当 } i+j-1 > m \text{ 时}, \tag{6}$$

其中 $a_0=0$; a_i 由下面确定(记 m_0 为正整数):

(1) 当 $m=2m_0$ (偶数) 时, $a_i=j, i=2j-1, 2j=1, \dots, m_0-1; a_{m-1}=m_0, a_m=m_0; a_{m+i}=a_{m-i}, 0 < i < m$.

(2) 当 $m=2m_0-1$ (奇数) 时, $a_i=j, i=2j-1, 2j=1, \dots, m_0-1; a_m=m_0; a_{m+i}=a_{m-i}, 0 < i < m$.

4 优先级插值

前面对于优先级表的讨论都是针对任务具有特定的相对截止期和特定的空闲时间来进行优先级设计的。对于任一任务 T ,其相对截止期 u 或空闲时间 v 不一定是特定的相对截止期或特定的空闲时间,这时 T 的优先级可通过对事先确定的优先级表进行线性插值得到。不妨假设 $d_1 \leq u \leq d_m, s_1 \leq v \leq d_n$,下面介绍采用二元三点插值公式来计算 T 的优先级 $P(T,u,v)$ 的方法。

对于相对截止期 u ,我们可以在 m 个典型相对截止期 d_1, d_2, \dots, d_m 中找到离 u 值最近的 3 个典型相对截止期,记为 $d_i(i=p, p+1 \text{ 和 } p+2)$; 同样地,对于空闲时间 v ,我们也可以在 n 个典型空闲时间 s_1, s_2, \dots, s_n 中找到离 v 值最近的 3 个典型空闲时间,记为 $s_j(j=q, q+1, q+2)$ 。这样,我们可以得到点 (u,v) 周围的 9 个点 $(d_i, s_j), (i=p, p+1, p+2; j=q, q+1, q+2)$,每个点对应一个典型的任务 T_{ij} ,每个任务 T_{ij} 具有自己惟一分配的优先级值 $P(T_{ij}, d_i, s_j)$ 。然后利用这 9 个点对应的任务优先级值进行 Lagrange 插值计算 $P(T,u,v)$:

$$P(T, u, v) = \sum_{i=p}^{p+2} \sum_{j=q}^{q+2} \left(\prod_{\substack{k=p \\ k \neq i}}^{p+2} \frac{u - d_k}{d_i - d_k} \right) \left(\prod_{\substack{l=q \\ l \neq j}}^{q+2} \frac{v - s_l}{s_j - s_l} \right) P(T_{ij}, d_i, s_j). \tag{7}$$

5 举例

在仿真中,考虑使用截止期错失率(missed deadline percentage,简称 MDP)作为衡量调度策略好坏的一个性能指标。所谓截止期错失率就是指系统中未被调度成功的任务个数与参加调度的任务个数之比,它与调度成功率成反比,截止期错失率越高,则调度成功率越低。

5.1 特定任务的调度

考虑 5 个任务: $T_1(0,4,6), T_2(1,2,9), T_3(1,6,10), T_4(1,5,9)$ 和 $T_5(2,3,7)$ 。调度原则:在相同的相对截止期和空闲时间条件下,按谁先到就先调度谁;如果到达时间也相同,则按先后次序调度的原则。

(1) 按 EDF 策略进行调度,调度过程如图 5(上)所示,调度结果是任务 T_1 和 T_5 得到提交,其他 3 个任务 T_2, T_3 和 T_4 错过截止期,截止期错失率达到 60%。

(2) 按 LSF 策略进行调度,调度过程如图 5(中)所示,调度结果是任务 T_1 和 T_2 得到提交,其他 3 个任务 T_3, T_4 和 T_5 错过截止期,截止期错失率达到 60%。表 1 给出的是从 $t=0$ 到 $t=5$ 时刻 5 个任务的空闲时间,其中*号表示任务还没有到达,空闲时间为负值表示该任务在其截止期前已不能提交。由表 1 可见,任务的空闲时间是动态变化的。

(3) 按 PTD 策略进行调度,调度过程如图 5(下)所示,调度结果是任务 T_1, T_2 和 T_5 得到提交,其他两个任务 T_3 和 T_4 错过截止期,截止期错失率降低到 40%。

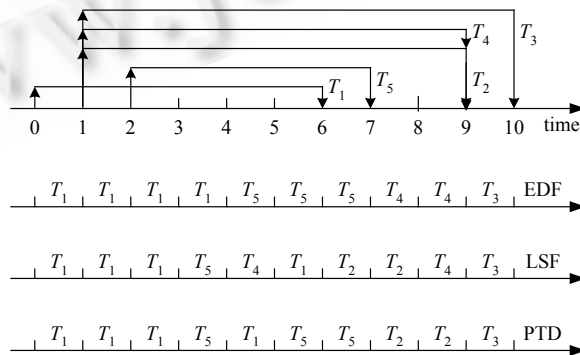


Fig.5 Scheduling sequences based on EDF (above), LSF (middle) and PTD (below)

图 5 基于 EDF 策略(上)、LSF 策略(中)和 PTD 策略(下)的调度次序

Table 1 Slack time of five tasks from $t=0$ to $t=5$

表 1 从 $t=0$ 到 $t=5$ 时刻 5 个任务的空闲时间

	Task T_1	Task T_2	Task T_3	Task T_4	Task T_5
$t=0$	2	*	*	*	*
$t=1$	2	6	3	3	*
$t=2$	2	5	2	2	2
$t=3$	2	4	1	1	1
$t=4$	1	3	0	0	1
$t=5$	0	2	-1	0	0

5.2 错失率统计与比较

记 $U(x,y)$ 表示在 x 和 y 之间的均匀随机整数.取 $a=U(0,3),de=U(5,10),c=fix(1+rand*(de-a-1))$,其中 fix 表示取整函数, $rand$ 为 0 和 1 之间的随机数.共仿真 10 批,每批 100 组,每组 10 个任务.对每组 10 个任务,分别采用 EDF,LSF 和 PTD 策略进行调度,并各自统计一次 MDP.针对这 3 种策略,分别统计每批 MDP 的平均值(见表 2)及其散布的标准差(见表 3).为了节省篇幅,表 2 和表 3 只给出前 5 批的统计结果.

记 MDP_{EDF} 为基于 EDF 的 MDP, MDP_{LSF} 为基于 LSF 的 MDP, MDP_{PTD} 为基于 PTD 的 MDP.图 6 和图 7 分别给出某批的 $MDP_{EDF}-MDP_{PTD}$ 曲线和 $MDP_{LSF}-MDP_{PTD}$ 曲线.由图 6 可知,在该批的 100 组任务中,满足 $MDP_{EDF}-MDP_{PTD}<0$ 的个数为 4,等于 0 的个数为 18,大于 0 的个数为 78(其中,41(0.1),29(0.2),7(0.3),1(0.4),括号内的数据是截止期错失率的差值,括号外的数据为该差值的个数).由图 7 可知,在此 100 组中,满足 $MDP_{LSF}-MDP_{PTD}<0$ 的个数为 0,等于 0 的个数为 83,大于 0 的个数为 17(其中,16(0.1),1(0.2)).

Table 2 Mean of MDP

表 2 截止期错失率的平均值

Batch	EDF	LSF	PTD
1	0.790 0	0.688 0	0.670 0
2	0.788 0	0.684 0	0.663 0
3	0.800 0	0.689 0	0.658 0
4	0.779 0	0.673 0	0.656 0
5	0.757 0	0.676 0	0.649 0

Table 3 Standard deviation of MDP

表 3 截止期错失率的标准差

Batch	EDF	LSF	PTD
1	0.113 3	0.089 1	0.090 5
2	0.110 4	0.100 2	0.103 1
3	0.118 9	0.092 0	0.111 2
4	0.110 4	0.102 4	0.110 4
5	0.124 1	0.101 6	0.114 1

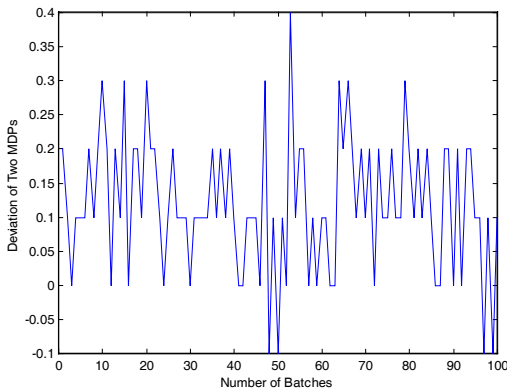


Fig.6 A curve of the deviation MDPEDF-MDPPTD

图 6 偏差 MDPEDF-MDPPTD 的曲线

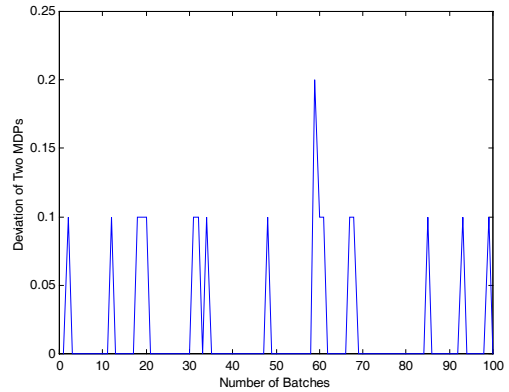


Fig.7 A curve of the deviation MDPLSF-MDPPTD

图 7 偏差 MDPLSF-MDPPTD 的曲线

(1) 基于 EDF 策略的调度, MDP_{EDF} 的平均值是在 $0.782 9\pm$,其散布的标准差在 $0.117 6\pm$;基于 LSF 策略的调度, MDP_{LSF} 的平均值是在 $0.685 2\pm$,其散布的标准差在 $0.096 4\pm$;基于 PTD 策略的调度, MDP_{PTD} 的平均值是在 $0.658 5\pm$,其散布的标准差在 $0.105 4\pm$.由此可见, MDP_{PTD} 比 MDP_{EDF} 平均小 12.44%, MDP_{PTD} 比 MDP_{LSF} 平均小 2.67%.

(2) 由图 6 可知,与 EDF 相比,通过 PTD 调度,除个别几组任务以外,大多数任务组的截止期错失率得到降低,在这 100 组任务中,得到降低的比例达到 78%,持平的比例达到 18%,而升高的比例只有 4%.

(3) 由图 7 可知,与 LSF 相比,通过 PTD 调度,在这 100 组任务中,截止期错失率得到降低的比例达到 17%,持平的比例达到 83%,但没有截止期错失率升高的现象出现。

由此可见,通过对优先级表的综合设计,任务的截止期错失率大大低于 EDF 的截止期错失率.但与 LSF 的截止期错失率相比,改进不是很突出的主要原因是:采用的优先级表(图 3 和图 4)是在满足式(4)的条件下,任务的优先级在相同等级的优先级层次上随空闲时间递增。

6 结 论

本文讨论了综合考虑任务的相对截止期和空闲时间两个特征参数的优先级表设计方法,所设计的任务优先级与不同的任务是一一对应的.首先对典型的特征参数,设计任务的优先级表;然后,对任意不同的特征参数,进行优先级插值,获得不同任务的惟一优先级值。

为了综合考虑这两个特征,使得截止期越早或空闲时间越短时,任务的优先级也越高,我们按相对截止期和空闲时间之间的一定线性关系来设置任务的优先级等级,在同一等级层次上,再按一定的规律分配任务的优先级,如截止期越早优先级越高,或空闲时间越短优先级越高.这种设计方法也适用于对其他不同的两种特征参数之间的综合.仿真结果表明,通过对优先级表的综合设计,任务的截止期错失率得到大大降低,从而任务调度的成功率得到很大提高。

References:

- [1] Liu YS, He XG, Tang CJ, Li L. Special Type Database Technology. Beijing: Science Press, 2000 (in Chinese).
- [2] Liu C, Layland J. Scheduling algorithms for multiprogramming in a hard real-time environment. *Journal of the ACM*, 1973,20(1): 46~61.
- [3] Haritsa JR, Livny M, Carey MJ. Earliest deadline scheduling for real-time database systems. In: *Proceedings of the 12th IEEE Real-Time Systems Symposium*. Los Alamitos, CA: IEEE Computer Society Press, 1991. 232~243.
- [4] Jensen ED, Locke CD, Toduda H. A time-driven scheduling model for real-time operating systems. In: *Proceedings of the IEEE Real-Time Systems Symposium*. Washington, DC: IEEE Computer Society Press, 1985. 112~122.
- [5] Abbott R, Garcia-Molina H. Scheduling real-time transactions. *ACM SIGMOD Record*, 1988,17(1):71~81.
- [6] Biyabani SR, Stankovic JA, Ramamritham K. The integration of deadline and criticalness in hard real-time scheduling. In: *Proceedings of the 9th IEEE Real-Time Systems Symposium*. Washington, DC: IEEE Computer Society Press, 1988. 152~160.
- [7] Huang J, Stankovic JA, Towsley DF, Ramamritham K. Experimental evaluation of real-time transaction processing. In: *Proceedings of the 10th IEEE Real-Time Systems Symposium*. Los Alamitos, CA: IEEE Computer Society Press, 1989. 144~153.
- [8] Kao B, Garcia-Molina H. An overview of real-time database systems. In: Son SH, ed. *Advances in Real-Time Systems*. Englewood Cliffs, NJ: Prentice Hall Inc., 1995. 463~486.
- [9] Buttazzo G, Spuri M, Sensini F. Value VS. deadline scheduling in overload conditions. In: *Proceedings of the 16th IEEE Real-Time Systems Symposium*. Los Alamitos, CA: IEEE Computer Society Press, 1995. 90~99.
- [10] Ramamritham K, Sivasankaran R, Stankovic JA, Towsley DF, Xiong M. Integrating temporal, real-time, and active databases. *ACM SIGMOD Record*, 1996,25(1):8~12.

附中文参考文献:

- [1] 刘云生,何新贵,唐常杰,李霖.特种数据库技术.北京:科学出版社,2000.