一种有限优先级的静态优先级分配算法*

宾雪莲[1,2+], 杨玉海[2], 金士尧[1]

[1](国防科学技术大学 并行与分布国家重点实验室,湖南 长沙 410073)

[2](空军雷达学院 计算机教研室,湖北 武汉 430010)

An Assignment Algorithm of Static Priority for Limited Priority Levels

BIN Xue-Lian[1,2+], YANG Yu-Hai[2], JIN Shi-Yao[1]

[1](National Laboratory for Parallel and Distributed Processing, National University of Defense Technology, Changsha 410073, China)

[2](Staff Room of Computer, Airforce Army Radar Academy, Wuhan 430010, China)

+ Corresponding author: E-mail: yhyang@wtwh.com.cn, http://www.nudt.edu.cn

Received 2003-05-12; Accepted 2004-01-14

Bin XL, Yang YH, Jin SY. An assignment algorithm of static priority for limited priority levels. *Journal of Software*, 2004,15(6):815~822.

http://www.jos.org.cn/1000-9825/15/815.htm

Abstract: Static priority scheduling is widely used in real-time systems. But its schedulability will be reduced if priority levels of the system are insufficient. A task set may require more priority levels than the system can support. In this case, more than one task must be grouped into the same priority. This paper presents necessary and sufficient conditions for analyzing the schedulability of static priority algorithms on resources with limited priority levels. A static priority assignment algorithm (AGP) with limited priority levels is developed. As it turns out, AGP is optimal for the basic task set in the sense that the number of priority levels required by AGP is minimal and no other static priority rule can schedule a basic task set which cannot scheduled by AGP. Simulation results show that the schedulability of AGP is much higher than that of Constant Ratio Grid algorithm. AGP is significant for solving the problem of assigning priorities of tasks in embedded real-time systems.

Key words: real-time system; limited priority level; schedulability; static priority scheduling; task priority

摘 要: 静态优先级调度在实时系统中得到了广泛应用.然而,静态优先级调度受到系统支持的优先级个数的限制.当任务的个数大于优先级个数时,需要将多个任务映射到同一个优先级.针对优先级个数有限的情况,给出了在截止期限大于周期时任务可调度的充分必要条件,并提出了基于有限优先级的静态优先级分配算法(AGP).AGP 算法对于基本任务集合是最优的静态优先级分配算法.其最优性表现在,所需的优先级个数最小,并且若采用 AGP 算法不可调度某个任务集,则采用其他静态优先级分配算法也不可调度该任务集.模拟结果表明,AGP 算法的可调度性要远远大于常量法. AGP 算法对于解决在嵌入式实时系统中任务的优先级分配问题具有重要意义.

* Supported by the National Natural Science Foundation of China under Grant No.60073003 (国家自然科学基金)

作者简介: **宾雪莲**(1974－),女,湖南湘潭人,博士生,主要研究领域为实时调度技术;**杨玉海**(1971－),男,讲师,主要研究领域为计算机网络技术,智能化高速存储系统;**金士尧**(1937－),男,教授,博士生导师,主要研究领域为仿真、实时、并行与分布系统并行计算.

关键词: 实时系统;有限优先级;可调度性;静态优先级调度;任务优先级
中图法分类号: TP316 文献标识码: A

调度理论是实时系统需要研究的最重要的问题之一.实时调度算法主要分为两类:静态调度算法和动态调度算法.静态调度算法实现简单,调度的额外开销小,在系统超载时可预测性好,因此静态调度算法比动态调度算法得到更加广泛的应用.然而静态调度算法具有很大的局限性.例如,资源利用率低、受系统支持的优先级个数限制以及灵活性和自适应性差等.本文将给出在优先级个数有限的情况下静态优先级分配的有效方法.

已有的静态优先级分配算法[1~3]假设系统支持的优先级个数是无限的,有多少个任务,系统就支持多少个优先级.此类算法包括 RMA 算法[1]、DMA 算法[2]和 MRMA 算法[3]等.文献[4]指出,当所有任务的周期等于其截止期限时,RMA 算法是最优静态优先级分配算法;当所有任务的周期小于等于其截止期限时,DMA 算法是最优静态优先级分配算法.文献[3]指出,当每个任务的截止期限不超过 $\max(1,\gamma-1)^*$ 与其周期之积时,MRMA(modified rate-monotonic algorithm)是最优静态优先级分配算法.其中γ表示等于任务集合中任务的最大周期除以任务集合中任务的最小周期.

实际上,系统支持的优先级个数总是有限的.例如,在 Tru64 UNIX[5]中可用于实时调度的优先级个数为 32,LinuxRT[6]支持 256 个优先级,实时 POSIX[7]支持 32 个优先级,MSI STD BUS[8]支持 2 个优先级.当任务个数大于系统支持的优先级个数时,一个优先级可能对应多个任务.文献[9]指出,当系统支持的优先级个数很小时,任务的可调度性将大大降低.因此在优先级个数有限的情况下,研究任务的优先级分配具有重要意义.

目前,对于在有限优先级情况下任务的优先级分配方法的研究还很不充分.文献[10]给出一个在有限优先级情况下的优先级分配算法(常量法),即按照任务集合中任务周期的大小分配优先级.每个优先级都对应一定周期范围的任务,且该周期范围由任务集合中的最小周期和最大周期决定,在优先级分配过程中不发生变化.该算法实现简单,但是采用该算法的任务集合可调度性较低.当优先级个数为 4 时,任务集合的可调度性小于 0.1.文献[9]针对截止期限小于等于周期的任务集合,给出了在有限优先级的情况下,任务可调度的充分必要条件,并指出最优优先级分配算法的时间复杂度呈指数形式,但未给出相应的优先级分配算法.文献[11]给出了当优先级有限时 RMA 算法所需最少优先级的条件,然而同样没有给出相应的优先级分配算法.

本文针对现有研究的不足,给出在有限优先级的情况下,当任务的截止期限大于周期时,任务可调度的充分必要条件,提出一个优先级分配算法(AGP).模拟结果表明,本文所提出的算法在可调度性上远远高于常量法.

1 基于有限优先级的任务可调度的充分必要条件

考虑一个单处理机系统,系统支持的优先级个数为 N.可以将任务与优先级的关系看作函数映射的关系.将系统支持的优先级看作不同的集合,每个集合称为一个优先级组.给任务分配优先级,可以看作将任务映射到一个优先级组中.以 g_z 表示优先级组,z 表示相应的优先级.任务集合 $\Gamma=\{\tau_i|1\le i\le n\}$ 中所有任务都为周期性任务,并且任务之间相互独立.系统采用静态优先级抢占式调度策略.任务 τ_i 由三元组〈周期 T_i,计算时间 C_i,截止期限 D_i〉表示.任务 τ_i 的截止期限 D_i 是任意的,即 D_i 可以小于等于周期 T_i,也可以大于周期 T_i.称 τ_i 的第 k 次请求为任务实例 τ_{ik}.为了方便起见,设 τ_{ik} 的下标 k 从 0 开始.任务实例 τ_{ik} 的截止期限 $d_i(k)=k*T_i+D_i$.在不混淆的情况下,τ_{ik} 的截止期限是指 $d_i(k)$,而任务 τ_i 的截止期限是指 D_i.不妨认为任务 τ_i 具有两种优先级属性:自然优先级和组优先级.自然优先级是指假设系统支持的优先级个数无限时采用文献[1~3]中方法分配的优先级.以 $p(\tau_i)$表示 τ_i 的自然优先级.组优先级是指当系统中优先级个数有限时给任务分配的优先级.以 $l(\tau_i)$表示 τ_i 的组优先级.

为方便起见,设 g_1 表示最高优先级组,g_N 表示最低优先级组,1 表示最高的组优先级,N 表示最低的组优先级.当 $p(\tau_i)>p(\tau_j)$时,表示任务 τ_i 的自然优先级高于 τ_j 的自然优先级.当 $l(\tau_i)>l(\tau_j)$时,表示任务 τ_i 的组优先级高于任务 τ_j 的组优先级.

如果任务集合中的所有任务都可调度,那么称该任务集合可调度.如果任务集合中至少存在一个不可调度的任务,那么称该任务集合不可调度.

当 $D_i \le T_i$,任务 τ_i 映射到优先级组 g_z 时,τ_i 可调度的充分必要条件如下[9]:

$$\min_{0<t\le D_i} W_i(t)/t \le 1 \tag{1}$$

其中

$$W_i(t) = \sum_{\tau_j \in g_p, p<z} C_j \left\lceil \frac{t}{T_j} \right\rceil + \sum_{\forall \tau_q \in g_z} C_q \tag{2}$$

下面将给出当 $D_i > T_i$,任务 τ_i 映射到优先级组 g_z 时,τ_i 可调度的充分必要条件.

如果截止期限大于周期的任务可调度,那么它的最大响应时间一定小于等于其截止期限.设 τ_i 为截止期限大于周期的任务.τ_i 的最大响应时间发生在当所有组优先级高的任务和任务 τ_i 同时在时刻 0 释放,以时刻 0 为起始时刻的忙周期中[12],将该忙周期简称为第一忙周期.因此,判断 τ_i 是否可调度只需检查任务 τ_i 在第一忙周期中释放的请求中是否有请求的完成时间大于截止期限.

设 k 为在第一忙周期中,任务 τ_i 释放的第 k 个请求的序号.$S_i(k)$表示任务 τ_i 的第 k 个请求的到达时间且 $S_i(k)=k\times T_i$.

一个优先级组对应着一个优先级队列.在同一个优先级队列中,在 $S_i(k)$之前到达的组优先级等于 $l(\tau_i)$的请求排在 τ_{ik} 的前面,该任务实例将推迟 τ_{ik} 的执行;而在 $S_i(k)$之后到达的组优先级等于 $l(\tau_i)$的请求排在 τ_{ik} 的后面,不会影响 τ_{ik} 的执行;若组优先级等于 $l(\tau_i)$的请求的到达时间与 $S_i(k)$相同,由于各个系统处理方法的不同,该请求可能排在 τ_{ik} 的后面,也可能排在 τ_{ik} 的前面,最坏情况是到达时间为 $S_i(k)$、组优先级等于 $l(\tau_i)$的请求都排在 τ_{ik} 的前面.不妨假设在同一优先级队列中,与 τ_{ik} 同时到达的组优先级等于 $l(\tau_i)$的请求排在 τ_{ik} 的前面.由静态调度算法的特性可知,系统只有先完成组优先级高于 $l(\tau_i)$的请求,并进一步完成组优先级等于 $l(\tau_i)$且排在 τ_{ik} 前的请求,才能完成 τ_{ik}.

令 $W_i(k)$表示 τ_{ik} 的完成时间.由于在第一忙周期中 CPU 不会运行组优先级低于 $l(\tau_i)$的任务实例.因此,τ_{ik} 的完成时间 $W_i(k)$是由以下 3 部分构成的.

(1) 在 $W_i(k)$中释放的 τ_i 的请求运行时间.由于 τ_i 请求的序号是由 0 开始的,因此在 $W_i(k)$时,已运行了 τ_i 的 $(k+1)$个请求.在 $W_i(k)$中释放的 τ_i 的请求运行时间为 $(k+1)\times C_i$.

(2) 在 $W_i(k)$中释放的组优先级高于 $l(\tau_i)$的请求执行时间.在 $W_i(k)$前到达的组优先级高于 $l(\tau_i)$的任务 τ_j 的请求个数为 $\left\lceil \frac{W_i(k)}{T_j} \right\rceil$.这些请求的运行时间为 $\sum_{\tau_j \in g_P, p<z} \left\lceil \frac{W_i(k)}{T_j} \right\rceil \times C_j$.

(3) 在 $W_i(k)$中释放的组优先级等于 $l(\tau_i)$的请求运行时间.设 τ_q 的组优先级与 τ_i 的组优先级相同.当任务 τ_q 请求的到达时间都不等于 $S_i(k)$时,在 $S_i(k)$前到达的任务 τ_q 的请求个数为 $\left\lceil \frac{S_i(k)}{T_q} \right\rceil = \left\lfloor \frac{S_i(k)}{T_q} \right\rfloor + 1$;当任务 τ_q 有一个请求 τ_{qr} 的到达时间与 $S_i(k)$相同时,$S_i(k)$一定能整除 T_q.τ_q 请求的下标也从 0 开始,则 $r = \frac{S_i(k)}{T_q} = \left\lfloor \frac{S_i(k)}{T_q} \right\rfloor$.由于与 τ_{ik} 同时到达的组优先级等于 $l(\tau_i)$的请求排在 τ_{ik} 的前面,因此这时在同一优先级队列中,排在 τ_{ik} 前面的 τ_q 的请求个数应为 $\frac{S_i(k)}{T_q} + 1 = \left\lfloor \frac{S_i(k)}{T_q} \right\rfloor + 1$.从而,在 $W_i(k)$中释放的组优先级等于 $l(\tau_i)$的 τ_i 的请求运行时间为 $\left(\left\lfloor \frac{S_i(k)}{T_q} \right\rfloor + 1 \right) \times C_q$.

因此,$W_i(k) = (k+1)C_i + \sum_{\tau_j \in g_P, p<z} \left\lceil \frac{W_i(k)}{T_j} \right\rceil \times C_j + \sum_{\tau_q \in g_z, q \ne i} \left(\left\lfloor \frac{S_i(k)}{T_q} \right\rfloor + 1 \right) \times C_q$.

令 l 为第一忙周期中任务 τ_i 释放的最后一个请求的序号,$W_i(l)$为第一忙周期的结束时刻.当 $W_i(l)-(l+1)T_i \le 0$ 时,表示第一忙周期结束.

如果在第一忙周期中,任务 τ_i 释放的请求其完成时间都小于等于其截止期限,那么任务 τ_i 一定可调度.因此,如果任务 τ_i 可调度,则 $\max_{k=0,1,2,\ldots,l} W_i(k) \le D_i + kT_i$ 成立.

根据以上分析可以得出,当 $D_i > T_i$,任务 τ_i 映射到优先级组 g_z 时,τ_i 可调度的充分必要条件必须满足公式(3):

$$\max_{k=0,1,2,\ldots,l} W_i(k) \le D_i + kT_i \tag{3}$$

其中

$$W_i(l)-(l+1)T_i \le 0 \tag{4}$$

$$W_i(k) = (k+1)C_i + \sum_{\tau_j \in g_p, p<z} \left\lceil \frac{W_i(k)}{T_j} \right\rceil \times C_j + \sum_{\tau_q \in g_z, q \ne i} \left(\left\lfloor \frac{S_i(k)}{T_q} \right\rfloor + 1 \right) \times C_q \tag{5}$$

$$S_i(k)=k\times T_i \tag{6}$$

2 基于有限优先级的优先级分配算法

下文将基于有限优先级的优先级分配算法简称为组优先级分配算法.

当系统支持的优先级个数无限时,如果任务集合不可调度,那么无论采用哪种静态优先级分配算法,该任务集合都不可调度.因此,以下只讨论假设系统支持的优先级个数为无限时任务集合可以调度的情况.由于任务的自然优先级可由文献[1~3]中的优先级分配算法分配.因此,以下假设任务的自然优先级已知.

当组优先级个数为 N,任务个数为 n 时,需要在$(n-1)!/((N-1)!(n-N)!)$种组合中找到一个最优的组优先级分配方法[9],其时间复杂度为指数形式,因此有必要采用有效的启发式方法,以降低算法的复杂度和开销.

2.1 组优先级分配算法AGP

引理. 组优先级低于 $l(\tau_i)$的任务不会干扰任务 τ_i 的执行.

由静态优先级抢占式调度策略可知,不论任务 τ_j 的自然优先级是否高于任务 τ_i,当 τ_j 的组优先级低于 $l(\tau_i)$时,τ_j 都不会干扰 τ_i 的执行.

定理 1. 对于一个给定的任务 τ_i,如果将其从优先级组 g_k 中去掉,并将其加入到优先级组 g_j 中,那么组优先级介于 k 和 j 之间的任务以及在 g_k 和 g_j 中任务的最大响应时间会发生改变,而其他优先级组中任务的最大响应时间不会发生改变.

证明:根据引理可知,由于任务的最大响应时间只与组优先级高的任务、同一优先级组中的任务以及任务的本身执行时间有关,因此,当任务 τ_i 从优先级组 g_k 中去掉,并加入到优先级组 g_j 中时,被映射到优先级组 g_k,g_j 以及 g_k 和 g_j 之间的任务的最大响应时间都将发生变化,而映射到其他优先级组中的任务的最大响应时间都不会发生变化.定理 1 成立. □

定义 1. 如果任务集合 Γ 的两个任务交换自然优先级后,这两个任务以及自然优先级界于两者之间的任务都可调度,那么称这两个任务可以互换自然优先级;如果任务集合Γ的任意两个任务交换自然优先级,都将导致至少有一个任务 τ_i 不可调度(τ_i 属于由这两个任务和自然优先级界于两者之间的任务组成的任务集合),那么称这两个任务不可互换自然优先级.

定义 2. 若一个任务的自然优先级与任务集合中所有其他任务的自然优先级都不相同,则称该任务为唯一自然优先级任务.当任务集合中存在自然优先级相同的任务时,则采取某种策略,保持其他任务的自然优先级高低顺序不变,使这些任务的自然优先级各不相同且使所有任务都可调度,则称这些自然优先级相同的任务为可转化任务,将这种过程简称为可转化任务,转化为唯一自然优先级任务;若只能将一部分自然优先级相同的任务转化为唯一自然优先级任务,而对其他一些自然优先级相同的任务,无论采取哪种策略都不能使这些任务自然优先级各不相同且可调度,则称这些任务为不可转化任务.

定义 3. 如果任务集合中不存在可互换自然优先级的任务,并且任务集合中不存在可转化任务,则称该任务集合为基本任务集合.否则,称该任务集合为扩展任务集合.

当任务个数大于组优先级个数时,需要将多个任务映射到同一个优先级组中,采用如下规则分配组优先级:

(1) 保序规则:若任务 τ_i 的自然优先级高于任务 τ_j 的自然优先级,则要么 τ_i 与 τ_j 的组优先级相同,要么 τ_i 的组优先级高于 τ_j 的组优先级.

(2) 可调度性规则:同一个组的所有任务都必须满足可调度的充分必要条件.当 $\tau_i \in g_k$ 时,g_k 中所有任务都应

满足可调度的充分必要条件,即当 $D_i \le T_i$ 时,满足公式(1);当 $D_i > T_i$ 时,满足公式(3).

(3) 不可转化任务映射规则:将不可转化且自然优先级相同的任务映射到同一组中.

根据以上分配规则,给出一个组优先级分配算法 AGP.

AGP 算法思想是:根据保序规则,组优先级较高的任务,其自然优先级也较高,因此最高的自然优先级对应于最高的组优先级.首先从最高的组优先级开始进行分配.按照自然优先级从高到低的顺序将任务映射到优先级组中.将最高自然优先级任务映射到组优先级最高的优先级组中.根据式(1)或式(3),判断次高自然优先级任务能否映射到组优先级最高的优先级组中.若可以,则将次高自然优先级任务映射到组优先级最高的优先级组中;否则,将次高自然优先级任务映射到组优先级次高的优先级组中.对于不可转化任务,需判断所有自然优先级相同的不可转化任务能否映射到当前组中.若可以,则将所有自然优先级相同的不可转化任务映射到当前组中,否则将其映射到下一组中.依此类推.

2.2 基本任务集合

本节将针对基本任务集合进行分析.

定义 4. 采用一个组优先级分配算法给任务集合中任务分配组优先级,并且使该任务集合中至少存在这样两个任务 τ_i 和 τ_j($p(\tau_i)>p(\tau_j)$且 $l(\tau_i)<l(\tau_j)$),称这两个任务不具有保序性,称该组优先级分配算法为不保序组优先级分配算法;否则,称该组优先级分配算法为保序组优先级算法.

定义 5. 将采用一个组优先级分配算法给任务集合中的任务分配组优先级,并使得该任务集合可调度简称为该算法使该任务集合可调度;否则,简称为该算法使该任务集合不可调度.

定理 2. 对于基本任务集合,只有保序静态组优先级分配算法才能使该任务集合可调度.

证明:采用反证法.

假设存在一个不保序静态组优先级分配算法 B 使基本任务集合 Γ 可调度,那么至少存在两个任务不具有保序性.

(1) 在任务集合 Γ 中有两个任务 τ_i,τ_j 不具有保序性.不妨设 $p(\tau_i)>p(\tau_j)$,而 $l(\tau_i)<l(\tau_j)$.τ_i 的组优先级一定比 τ_j 的组优先级低 1,并且在 τ_i 与 τ_j 之间一定不存在自然优先级界于两者之间的任务.否则,将会有多个任务不具有保序性,而不是只有两个任务不具有保序性.由任务集合 Γ 中不存在可以互换自然优先级的任务可知,τ_i,τ_j 互换自然优先级将导致 τ_i 不可调度.当 τ_j 的组优先级高于 τ_i 时,由于此时组优先级高于 τ_i 的任务集合与在 τ_i,τ_j 互换了自然优先级后,自然优先级高于 τ_i 的任务集合相等,从而 τ_i 的最大响应时间仍将大于其截止期限.因此,当任务集合 Γ 中有两个任务不具有保序性时,B 使任务集合 Γ 不可调度,与假设矛盾.

(2) 在任务集合 Γ 中有 3 个任务 τ_i,τ_j,τ_p 不具有保序性,设 $p(\tau_i)>p(\tau_p)>p(\tau_j)$.可以分以下不同的情况考虑:

- τ_j,τ_p,τ_i 分别在不同的组中,它们的组优先级依次相差 1 且 $l(\tau_j)>l(\tau_p)>l(\tau_i)$ 或 $l(\tau_p)>l(\tau_j)>l(\tau_i)$ 或 $l(\tau_j)>l(\tau_i)>l(\tau_p)$;
- τ_j,τ_p 在同一组中,τ_i 在下一组中;
- τ_j 在一个组中,而 τ_i,τ_p 在下一组中.

由任务集合 Γ 中不存在可互换自然优先级的任务可知:对于以上情况,都将至少导致 τ_i,τ_p 中的 1 个不可调度.因此,当有 3 个任务不具有保序性时,算法 B 使 Γ 不可调度.与假设矛盾.

(3) 假设在任务集合 Γ 中有 $k(k\le n)$ 个任务不具有保序性,设 $p(\tau_i)>p(\tau_{i+1})>\ldots>p(\tau_{i+k-1})$.考虑在这 k 个任务中组优先级最高的任务 $\tau_{i+m}(m<>0)$ 可知,$\tau_i,\tau_{i+1},\ldots,\tau_{i+m-1}$ 的组优先级必低于(或等于)$l(\tau_{i+m})$.由任务集合 Γ 中不存在可互换自然优先级的任务可知,不论 $\tau_i,\tau_{i+1},\ldots,\tau_{i+m-1}$ 的组优先级如何分配,都将导致在 $\tau_i,\tau_{i+1},\ldots,\tau_{i+m-1}$ 中至少有 1 个任务不可调度.因此当任务集合 Γ 中有 k 个任务不具有保序性时,至少导致这 k 个任务中的一个任务不可调度.

综上所述,如果任务集合 Γ 中不存在可以互换自然优先级的任务,不保序静态组优先级分配算法 B 一定使任务集合 Γ 不可调度,与假设矛盾.

因此定理 2 成立. □

定理 3. 对于给定的基本任务集合,在所有组优先级分配算法中,AGP 算法所需优先级个数最少.

证明:采用反证法.

对于一个给定的基本任务集合Γ,设 AGP 算法使得该任务集合可调度,且 AGP 算法所需组优先级个数为 p.以 $A(i)$表示采用 AGP 算法的优先级组.

设存在算法 B 使任务集合Γ可调度,且算法 B 所需组优先级个数为 $q(q<p)$.

根据定理 2 可知,算法 B 一定满足保序规则.

对于算法 B 来说,至少有一个优先级组 $A(l)$中的任务被映射到其他组.如果将 $A(l)$中第 1 个任务τ_j移到$A(l-1)$优先级组中,若τ_j为唯一自然优先级任务,则根据 AGP 算法可知,将τ_j移到$A(l-1)$中将导致$A(l-1)$组中有任务不满足可调度条件,否则τ_j应被映射到$A(l-1)$组,而不是映射到$A(l)$组.若τ_j为不可转化任务,将τ_j移到组$A(l-1)$中,则导致$A(l)$中自然优先级与τ_j相等的不可转化任务不可调度.因此,只能将τ_j移到$A(l+1)$优先级组中,为了满足任务之间的保序性,需要将$A(l)$组中的所有任务移至$A(l+1)$.这时将导致原来$A(l)$组中的任务不满足可调度条件,将 $A(l+1)$的原有任务移至 $A(l+2)$中.此时,$A(l+1)$的组优先级对应着原来的 $A(l)$的组优先级.依次将 $A(l+2)$的原有任务将移至 $A(l+3)$中,….,$A(p)$的原有任务移至 $A(p+1)$中.此时,$A(l+2)$,$A(p+1)$的组优先级分别对应着原来的$A(l+1)$,….,$A(p)$的组优先级.可以看出,将 $A(l)$组中的任务移至下一组 $A(l+1)$中,并不能减少所需组优先级个数.从而得到 $q\geq p$.与假设 $q<p$ 的假设矛盾.

因此可知,当算法 B 使得任务集合 Γ 可调度时,算法 B 所需组优先级个数一定大于等于 AGP 算法的组优先级个数.故定理 3 成立. □

定理 4. 对于组优先级个数已定的基本任务集合,如果存在静态组优先级分配算法使得该任务集合可调度,则 AGP 算法也一定能使该任务集合可调度.

定理 4 等价于:对于组优先级个数已定的基本任务集合,如果 AGP 算法使得该任务集合不可调度,那么其他静态组优先级分配算法也一定使该任务集合不可调度.

证明:采用反证法.

设基本任务集合$\Gamma=\{\tau_i|1\leq i\leq n\}$,系统支持的组优先级个数为 m 且 $m<n$.不失一般性,设Γ中的任务按照自然优先级从高到低的顺序排列.

在给定组优先级个数的情况下,设静态优先级分配算法 B 使得任务集合Γ可调度,B 所需的组优先级个数为 q.根据定理 2 可知,算法 B 一定是保序算法.

设采用 AGP 算法对任务集合Γ中的任务进行组优先级分配时,AGP 所需组优先级个数为 p.

由定理 3 可知,在所有组优先级分配算法中,AGP 算法所需组优先级个数最小.因此 $p\leq q$.

由于在组优先级个数已定的情况下,B 使得任务集合 Γ 可调度.因此 $q\leq m$,从而 $p\leq m$.因此 AGP 也使得任务集合 Γ 可调度.故定理 4 成立. □

对于基本任务集合,当一个任务映射到一个优先级组时,需判断该优先级组中是否因为该任务的加入而导致组中有任务不满足可调度条件,因此需要检查该组中的所有任务.当组中第 1 个任务映射到该组时,需检查该任务是否满足调度条件;当第 2 个任务映射到该组时,需检查该组中第 1,2 个任务是否满足条件;当第 m 个任务映射到该组时,需检查该组中 m 个任务是否满足可调度条件.因此,当组中有 m 个任务,判断组中是否有任务不满足调度条件的次数为 $1+2+\ldots+m=(m+1)m/2$ 次.当所有任务都平均分配到 N 个优先级组中时,需要检查的次数之和最小.最小检查次数之和为$(n/N+1)(n/N)\times N/2=(n+N)n/(2N)$.当所有的任务都映射到同一个优先级组中时,需要检查的次数最大.最大检查次数之和为$(n+1)n/2$ 次.因此,AGP 算法的时间复杂度为 $O(n^2)$.

2.3 扩展任务集合

当任务集合中存在自然优先级相同的任务时,首先将所有可转化任务转化成唯一自然优先级任务,使任务集合中只包括唯一自然优先级任务和不可转化任务.

由于扩展任务集合中存在可以互换自然优先级的任务、可转化任务,当可互换自然优先级的任务分配的自然优先级不同或者任务转化策略不同,采用 AGP 算法有可能得到不同的组优先级分配结果.

对于扩展任务集合仍然可以采用 AGP 算法分配组优先级.如果 AGP 算法所需组优先级个数大于给定的组

优先级个数,那么改变可互换自然优先级的任务或可转化任务的自然优先级,重新采用 AGP 算法进行组优先级分配,直到AGP算法所需组优先级个数小于等于给定的组优先级个数,这时AGP算法成功地给任务集合中任务分配组优先级.或者,对于所有的自然优先级分配方案,AGP 算法所需组优先级个数都大于给定的组优先级个数,这表明没有一个组优先级分配算法能使该任务集合可调度.扩展任务集合中的可以互换自然优先级的任务、可转化任务的个数决定了 AGP 算法的时间复杂度.在最坏情况下,所有的任务都可互相转化且所有的任务都可互换自然优先级,这时采用 AGP 算法分配组优先级相等于从$(n-1)!/((N-1)!(n-N)!)$组合中选出一种组合,因此在最坏情况下,AGP 的时间复杂度为 $O((n-1)!/((N-1)!(n-N)!))$.

3 性能比较

令不可调度率等于不可调度的任务集合个数除以任务集合总数.采用不可调度率作为算法的评价标准.将本文提出的算法与常量法进行比较.

以任务集合中任务个数分类,共生成 49 类任务集合.任务个数从 2 依次递增到 50.每类集合中共有 50 个任务集合.测试的任务集合共有 980 个任务集合.

任务集合中的任务满足以下条件:

(1) 任务的周期服从[10,1000]的均匀分布;

(2) 任务的截止期限满足(3/4×周期,周期)的均匀分布;

(3) 要求系统的资源利用率小于等于 1.为了保证资源利用率不超过 1,令任务的执行时间满足$(1,a)$的均匀分布,其中 $a=\frac{2.5\times\text{截止期限}}{\text{任务个数}}$;

(4) 任务在无限组优先级个数的情况下可调度;

(5) 在任务集合中,任务按照自然优先级从高到低的顺序排列.

图 1~图 3 为当组优先级个数分别为 2,4,8 时常量法与 AGP 算法的不可调度率分布图.从图中可以看出,随着优先级个数的增加,任务集合的可调度率随之增加.其中,采用常量法的可调度率最低,且 AGP 的可调度率要远远大于常量法的可调度率.

4 结束语

静态优先级调度在实时系统中得到了广泛应用.然而,静态优先级调度受到系统支持的优先级个数的限制.本文针对系统优先级个数有限的情况,给出了当任务截止期限大于其周期的任务可调度的充分必要条件,并提出一个组优先级分配算法 AGP.

AGP算法对于基本任务集合来说是最优组优先级分配算法:AGP所需组优先级个数最少,且若AGP算法不能使一个基本任务集合调度,则其他静态优先级分配算法也不能使该任务集合可调度.AGP 算法对于扩展任务集合来说不一定是最优,但是仍然可以通过 AGP 算法有效分配任务的优先级或者判定该任务集合是否可调度.实验结果表明,AGP 算法性能要优于常量法,是目前具有较高可调度性的组优先级分配算法.本文提出的算法对于嵌入式实时系统来说具有一定的价值.

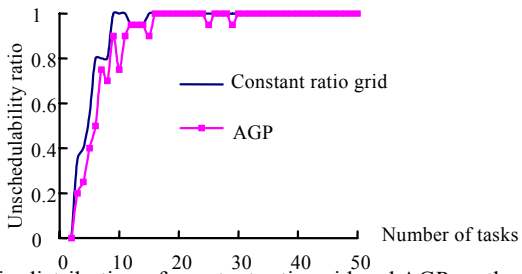

Fig.1 Unschedulabilility ratio distribution of constant ratio grid and AGP, as the number 2 of priority levels

图1 组优先级个数为2时AGP和常量法不可调度率分布图

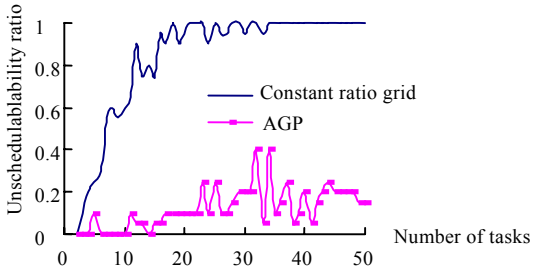

Fig.2 Unschedulabilility ratio distribution of constant ratio grid and AGP, as the number 4 of priority levels

图2 组优先级个数为4时AGP和常量法不可调度率分布图

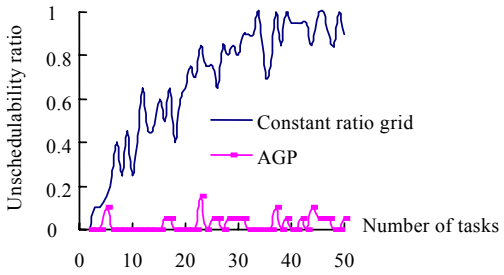

Fig.3 Unschedulabilility ratio distribution of constant ratio grid and AGP, as the number 8 of priority levels

图3 组优先级个数为8时AGP和常量法不可调度率分布图

致谢 在此,我们向对本文提出宝贵意见的评审老师表示感谢.

References:

[1] Liu CL, Layland JW. Scheduling algorithms for multiprogramming in a hard-real-time environment. Journal of the ACM, 1973,20(1):46~61.

[2] Audsley NC. Deadline monotonic scheduling. Technical Report, YCS 146, University of York, 1990.

[3] Shih WK, Liu JWS, Liu CL. Modified rate-monotonic algorithm for scheduling periodic jobs with deferred deadlines. IEEE Trans. on Software Engineering, 1993,19(12):1171~1179.

[4] Audsley NC. Optimal priority assignment and feasibility of static priority tasks with arbitrary start times. Technical Report, YCS 164, University of York, 1990.

[5] Tru64 Unix: Guide to real-time programming. Compaq Computer Corportation. 2000. http://www.tru64unix.compaq.com/

[6] The concise handbook of linux for embedded real-time systems version 1.0. Timesys Corporation, 2000. http://www.timesys.com

[7] Harbour MG. Real-Time POSIX: An overview. In: Proc. of the Int'l Conf. of VVConex'93. 1993. http://www.ctr.unican.es/publications/mgh-1993a.pdf

[8] MSI-C851 STD BUS 80C51 Microcontroller Card. Microcomputer systems. http://www.microcomputersystems.com.

[9] Katcher DI, Sathaye SS. Strosnider JK. Fixed priority scheduling with limited priority levels. IEEE Trans. on Computers, 1995, 44(9):1140~1144.

[10] Lehoczky JP, Sha L. Performance of real-time bus scheduling algorithms. ACM SIGMETRICS Performance Evaluation Review, 1986,14(1):44~53.

[11] Orozco J, Cayssials R, Santos J, Santos R. On the minimum number of priority levels required for the rate monotonic scheduling of real-time systems. In: Proc. of the 10th EUROMICRO Workshop on Real Time Systems. 1998. http://www.mrtc.mdh.se/emrt98/wip/proceedings/5.ps

[12] Lehoczky JP. Fixed priority scheduling of periodic task sets with arbitrary deadlines. In: Proc. of the 11th Real-Time Systems Symp. IEEE Computer Society Press, 1990. 201~213.