

基于 (m, k) 模型的任务可调度性分析^{*}

宾雪莲¹ 杨玉海¹ 宾 亚¹ 金士尧²

(空军雷达学院 武汉 430019)¹ (国防科技大学计算机学院 长沙 410073)²

摘 要 (m, k) 模型是降低系统负载的一种有效方法。EDF 算法由于其灵活性和自适应性而得到广泛应用。针对 EDF 算法, 首先研究并给出了基于 (m, k) 模型的任务集可调度的必要条件和充分条件, 然后给出了基于 (m, k) 模型的任务最大响应时间的一种新的计算方法。

关键词 实时, (m, k) 模型, EDF, 最大响应时间

Scheduling Analysis of Tasks with (m, k) Guarantee

BIN Xue-Lian¹ YANG Yu-Hai¹ BIN Ya¹ JIN Shi-Yao²

(Airforce Army Radar Academy, Wuhan 430019)¹

(School of Computer Science, National University of Defense Technology, Changsha 410073)²

Abstract (m, k) guarantee is an effective method to reduce the overload of the system. Because of the merits of EDF, such as high utilization, adaptability and flexibility, it has been applied in real-time system widely. The paper focuses on the scheduling analysis problem of EDF preemptive scheduled task sets with (m, k) guarantee. A sufficient condition and a necessary condition are derived in the paper for the EDF preemptive scheduled task sets with (m, k) guarantee. Finally, we give a method to compute the worst-case response time for task with (m, k) guarantee.

Keywords Real-time, (m, k) guarantee, EDF, Worst-case response time

1 引言

在实时系统中, 如果一个周期任务的请求未能在截止期限内完成, 那么称请求超时。强实时系统必须保证任务把所有请求都集中在其截止期限之前完成, 不允许任何一个请求超时。对于准实时系统, 一些请求超时, 不会导致系统失效, 只是降低了服务质量。如在多媒体应用中, 放弃一帧的运行不会对整个系统造成影响。允许放弃一些请求的执行, 可以增加系统的灵活性, 降低系统负载。根据准实时系统的特性, 研究人员提出 (m, k) 模型^[1]和跳跃因子模型^[2], 允许在系统超载时选择性地丢弃一些请求, 使系统在超载时能够缓慢降级。

在 (m, k) 模型和跳跃因子模型提出后, 研究人员针对该模型提出了各种优先级分配方法以及如何放弃请求的方法^[1~3, 8, 9], 而对基于 (m, k) 模型的任务的可调度分析研究较少。文[3]对采用固定优先级调度的基于 (m, k) 模型的任务的可调度性进行了分析。EDF 算法由于其资源利用率高, 具有较强的自适应性和灵活性的特点, 在实时系统中得到了广泛的应用。文[4, 5, 7, 10]针对 EDF 抢占式调度算法给出了任务集的可调度条件。目前没有对采用 EDF 的基于 (m, k) 模型的任务集可调度性的研究。本文将针对 EDF 可剥夺式算法, 给出基于 (m, k) 模型的任务集的可调度的必要条件和充分条件, 并研究了判断基于 (m, k) 模型的任务集是否可调度的一种新方法。最后给出计算基于 (m, k) 模型的任务的最大响应时间的方法。

2 模型定义

假设系统中的时间是离散的, 时间的基本单位为一个时钟滴答; 系统中任何时间都可用自然数表示。

假设单处理机系统, 采用 EDF 抢占式调度, 任务集 $\Gamma = \{\tau_i | 1 \leq i \leq n\}$ 中所有任务都为周期性任务, 任务之间相互独立。任务 τ_i 由 $($ 周期 T_i , 计算时间 C_i , 截止期限 D_i , 约束条件 (m_i, k_i) $)$ 表示, 其截止期限 D_i 小于等于周期 T_i 。任务的相位为任意的, 即任务的第一次请求可以在任何时候到达。任务 τ_i 第 j 次请求称为实例 τ_{ij} 。约束条件 (m_i, k_i) 表示在 τ_i 的连续 k_i 个请求中至少要有 m_i 个请求在截止期限前完成。实例可以看作具有红色或蓝色, 红色表示实例必须在截止期限前完成, 蓝色表示允许实例超过截止期限。

定义 1 基于 (m_i, k_i) 模型的任务 τ_i 的执行情况可以用一个二进制字符串 $II_i = \{a_{i1} a_{i2} \dots a_{ij} \dots\}$ 表示。其中 $a_{ij} = 1$ 表示实例 τ_{ij} 为红色; $a_{ij} = 0$ 表示实例 τ_{ij} 为蓝色。称字符串 II_i 为任务 τ_i 的执行字符串。以 α_i 表示 II_i 的一个子串。

定义 2 若字符串 II_i 的所有长度为 k_i 的子串 α_i 满足: $\sum_{j=1+p, p \geq 1}^{k_i+1-p} a_{ij} = m_i$, 则称子串 α_i 为最小子串, 称字符串 II_i 为最小执行串。

定义 3 若基于 (m_i, k_i) 模型的任务 τ_i 的最小执行串如图 1 所示, 则称该字符串为关键执行串。若其最小子串 α_i 如图 1(b)所示, 则称 α_i 为关键子串。

$$\begin{array}{ccccccc} \overbrace{11..}^m & \overbrace{100..}^{k-m} & \overbrace{011..}^m & \overbrace{100..}^{k-m} & \dots & \overbrace{11..}^m & \overbrace{100..}^{k-m} \\ 11..100..011..100..0\dots & & & & & & 11..100..0 \end{array}$$

(a) 关键执行串 (b) 关键子串

图 1 最小执行串

定义 4 d -忙周期: 在该忙周期中只执行截止期限小于等于 d 的实例。

3 可调度性分析

每个周期任务都是由无限个具有相同到达时间间隔的实

^{*} 本文受国家自然科学基金资助(项目标号: 60073003)。宾雪莲 博士, 主要研究方向为实时调度技术; 杨玉海 博士, 讲师, 主要从事讲算机网络技术研究; 宾 亚 硕士, 讲师, 主要从事网络通信技术研究; 金士尧 博士生导师, 教授, 主要从事仿真、实时、并行与分布系统等方面的研究。

例构成。在某一时刻,同一个任务只能有一个实例到达(或者释放);一个任务在某一时刻到达(或者释放)表示该任务的某个实例在该时刻到达(或者释放)。由于不存在释放抖动,实例的到达时间等于实例的释放时间。

由于红色实例的优先级高于蓝色实例的优先级,因此红色实例会抢占蓝色实例的执行。在以下讨论中,假设所有任务都以最小执行串执行,即放弃执行蓝色实例。

当采用某种 EDF 调度基于 (m, k) 模型的任务集合时,如果所有任务都满足 (m, k) 约束条件,则称该任务集合可调度,否则称该任务集合不可调度。

引理 1 一个任务集的可调度的必要条件为 CPU 的利用率小于等于 1^[4]。

由引理 1 很容易推出采用 EDF 抢占式调度算法基于 (m, k) 模型的任务集的可调度的必要条件。

定理 1 任务集 $\Gamma = \{\tau_i \langle T_i, C_i, D_i, (m_i, k_i) \rangle\}_{i=1}^n$ 可调度的必要条件为 $\sum_{i=1}^n \frac{C_i}{T_i} (\frac{m_i}{k_i}) \leq 1$ 。

定理 2 采用 EDF 调度的任务集 $\Gamma = \{\tau_i \langle T_i, C_i, D_i \rangle\}_{i=1}^n$ 的可调度的充分必要条件为: $\forall L, L > 0, L \geq \sum_{i=1}^n \lfloor \frac{L}{T_i} \rfloor C_i$ ^[6]。

定理 3 当在单 CPU 上采用 EDF 抢占式调度算法时,基于 (m, k) 模型的任务集的最坏情形出现在同时满足以下条件的情况下:(1)所有任务在时刻 t 同时释放;(2)所有的任务以关键执行串执行,且在时刻 t 释放的实例对应其关键子串中的第一个元素。

证明:设从时刻 t 到 CPU 出现第一次空闲的忙周期长度为 L 。

(1)假设除实例 τ_{ij} 外,所有的其它任务都在时刻 t 同时释放。 d 为 τ_{ij} 的截止期限(不妨设 $t < d < L$)。 τ_{ij} 的释放时间分为两种情况:第一种情况为在 t 之前释放,第二种情况为在 t 之后释放。

考虑一个截止期限大于 d 的实例的响应时间。

• 对于第一种情况:如果将 τ_{ij} 的释放时间推迟,则截止期限小于等于 d 的实例在 $[t, L]$ 中产生的负载随着 τ_{ij} 释放时间的推迟而增加。从而在 $[t, L]$ 中释放的截止期限大于 d 的请求的响应时间也将增加。当将 τ_{ij} 的释放时间推迟到 t 时,截止期限小于等于 d 的实例在 $[t, L]$ 中产生的负载达到最大,则在 $[t, L]$ 中释放的截止期限大于 d 的请求的响应时间也达到最大。

• 对于第二种情况:设 τ_{ij} 的释放时间为 t' 。随着 τ_{ij} 的释放时间的提前,截止期限小于等于 d 的实例在 $[t, t']$ 中产生的负载也随之增加。当将 τ_{ij} 的释放时间提前到 t 时,在 $[t, t']$ 中的截止期限小于等于 d 的实例产生的负载达到最大值,在 $[t, t']$ 中释放的截止期限大于 d 的实例的响应时间也达到最大。

因此条件 1 成立。

(2)当在时刻 t 到 d 中($d < L$)中存在一个任务 τ_i 不是以关键子串执行或者在 t 时刻释放的实例不是对应着关键子串中的第一个元素时,令 l_{end} 为在 $[t, d]$ 中释放的截止期限小于等于 d 的实例产生的负载,令 R 为在 $[t, d]$ 中释放的截止期限大于等于 d 的实例的响应时间。当所有任务都以关键执行串执行且在时刻 t 释放的请求对应关键子串中的第一个元素时,令 l'_{end} 为在 $[t, d]$ 中释放的截止期限小于等于 d 的实例产生的负载, R' 为在 $[t, d]$ 中释放的截止期限大于 d 的实例的响应时间。显然 $l'_{\text{end}} \geq l_{\text{end}}$, $R' \geq R$ 。因此条件 2 成立。

综上可知定理 3 成立。

定理 4 采用 EDF 调度任务集 $\Gamma = \{\tau_i \langle T_i, C_i, D_i, (m_i, k_i) \rangle\}_{i=1}^n$ 的充分条件为:

$$L \geq \sum_{i=1}^n D(i, [0, L], \forall L, L > 0 \quad (1)$$

$$\text{其中, } D(i, [0, L]) = (\lfloor \frac{L}{T_i} \rfloor - \lfloor \frac{L(k_i - m_i)}{T_i k_i} \rfloor) C_i \quad (2)$$

证明:假设任务集合中所有任务在时刻 0 同时释放,并以关键执行串执行。 $D(i, [0, L])$ 表示在时间段 $[0, L]$ 中任务 τ_i 的红色实例产生的负载。在 L 前能够完全执行完的所有红色实例的个数为 $\lfloor \frac{L}{T_i} \rfloor - \lfloor \frac{L(k_i - m_i)}{T_i k_i} \rfloor$, 从而式(2)成立。

假设任务集 Γ 在最坏情况下不可调度,那么必然存在一时刻 d , 有红色实例超时。令 $t_a (t_a \geq 0)$ 表示在 d 之前, CPU 出现的最后一个空闲时间的结束时刻。令 $t_b (t_b \geq 0)$ 表示在 d 之前, 截止期限大于 d 的红色实例的忙周期的结束时刻。令 $t = \max(t_a, t_b)$ 。在 $[t, d]$ 中 CPU 不会执行截止期限大于 d 的红色实例,即在 $[t, d]$ 中 CPU 只能执行截止期限小于等于 d 的红色实例,并且在 $[t, d]$ 中 CPU 也不会出现空闲时间。由于任务集合中的任务是以关键执行串执行,因此 $D(i, [t, d]) \leq D(i, [0, d-t])$ 。由 EDF 算法的性质和在时刻 d 时有红色实例超时可知,截止期限小于等于 d 的红色实例产生的负载应大于该时间段的长度。因此 $d-t < \sum_{i=1}^n D(i, [t, d])$ 。即 $d-t < \sum_{i=1}^n D(i, [t, d]) \leq \sum_{i=1}^n D(i, [0, d-t])$ 。因此,若基于 (m, k) 模型的任务集在最坏情况下不可调度,则式(1)不成立。

综上定理 4 成立。证毕。

根据定理 4 可知,判断基于 (m, k) 模型的任务集是否能够被 EDF 成功调度,只要判断当任务集合中任务以关键执行串执行时,在整个超周期(任务集合中所有任务周期的最小公倍数)中是否有红色实例超时。由于定理 4 的计算复杂度比较大,不适用于做控制条件,因此需要给出一个计算量少的控制条件。

判断基于 (m, k) 模型的任务集的可调度性是 NP-难问题^[2,9]。此结论是基于任务的优先级以及任务执行字符串都未知的假设。如果能够保证任务集在最坏情形下可以调度,那么就能保证任务集在其它情形下可调度。

定理 5 在单 CPU 上采用 EDF 算法调度任务时,若在时刻 t 时有溢出(overflow),则在 t 之前没有空闲时间^[4]。

推论 1 在单 CPU 上采用 EDF 算法调度任务集 $\Gamma = \{\tau_i \langle T_i, C_i, D_i, (m_i, k_i) \rangle\}_{i=1}^n$ 。若当任务集中任务的以某个最小执行串执行,在时刻 t 时溢出,那么当任务以最坏情形执行时,在时刻 t 之前没有空闲时间。

证明:假设任务集合中任务以某个最小执行串执行且在时刻 t 时溢出。设 t' 为在时刻 t 之前的最后一个空闲时间段的结束时刻。则在时刻 t' 时,必然有一个红色实例释放。如果使得所有的任务在 t' 时以最坏情形释放,那么显然这时在 $[t', t]$ 中负载将达到最大。由于在 $[t', t]$ 中没有空闲时间,因此当任务集合中的任务以最坏情形在 t' 释放时,在 t 时或 t 之前仍然有溢出。若考虑 t' 之后的任务的执行情况,可以知道在 $[t', t]$ 中也不会存在空闲时间。证毕。

由推论 1 可知,判断采用 EDF 算法调度基于 (m, k) 模型的任务集是否可调度时,只需判断任务以最坏情形在时刻 0 释放时,从时刻 0 到 CPU 第 1 次出现空闲时间的忙周期中是否有红色实例在该忙周期中超时。如果在该忙周期中所有红色实例都能在截止期限前完成,则该任务集合可调度。该忙周期长度 L 可由式(3)、(4)计算。

$$\begin{cases} L^{(0)} = \sum_{i=1}^n C_i \\ L^{(m+1)} = W(L^{(m)}) \end{cases} \quad (3)$$

$$\text{其中, } W(t) = \sum_{i=1}^n C_i * \min(\beta_i(t), \eta_i(t)) \quad (4)$$

令 $\beta_i(t)$ 为在 $[0, t]$ 中任务 τ_i 红色实例的个数, 其计算公式如式(5)所示。

要求出在 $[0, L]$ 中释放的红色实例的个数。设任务 τ_i 的一个红色实例的截止期限为 d 。令 $\eta_i(d)$ 为任务 τ_i 在 $[0, d]$ 中释放的截止期限小于等于 d 的红色实例的个数, 其计算公式如式(6)所示。因此 $[0, d]$ 中截止期限小于等于 d 的任务 τ_i 的红色实例产生的负载为 $\sum_{D_i \leq d} \min(\beta_i(d), \eta_i(d)) * C_i$ 。

本文没有考虑截止期限相等的情况。只是假定所有截止期限等于 d 的请求, 其优先级都高于正在判断的任务。因此, 在时刻 0 到 CPU 第 1 次出现空闲时间的忙周期中截止期限小于等于 d 的红色实例产生的负载 $D(d)$ 可由式(7)计算。

由此可以得出采用 EDF 调度算法, 任务集 $\Gamma = \{\tau_i \langle T_i, C_i, D_i, (m_i, k_i) \rangle\}_{i=1}^n$ 可调度的充分条件为: 当任务集合中的任务以最坏情形在时刻 0 释放时, 在时刻 0 到 CPU 第 1 次出现空闲时间的忙周期中截止期限小于等于 d 的红色实例产生的负载 $D(d)$ 应小于等于 d , 即 $d \geq D(d)$ 。

$$\beta_i(t) = \begin{cases} \lceil \frac{t}{T_i} \rceil & \lceil \frac{t}{T_i} \rceil \leq m \\ \lceil \lceil \frac{t}{T_i} \rceil * \frac{m_i}{k_i} \rceil & \text{otherwise} \end{cases} \quad (5)$$

$$\eta_i(d) = \begin{cases} 1 + \lfloor \frac{d-D_i}{T_i} \rfloor & 1 + \lfloor \frac{d-D_i}{T_i} \rfloor \leq m \\ \lceil (1 + \lfloor \frac{d-D_i}{T_i} \rfloor) * \frac{m_i}{k_i} \rceil & \text{otherwise} \end{cases} \quad (6)$$

$$D(d) = \sum_{i=1}^n C_i * \min(\beta_i(d), \eta_i(d)) \quad (7)$$

4 最大响应时间分析

在分布式实时系统中, 任务的释放抖动由其前驱子任务的最大响应时间和最小响应时间决定。为了能够确定端到端任务的最大释放抖动, 需要先求出其前驱子任务的最大响应时间。本节将给出基于 (m, k) 模型的任务的最大响应时间的计算公式。

对于采用 EDF 算法调度的基于 (m, k) 模型的任务的最大响应时间不一定发生在所有任务都同时释放的时刻。如当表 1 中任务 τ_2 中请求释放时间为 5 时, 可以得到 τ_2 的最大响应时间为 4。

表 1 任务集 Γ

τ_i	T_i	C_i	D_i	(m_i, k_i)
τ_1	10	7	10	(9, 10)
τ_2	5	2	5	(1, 2)

由定理 3 和推论 1 可以得出推论 2。

推论 2 基于 (m, k) 模型的任务 τ_i 的最大响应时间发生在这样的忙周期中, 所有的任务都以关键字串执行, 除 τ_i 外, 所有其它任务都以最坏情形在时间 t 同时释放, 且该忙周期的起始时间为时间 t 。

令 a 表示当任务 τ_i 的响应时间获得最大值时该实例的释放时间, d 表示该实例的截止期限。对于给定时间 $a \geq 0$, 因此任务 τ_i 的第一次请求的到达时间 $S_i(a)$ 为 $S_i(a) = a - \lfloor \frac{a}{T_i} \rfloor T_i$ ($S_i(a) \geq 0$)。如果其它所有红色实例都在时刻 0 同时释放, 则在 $[0, t]$ 中释放的任务 τ_j ($j \neq i$) 的红色实例的个数 $\beta_j(t)$ 可以通过式(5)求出。在 $[0, t]$ 中的释放的红色实例的截止期限小于等于 d 任务 τ_j ($j \neq i$) 红色实例个数 $\eta_j(d)$ 可通过式(6)求出。由于 $d = a + D_i$, 可以将 $\eta_j(d)$ 改写成式(8)。因此实际上, 在 $[0, t]$ 中释放的截止期限小于等于 d 任务 τ_j ($j \neq$

i) 红色实例个数为: $\min(\beta_j(t), \eta_j(a + D_i))$ 。

令 $\delta_i(a, t)$ 表示在 $[0, t]$ 中任务 τ_i 的红色实例的个数。可知, 当 $t \leq S_i(a)$ 时, $\delta_i(a, t) = 0$; 当 $t > S_i(a)$ 时, $\delta_i(a, t) = \min(\beta_i(t - S_i(a)), \eta_i(a + D_i))$ 。

$$\eta_j(a + D_i) = \begin{cases} 1 + \lfloor \frac{a + D_i - D_j}{T_j} \rfloor, & 1 + \lfloor \frac{a + D_i - D_j}{T_j} \rfloor \leq m_j \\ \lceil (1 + \lfloor \frac{a + D_i - D_j}{T_j} \rfloor) * \frac{m_j}{k_j} \rceil, & \text{otherwise} \end{cases} \quad (8)$$

因此截止期限小于等于 d 的红色实例 ($j \neq i$) 产生的负载为:

$$W_i(a, t) = \sum_{D_j \leq a + D_i} \min(\beta_j(t), \eta_j(a + D_i)) C_j + \delta_i(a, t) C_i$$

包含以 a 为释放时间的任务 τ_i 请求的 d -忙周期的长度可以通过以下叠代方法进行计算。

$$\begin{cases} L_i^{(0)}(a) = \sum_{D_j \leq a + D_i} C_j + I_{\{S_i(a)=0\}} C_i \\ L_i^{m+1}(a) = W_i(a, L_i^{(m)} + a) \end{cases}$$

其中, 当 $S_i(a) = 0$ 时, $I_{\{S_i(a)=0\}} = 1$; 否则, $I_{\{S_i(a)=0\}} = 0$ 。

因此对于给定 a , 任务 τ_i 的最大响应时间为: $r_i(a) = \max\{C_i, L_i(a) - a\}$ 。从而任务 τ_i 的最大响应时间为: $\gamma_i = \max_{a \geq 0} \{r_i(a)\}$ 。

由于从时刻 0 到 CPU 出现第 1 次空闲的时间段为最长的忙周期, 所有其它忙周期都小于等于 L 。因此可知, 当 a 在 $[0, L_i - C_i]$ 中取值, 必然能够得到任务的最大响应时间。

采用以上方法求表 1 任务集 Γ 中任务 τ_2 的最大响应时间。该任务集 Γ 中任务以最好情形在时刻 0 释放时, 从时刻 0 到 CPU 出现第一个空闲时间的忙周期长度 $L = 9$ 。在 $a = 0, 1, 2, 3, 4, 5, 6, 7$ 的情况下任务 τ_2 的响应时间分别为 (2, 2, 2, 2, 4, 3, 2)。因此可知任务 τ_2 的最大响应时间为 4。

结论 针对 EDF 算法, 研究了基于 (m, k) 模型的任务集在最坏情况下可调度的必要和充分条件, 并给出了一个简化的判断任务集可调度的方法, 最后给出了基于 (m, k) 模型的任务的最大响应时间的计算方法。本文所给方法解决了基于 (m, k) 模型的实时任务的可调度分析问题, 对于分析基于 (m, k) 模型的实时系统具有一定的理论指导作用。

参考文献

- Hamdaoui M, Ramanathan P. A Dynamic Priority Assignment Technique for Streams with (m, k) -Firm Deadlines. IEEE Transactions on Computer, 1995, 44(12): 1443~1451
- Koren G, Shasha D. Skip-Over: Algorithms and Complexity for Overloaded Systems that Allow Skips. In: Proceedings of the 16th IEEE Real-Time Systems Symposium, Pisa, Italy, 1995. 110~117
- Bernat G, Cayssials R. Guaranteed On-Line Weakly-Hard Real-Time Systems. In: Proceedings 22nd Real-Time Systems Symposium, London, England, 2001. 25~35
- Liu C L, Layland J. Scheduling Algorithm for Multiprogramming in a Hard Real-Time Environment. Journal of the ACM, 1973, 20(1): 46~61
- Spuri M. Analysis of Deadline Scheduled Real-Time Systems. [Technique report]. 772, Inria, 1996
- Jeffay K, Stone D. Accounting for interrupt handling costs in dynamic priority task systems. In: Proceedings of the 14th IEEE Symposium on Real-Time Systems, 1993. 212~221
- Spuri M. Holistic Analysis for Deadline Scheduled Real-Time Distributed System. [Technique report]. 873, Inria, 1996
- Ramanathan P. Overload Management in Real-Time Control Applications Using (m, k) -Firm Guarantee. IEEE Transactions on Parallel and Distributed Systems, 1999, 10(6): 549~559
- Quan G, Hu X. Enhanced Fixed-Priority Scheduling with (m, k) -Firm Guarantee. In: IEEE Real-Time Systems Symposium, 2000. 79~88
- George L, Rivierre N, Spuri M. Preemptive and non-preemptive real-time uniprocessor scheduling. [Technique report]. 873, Inria, 1996