Análisis de Factibilidad de Sistemas de Tiempo Real PMC con Granularidad Gruesa

Javier Orozco, Ricardo Cayssials, Rodrigo Santos y Edgardo Ferro

Instituto de Ciencias e Ingeniería de la Computación
Dpto. de Ing. Eléctrica, Universidad Nacional del Sur
Bahía Blanca, Argentina
e-mail: {ieorozco, iecayss}@criba.edu.ar

Resumen

La disciplina de Prioridades Fijas por Períodos Monotónicos Crecientes PMC es una norma de facto impuesta por el Departamento de Defensa de USA y aceptada, en consecuencia, por IBM, Honeywell, Mc Donnell Douglas, Boeing, General Electric, General Dynamics, NASA entre otras. En numerosas implementaciones de diagramadores PMC, el principal inconveniente que debe afrontarse es que el número disponible de prioridades es limitado y, en general puede ser menor al numero de usuarios del sistema reduciendo así, los posibles niveles de utilización del mismo. En lo que sigue se propone un método para determinar la factibilidad de un sistema multitarea-monoprocesador con limitado número de prioridades operando un diagramador por PMC. Por otro lado se propone un método para la determinación de la factibilidad de sistemas PMC con una complejidad menor a la de los métodos tradicionales.

Palabras Clave: Tiempo-Real, Diagramadores, PMC

1. Introducción y definiciones

Consideramos un conjunto de tareas periódicas y apropiativas S(m)={τ_1, τ_2, …., τ_m} con períodos T(m) = {T_1, T_2, …., T_m}, vencimientos D(m) = {D_1, D_2, …., D_m} y tiempos de ejecución C(m) = {C_1, C_2, …., C_m} a ser ejecutado en un sistema monoprocesador con P niveles de prioridad.

Si $P < m$, para solucionar el problema del número limitado de niveles de prioridad, debe realizarse una partición sobre el conjunto de m tareas en clases de prioridad indistinguible, de manera que el número de clases resultante sea menor o igual al número de niveles de prioridad disponibles. En principio las tareas pertenecientes a una clase tienen la misma prioridad para el diagramador y, frente a un requerimiento simultáneo de un par o más de ellas podrá utilizarse un criterio arbitrario de desempate.

En [3] se propone una técnica para determinar la diagramabilidad PMC de sistemas de tareas periódicas, apropiativas e independientes con un número limitado de niveles de prioridad mediante la utilización de una grilla logarítmica. El ésta, Se asumen P niveles de prioridad correspondientes a una grilla {0, L_1, ..., L_j, …, L_P}. Si el período T_i de la tarea τ_i es tal que $T_i \in (L_{j-1}, L_j]$, para $1 \leq j \leq P$ luego, se asignará la prioridad j a τ_i.

2. Efecto del número limitado de niveles de prioridad.

Al reducir el número de niveles de prioridad, varias tareas compartirán una misma clase resultando por lo tanto de prioridades indistinguibles para el diagramador. Si más de una tarea perteneciente a una misma clase se encuentra lista para ser ejecutada, el diagramador podrá realizar una asignación aleatoria produciendo un efecto similar al que producen las inversiones de prioridad.

El concepto de inversión de prioridad aplicable en el contexto de diagramadores con ilimitado número de niveles de prioridad debe interpretarse ahora como la utilización del procesador por tareas que, aún teniendo mayor período, utilizan el procesador antes que otra de menor período, por pertenecer a una misma clase. Aún así, el concepto de inversión de prioridad podría extenderse a inversiones entre clases.

2.1 El método de las ranuras vacías con inversión de prioridades

En [8] se ha demostrado que la j-ésima ranura que deja libre un sistema $\mathbf{S}(m)$ está dada por la expresión:

$$e_{j(\mathbf{S}(m))} = \text{ menor } \; t \mid t = j + W_m(t) = j + \sum_{h=1}^{m} C_h . \left\lceil \frac{t}{T_h} \right\rceil \tag{1}$$

y que un sistema $\mathbf{S}(m)$ será factible bajo la disciplina PMC sss

$$\forall i \in \{2,...,m\} \quad \sum_{h=1}^{i-1} \frac{C_i}{T_h} < 1 \quad y \quad T_i \geq D_i \geq e_{C_i(\mathbf{S}(i-1))} \tag{2}$$

donde la primera desigualdad indica la condición para que un subsistema $\mathbf{S}(i\text{-}1)$ admita la incorporación de una nueva tarea. La segunda expresión indica la condición que debe cumplir el período y vencimiento de τ_i para que pueda completar su ejecución antes de alcanzar su vencimiento. Este análisis presupone la existencia de tantos niveles de prioridad como tareas tenga el sistema.

En [6] y [7] se ha demostrado que en sistemas en los que el cumplimiento estricto de una disciplina PMC se realiza luego de un intervalo en el cual se produce un cierto número de asignaciones aleatorias. Este comportamiento aleatorio puede producir *inversiones de prioridad* en el sentido de que una tarea de baja prioridad se le asigna el procesador cuando compite con otra de mayor prioridad. Si el intervalo en el que pueden producirse las inversiones es [1, k] y, a partir de la ranura k+1 el diagramador cumple estrictamente con la disciplina PMC, el sistema es factible si se cumple:

$$\forall i \in \{1,2,\ldots,(m-I)\} \qquad T_i \geq e_{(C_i+k)(\mathbf{S}(i-1))} \tag{3}$$

$$\text{donde } I = \max_{1\leq s\leq m} s \mid \sum_{h=m}^{m-s+1} \frac{C_h}{k} \leq 1$$

Notemos que, ante el peor estado de carga y de inversión de prioridades, sólo las primeras m-I tareas pueden ser afectadas por k inversiones de prioridad y, en consecuencia, podrán ejecutarse antes de sus vencimientos si los mismos son mayores o iguales que la primera ranura que dejan libre las tareas que las preceden en prioridad (ec.(2)) a los cuales deben sumarse las k ranuras ocupadas por inversión de prioridad (ec.(3)).

Para lo que sigue, se introduce la siguiente notación τ_i^j indica el i-ésimo elemento dentro de la clase j; C_i^j representa su tiempo de ejecución y T_i^j y D_i^j sus vencimientos respectivamente. τ_i indica el i-ésimo elemento se **S(*m*).** La equivalencia entre la nueva notación y la anterior está dada por: $C_1^2 = C_{|Q_1|+1}$, $T_1^2 = T_{|Q|+1}$.

<u>Teorema 1:</u>

Dada una partición sobre **S**(*m*) ordenado PMC tal que:

$$\mathbf{Q}(\mathbf{S}(m))=\{Q_1, Q_2, \ldots. Q_P\},\ \sum_{h=1}^{P}|Q_h| = m \text{ y } \forall\tau\in \mathbf{S}(m)\ \exists!\ Q|\ \tau\in Q.$$

S(*m*) será factible por un diagramador PMC si y sólo si

$$\forall\tau\in\{\tau_1^1,\tau_1^2,\ldots.,\tau_1^p\} \qquad T_1^j \geq e_{(C_1^j+\varepsilon_1^j)(\tilde{\mathbf{Q}}(j))} \tag{4}$$

donde

$$\varepsilon_1^j = \sum_{h|\tau_h^j\neq\tau_1^j} C_h^j \quad \text{y} \quad \tilde{\mathbf{Q}}(j) = \bigcup_{v=1}^{j-1} Q_v$$

ε^j representa el número de ranuras necesarias para ejecutar todas las tareas pertenecientes a la clase j excluyendo a τ_1^j y $\tilde{\mathbf{Q}}(j)$ el subconjunto de tareas de **S**(*m*) pertenecientes a las clases de mayor prioridad que τ_1^j.

<u>Demostración:</u>

En (4), la tarea de menor período de cada clase deberá soportar no sólo la ejecución de las tareas pertenecientes a clases de mayor prioridad sino la ejecución de aquellas que, perteneciendo a la misma clase se ejecutan previamente.

Como el número de tareas de mayor período que pueden ejecutarse previamente a $\tau^1(j)$ está limitado a ε_1^j, si se verifica (3) $\forall\,\tau_1^j$ con $k=\ \varepsilon_1^j$, se cumple (4) $\forall\,\tau_1^j$ con lo que, la primer tarea de cada clase resulta factible.

Si se verifica (4) $\forall\,\tau_1^j$, luego para una clase cualesquiera:

$$T_1^j \geq \min t' \Big| t' = C_1^j + \sum_{2\leq h\leq|Q_j|} C_h^j + \sum_{\forall\tau_l\in\tilde{\mathbf{Q}}(j)} C_l . \left\lceil \frac{t'}{T_l} \right\rceil \tag{5}$$

Para resultar factible, una tarea $\tau^n(j)$ completará su ejecución en el peor caso de carga luego que: todas las tareas pertenecientes a las clases de mayor prioridad lo hayan hecho más todas las de su propia clase. En este último caso de deberá considerar que se producirá más de un arribo de las tareas de menor período que $T^n(j)$, aún cuando pertenezcan a la misma clase. Con lo que:

$$T_n^j \geq \min t'' | t'' = C_n^j + \sum_{n+1\leq h\leq|Q_j|} C_h^j + \sum_{1\leq h\leq n-1} C_h^j . \left\lceil \frac{t''}{T_h^j} \right\rceil + \sum_{\forall\tau_l\in\tilde{\mathbf{Q}}(j)} C_l . \left\lceil \frac{t''}{T_l} \right\rceil \tag{6}$$

calculando la expresión anterior para $t'' = t'$

$$C_n^j + \sum_{n+1 \le h \le |Q_j|} C_h^j + \sum_{1 \le h \le n-1} C_h^j . \left\lceil \frac{t}{T_h^j} \right\rceil + \sum_{\forall \tau_l \in \tilde{\mathbf{Q}}(j)} C_l . \left\lceil \frac{t}{T_l} \right\rceil =$$

$$= \sum_{n \le h \le |Q_j|} C_h^j + \sum_{1 \le h \le n-1} C_h^j . \left\lceil \frac{t'}{T_h^j} \right\rceil + \sum_{\forall \tau_l \in \tilde{\mathbf{Q}}(j)} C_l . \left\lceil \frac{t}{T_l} \right\rceil$$

como $t \le T_1^j$, $\left\lceil \frac{t}{T_1^j} \right\rceil = 1$ luego, para $T_l \ge T_1^j$

$$\sum_{1 \le l \le n-1} C_l^j . \left\lceil \frac{t'}{T_l^j} \right\rceil = \sum_{1 \le l \le n-1} T_l^j$$

con lo que

$$\sum_{1 \le l \le |Q_j|} C_l^j + \sum_{\forall \tau_l \in \tilde{\mathbf{Q}}(j)} C_l . \left\lceil \frac{t}{T_l} \right\rceil \le T_1^j \le T_n^j$$

Por lo tanto $t = t'$ resulta ser solución de (6). Luego basta que se cumpla (4) para determinar la diagramabilidad de $\mathbf{S}(m)$.

Corolario: Si un sistema $\mathbf{S}(m)$ es factible PMC, existirá al menos una partición $\mathbf{Q}(\mathbf{S}(m))=\{Q_1, Q_2, \ldots. Q_P\}$ con $P \le m$.

Demostración: trivial a partir del Teorema 1.

<u>Ejemplo 1:</u>

$\mathbf{S}(5)=\{\tau_1, \tau_2, \ldots, \tau_5\}$, $\mathbf{T}(5) = \mathbf{D}(5) = \{6, 10, 14, 18, 18\}$, $\mathbf{C}(5) = \{2, 2, 2, 2, 2\}$, $\mathbf{Q}(\mathbf{S}(5)) = \{Q_1, Q_2\}=\{\{\tau_1, \tau_2, \tau_3\}, \{\tau_4, \tau_5\}\}= \{\{\tau_1^1, \tau_2^1, \tau_3^1\}, \{\tau_1^2, \tau_2^2\}\}$

S(5)	T_i	C_i	Q
τ_1	6	2	1
τ_2	10	2	1
τ_3	14	2	1
τ_4	18	2	2
τ_5	18	2	2

Aplicando (4): $\forall \tau \in \{\tau_1^1, \tau_1^2\}$ $\qquad T_1^j \ge e_{(C_1^j + \varepsilon_1^j)(\tilde{\mathbf{Q}}(j))}$

para τ_1^1 $\qquad 6 \ge e_{(2+4)(\varnothing)} \Rightarrow 6 \ge \text{menor } t | t = 6 + 0 = 6$

para τ_1^2 $\qquad 18 \ge e_{(2+2)(Q_1)} \Rightarrow 18 \ge \text{menor } t | t = 4 + 2.\left\lceil \frac{t}{6} \right\rceil + 2.\left\lceil \frac{t}{10} \right\rceil + 2.\left\lceil \frac{t}{14} \right\rceil = 18$

dentro de un período de τ_2^1, τ_1^1 producirá $\lceil 10/6 \rceil$=2 arribos, requiriendo (2.$\lceil 10/6 \rceil$ = 4) ranuras para su ejecución.

Aún cuando τ_1^1 se ejecute completamente en todos sus arribos antes que τ_2^1 y que esta última soporte la ejecución de τ_3^1, se ve que la ec. (4) es suficiente para analizar la diagramabilidad del sistema. Por otro lado y en general, si τ_1^j puede soportar la ejecución del resto de las tareas de su propia clase más su propia ejecución más la de las clases superiores, también podrán hacerlo τ_2^j, τ_3^j,.... que tienen un período mayor que τ_1^1y soportan la misma carga. Luego aplicando (4)

$$\text{para } \tau_1^2 \qquad 10 \geq e_{(2+2)(\{\tau_1^1\})} \Rightarrow 10 \geq \text{menor } t | t = 4 + 2.\left\lceil \frac{t}{6} \right\rceil = 6$$

$$\text{para } \tau_1^3 \qquad 14 \geq e_{(2+0)(\{\tau_1^1, \tau_1^2\})} \Rightarrow 14 \geq \text{menor } t | t = 2 + 2.\left\lceil \frac{t}{6} \right\rceil + 2.\left\lceil \frac{t}{10} \right\rceil = 6$$

En forma idéntica se pueden obtener los resultados para Q_2.

Utilizando los resultados previos, se presenta un método para determinar la diagramabilidad de sistemas PMC con una complejidad en general menor a la de los métodos tradicionales.

<u>Lema 1:</u>

Si existe una partición $\mathbf{Q}(\mathbf{S}(m))=\{Q_1, Q_2, \ldots. Q_P\}$ con $P \leq m$, $\mathbf{S}(m)$ es factible PMC.

<u>Demostación:</u>

Su demostración resulta trivial a partir del Teorema 1.

Por la ec. (2) si $T_i \geq e_{C_i(\mathbf{S}(i-1))}$, en el intervalo [1, $e_{C_i(\mathbf{S}(i-1))}$] existirán suficientes ranuras como para atender los requerimientos de las tareas de mayor prioridad más los de τ_i, de lo contrario el sistema es no factible. Por el Teorema 1 si $T_{i-1} \geq e_{C_i(\mathbf{S}(i-1))}$, τ_{i-1} podrá pertenecer a la clase $\{\tau_{i-1}, \tau_i\}$ y resultar aún factible. Nótese que esta incorporación no afecta a la diagramabilidad de τ_i ya que ha sido considerada la carga impuesta por τ_{i-1}, en el cálculo de $e_{C_i(\mathbf{S}(i-1))}$. Continuando el razonamiento, se podrán seguir incorporando tareas a la clase mientras $T_{i-l} \geq e_{C_i(\mathbf{S}(i-1))}$, $l \in$ [1, $(i$-1)].

Si, para una dada τ_x, $x<i$, $T_x < e_{C_i(\mathbf{S}(i-1))}$, como no puede incorporarse a la clase a la que pertenece τ_i , se calcula $e_{C_x(\mathbf{S}(x-1))}$ y se procede de igual manera intentando incorporar tareas a la nueva clase. Luego, si se inicia el procedimiento en τ_m y se obtiene una partición $\mathbf{Q}(\mathbf{S}(m))$, $\mathbf{S}(m)$ es PMC factible.

<u>Ejemplo 2:</u>

Sobre el ejemplo anterior $\mathbf{S}(5)=\{\tau_1, \tau_2, \ldots, \tau_5\}$, $\mathbf{T}(5)=\mathbf{D}(5)=\{6, 10, 14, 18, 18\}$, $\mathbf{C}(5)=\{2, 2, 2, 2, 2\}$

$$e_{C_5(\mathbf{S}(4))} = \text{menor } t | t = 2 + 2.\left\lceil \frac{t}{6} \right\rceil + 2.\left\lceil \frac{t}{10} \right\rceil + 2.\left\lceil \frac{t}{14} \right\rceil + 2.\left\lceil \frac{t}{18} \right\rceil = 18$$

con lo que podrá crearse la clase $\{\tau_4, \tau_5\}$

$$e_{C_3(\mathbf{S}(2))} = \text{menor } t | t = 2 + 2.\left\lceil \frac{t}{6} \right\rceil + 2.\left\lceil \frac{t}{10} \right\rceil = 6$$

con lo que podrá crearse la clase $\{\tau_1, \tau_2, \tau_3\}$. Como ambas clases cubren al sistema, el mismo es factible.

Ejemplo 3:

$\mathbf{S}(9)$	T_i	$e_{C_i(\mathbf{S}(i-1))}$	Verificación	clase
9	54	34	54≥34	4
8	34		34≥34	4
7	20	18	20≥18	3
6	12	12	12≥12	2
5	9	5	9≥5	1
4	7		7≥5	1
3	7		7≥5	1
2	6		6≥5	1
1	5		5≥5	1

En el ejemplo se observa que el sistema resulta factible aún con una partición en cuatro clases. Luego el sistema es PMC factible. Nótese que el cálculo de la factibilidad del sistema se limitó a calcular la expresión (4) para τ_i = (9, 8, 7, 6).

Si fuese T_6=10 el sistema es no factible ya que 10 < 12.

i	T_i	$e_{C_i(\mathbf{S}(i-1))}$	Verificación	clase
9	54	54	54≥54	n
8	34	34	34≥34	n-1
7	20	18	20≥18	n-2
6	10	12	**10<12**	-

A fin de determinar la factibilidad de sistemas PMC es usual la utilización de la expresión (2). En este método la complejidad está dada por: $\mathbf{O}=m^2.T_m$, donde T_m representa el máximo período del sistema. En el método propuesto la misma está dada por: $\mathbf{O'}=P.m.T_m$ siendo $\mathbf{O'} \leq \mathbf{O}$.

3. Conclusiones

En este trabajo se presenta un método para determinar la factibilidad de sistemas PMC con limitado número de clases de prioridad. En el contexto de los sistemas operativos de tiempo real esta restricción es posible salvarla ya que, en principio, podría diseñarse un diagramador con tantos niveles de prioridad como sea necesario o un número considerable de ellas (256 niveles en el estándar POSIX de Tiempo Real 1003.b). En redes de comunicaciones, desarrollos específicos basados en microcontroladores, etc., el diagramador es implementado por medio de un hardware vinculado al control de acceso al medio o al diagramador, resultando entonces razonable pensar en un conjunto limitado de recursos.

Por otro lado se propone un método para determinar la factibilidad PMC con una complejidad menor que la utilizada por los métodos tradicionales.

Referencias

[1] Liu, and J.W. Layland. 1973. Scheduling algorithms for multiprogramming in hard real time enviroments. *J. ACM*, 20(1):46-61.

[2] Joseph and P. Pandya. 1986. Finding response times in a real time system. *The Computer Journal*, 29(5):390-395.

[3] Lehosky J.P. and Sha L. 1986. Perfomance of Real-time Bus Scheduling Algorithms. *ACM Performance*, Vol.14 , pag. 44-53.

[4] Lehozcky, L. Sha and Y. Ding. 1989. The rate monotonic scheduling algorithm: Exact characterization and average case behavior. In *Proc. Real Time Systems Symp. IEEE CS*, Los Alamitos, CA.

[5] Obenza. 1993. Rate monotonic analysis for real-time systems. *IEEE Computer*, 26(3):73-74.

[6] Orozco and J. Santos. 1994. Diagramación de tareas en sistemas operativos de tiempo real PMC con inversión de prioridades. In *Proc.* 23 *Jornadas Argentinas de Informática e Investigación Operativa*, pp. 1.25-1.35.

[7] Santos and J. Orozco. 1994. On the priority mechanism of 802.4, in hard real-time factory communications. In *Proc. IEEE International Symposium on Industrial Electronics*, pp. 281-285.

[8] Santos, M.L. Gastaminza, J. Orozco and C. Matrangolo. 1994. 802.5 priority mechanism in hard real-time RMS applications. *Computer Communications*, 17(6):439-442.

[9] Santos, M.L. Gastaminza, J. Orozco, D. Picardi and O. Alimenti. 1991. Priorities and protocols in hard real-time LANs: implementing a crisis-free system. *Computer Communications*, 14(9):507-514.