Sistemi Embedded e Real-time (M. Cesati)

Compito scritto del 8 luglio 2011

Esercizio 1. Si consideri il seguente sistema di task periodici in fase schedulato su un processore con un algoritmo "cyclic schedule": $T_1 = (9, 1, 10)$, $T_2 = (10, 2)$, $T_3 = (15, 2, 10)$, $T_4 = (18, 3, 8)$. I job sono non interrompibili.

(a) Determinare la dimensione del frame che minimizza l'overhead dello scheduler.

(b) Qual è la minima dimensione del frame che garantisce l'esistenza di una schedulazione ciclica fattibile? Giustificare la risposta.

Esercizio 2. Un server procastinabile con periodo $p_s = 6$, budget $e_s = 2$ e fase indeterminata è schedulato su un singolo processore insieme a due task periodici, indipendenti e interrompibili: $T_1 = (10,\ 1)$ e $T_2 = (11,\ 1)$.

(a) Determinare analiticamente se il sistema è schedulabile con EDF.

(b) Determinare analiticamente se il sistema è schedulabile con RM.

(c) Il server procastinabile è sostituito da un server sporadico semplice con gli stessi parametri $p_s = 6$ e $e_s = 2$: cosa si può concludere sulla schedulabilità con EDF e con RM?

Esercizio 3. In fase di progettazione di un sistema real-time multiprocessore si prevede di dover eseguire al massimo 10 task periodici $T_i = (2, 1)$, 10 task periodici $T_i' = (3, 1/5)$ e 10 task periodici $T_i'' = (5, 1/3)$.

(a) Supponendo di utilizzare uno scheduler partizionato RM-FF, determinare un numero minimo di processori che garantisca il rispetto di tutte le scadenze dei vari task.

(b) Ripetere l'analisi del punto (a) supponendo di utilizzare uno scheduler partizionato EDF-FF.

Sistemi Embedded e Real-time (M. Cesati)

Soluzioni del compito scritto del 8 luglio 2011

Esercizio 1. *Si consideri il seguente sistema di task periodici in fase schedulato su un processore con un algoritmo "cyclic schedule":* $T_1 = (9, 1, 10)$, $T_2 = (10, 2)$, $T_3 = (15, 2, 10)$, $T_4 = (18, 3, 8)$. *I job sono non interrompibili.*

(a) *Determinare la dimensione del frame che minimizza l'overhead dello scheduler.*

Determiniamo la dimensione del frame f più appropriata:

- Vincolo sulle fasi dei task: tutti i task sono in fase, quindi le fasi non impongono alcun vincolo per f.
- Vincolo sui tempi d'esecuzione dei job:
$$f \geq \max\{1, 2, 2, 3\} = 3$$
- Vincolo sulla divisibilità della lunghezza dell'iperperiodo ($\text{mcm}\{9, 10, 15, 18\} = 90$):
$$f \in \{1, 2, 3, 5, 6, 9, 10, 15, 18, 30, 45, 90\}.$$
Quindi combinando con le due condizioni precedenti si ha:
$$f \in \{3, 5, 6, 9, 10, 15, 18, 30, 45, 90\}.$$
- Vincolo sul task T_1 ($2f - \gcd\{9, f\} \leq 10$):
$$\begin{array}{lll} 2 \cdot 3 - \gcd\{9, 3\} = 3 & \leq 10 & \text{ok} \\ 2 \cdot 5 - \gcd\{9, 5\} = 9 & \leq 10 & \text{ok} \\ 2 \cdot 6 - \gcd\{9, 6\} = 9 & \leq 10 & \text{ok} \\ 2 \cdot 9 - \gcd\{9, 9\} = 9 & \leq 10 & \text{ok} \\ 2 \cdot 10 - \gcd\{9, 10\} = 19 & > 10 & \text{no} \end{array}$$
Per ogni $f \geq 10$, $2f - \gcd\{9, f\} \geq 20 - 9 = 11 > 10$, quindi i vincoli imposti dal task T_1 comportano:
$$f \in \{3, 5, 6, 9\}.$$
- Vincolo sul task T_2 ($2f - \gcd\{10, f\} \leq 10$):
$$\begin{array}{lll} 2 \cdot 3 - \gcd\{10, 3\} = 5 & \leq 10 & \text{ok} \\ 2 \cdot 5 - \gcd\{10, 5\} = 5 & \leq 10 & \text{ok} \\ 2 \cdot 6 - \gcd\{10, 6\} = 10 & \leq 10 & \text{ok} \\ 2 \cdot 9 - \gcd\{10, 9\} = 17 & > 10 & \text{no} \end{array}$$

- Vincolo sul task T_3 ($2f - \gcd\{15, f\} \leq 10$):

$$\begin{array}{lll} 2 \cdot 3 - \gcd\{15,3\} = 3 & \leq 10 & \text{ok} \\ 2 \cdot 5 - \gcd\{15,5\} = 5 & \leq 10 & \text{ok} \\ 2 \cdot 6 - \gcd\{15,6\} = 9 & \leq 10 & \text{ok} \end{array}$$

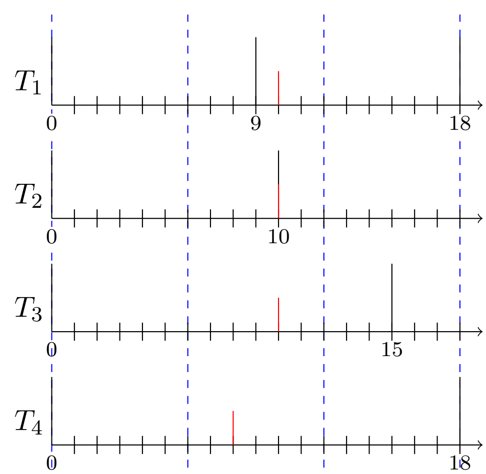
- Vincolo sul task T_4 ($2f - \gcd\{18, f\} \leq 8$):

$$\begin{array}{lll} 2 \cdot 3 - \gcd\{18,3\} = 3 & \leq 8 & \text{ok} \\ 2 \cdot 5 - \gcd\{18,5\} = 9 & > 8 & \text{no} \\ 2 \cdot 6 - \gcd\{18,6\} = 6 & \leq 8 & \text{ok} \end{array}$$

In conclusione, sono ammissibili come dimensione del frame $f = 3$ e $f = 6$. Il valore che minimizza l'overhead dello scheduler è $f = 6$.

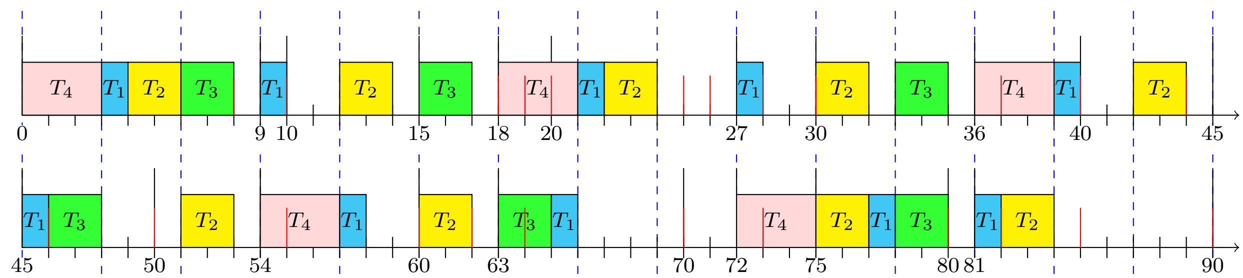
(b) *Qual è la minima dimensione del frame che garantisce l'esistenza di una schedulazione ciclica fattibile? Giustificare la risposta.*

Consideriamo i primi tre frame con dimensione $f = 6$:



All'istante $t = 0$, all'inizio del primo frame, sono rilasciati tutti i job dei quattro task. Poiché le scadenze di tutti i quattro job cadono all'interno del secondo frame, tutti i job dovrebbero essere eseguiti nel primo frame. Ciò è evidentemente impossibile, in quanto il frame di lunghezza 6 non consente di eseguire tutti i quattro job aventi tempo di esecuzione totale pari a $1 + 2 + 2 + 3 = 8$.

Consideriamo dunque la dimensione del frame $f = 3$. In questo caso esiste una schedulazione ciclica fattibile. A riprova di ciò si consideri la seguente schedulazione nell'iperperiodo tra 0 e 90:



Esercizio 2*. Un server procastinabile con periodo $p_s = 6$, budget $e_s = 2$ e fase indeterminata è schedulato su un singolo processore insieme a due task periodici, indipendenti e interrompibili:* $T_1 = (10, 1)$ *e* $T_2 = (11, 1)$.

(a) *Determinare analiticamente se il sistema è schedulabile con EDF.*

Utilizzando il fattore di utilizzazione, condizione sufficiente per la schedulabilità di T_i è:

$$\sum_{k=1}^{n} \frac{e_k}{\min(D_k, p_k)} + \frac{e_s}{p_s}\left(1 + \frac{p_s - e_s}{D_i}\right) \le 1$$

- Server procastinabile: $1/10 + 1/11 + 2/6 = 173/330 \le 1 \quad \Rightarrow$ schedulabile
- Task T_1: $1/10 + 1/11 + (2/6)(1 + (6-2)/10) = 217/330 \le 1 \quad \Rightarrow$ schedulabile
- Task T_2: $1/10 + 1/11 + (2/6)(1 + (6-2)/11) = 71/110 \le 1 \quad \Rightarrow$ schedulabile

Il sistema di task è dunque schedulabile con EDF.

(b) *Determinare analiticamente se il sistema è schedulabile con RM.*

Utilizzando il fattore di utilizzazione, condizione sufficiente per la schedulabilità di T_i è:

$$\sum_{k=1}^{i} \frac{e_k}{p_k} + \frac{e_s}{p_s} + \frac{e_s}{p_i} \le U_{\text{RM}}(i+1)$$

(a condizione che per ogni task $T_1, \ldots, T_i$ la scadenza relativa coincida con il periodo)

- Server procastinabile: $e_s/p_s = 2/6 \le 1 \quad \Rightarrow$ schedulabile
- Task T_1: $1/10 + 2/6 + 2/10 = 19/30 < 0.64 < 0.828 < U_{\text{RM}}(2) \quad \Rightarrow$ schedulabile
- Task T_2: $1/10 + 1/11 + 2/6 + 2/11 = 233/330 < 0.71 < 0.779 < U_{\text{RM}}(3) \quad \Rightarrow$ schedulabile

Il sistema di task è perciò schedulabile con RM.

In alternativa: si verifica che le condizioni del teorema di Lehoczky, Sha e Strosnider sono soddisfatte: $p_s < p_1 < p_2 < 2p_s$ e $p_2 > p_s + e_s$. Perciò è possibile confrontare il fattore di utilizzazione totale del sistema (incluso il server procastinabile) con il valore di soglia

$$U_{\mathrm{RM/DS}}(n) = \frac{e_s}{p_s} + n \cdot \left[\left(\frac{e_s + 2p_s}{p_s + 2e_s} \right)^{1/n} - 1 \right]$$

Si ha: $U_{\mathrm{RM/DS}}(2) = e_s/p_s + 2\left(\sqrt{(e_s + 2p_s)/(p_s + 2e_s)} - 1\right) = 1/3 + 2\sqrt{1 + 2/5} - 2 \approx 1/3 + 0.366$. Poiché il fattore di utilizzazione totale è pari a $e_s/p_s + e_1/p_1 + e_2/p_2 = 1/3 + 1/10 + 1/11 \approx 1/3 + 0.191$, il sistema è schedulabile con RM.

(c) *Il server procastinabile è sostituito da un server sporadico semplice con gli stessi parametri $p_s = 6$ e $e_s = 2$: cosa si può concludere sulla schedulabilità con EDF e con RM?*

Poiché è stato dimostrato analiticamente che il sistema di task è schedulabile con un server procastinabile sia con EDF che con RM, sostituendo il server procastinabile con un server sporadico semplice avente le stesse dimensioni si continuerà ad avere la garanzia di schedulabilità con entrambi gli algoritmi. Infatti, il server sporadico semplice si comporta nel caso peggiore come un qualunque task periodico, e quindi a differenza del server procastinabile non richiede di considerare nell'analisi di schedulabilità tempi di blocco aggiuntivi. Perciò le conclusioni dell'analisi svolta precedentemente sono valide, a maggior ragione, anche per un sistema con un server sporadico semplice.

Esercizio 3*. In fase di progettazione di un sistema real-time multiprocessore si prevede di dover eseguire al massimo 10 task periodici $T_i = (2, 1)$, 10 task periodici $T_i' = (3, 1/5)$ e 10 task periodici $T_i'' = (5, 1/3)$.*

(a) *Supponendo di utilizzare uno scheduler partizionato RM-FF, determinare un numero minimo di processori che garantisca il rispetto di tutte le scadenze dei vari task.*

Il fattore di utilizzazione dell'algoritmo di schedulazione partizionato RM-FF è

$$U_{\mathit{RM\text{-}FF}}(m) = m \cdot \left(\sqrt{2} - 1\right)$$

ove m è il numero di processori nel sistema. I task periodici da eseguire hanno un fattore di utilizzazione pari a

$$U_T = 10 \cdot \frac{1}{2} + 10 \cdot \frac{1/5}{3} + 10 \cdot \frac{1/3}{5} = 6 + \frac{1}{3}$$

Se $U_T \leq U_{\mathit{RM\text{-}FF}}(m)$, tutte le scadenze dei task sono rispettate, il che significa che m processori sono sufficienti. Si ottiene perciò:

$$m \geq \frac{6 + 1/3}{\sqrt{2} - 1} \approx 15.28$$

È quindi necessario includere nel sistema almeno 16 processori.

(b) *Ripetere l'analisi del punto (a) supponendo di utilizzare uno scheduler partizionato EDF-FF.*

Il fattore di utilizzazione dell'algoritmo di schedulazione partizionato EDF-FF è

$$U_{\textit{EDF-FF}}(m, \beta) = \frac{\beta \cdot m + 1}{\beta + 1}$$

ove m è il numero di processori e $\beta = \lfloor 1/ \max_k \{e_k/p_k\} \rfloor$.

Nel caso in esame, il massimo fattore di utilizzazione dei task è 1/2, quindi $\beta = 2$. Di conseguenza, un numero sufficiente di processori per garantire la fattibilità del sistema di task si ottiene risolvendo $U_T \leq U_{\textit{EDF-FF}}(m, 2)$, ossia:

$$6 + \frac{1}{3} \leq \frac{2\,m + 1}{2 + 1}$$

che equivale a

$$m \geq \frac{3 \cdot (6 + 1/3) - 1}{2} = 9.$$

È quindi necessario includere nel sistema almeno 9 processori.