Sistemi Embedded e Real-time (M. Cesati)

Compito scritto del 9 luglio 2010

Esercizio 1. Il seguente sistema di task periodici in fase è schedulato su un processore con un algoritmo "cyclic schedule": $T_1 = (5, 2)$, $T_2 = (4, 1, 9)$, $T_3 = (8, 2, 9)$, $T_4 = (10, 1)$.

(a) Determinare le possibili dimensioni intere appropriate per il frame assumendo che i job sono non interrompibili.

(b) Dimostrare che non esiste alcuna schedulazione fattibile per i task periodici utilizzando la dimensione del frame più grande tra quelle determinate nel punto (a).

Esercizio 2. Si consideri un sistema di task periodici e indipendenti schedulati su un processore:

$$\begin{aligned}
T_1 &= (5, 1.5, 4), \quad x_1 = 0.1, \quad K_1 = 1, \quad \theta_1 = 0.2 \\
T_2 &= (6, 0.2, 5), \quad x_2 = 1.25, \quad K_2 = 2, \quad \theta_2 = 0.1 \\
T_3 &= (7, \ 1 \ , 3), \quad x_3 = 0, \quad K_3 = 0, \quad \theta_3 = 0.5
\end{aligned}$$

ove x_i è il tempo massimo di auto-sospensione, K_i è il numero massimo di auto-sospensioni, e θ_i è la durata della più lunga sezione di codice non interrompibile.

Determinare analiticamente se il sistema è schedulabile con EDF.

Esercizio 3. Si consideri il seguente sistema di task:

$$T_1 = (10, 3),\ T_2 = (8, 3),\ T_3 = (6, 2),\ T_4 = (5, 2),\ T_5 = (2, 1),\ T_6 = (40, 15).$$

(a) Determinare analiticamente se l'insieme di task è schedulabile su 3 (**tre**) processori identici utilizzando l'algoritmo partizionato EDF-FF.

(b) Considerando l'ordine dei task indicato nel testo dell'esercizio, determinare il partizionamento dei task sui processori indotto da EDF-FF.

Sistemi Embedded e Real-time (M. Cesati)

Soluzioni del compito scritto del 9 luglio 2010

Esercizio 1. *Il seguente sistema di task periodici in fase è schedulato su un processore con un algoritmo "cyclic schedule":* $T_1 = (5, 2)$, $T_2 = (4, 1, 9)$, $T_3 = (8, 2, 9)$, $T_4 = (10, 1)$.

(a) *Determinare le possibili dimensioni intere appropriate per il frame assumendo che i job sono non interrompibili.*

- Vincolo sui tempi d'esecuzione dei job:

$$f \geq \max\{2, 1, 2, 1\} = 2$$

- Vincolo sulla divisibilità della lunghezza dell'iperperiodo ($\text{mcm}\{5, 4, 8, 10\} = 40$):

$$f \in \{2, 4, 5, 8, 10, 20, 40\}$$

- Vincolo sul task T_1 ($2f - \gcd\{5, f\} \leq 5$):

$$\begin{array}{ll}
2 \cdot 2 - \gcd\{5, 2\} = 3 \leq 5 & \\
2 \cdot 4 - \gcd\{5, 4\} = 7 > 5 & \Rightarrow f \neq 4 \\
2 \cdot 5 - \gcd\{5, 5\} = 5 \leq 5 & \\
2 \cdot x - \gcd\{5, x\} > 5 \text{ if } x > 5 & \Rightarrow f \leq 5
\end{array}$$

- Vincolo sul task T_2 ($2f - \gcd\{4, f\} \leq 9$):

$$\begin{array}{l}
2 \cdot 2 - \gcd\{4, 2\} = 2 \leq 9 \\
2 \cdot 5 - \gcd\{4, 5\} = 9 \leq 9
\end{array}$$

- Vincolo sul task T_3 ($2f - \gcd\{8, f\} \leq 9$):

$$\begin{array}{l}
2 \cdot 2 - \gcd\{8, 2\} = 2 \leq 9 \\
2 \cdot 5 - \gcd\{8, 5\} = 9 \leq 9
\end{array}$$

- Vincolo sul task T_4 ($2f - \gcd\{10, f\} \leq 10$):

$$\begin{array}{l}
2 \cdot 2 - \gcd\{10, 2\} = 2 \leq 10 \\
2 \cdot 5 - \gcd\{10, 5\} = 5 \leq 10
\end{array}$$

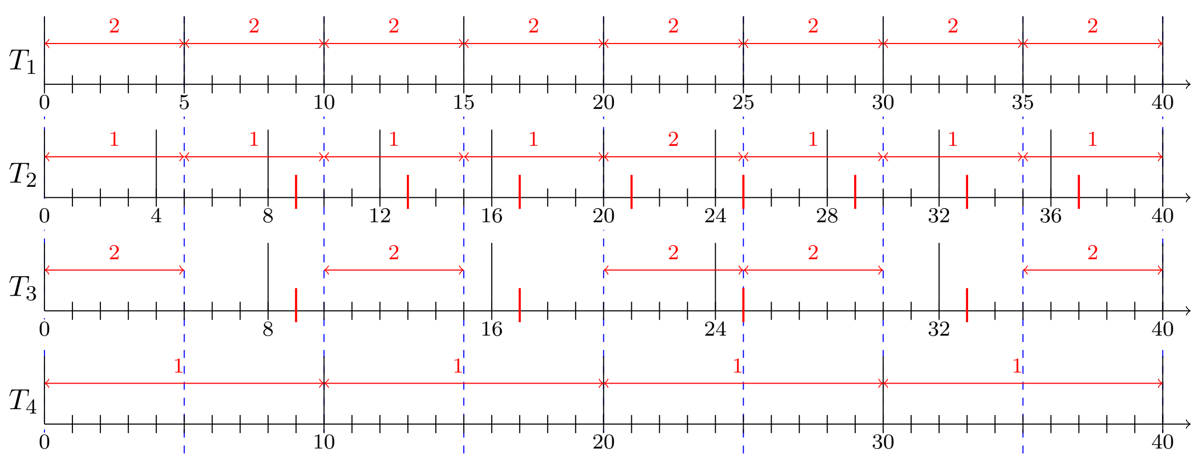
Le dimensioni intere ammissibili per il frame sono dunque $f = 2$ e $f = 5$.

(b) *Dimostrare che non esiste alcuna schedulazione fattibile per i task periodici utilizzando la dimensione del frame più grande tra quelle determinate nel punto (a).*

Per dimostrare formalmente la non esistenza di una schedulazione fattibile con dimensione del frame $f = 5$ si deve tenere presente che:

- Un job può essere schedulato solo se il suo istante di rilascio cade in un frame precedente oppure coincide con l'inizio stesso del frame.
- Lo scheduler deve controllare l'avvenuto rispetto delle scadenze dei vari job; per questo motivo, un job deve essere completato (e dunque deve essere completamente eseguito) in un frame precedente a quello contenente la sua scadenza oppure in un frame che termina esattamente nell'istante di scadenza del job.

Quindi per ciascun job è possibile determinare in quali frame esso deve essere eseguito. Si ottiene così lo schema seguente, in cui il lavoro da effettuare per ciascun task è rappresentato da frecce che si estendono tra uno o più frame e valori numerici che indicano la somma dei tempi d'esecuzione dei job in quei frame.



Nel frame iniziante all'istante 20 il lavoro totale da compiere è maggiore della lunghezza del frame. In effetti osserviamo che nei due frame compresi tra l'istante 20 e l'istante 30 il lavoro totale da compiere è pari a $2 + 2 + 2 + 1 + 2 + 2 + 2 + 1 = 12$, e di conseguenza il fattore di utilizzazione è

$$\frac{12}{10} > 1.$$

Perciò in qualunque schedulazione almeno uno dei job che devono essere eseguiti in questi due frame non potrà essere completato. Ciò conclude la dimostrazione.

Esercizio 2. *Si consideri un sistema di task periodici e indipendenti schedulati su un processore:*

$$\begin{array}{rcllll} T_1 & = & (5, 1.5, 4), & x_1 = 0.1, & K_1 = 1, & \theta_1 = 0.2 \\ T_2 & = & (6, 0.2, 5), & x_2 = 1.25, & K_2 = 2, & \theta_2 = 0.1 \\ T_3 & = & (7,\ 1\ , 3), & x_3 = 0, & K_3 = 0, & \theta_3 = 0.5 \end{array}$$

ove x_i è il tempo massimo di auto-sospensione, K_i è il numero massimo di auto-sospensioni, e θ_i è la durata della più lunga sezione di codice non interrompibile.

Determinare analiticamente se il sistema è schedulabile con EDF.

I task hanno scadenze inferiori od uguali al periodo e possono bloccare. Di conseguenza per determinare la schedulabilità tramite EDF dobbiamo controllare ciascun task singolarmente verificando la condizione:

$$\sum_{k=1}^{n} \frac{e_k}{\min(p_k, D_k)} + \frac{b_i}{\min(p_i, D_i)} \leq 1$$

Per determinare i tempi di blocco di ciascun task possiamo utilizzare il teorema di Baker che afferma che un job di un task con scadenza relativa D può bloccare un job di un altro task con scadenza relativa D' solo se $D > D'$.
Riordiniamo dunque i task in base alla loro scadenza relativa, ottenendo:

$$T'_1 = T_3, \quad T'_2 = T_1, \quad T'_3 = T_2.$$

Verifica della schedulabilità di T'_1:

$$\begin{aligned} b'_1(ss) &= x'_1 = x_3 = 0 \\ b'_1(np) &= \max\{\theta'_2, \theta'_3\} = \max\{\theta_1, \theta_2\} \\ &= \max\{0.2, 0.1\} = 0.2 \\ b'_1 &= b'_1(ss) + (1 + K'_1) \cdot b'_1(np) \\ &= 0 + (1 + K_3) \cdot 0.2 = 0.2 \end{aligned}$$

$$\begin{aligned} \sum_{k=1}^{n} \frac{e'_k}{\min(p'_k, D'_k)} + \frac{b'_1}{\min(p'_1, D'_1)} &= \frac{1}{\min(7,3)} + \frac{1.5}{\min(5,4)} + \frac{0.2}{\min(6,5)} + \frac{0.2}{\min(7,3)} \\ &= \frac{1}{3} + \frac{3}{8} + \frac{1}{25} + \frac{1}{15} = \frac{163}{200} < 1 \quad \Rightarrow \text{schedulabile!} \end{aligned}$$

Verifica della schedulabilità di T_2':

$$\begin{aligned}
b_2'(ss) &= x_2' + \min(x_1', e_1') = x_1 + \min(x_3, e_3) \\
&= 0.1 + \min(0, 1) = 0.1 \\
b_2'(np) &= \theta_3' = \theta_2 = 0.1 \\
b_2' &= b_2'(ss) + (1 + K_2') \cdot b_2'(np) \\
&= 0.1 + (1 + K_1) \cdot 0.1 = 0.1 + 2 \cdot 0.1 = 0.3 \\
\sum_{k=1}^{n} \frac{e_k'}{\min(p_k', D_k')} + \frac{b_2'}{\min(p_2', D_2')} &= \frac{1}{\min(7,3)} + \frac{1.5}{\min(5,4)} + \frac{0.2}{\min(6,5)} + \frac{0.3}{\min(5,4)} \\
&= \frac{1}{3} + \frac{3}{8} + \frac{1}{25} + \frac{3}{40} = \frac{247}{300} < 1 \quad \Rightarrow \text{schedulabile!}
\end{aligned}$$

Verifica della schedulabilità di T_3':

$$\begin{aligned}
b_3'(ss) &= x_3' + \min(x_1', e_1') + \min(x_2', e_2') \\
&= x_2 + \min(x_3, e_3) + \min(x_1, e_1) \\
&= 1.25 + \min(0, 1) + \min(0.1, 1.5) = 1.35 \\
b_3'(np) &= 0 \\
b_3' &= b_3'(ss) + (1 + K_3') \cdot b_3'(np) = 1.35 \\
\sum_{k=1}^{n} \frac{e_k'}{\min(p_k', D_k')} + \frac{b_3'}{\min(p_3', D_3')} &= \frac{1}{\min(7,3)} + \frac{1.5}{\min(5,4)} + \frac{0.2}{\min(6,5)} + \frac{1.35}{\min(6,5)} \\
&= \frac{1}{3} + \frac{3}{8} + \frac{1}{25} + \frac{27}{100} = \frac{611}{600} > 1 \quad \Rightarrow \textbf{no}!
\end{aligned}$$

Se ne conclude che non è possibile determinare analiticamente se il sistema di task è schedulabile con EDF. Infatti la presenza di scadenze inferiori al periodo e di tempi di blocco non nulli implica che l'analisi svolta è una condizione solo sufficiente per la schedulabilità (non necessaria).

Esercizio 3. *Si consideri il seguente sistema di task:*

$$T_1 = (10, 3),\ T_2 = (8, 3),\ T_3 = (6, 2),\ T_4 = (5, 2),\ T_5 = (2, 1),\ T_6 = (40, 15).$$

(a) *Determinare analiticamente se l'insieme di task è schedulabile su 3 (**tre**) processori identici utilizzando l'algoritmo partizionato EDF-FF.*

Il fattore di utilizzazione dell'algoritmo partizionato EDF-FF dipende dal numero di processori m e da $u_{\max} = \max_k \{ e_k/p_k \}$:

$$\begin{aligned}
m &= 3 \\
u_{\max} &= \max_k \frac{e_k}{p_k} = \max\left\{\frac{3}{10}, \frac{3}{8}, \frac{2}{6}, \frac{2}{5}, \frac{1}{2}, \frac{15}{40}\right\} = \frac{1}{2} \\
\beta &= \left\lfloor \frac{1}{u_{\max}} \right\rfloor = 2 \\
U_{\text{EDF-FF}} &= \frac{\beta \cdot m + 1}{m + 1} = \frac{7}{3} \\
U_T &= \sum_{k=1}^{6} \frac{e_k}{p_k} = \frac{3}{10} + \frac{3}{8} + \frac{2}{6} + \frac{2}{5} + \frac{1}{2} + \frac{15}{40} = \frac{137}{60} < \frac{140}{60} = U_{\text{EDF-FF}}
\end{aligned}$$

Se ne conclude che il sistema di task è schedulabile con EDF-FF.

(b) *Considerando l'ordine dei task indicato nel testo dell'esercizio, determinare il partizionamento dei task sui processori indotto da EDF-FF.*

Indichiamo i tre processori come P_1, P_2 e P_3.

- T_1: allocato su P_1
- T_2: allocato su P_1 poiché $\frac{3}{10} + \frac{3}{8} = \frac{27}{40} < 1$
- T_3: allocato su P_2 poiché $\frac{27}{40} + \frac{2}{6} = \frac{121}{120} > 1$
- T_4: allocato su P_2 poiché $\frac{27}{40} + \frac{2}{5} = \frac{43}{40} > 1$ mentre $\frac{2}{6} + \frac{2}{5} = \frac{22}{30} < 1$
- T_5: allocato su P_3 poiché $\frac{27}{40} + \frac{1}{2} = \frac{47}{40} > 1$ e $\frac{22}{30} + \frac{1}{2} = \frac{37}{30} > 1$

- T_6: allocato su P_3 poiché $\dfrac{27}{40} + \dfrac{15}{40} = \dfrac{42}{40} > 1$ e $\dfrac{22}{30} + \dfrac{15}{40} = \dfrac{133}{120} > 1$

  (dall'analisi del punto (a) è certo che il carico di P_3 è ≤ 1, infatti $\dfrac{1}{2} + \dfrac{15}{40} = \dfrac{35}{40} < 1$)

Se ne conclude che EDF-FF determina il seguente partizionamento:

$$\begin{array}{rcl} P_1 & : & \{T_1, T_2\} \\ P_2 & : & \{T_3, T_4\} \\ P_3 & : & \{T_5, T_6\} \end{array}$$