Sistemi Embedded e Real-time (M. Cesati)

Compito scritto del 19 luglio 2012

Esercizio 1. Si consideri il seguente sistema di task periodici in fase con job non interrompibili: $T_1 = (2, 0.3, 9)$, $T_2 = (4, 1, 9)$, $T_3 = (5, 3, 5)$.

(a) È possibile determinare una schedulazione ciclica struttura per il sistema di task? Giustificare la risposta.

(b) Ripetere l'esercizio assumendo che il job del task T_3 possa essere interrotto suddividendo il lavoro in due job di uguale durata.

Esercizio 2. Un server sporadico semplice con periodo $p_s = 5$ e budget $e_s = 2$ è schedulato insieme a tre task periodici con scadenze uguali al periodo: $T_1 = (5, 1)$, $T_2 = (10, 3)$, $T_3 = (20, 1)$.

(a) Determinare se il sistema è schedulabile con l'algoritmo RM assumendo come trascurabile l'overhead dello scheduler.

(b) Ripetere l'esercizio sostituendo il server sporadico semplice con un server procastinabile avente identico periodo e budget.

Esercizio 3. Si consideri un sistema di task periodici e indipendenti schedulati su un processore:

$$\begin{aligned}
T_1 &= (5, 1.5, 4), & x_1 &= 0.1, & K_1 &= 1, & \theta_1 &= 0.2 \\
T_2 &= (6, 0.2, 5), & x_2 &= 1.15, & K_2 &= 2, & \theta_2 &= 0.1 \\
T_3 &= (7, 1, 3), & x_3 &= 0, & K_3 &= 0, & \theta_3 &= 0.5
\end{aligned}$$

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.

Sistemi Embedded e Real-time (M. Cesati)

Soluzioni del compito scritto del 19 luglio 2012

Esercizio 1. *Si consideri il seguente sistema di task periodici in fase con job non interrompibili:* $T_1 = (2, 0.3, 9)$, $T_2 = (4, 1, 9)$, $T_3 = (5, 3, 5)$.

(a) *È possibile determinare una schedulazione ciclica struttura per il sistema di task? Giustificare la risposta.*

Determiniamo la dimensione del frame più opportuna:

- La fase identica per tutti i task non impone alcun vincolo alla dimensione del frame.
- Vincolo sui tempi d'esecuzione dei job:

$$f \geq \max\{0.3, 1, 3\} = 3$$

- Vincolo sulla divisibilità della lunghezza dell'iperperiodo ($\text{mcm}\{2, 4, 5\} = 20$):

$$f \in \{1, 2, 4, 5, 10, 20\} \quad \Rightarrow f \in \{4, 5, 10, 20\}$$

- Vincolo sul task T_1 ($2f - \gcd\{2, f\} \leq 9$):

$$\begin{array}{l} 2 \cdot 4 - \gcd\{2, 4\} = 6 \leq 9 \\ 2 \cdot 5 - \gcd\{2, 5\} = 9 \leq 9 \\ 2 \cdot 10 - \gcd\{2, 10\} = 18 > 9 \Rightarrow f \neq 10 \\ 2 \cdot 20 - \gcd\{2, 20\} = 38 > 9 \Rightarrow f \neq 20 \end{array}$$

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

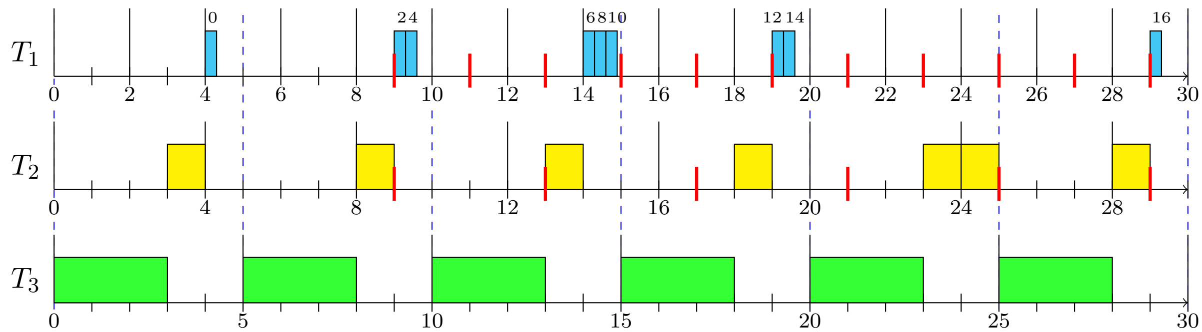
$$\begin{array}{l} 2 \cdot 4 - \gcd\{4, 4\} = 4 \leq 9 \\ 2 \cdot 5 - \gcd\{4, 5\} = 9 \leq 9 \end{array}$$

- Vincolo sul task T_3 ($2f - \gcd\{5, f\} \leq 5$):

$$\begin{array}{l} 2 \cdot 4 - \gcd\{5, 4\} = 7 > 5 \Rightarrow f \neq 4 \\ 2 \cdot 5 - \gcd\{5, 5\} = 5 \leq 5 \end{array}$$

L'unica dimensione intera ammissibile per il frame è dunque $f = 5$.

Per determinare l'esistenza di una schedulazione ciclica strutturata con $f = 5$ osserviamo preliminarmente che in ciascun frame deve trovare posto obbligatoriamente un job di T_3 ed almeno un job di T_2. Inoltre nell'intervallo tra 0 e 30 devono essere eseguiti almeno 7 job di T_2. Si ottiene pertanto:



(Per maggior chiarezza è stato indicato sopra a ciascun job di T_1 il corrispondente istante di rilascio.) Il job di T_1 rilasciato all'istante 16 manca la scadenza.

Per dimostrare formalmente che non esiste alcuna schedulazione ciclica strutturata osserviamo che il fattore di utilizzazione del sistema di task è

$$U_T = \frac{0.3}{2} + \frac{1}{4} + \frac{3}{5} = 1.$$

Pertanto, qualunque sia l'eventuale schedulazione ciclica, a regime il processore non potrà essere mai *idle*. D'altra parte nei frame con job di T_1 possono trovare posto al massimo 1 job di T_3, un job di T_2 e tre job di T_1, con un tempo inutilizzato pari a 0.1. Quindi non può esistere una schedulazione ciclica strutturata valida.

(b) *Ripetere l'esercizio assumendo che il job del task T_3 possa essere interrotto suddividendo il lavoro in due job di uguale durata.*

L'analisi svolta al punto (a) resta valida, tranne per il vincolo sui tempi di esecuzione dei job. Poiché ora il job di T_3 può essere spezzato in due job di durata 1.5 ciascuno, è possibile prendere in esame anche la dimensione del frame $f = 2$. Questa dimensione è in effetti ammissibile, in quanto:

- Vincolo sul task T_1 ($2f - \gcd\{2, f\} \leq 9$):

$$2 \cdot 2 - \gcd\{2, 2\} = 2 \leq 9$$

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

$$2 \cdot 2 - \gcd\{4, 2\} = 2 \leq 9$$

- Vincolo sul task T_3 ($2f - \gcd\{5, f\} \le 5$):

$$2 \cdot 2 - \gcd\{5, 2\} = 3 \le 5$$

D'altra parte si può applicare lo stesso ragionamento svolto nel punto (a) per concludere che non può esistere alcuna schedulazione ciclica strutturata per $f = 2$. Infatti, nei frame in cui è eseguito almeno un job di T_1 una porzione del tempo di processore è certamente "sprecata": nessuna delle quantità $f = 2$, $f - e_3/2 = 0.5$, $f - e_2 = 1$ è un multiplo intero di $e_1 = 0.3$.

Esercizio 2. *Un server sporadico semplice con periodo $p_s = 5$ e budget $e_s = 2$ è schedulato insieme a tre task periodici con scadenze uguali al periodo: $T_1 = (5, 1)$, $T_2 = (10, 3)$, $T_3 = (20, 1)$.*

(a) *Determinare se il sistema è schedulabile con l'algoritmo RM assumendo come trascurabile l'overhead dello scheduler.*

Nell'analisi di schedulabilità il server sporadico semplice è equivalente ad un task periodico con periodo p_s e tempo di esecuzione e_s. Consideriamo pertanto il fattore di utilizzazione totale del sistema:

$$U_T = \frac{e_s}{p_s} + \sum_{i=1}^{3} \frac{e_i}{p_i} = \frac{2}{5} + \frac{1}{5} + \frac{3}{10} + \frac{1}{20} = \frac{19}{20}.$$

Osserviamo inoltre che i task sono armonici, quindi l'algoritmo RM è ottimale e la condizione di schedulabilità è $U_T \le 1$. Se ne conclude perciò che il sistema di task è schedulabile.

(b) *Ripetere l'esercizio sostituendo il server sporadico semplice con un server procastinabile avente identico periodo e budget.*

Il server procastinabile si comporta come un task periodico che può infliggere un tempo di blocco pari a e_s/p_i ad ogni task di priorità inferiore e periodo p_i. Valutiamo la schedulabilità del sistema task per task:

- Il server procastinabile è schedulabile in quanto $e_s/p_s = 2/5 \le 1$.
- Il task T_1 è schedulabile in quanto $e_s/p_s + e_1/p_1 + e_s/p_1 = 2/5 + 1/5 + 2/5 = 1$.
- Il task T_2 non è schedulabile in quanto $e_s/p_s + e_1/p_1 + e_2/p_2 + e_s/p_2 = 2/5 + 1/5 + 3/10 + 2/10 = 11/10 > 1$.
- Il task T_3 non è schedulabile in quanto $U_T + e_s/p_3 = 19/20 + 2/20 = 21/20 > 1$.

Concludiamo che il sistema di task con il server procastinabile non è schedulabile.

Esercizio 3. *Si consideri un sistema di task periodici e indipendenti schedulati su un processore:*

$$\begin{array}{llll} T_1 = (5, 1.5, 4), & x_1 = 0.1, & K_1 = 1, & \theta_1 = 0.2 \\ T_2 = (6, 0.2, 5), & x_2 = 1.15, & K_2 = 2, & \theta_2 = 0.1 \\ T_3 = (7, 1, 3), & x_3 = 0, & K_3 = 0, & \theta_3 = 0.5 \end{array}$$

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 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':

$$\begin{aligned}
b_2'(ss) &= x_2' + \min(x_1', e_1') = x_1 + \min(x_3, e_3) \\
&= 0.1 + \min(0, 1) = 0.1 \\
b_2'(np) &= \theta_3' = \theta_2 = 0.1 \\
b_2' &= b_2'(ss) + (1 + K_2') \cdot b_2'(np) \\
&= 0.1 + (1 + K_1) \cdot 0.1 = 0.1 + 2 \cdot 0.1 = 0.3 \\
\sum_{k=1}^{n} \frac{e_k'}{\min(p_k', D_k')} + \frac{b_2'}{\min(p_1', D_1')} &= \frac{1}{\min(7,3)} + \frac{1.5}{\min(5,4)} + \frac{0.2}{\min(6,5)} + \frac{0.3}{\min(5,4)} \\
&= \frac{1}{3} + \frac{3}{8} + \frac{1}{25} + \frac{3}{40} = \frac{247}{300} < 1 \quad \Rightarrow \text{schedulabile!}
\end{aligned}$$

Verifica della schedulabilità di T_3':

$$\begin{aligned}
b_3'(ss) &= x_3' + \min(x_1', e_1') + \min(x_2', e_2') = \\
&= x_2 + \min(x_3, e_3) + \min(x_1, e_1) \\
&= 1.15 + \min(0, 1) + \min(0.1, 1.5) = 1.25 \\
b_3'(np) &= 0 \\
b_3' &= b_3'(ss) + (1 + K_3') \cdot b_3'(np) = 1.25 \\
\sum_{k=1}^{n} \frac{e_k'}{\min(p_k', D_k')} + \frac{b_3'}{\min(p_1', D_1')} &= \frac{1}{\min(7,3)} + \frac{1.5}{\min(5,4)} + \frac{0.2}{\min(6,5)} + \frac{1.25}{\min(6,5)} \\
&= \frac{1}{3} + \frac{3}{8} + \frac{1}{25} + \frac{1}{4} = \frac{599}{600} < 1 \quad \Rightarrow \text{schedulabile!}
\end{aligned}$$

Se ne conclude che il sistema è schedulabile con EDF.