Ordonnancement dans les STR

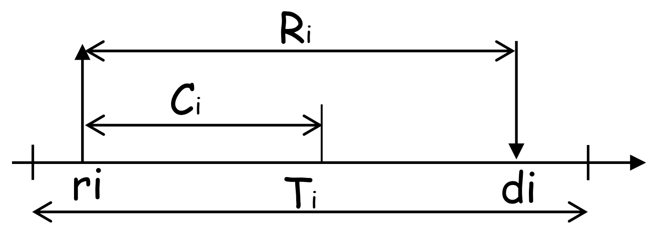
Cours N°2

Boris Vidolov

UMR CNRS 7253 HeuDiaSyC
Département Génie Informatique

bvidolov@utc.fr

Caractéristiques d'un processus τ_i périodique



$\tau_i : (r_i, C_i, R_i, T_i)$

Ti : période de τ_i

ri : première date de disponibilité de τ_i.

date de début de la k[ème] période : $r_i^{(k)} = r_i + (k-1)T_i ;\ \forall k \geq 0$

Ci : durée d'exécution sans préemption de τ_i sur un processeur

Ri : délai critique de τ_i

$0 < Ci < Ri < Ti$

Pi : priorité du processus

ui : facteur d'utilisation de τ_i : $u_i = Ci/Ti < 1$

di : prochaine date d'échéance du processus : $d_i = r_i^{(k)} + R_i$

Ri(t) : temps de réponse dynamique : Ri (t) = di - t

Ci(t) : durée d'exécution dynamique : Ci (t) = Ci - t

Li : laxité maximale : Li = di -Ci

Li(t) : laxité dynamique : Li (t) = di -Ci (t)

Rate Monotonic

- algorithme préemptif à priorité statique
- le processus le plus prioritaire est celui disponible avec la plus petite période
- les instants de prise en compte des priorités sont les dates de disponibilités et les dates de terminaisons des processus.

Théorème : Pour n processus périodiques à échéance sur requête (Ti = Ri), une condition suffisante d'acceptation d'une configuration par l'algorithme Rate Monotonic est :

$$U \leq n\left(2^{1/n} - 1\right)$$

$U = \sum_{i=1}^{n} u_i = \sum_{i=1}^{n} C_i / T_i$ facteur d'utilisation d'une configuration.

<u>Exemple</u> : Trois processus à échéance sur requête
(ri, Ci, Ti) : A(0, 3, 20), B(0, 2, 5), C(0, 2, 10).

$U \leq n\left(2^{1/n} - 1\right)$ => $0.75 < 0.78$

Deadline Monotonic

- même hypothèse que pour le RM
- $d_i < T_i$
- le processus le plus prioritaire est celui disponible avec le plus petit délai critique
- condition suffisante d'acceptation d'une configuration :

$$\boxed{\sum_{i=1}^{n} C_i / d_i \leq n\left(2^{1/n} - 1\right)}$$

<u>Exemple</u> : (ri, Ci, Ri, Ti) : A(0, 3, 7, 20), B(0, 2, 4, 5) C(0, 2, 9, 10).

Théorème de la zone critique

- condition de Rate Monotonic très restrictive

- condition nécessaire et suffisante

  Si toutes les tâches arrivent initialement dans le système simultanément et si elles respectent leur premières échéances, alors toutes les échéances seront respectées par la suite.

- Test de terminaison (cas échéance sur requête):

$$\forall i, 1 \le i \le n \quad \min_{0 \le t \le d_i} \sum_{j \in hp(i)} \frac{C_j}{t} \left\lceil \frac{t}{T_j} \right\rceil \le 1$$

$$\lceil x \rceil - \text{opérateur plafond}$$

Si Di < Ti alors $t = \sum C_j \left\lceil \frac{t}{T_j} \right\rceil + E_i$ avec $E_i = T_i - R_i$

Méthode itérative

$C_j \left\lceil \frac{t}{T_j} \right\rceil$ - utilisation processeur par la tâche j dans [0,t]

$W_i(t) = C_i + \sum_{j \in hp(i)} \left\lceil \frac{t}{T_j} \right\rceil C_j$ - temps processeur de toutes les tâches dans [0,t]

$W_i(t) = t$ avec t ≤ di

$$\text{Init } t_0 = \sum_{j=1}^{i} C_j$$

$$W_i^{n+1} = C_i + \sum_{j \in hp(i)} \left\lceil \frac{t}{T_j} \right\rceil C_j$$

<u>Exemple</u> : (ri, Ci, Ti) : A(0, 40, 80), B(0, 10, 40) C(0, 5, 20); CS non satisfaite

Earliest Deadline First - EDF

- algorithme préemptif à priorités dynamiques
- le processus le plus prioritaire est celui disposant du plus petit temps de réponse dynamique Ri(t)
- il permet de garantir un plus grand nombre de configuration que le RM

<u>Théorème</u> : Pour des processus périodiques à échéance sur requête, une condition suffisante d'ordonnancement par l'algorithme Ealiest Deadline est U < 1.

<u>Exemple</u> : (ri, Ci, Ri, Ti) : A(0, 3, 7, 20), B(0, 2, 4, 5), C(0, 1, 8, 10)

Laxité Minimale - Least Laxity First

- algorithme préemptif à priorités dynamiques
- le processus le plus prioritaire est celui qui a la plus petite laxité dynamique $L_i(t)$.
- laxité – la date au plus tard au-delà de laquelle un processus ne peut pas reprendre son exécution sans commettre une violation de contraintes temporaires

Théorème : Pour des processus périodiques à échéance sur requête, une condition suffisante d'ordonnancement par l'algorithme LLF est $U < 1$.

Exemple : (ri, Ci, Ri, Ti) : A(0, 3, 7, 20), B(0, 2, 4, 5), C(0, 1, 8, 10)

Cheddar

Outil d'analyse basé sur la théorie de l'ordonnancement

http://beru.univ-brest.fr/cheddar/