

Méthode d'évaluation du pire temps de réponse de tâches à offset

RAHNI Ahmed, RICHARD Michael, GROLLEAU Emmanuel.
Laboratoire d'Informatique Scientifique et Industrielle - LISI/ENSMA
1, av Clément Ader - Téléport 1 - BP 40109
86961 Futuroscope Chasseneuil Cedex
{rahni,richardm,grolleau}@ensma.fr

Résumé

Dans cet article nous présentons une nouvelle méthode [8] d'évaluation de pire temps de réponse dans le contexte des tâches à offset (transactions). Cette nouvelle méthode améliore la borne supérieure du temps de réponse fournie par l'approche approximative présentée dans [10, 6]. L'idée mise en œuvre dans cette approche est de combiner le principe du calcul exact et le principe d'approximation afin de diminuer le pessimisme de l'analyse pire cas, permettant ainsi d'améliorer la borne supérieure du temps de réponse fournie tout en conservant une complexité pseudo-polynomiale.

Mots Clés : Temps de Réponse, Transactions, Analyse pire cas.

1. Introduction

L'analyse des temps de réponse [1] est l'une des approches de validation temporelle utilisées pour vérifier l'ordonnabilité d'un système de tâches temps réels. Le modèle de tâches à offset (transactions) [10] proposé par Tindell est une extension du modèle de Liu et Layland [4]. La principale difficulté de cette approche est de déterminer le pire scénario pour une tâche donnée puisque les tâches à offset (transaction) ne sont pas concrètes (i.e. les dates d'activation ne sont pas fixées a priori). Tindell a montré qu'un instant critique pour une tâche analysée, notée (τ_{ua}), se produit quand une tâche plus prioritaire que τ_{ua} dans chaque transaction est activée en même temps que τ_{ua} .

Une méthode exacte de calcul du pire temps de réponse (traitant tous les instants critiques possibles) à été proposée dans [9]; mais cette méthode est inapplicable sur des systèmes de tâches réels car d'une complexité exponentielle. Tindell [10] a proposé une méthode approximative de complexité pseudo-polynomiale fournissant une borne supérieure pour le pire temps de réponse. Cette méthode à été formalisée par Palencia [3] et améliorée par Turja et Nolin dans [5, 6, 7].

Dans la suite, nous rappelons le modèle de tâches à offset (systèmes de transactions), puis présentons brièvement

la méthode approximative proposée dans [10, 6]. Nous développons ensuite les principes de notre nouvelle méthode et les résultats de comparaison avec les autres approches.

2. Système de transactions

Dans le contexte des tâches à offset, un système Γ de tâches est composé d'un ensemble de $|\Gamma|$ transactions Γ_i , avec $1 \leq i \leq |\Gamma|$ (tel que $|\Gamma_i|$ est le nombre d'éléments dans l'ensemble Γ_i). Formellement un système de transactions est défini comme suit :

$$\begin{aligned}\Gamma & : \{ \Gamma_1, \Gamma_2, \dots, \Gamma_{|\Gamma|} \} \\ \Gamma_i & : \{ \tau_{i1}, \tau_{i2}, \dots, \tau_{i|\Gamma_i|}, T_i \} \\ \tau_{ij} & : \langle C_{ij}, O_{ij}, D_{ij}, J_{ij}, B_{ij}, P_{ij} \rangle\end{aligned}$$

Chaque transaction Γ_i (cf. 1) est constituée d'un ensemble de $|\Gamma_i|$ tâches τ_{ij} possédants la même période T_i , avec $0 < j \leq |\Gamma_i|$.

Une tâche τ_{ij} est définie par son pire temps d'exécution (WCET) C_{ij} , un intervalle de temps séparant la date d'activation de la transaction et la date d'activation de la tâche, appelé offset O_{ij} , un délai critique D_{ij} , un gigue sur activation J_{ij} , un facteur de blocage maximum B_{ij} , et une priorité P_{ij} (on suppose une politique d'ordonnement à priorités fixes). B_{ij} modélise le pire temps de blocage induit par un éventuel partage de ressources et J_{ij} permet de prendre ne compte un retard sur l'activation dans le cas par exemple d'application distribuée. La figure 1 présente un exemple de transaction Γ_i contenant trois tâches avec une même période $T_i = 16$. Afin de simplifier et clarifier les formules de calcul pour les méthodes d'analyse, nous considérons dans la suite que la tâche analysée τ_{ua} est la seule tâche de la transaction Γ_u et qu'une seule instance de cette tâche est activée dans n'importe quel intervalle de temps t . Soit $hp_i(\tau_{ua})$ l'ensemble d'indices des tâches de Γ_i ayant une priorité supérieure à celle de τ_{ua} .

3. Méthode approximative de Tindell-Nolin

Cette méthode [10] fournit une borne supérieure du pire temps de réponse. Au lieu de traiter tous les instants

critiques possibles, elle traite une seule fonction d'approximation avec une complexité pseudo-polynomiale. Cette méthode consiste à utiliser une borne supérieure de l'interférence $W_i(t)$ d'une transaction Γ_i dans le processus de calcul :

$$R_{ua}^{(n+1)} = C_{ua} + \sum_{\Gamma_i \in \Gamma} (W_i(R_{ua}^n)) \quad (1)$$

$$W_i(t) = \max_{c \in hp_i(\tau_{ua})} \{W_{ic}(t)\}$$

avec $W_{ic}(t)$, l'interférence de l'ensemble des tâches de la transaction Γ_i qui sont plus prioritaires que τ_{ua} à l'instant critique initié par la tâche τ_{ic} .

Cette méthode à été améliorée dans [6] par le calcul de l'interférence effective d'une transaction dans une période d'activité de longueur t en supprimant la surestimation causée par les unités de temps d'exécution des tâches après la date t .

$$W_{ijc}(t) = \left\lfloor \frac{J_{ij} + \Phi_{ijc}}{T_i} \right\rfloor C_{ij} + \left\lceil \frac{t^*}{T_i} \right\rceil C_{ij} - x$$

$$t^* = t - \Phi_{ijc}$$

$$x = \begin{cases} C_{ij} - (t^* \bmod T_i) & \text{if } t^* > 0 \wedge (0 < t^* \bmod T_i) < C_{ij} \\ 0 & \text{otherwise} \end{cases}$$

3.1. Exemple

Soit le système de transactions illustré par la figure 2. la tâche analysée possède une charge de $C_{ua} = 1$, son pire temps de réponse exacte $R_{ua} = 8$. Pour l'obtenir, $|\Gamma_1| * |\Gamma_2| = 3 * 2 = 6$ instants critiques possibles ont été considérés.

Nous appliquons la méthode approximative de Tindell-Nolin pour calculer le pire temps de réponse de la tâche τ_{ua} . le pire temps de réponse obtenu est supérieur au temps de réponse exacte.

$$\begin{aligned} R_{ua}^{(0)} &= C_{ua} = 1 \\ R_{ua}^{(1)} &= C_{ua} + W_1(1) + W_2(1) = 1 + 4 + 2 = 7 \\ R_{ua}^{(2)} &= C_{ua} + W_1(7) + W_2(7) = 1 + 4 + 3 = 8 \\ R_{ua}^{(3)} &= C_{ua} + W_1(8) + W_2(8) = 1 + 5 + 3 = 9 \\ R_{ua}^{(4)} &= C_{ua} + W_1(9) + W_2(9) = 1 + 5 + 4 = 10 \\ R_{ua}^{(5)} &= C_{ua} + W_1(10) + W_2(10) = 1 + 5 + 4 = 10 \end{aligned}$$

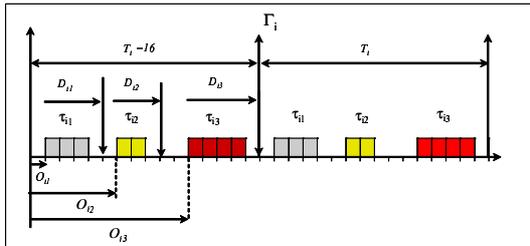


FIG. 1 – Exemple d'une transaction.

3.2. Discussion

Dans l'approche précédente, le pessimisme est produit par la fonction d'approximation $W_i(t)$ qui prend le maximum des interférences de la transaction Γ_i . L'effet négatif de cette fonction est le changement de la tâche initiant l'instant critique d'une itération à une autre. Pour une tâche analysée donnée, le pessimisme peut se produire par l'application de la fonction W_i sur une ou plusieurs transactions.

Ainsi, nous remarquons dans l'exemple précédent que le temps de réponse exacte $t = 8$ (fin de l'exécution) de la tâche sous analyse correspond à la date où l'interférence de la transaction Γ_1 , quand l'instant critique est initié par la tâche τ_{12} , rattrape et dépasse son interférence, quand la tâche τ_{11} initie l'instant critique ; à l'instant $t = 8$ il y a un changement de la tâche initiant l'instant critique.

Afin d'éviter ce phénomène, produit par la fonction d'approximation $W_i(t)$ pour l'interférence d'une transaction Γ_i , nous traitons tous les instants critiques correspondant aux tâches candidates de Γ_i (la fonction d'approximation d'interférence est appliquée pour toutes les transactions, exceptées Γ_i). En se basant sur cette idée, nous proposons dans la section suivante une nouvelle approche moins pessimiste que cette dernière.

4. Nouvelle méthode approximative [8]

Pour une transaction Γ_i (un scénario) nous avons $|hp_i(\tau_{ua})|$ instants critiques (tâches candidates). Nous calculons le temps de réponse $R_{ua,ic}$ de τ_{ua} pour chaque tâche candidate τ_{ic} , puis nous choisissons $R_{ua,i}$ défini comme le maximum des temps de réponses obtenus.

$$R_{ua,ic}^{(n+1)} = C_{ua} + W_{ic}(R_{ua}^n) + \sum_{\Gamma_k \in \Gamma, \Gamma_k \neq \Gamma_i} (W_k(R_{ua}^n))$$

$$R_{ua,i} = \max_{c \in hp_i(\tau_{ua})} \{R_{ua,ic}\}$$

$$W_k(t) = \max_{c \in hp_k(\tau_{ua})} \{W_{kc}(t)\}$$

Le temps de réponse $R_{ua,i}$ obtenu dans chaque scénario est toujours compris entre le temps de réponse exacte et la borne de temps réponse calculée par la

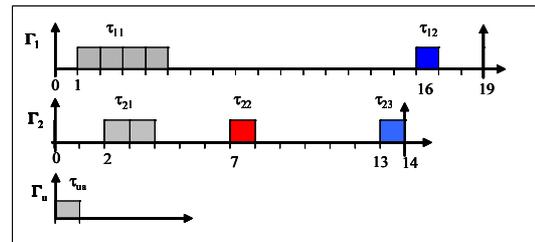


FIG. 2 – Exemple d'un Système.

méthode approximative de Tindell-Nolin (preuve dans [8]).

Pour un système, les transactions pour lesquelles l'application de la fonction approximative de Tindell produit un pessimisme sur l'évaluation du pire temps de réponse de la tâche sous analyse ne sont pas connues. Pour cela nous devons appliquer notre nouvelle méthode sur toutes les transactions du système une à une, d'où un nombre de scénarii à traiter égal au nombre de transactions ($|\Gamma|$).

Il est démontré [8] que le temps de réponse obtenu pour chaque scénario (transaction) est une borne supérieure de pire temps de réponse. Ainsi la meilleure borne de pire cas correspond à la plus petite borne parmi tous les scénarios.

$$R_{ua} = \min_{i \in 1..|\Gamma|} \{R_{ua,i}\}$$

Pour calculer une borne supérieure de temps réponse par notre méthode approximative, le nombre totale de cas possibles à traiter est égal au nombre de tâches du système $|\Gamma_1| + |\Gamma_2| + \dots + |\Gamma_{|\Gamma|}|$

4.1. Exemple

Afin de montrer les différentes étapes de calcul, nous appliquons notre méthode sur l'exemple de la figure 2.

Premier scénario correspond à la transaction Γ_1 : Nous calculons le temps de réponse pour les deux tâches candidates possibles :

pour la tâche τ_{11} :

$$\begin{aligned} R_{ua,11}^{(0)} &= C_{ua} = 1 \\ R_{ua,11}^{(1)} &= C_{ua} + W_{11}(1) + W_2(1) = 1 + 4 + 2 = 7 \\ R_{ua,11}^{(2)} &= C_{ua} + W_{11}(7) + W_2(7) = 1 + 4 + 3 = 8 \\ R_{ua,11}^{(3)} &= C_{ua} + W_{11}(8) + W_2(8) = 1 + 4 + 3 = 8 \end{aligned}$$

pour la tâche τ_{12} :

$$\begin{aligned} R_{ua,12}^{(0)} &= C_{ua} = 1 \\ R_{ua,12}^{(1)} &= C_{ua} + W_{12}(1) + W_2(1) = 1 + 1 + 2 = 4 \\ R_{ua,12}^{(2)} &= C_{ua} + W_{12}(4) + W_2(4) = 1 + 1 + 2 = 4 \end{aligned}$$

La borne du pire temps de réponse de τ_{ua} dans ce scénario est : $R_{ua,1} = \max \{R_{ua,11}, R_{ua,12}\} = \max \{8, 4\} = 8$

Deuxième scénario correspond à la transaction Γ_2 : même principe de calcul pour chaque tâche candidate :

pour la tâche τ_{21} :

$$\begin{aligned} R_{ua,21}^{(0)} &= C_{ua} = 1 \\ R_{ua,21}^{(1)} &= C_{ua} + W_{21}(1) + W_1(1) = 1 + 2 + 4 = 7 \\ R_{ua,21}^{(2)} &= C_{ua} + W_{21}(7) + W_1(7) = 1 + 3 + 4 = 8 \\ R_{ua,21}^{(3)} &= C_{ua} + W_{21}(8) + W_1(8) = 1 + 3 + 5 = 9 \\ R_{ua,21}^{(3)} &= C_{ua} + W_{21}(9) + W_1(9) = 1 + 3 + 5 = 9 \end{aligned}$$

pour la tâche τ_{22} : $R_{ua,22} = 6$

pour la tâche τ_{23} : $R_{ua,23} = 10$

La borne du pire temps de réponse de τ_{ua} dans ce scénario est : $R_{ua,2} = \max \{R_{ua,21}, R_{ua,22}, R_{ua,23}\} = \max \{9, 6, 10\} = 10$

$R_{ua} = \min \{R_{ua,1}, R_{ua,2}\} = \min \{8, 10\} = 8$ égal au temps de réponse exacte. Dans ce système, le pessimisme est induit par la fonction d'approximation de l'interférence sur la première transaction.

5. Évaluation

Afin d'évaluer et quantifier l'amélioration sur la borne de temps de réponse apportée par notre méthode par rapport aux autres méthodes, nous avons implémenté ces trois méthodes (Exacte, Turja-Nolin, notre méthode). Le calcul effectué correspond au calcul du temps de réponse d'une seule tâche qui est moins prioritaire que toutes les tâches des systèmes générés aléatoirement par un générateur de transactions.

Le générateur prend en entrée plusieurs paramètres réglables : charge totale du système, le nombre de transaction par système et le nombre de tâches par transaction. En fonction de ces paramètres les autres propriétés des tâches sont générées :

- La charge totale du système est distribuée sur les transactions suivant l'algorithme UUniFast présenté dans [2].
- Les périodes T_i des transactions sont générées aléatoirement entre 1000 et 1000000 (distribution uniforme)
- La charge d'une transaction Γ_i est distribuée sur les tâches suivant l'algorithme UUnifast, les C_{ij} sont calculés en fonction de T_i et de la charge du système.
- L'offset est distribué aléatoirement dans la période (distribution uniforme)
- B_{ij} et J_{ij} sont nuls.

Dans tous les graphes d'évaluation chaque point correspond à la moyenne des résultats pour 1000 configurations. La figure 3 présente l'influence du nombre de tâches par

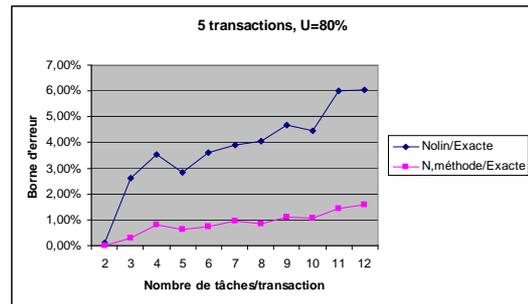


FIG. 3 – influence du nombre de tâches.

transaction sur la borne d'erreur (pessimisme du temps de réponse) des deux méthodes approximatives par rapport à la méthode optimale (exacte) ; nous constatons que la nouvelle méthode fournit toujours une borne au moins aussi bonne que celle fournie par la méthode de Tindell-Nolin, et le pessimisme croît avec le nombre de tâches.

Mais le taux d'accroissement de la nouvelle méthode est plus faible que celui de la méthode de Tindell-Nolin, en effet, le nombre de tâches dans une transaction augmente la probabilité de pessimisme et par conséquent augmente le nombre de transactions induisant ce pessimisme pour la tâche analysée.

L'influence du nombre de transactions sur le taux de pessimisme (borne d'erreur) du temps de réponse pour une tâche analysée par les deux méthodes approximatives est présenté dans figure 4 ; la qualité de la borne fournie par notre méthode n'est pas très sensible au nombre de transactions, ceci pouvant être expliqué par le nombre fixe de tâches par transaction qui stabilise la probabilité qu'une transaction soit une source de surestimation. Par contre le pessimisme décroît pour la méthode de Tindell-Nolin lorsque on augmente le nombre de transactions. Les deux méthodes approximatives ont une com-

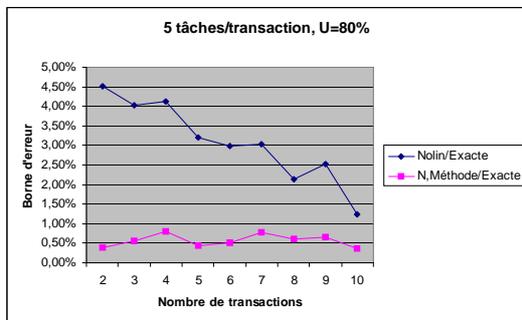


FIG. 4 – influence du nombre de transactions.

plexité pseudo-polynomiale sauf que notre méthode met $|\Gamma_1| + |\Gamma_2| + \dots + |\Gamma_{|\Gamma|}|$ opérations de calcul de temps de réponse tandis que la méthode de Turja-Nolin n'en fait qu'une. Ces différences de temps de traitement en fonction du nombre de transactions sont présentées respectivement sur la figure 5.

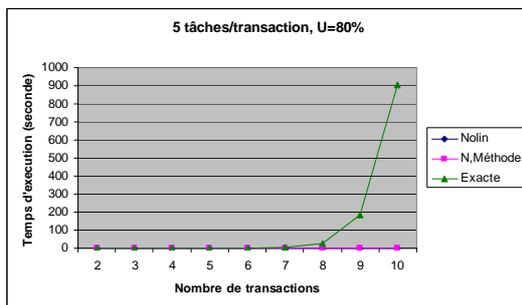


FIG. 5 – Temps d'exécution/Nbr transaction.

6. Conclusion

Nous avons présenté une nouvelle approche approximative pseudo-polynomiale [8] qui fournit une borne du

pire temps de réponse au moins aussi bonne que celle fournie par la méthode présentée dans [10, 6]. Une évaluation de performance en qualité de temps de réponse et en temps de traitement de la cette nouvelle méthode par rapport à l'approche de [10, 6] à été présentée plus haut.

Dans notre travaux futures nous essayerons d'améliorer le temps de réponse fourni et le temps de traitement de la nouvelle méthode en utilisant des propriétés de monotonicité et de dominance [11].

Références

- [1] N. Audsley, A. Burns, R. Davis, K. Tindell, and A. Welings. Fixed priority pre-emptive scheduling : An historical perspective. *Real-Time Systems* 8, pages 129–154, 1995.
- [2] E. Bini and G. Buttazzo. Biasing effects in schedulability measures. *IEEE Proceedings of the 16th Euromicro Conference on Real-Time Systems (ECRTS'04), Catania, Italy*, (16), July 2004.
- [3] J. P. Gutierrez and M. G. Harbour. Schedulability analysis for tasks with static and dynamic offsets. *Proc IEEE Real-time System Symposium (RTSS)*, (19), December 1998.
- [4] C. Liu and J. Layland. Scheduling algorithms for multiprogramming in real-time environment. *Journal of ACM*, 1(20) :46–61, October 1973.
- [5] J. Maki-Turja and M. Nolin. Faster response time analysis of tasks with offsets. *Proc 10th IEEE Real-Time Technology and Applications Symposium (RTAS)*, May 2004.
- [6] J. Maki-Turja and M. Nolin. Tighter response time analysis of tasks with offsets. *Proc 10th International Conference on Real-Time Computing and Applications (RTC-SA'04)*, August 2004.
- [7] J. Maki-Turja and M. Nolin. Fast and tight response-times for tasks with offsets. *17th EUROMICRO Conference on Real-Time Systems IEEE Palma de Mallorca Spain*, July 2005.
- [8] A. Rahni, M. Richard, and E. Grolleau. Nouvelle approche approximative d'analyse de temps de réponse. *Rapport interne 2007-001 LISI/ENSMA*, Université de Poitiers.
- [9] K. Tindell. Using offset information to analyse static priority pre-emptively scheduled task sets. *Technical Report YCD-182, Dept of Computer Science, Unoversity of York, England*, 1992.
- [10] K. Tindell. Adding time-offsets to schedulability analysis. *Technical Report YCS 221, Dept of Computer Science, University of York, England*, January 1994.
- [11] K. Traore, E. Grolleau, and F. Cottet. Characterization and analysis of tasks with offsets : Monotonic transactions. *Proc 12th International Conference on Embedded and Real-Time Computing Systems and Applications. RTCSA'06*, (12), August 16-18th Sydney, Australia 2006.