Ordonnancement non-préemptif pour systèmes temps réel à contraintes de précédences et de latences

Liliana Cucu[1], *Yves Sorel*[1]

(1) INRIA Rocquencourt,
BP 105 - 78153 Le Chesnay Cedex, France
liliana.cucu@inria.fr, yves.sorel@inria.fr

1 Introduction

Les systèmes temps réel sont de plus en plus utilisés dans le monde contemporain. Longtemps réservés aux équipements industriels lourds (centrale nucléaire, chaîne de fabrication, avionique, systèmes d'armes), leurs domaines d'utilisation sont aujourd'hui très variés, on peut les trouver dans des produits grand public (automobile, téléphone, domotique) pour lesquels les temps de développement conditionnant la mise sur le marché doivent être minimisés. Ces systèmes ne sont pas seulement contraints de réagir correctement, mais aussi, d'avoir une réaction bornée en temps. Un délai est donc imposé entre la consommation d'un stimulus par le système et la production de la réaction correspondant à ce stimulus, les deux étant liées par des contraintes de précédences à travers d'opérations intermédiaires. Pour cela on a proposé [1] une nouvelle contrainte appelée « contrainte de latence » mieux adaptée que d'autres contraintes pour décrire ce délai.

Dans la littérature sur l'ordonnancement temps réel, le mot "latence" a déjà été utilisé à plusieurs reprises. Par exemple par Hagmann [2] pour exprimer le temps nécessaire pour obtenir une donnée lors de la lecture depuis un disque dur. Des définitions plus théoriques ont été données par Goddard [3] dans le cas de graphes flot de données ou par Van Beek [4] dans le cas de précédences répétitives. Les définitions théoriques existant se limitent à des cas particuliers, on ne peut pas les utiliser pour exprimer le temps écoulé entre deux opérations quelconques d'un système d'opérations. Evidemment si on veut que la latence exprime ce temps écoulé entre deux opérations, alors elle fait partie de la classe des contraintes relatives (contrainte définie au moins entre deux opérations), et l'intérêt des contraintes relatives a très bien été justifié par Gerber [5]. Notre définition de la latence généralise la notion de contrainte de bout en bout qui est imposée entre une entrée et une sortie du système temps réel.

2 Notre problème d'ordonnancement

Le problème traité dans cet article est le problème d'ordonnancement non-préemptif des systèmes temps réel avec contraintes de précédences et de latences. On définit les contraintes de précédences à l'aide d'un graphe orienté acyclique $\mathcal{G} = (\mathcal{V},\mathcal{E})$ où $\mathcal{V}$ est l'ensemble des opérations et $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ l'ensemble des arcs qui représentent des précédences entre les opérations. Les contraintes de précédences induisent un ordre partiel et un ordre total correspondant à cet ordre partiel s'appelle ordonnancement. La définition suivante est la définition formelle de cette dernière notion et elle est suivie par la définition de la contrainte de latence.

Définition 1 *pour un système d'opérations, un ordonnancement S est un ordre total sur l'ensemble des opérations qui associe à chaque opération une date de début qui exprime l'instant auquel l'opération sera exécutée. Cela signifie qu'un ordonnancement S est égal à l'ensemble $\{s_A \in \mathbb{N}, A \in \mathcal{V}\}$.*

Définition 2 *deux opérations différentes A et B , telles que $\exists P(A,B) \in \mathcal{P}$, ont une contrainte de latence $L_{AB} \in \mathbb{Z}^+$ si les opérations sont ordonnancées tel que $s_B + C_B - s_A \leq L_{AB}$.*

Résoudre le problème d'ordonnancement non-préemptif des systèmes avec contraintes de précédences et de latences revient à trouver un ordonnancement qui satisfait toutes les contraintes de latences et cela d'une manière non-préemptive c'est à dire aucune opération ne peut arrêter l'exécution d'une autre opération pour s'ordonnancer.

L'étude d'ordonnançabilité [6] nous a permis d'identifier les opérations qui interviennent dans une contrainte de latence et l'influence réciproque de ces opérations dans le cas où l'on a plusieurs contraintes de latences. On a formalisé cela à l'aide de trois relations ||, Z et X entre des paires d'opérations sur lesquelles une contrainte de latence a été imposée.

A l'aide de ces trois relations, nous donnons dans cet article un algorithme d'ordonnancement optimal dans le sens où s'il y a un ordonnancement, l'algorithme le trouvera. L'algorithme d'ordonnancement utilise un marquage des opérations du graphe. Ce marquage des sommets "prévoit" quelles sont pour chaque contrainte de latence les opérations importantes appartenant aux autres contraintes de latences. On prouve l'optimalité de l'algorithme d'ordonnancement en montrant qu'il construit des ordonnancements qui satisfont les conditions d'ordonnançabilité et donc si ces ordonnnancements ne satisfont pas les contraintes de latence alors aucun autre ordonnancement ne le fera.

3 Perspectives

Une perspective ouverte par ces résultats concerne les contraintes relatives. La définition de la latence permet d'imposer une contrainte de latence sur deux opérations quelconques. Son caractère général permet d'étendre les résultats obtenus sur l'ordonnançabilité et l'ordonnancement de tels systèmes pour d'autres contraintes relatives.

Références

[1] L. Cucu, R. Kocik, and Y. Sorel. Real-time scheduling for systems with precedence, periodicity and latency constraints. *Real-time and Embedded Systems*, 2002.

[2] R.B. Hagmann. Low latency logging. Technical report, Palo Alto Research Center, 1991.

[3] S. Goddard. *Constraints on data-flow*. PhD thesis, University of North Carolina, 2000.

[4] P. van Beek and K. Wilken. Fast optimal instruction scheduling for single-issue processors with arbitrary latencies. *Springer-Verlag*, 2001.

[5] Gerber R., S. Hong, and Saksena M. Guaranteeing real-time requirements with resource-based calibration of periodic processes. *IEEE Transactions on Software Engineering(21)7*, 1995.

[6] L. Cucu and Y. Sorel. Schedulability condition for real-time non-preemptive systems with precedence and latency contraints. *à apparaître*.