

Evaluation du débit maximum d'un Synchronous DataFlow

Bruno Bodin¹, Benoît Dupont de Dinechin¹, Alix Munier-Kordon²

¹ KALRAY, 445 rue Lavoisier, 38330 Montbonnot France
{bruno.bodin, benoit.dinechin}@kalray.eu

² LIP6-UPMC, 4 place Jussieu, 75005 Paris, France
alix.munier@lip6.fr

Mots-clés : *Synchronous DataFlow, ordonnancement K-périodique, systèmes embarqués*

1 Présentation du problème

Les systèmes embarqués sont des équipements soumis à de fortes exigences : ils doivent être réactifs et performants tout en respectant des contraintes de ressources très strictes (mémoire, énergie, ...). Le développement d'applications destinées à ces systèmes est en conséquence très contraint.

Les outils de conception s'appuient fréquemment sur des modèles à l'expressivité limitée permettant d'analyser efficacement les applications ainsi conçues. Les *Synchronous Dataflow* (SDF en abrégé), introduits par Lee et Messerschmitt [3] permettent dans ce contexte de modéliser simplement les échanges de données entre les différents processus qui composent une application embarquée.

Un SDF est un graphe orienté $\mathcal{G} = (\mathcal{T}, \mathcal{B})$ où les noeuds sont des tâches qui s'exécutent un grand nombre de fois. Toute tâche $t_i \in \mathcal{T}$ possède une durée $\ell(t_i)$ correspondant à celle de ses exécutions et une valeur $Z_{t_i} \in \mathbb{N}^*$. Tout arc $b = (t_i, t_j) \in \mathcal{B}$ entre deux tâches t_i et t_j est associé à un canal de communication. A toute exécution de t_i , Z_{t_i} valeurs sont placées dans b . De même, t_j requiert Z_{t_j} valeurs dans b pour s'exécuter. Le problème traité dans cet exposé est de développer un algorithme efficace pour évaluer une borne inférieure du débit maximum d'un SDF quelconque.

Il n'existe pas aujourd'hui d'algorithme polynomial pour évaluer le débit maximum d'un SDF. Compte tenu de l'importance des applications modélisées par ce formalisme, deux familles de méthodes sont employées pour le calculer. La première consiste à calculer un ordonnancement au plus tôt et d'en évaluer le débit quand le régime permanent est atteint. Le problème de cette approche est que cet algorithme est de complexité exponentielle [4, 6]. La seconde consiste à limiter l'étude aux ordonnancements périodiques : chaque tâche $t \in \mathcal{T}$ est associée à deux valeurs, s_t (la date de première exécution de t) et ω_t (sa période). La date de la n -ième exécution de t est alors $s_t^n = s_t + (n - 1) \times \omega_t$. Le gros défaut de cette approche est que le débit d'un ordonnancement périodique est une borne inférieure du débit optimum qui peut en être très éloignée [4].

Le but de cet exposé est de présenter une méthode alternative moins restrictive sur la structure des ordonnancements recherchés de sorte à avoir une meilleure évaluation du débit tout en limitant l'explosion combinatoire.

2 Résultats

Soit $K = (k_1, \dots, k_{|\mathcal{T}|}) \in (\mathbb{N}^*)^{|\mathcal{T}|}$. Un ordonnancement K -périodique associé est défini par k_i dates d'exécutions $s_{t_i}^1 < s_{t_i}^2 < \dots < s_{t_i}^{k_i}$ et une période ω_{t_i} pour toute tâche $t_i \in \mathcal{T}$. Pour tout entier n sous la forme $n = q \times k_i + r$ avec $r \in \{1, \dots, k_i\}$, la date d'exécution de la n -ième exécution de t_i est alors de la forme $s_{t_i}^n = s_{t_i}^r + q \times \omega_{t_i}$.

L'ordonnancement au plus tôt d'un SDF est K -périodique pour lequel le vecteur K dépend du graphe [5]. Le but ici est de fixer le vecteur K a priori, et de calculer un ordonnancement K -périodique de débit maximum. Une partie des résultats présentés a été publiée dans [2].

Nous montrons que tout ordonnancement K -périodique réalisable possède une période propre liant les périodes de toutes les tâches et définie par $\Omega = \frac{w_{t_i}}{k_i Z_{t_i}}$. A tout buffer $b = (t_i, t_j) \in \mathcal{B}$ sont associées au plus $k_i \times k_j$ équations liant les dates de premières exécutions des tâches t_i et t_j .

Pour tout vecteur K fixé, la recherche du débit optimum d'un ordonnancement K -périodique se modélise alors par la recherche d'un circuit de ratio maximum dans un graphe. La complexité d'un tel algorithme dans le cas présent est en $\mathcal{O}((\sum_{t \in \mathcal{T}} k_t)^2)$.

Dans la dernière partie, nous présenterons quelques pistes qui permettent de sélectionner les bonnes valeurs de K pour évaluer le plus précisément possible le débit. Nous montrerons sur un exemple simple que l'amélioration du débit ne varie pas de manière linéaire en fonction de l'augmentation de K . Nous limiterons également les valeurs de K à explorer pour améliorer l'évaluation du débit à partir d'une propriété de dominance. Nous terminerons par une évaluation expérimentale de K sur des exemples académiques et industriels.

3 Perspectives

D'un point de vue académique, le problème essentiel reste de déterminer s'il existe un algorithme de complexité polynomiale pour évaluer le débit maximum d'un SDF. Notre étude propose une méthodologie ; la prochaine étape est d'exprimer des propriétés de dominance plus efficaces, ou à défaut des heuristiques pour déterminer des valeurs de K suffisamment petites (polynomiales si possible) pour évaluer le débit maximum le plus efficacement possible.

La valorisation de ce travail d'un point de vue industriel est son intégration à terme dans la chaîne de compilation associée à l'architecture MPPA256, un MPSoC composé de 256 coeurs et développé par la société Kalray. Les premières expérimentations sont encourageantes, et le passage à un modèle de communication plus proche des applications (les *Cyclo-Static Dataflow*) introduits par [1] est envisagé.

Références

- [1] Greet Bilsen, Marc Engels, Rudy Lauwereins, and J A Peperstraete. Cycle-static data flow. *IEEE Transactions on Signal Processing*, pages 3255–3258, 1995.
- [2] Bruno Bodin, Alix Munier-Kordon, and Benoît Dupont de Dinechin. K-Periodic Schedules for Evaluating the Maximum Throughput of a Synchronous Dataflow Graph. In *International Conference on Embedded Computer Systems : Architectures, Modeling, and Simulation, SAMOS XII*, pages 152–159, 2012.
- [3] Edward A. Lee and David G. Messerschmitt. Synchronous dataflow. *Proceedings of the IEEE*, 75(9) :1235–1245, 1987.
- [4] Olivier Marchetti and Alix Munier Kordon. Cyclic Scheduling for the Synthesis of Embedded Systems. In Yves Vivien and Robert Frederic, editors, *Introduction to scheduling*, chapter 6, pages 135–164. Chapman and Hall/CRC Press, 2009.
- [5] Alix Munier. Régime asymptotique optimal d'un graphe d'événements temporisé généralisé : Application à un problème d'assemblage. *RAIRO-Automatique Productique Industrielle*, 25(5) :487–513, 1993.
- [6] S. Sriram and S.S. Bhattacharyya. *Embedded multiprocessors : Scheduling and synchronization*. CRC, 2009.