

Modélisation et dimensionnement des systèmes mixtes flot de données/temps réel

Mots clés :

- **Directeur de thèse** : Alix Munier
- **Co-encadrant(s)** :
- **Unité de recherche** : Laboratoire d'informatique de Paris 6
- **Ecole doctorale** : École Doctorale Informatique, Télécommunications, Électronique de Paris
- **Domaine scientifique principal**: Divers

Résumé du projet de recherche (Langue 1)

L'émergence d'applications intégrant des tâches temps réel et du calcul intensif nécessite un effort de recherche pour les modéliser et les exécuter efficacement. A titre d'exemples, les applications de la réalité augmentée (Google Glass) ou d'aide à la conduite ADAS (Advanced Driver Assistance Systems) mélangent des tâches de calcul intensif (le traitement d'image pour l'analyse de scènes complexes par exemple) et des tâches critiques de prise de décision ou de contrôle (le freinage d'urgence ou le régulateur de vitesse ACC). Les propriétés telles que le déterminisme, l'absence d'interblocage (deadlock), la vivacité d'une application etc. ne sont pas garanties de la même manière dans les systèmes temps réels et les systèmes de calcul « best effort ». Concrètement dans les systèmes dits mixtes des contraintes temporelles en plus des contraintes de données doivent être satisfaites comme par exemple un temps de réponse minimum entre la réception d'une donnée en provenance d'un capteur et la réponse du système. La programmation optimisée des processeurs massivement parallèles pour les applications de calcul intensif est un processus compliqué. Une manière d'aborder cette problématique est de représenter les applications est l'utilisation des « Synchronous DataFlow Graph » (SDFG en bref) [1] et ses extensions qui sont fréquemment utilisés pour décrire les échanges de données au sein d'une application. Un SDFG est un graphe orienté dont les nœuds sont associés à des acteurs (ou tâches) et les arcs représentant des liens de communication (ou buffers). De plus, des poids entiers spécifient la quantité de données produite et consommée par les acteurs à chacune de leurs exécutions. La limite de ce formalisme est qu'il ne permet pas d'écrire directement des contraintes temporelles (i.e. temps maximum entre l'activité de deux acteurs), comme c'est le cas pour les automates temporisés [2].

Résumé du projet de recherche (Langue 2)

Le but de la thèse est de construire une extension des SDFG qui permette de prendre en charge des contraintes de temps spécifiques (débit imposé aux acteurs, temps maximum entre l'activation de deux acteurs, dates de disponibilité, dates d'échéances...), en partant de travaux de Cucu et al. [3] qui ont montré que des contraintes de ce type peuvent être modélisées par une contrainte de latence. Une fois le nouveau modèle défini, la généralisation des outils mathématiques et algorithmiques actuels pour évaluer l'existence d'un ordonnancement des acteurs et du débit maximum de l'application [4] sera aussi abordée. Les outils résultants seront expérimentés sur des jeux de tests pertinents du point de vue des applications.

Informations complémentaires (Langue 2)

Références [1] Edward A. Lee and David G. Messerschmitt. Synchronous data flow. Proceeding of the IEEE, vol. 75(no. 9) :pp. 1235-1245, 1987. [2] Matthieu Lemerre, Vincent David, Christophe Aussagues, and Guy Vidal-Naquet. An introduction to time-constrained automata. In Simon Bliudze, Roberto Bruni, Davide Grohmann, and Alexandra Silva, editors, ICE, volume 38 of EPTCS, pages 83-98, 2010.1 [3] Liliana Cucu, Nicolas Pernet, and Yves Sorel. Periodic real-time scheduling: from deadline based model to latency-based model. Annals OR, 159(1) :41–51, 2008. [4] Olivier Marchetti and Alix Munier Kordon. Cyclic Scheduling for the Synthesis of Embedded Systems, chapter 6. Introduction to scheduling. Chapman and Hall/CRC Press, November 2009. ISBN : 978-1420072730