

Proposition de sujet de thèse de doctorat au LIAS

Titre : Un framework d'aide à l'optimisation multi-objectifs pour le déploiement et le dimensionnement des systèmes temps réel

Laboratoire : LIAS/ISAE-ENSMA & Université de Poitiers (<https://www.lias-lab.fr/>)

Encadrants : Emmanuel Grolleau (grolleau@ensma.fr), Yassine OUHAMMOU (yassine.ouhammou@ensma.fr) et Antoine Bertout (antoine.bertout@univ-poitiers.fr)

Mots clés : ordonnancement, validation temporelle, ingénierie dirigée par les modèles, optimisation.

Date de début de thèse : Octobre 2018

Contexte. Les approches utilisées industriellement pour le développement d'applications temps réel critiques se basent de plus en plus sur des approches orientées modèle (UML-MARTE, AADL, SysML, etc.) afin d'appréhender la complexité de ces applications. Deux aspects majeurs doivent être considérés lors du déploiement : la préservation de la sémantique fonctionnelle de l'application et le respect des contraintes non-fonctionnelles y compris temporelles. La sémantique est généralement exprimée par des dépendances entre des fonctions atomiques. Les dépendances définissent un ordre d'activation des fonctions qui doit être respecté lorsque ces fonctions sont déployés sur une plateforme d'exécution (composée d'un ensemble de noeuds de calcul communiquant par des bus). Les fonctions doivent être concrètement placées dans des tâches, elles-mêmes ordonnancées sur des noeuds de calcul. Si deux fonctions échangeant des données sont placées sur des tâches différentes (elles-mêmes placées sur différents noeuds) alors cet échange de données sera implémenté par un message acheminé sur le bus. La préservation sémantique garantit que l'ensemble de tâches communicantes conserve l'ordre initial d'activation des fonctions qui leur sont assignées.

Problématique. Dans le domaine de l'ordonnancement temps réel, chaque choix de déploiement effectué a un impact essentiel sur la validité et la qualité du système. Plusieurs travaux ont proposé de mettre en place des méthodes d'analyse et de synthèse automatiques permettant de guider le concepteur vers une architecture opérationnelle valide et optimisée. Néanmoins, ces travaux visent généralement à optimiser un type particulier de métriques (souvent le placement, la partition et l'ordonnancement) en laissant de côté des métriques liées au domaine de la performance temporelle (comme la latence de chaînes de tâches communicantes) ou à d'autres domaines (comme la consommation énergétique ou la sûreté de fonctionnement). De plus, le concepteur se voit offrir une solution sans moyen d'imposer des contraintes spécifiques (de co-placement par exemple de certaines fonctions, ou de ségrégation d'autres). Il n'existe pas aujourd'hui de langage de haut niveau permettant d'exprimer des contraintes liées à l'optimisation d'architecture.

Objectifs & travail à faire. Comme il n'existe pas de méthode générique permettant de résoudre le problème de déploiement sans compromis, il serait intéressant d'avoir un support flexible et facile d'utilisation permettant de guider le concepteur durant l'exploration de l'espace des architectures possibles. La flexibilité consiste ici à aider le concepteur à choisir parmi les méthodes proposées dans la littérature celles qui sont les plus adaptées à sa situation de déploiement. Par facilité d'utilisation, nous entendons qu'un utilisateur non expert en optimisation ou en ordonnancement temps réel doit être capable de correctement formaliser son problème d'optimisation.

La thèse sera centrée sur la définition d'un framework d'optimisation pour le déploiement automatisé d'un système temps réel et de propositions novatrices dans le domaine de la synthèse des paramètres d'ordonnancement de tâches communicantes. Les activités suivantes sont envisagées :

- L'étude de l'impact de l'*assignation de priorités* et des *paramètres* de tâches sur les latences et la proposition de méthodes d'assignation novatrices dans le domaine.
- La définition d'un *méta-modèle pour l'optimisation*. Le méta-modèle doit inclure toutes les informations pertinentes permettant d'optimiser le déploiement d'applications temps réel.
- La définition d'un *moteur de contraintes* capable de générer automatiquement les contraintes d'optimisation en inspectant le modèle d'entrée (dépendances fonctionnelles, contraintes d'allocation, échéances, activations, etc.).
- La définition de *ponts* entre différents solveurs et algorithmes d'optimisation. Les ponts réaliseront des transformations entre le formalisme d'optimisation et les modèles d'entrée de ces solveurs/algorithmes.

Ces fonctionnalités seront proposées à travers l'usage de l'ingénierie dirigée par les modèles (IDM), de langages et d'outils. Des techniques d'apprentissage automatique pourront également être employées.