



Offre de thèse

Ordonnancement temps réel de plateformes hétérogènes

Financement : ANR JCJC

Laboratoire : Laboratoire d'Automatique et d'Informatique pour les Systèmes (LIAS), Poitiers

Date de début : Septembre 2022

Encadrement : Pascal RICHARD et Antoine BERTOUT (Équipe temps réel)

(L'encadrant dont le nom est souligné est l'encadrant référent, contact : antoine.bertout@univ-poitiers.fr)

Contexte

Les systèmes sur puce multiprocesseur (MPSoCs) embarqués dans les systèmes temps réel sont de plus en plus hétérogènes (dotés de CPUs, GPUs ou NPU par ex.). Cette hétérogénéité permet des gains de performances (calcul, consommation énergétique) mais complexifie leur analyse. Les systèmes temps réel doivent assurer des garanties logiques mais aussi temporelles.

La partie applicative de ces systèmes est représentée sous la forme de tâches ayant des contraintes temporelles, telle qu'une échéance, date avant laquelle l'exécution d'une tâche doit être terminée. Pour la partie matérielle, ces systèmes hétérogènes sont souvent décrits dans la littérature comme des plateformes "unrelated" [1]. Dans cette classification, il est possible d'assigner une vitesse d'exécution différente à chaque couple tâche/processeur. Cela généralise notamment la catégorie dite "homogène", où les processeurs peuvent avoir des vitesses différentes, mais constantes pour toutes les tâches.

Les caractéristiques temporelles de telles applications requièrent un traitement particulier de la part du système d'exploitation via son ordonnanceur. L'algorithme que ce dernier emploie doit permettre une gestion efficace des ressources (processeurs multi-coeurs, mémoire, etc.) et garantir le respect des contraintes de temps des applications quelle que soit la durée de fonctionnement du système.

Cette thèse est financée dans le cadre du projet SHRIMP de l'appel à projet 2021 Jeune Chercheur Jeune Chercheuse (JCJC) de l'Agence Nationale pour la Recherche (ANR).

Description du sujet

Objectif La thèse vise à proposer un ordonnanceur temps réel global (qui autorise la migration des tâches entre processeurs) et dynamique (en-ligne) pour des plateformes "unrelated". Cet ordonnanceur doit être capable de gérer des tâches sans date d'arrivée prédéfinie (tâches sporadiques) et de réagir en ligne aux événements, tout en assurant le respect des échéances des tâches. Les solutions existantes sont statiques. Elles ne permettent pas une utilisation satisfaisante des ressources. Par exemple, elles ne peuvent pas tirer profit de la fin d'exécution précoce (avant la fin de son temps d'exécution pire-cas) d'une tâche. De plus, les modèles de tâches considérés dans ces travaux ne sont pas adaptés aux caractéristiques des applications modernes (dépendances) et réalistes (temps d'exécution pire-cas monolithique pour une tâche s'exécutant sur possiblement sur différents processeurs). Le projet a pour objectif de considérer tout d'abord un cas particulier des plateformes "unrelated" nommées "consistent" pour lesquelles il existe un ordre de comparaison entre les processeurs mais où les vitesses

des processeurs ne sont pas nécessairement constantes (comme pour les plateformes homogènes). Cette catégorie permet notamment de représenter des architectures de type ARM big.LITTLE ayant des processeurs lents et rapides, d'architectures différentes mais dotées du même jeu d'instruction. Ensuite, il s'agira d'être critique vis-à-vis du modèle de tâches classiquement utilisé et de proposer un algorithme d'ordonnancement capable d'ordonner des tâches dépendantes. Ce dernier modèle permettrait de représenter plus fidèlement des tâches ayant des sections de code dont le temps d'exécution pourrait varier en fonction du processeur utilisé.

Validation Les solutions développées devront être validées formellement et comparées avec les ordonnanceurs existants. Par des simulations, l'attention sera portée sur les performances de l'ordonnanceur, par exemple sur la charge d'utilisation supportée ou sur le nombre de changements de contexte (préemptions, migrations) qui ont un impact fort sur l'applicabilité des résultats. Le chercheur recruté pourra se baser un prototype de simulateur existant développé en Python. Le projet met également l'accent sur l'évaluation pratique de la solution. Les algorithmes d'ordonnancement devront être implémentés sur un banc d'essai réaliste (OS temps réel ou patch par exemple). Cette dernière tâche sera spécifiquement portée par deux stagiaires financés par le projet, intervenant à différents moments de la thèse. Le chercheur recruté sera impliqué dans leur encadrement.

Collaboration internationale Dans la continuité des travaux déjà réalisés [3, 2] sur la thématique, la thèse sera réalisée en collaboration avec le Professeur Joël Goossens de l'Université Libre de Bruxelles.

Profil recherché

Le candidat devra être titulaire d'un master en informatique ou d'un diplôme d'ingénieur et posséder des connaissances en système, informatique théorique et mathématiques ainsi qu'en programmation. Un bon niveau d'anglais à l'écrit et l'oral est nécessaire.

Documents à fournir

- Curriculum Vitae ;
- Lettre de motivation ;
- Notes et classements de master ou équivalent ;
- Dernier rapport de stage ;
- Tout autre document jugé nécessaire par le candidat pouvant enrichir le dossier de candidature.

Références

- [1] S. K. BARUAH, « Feasibility Analysis of Preemptive Real-Time Systems upon Heterogeneous Multiprocessor Platforms », in Real-Time Systems Symposium, IEEE, 2004, p. 37-46.
- [2] A. BERTOUT, J. GOOSSENS, E. GROLLEAU, R. JAMIL et X. POCZEKAJLO, « Workload assignment for global real-time scheduling on unrelated clustered platforms », Real-Time Systems, p. 1-32, 2021.
- [3] A. BERTOUT, J. GOOSSENS, E. GROLLEAU et X. POCZEKAJLO, « Template schedule construction for global real-time scheduling on unrelated multiprocessor platforms », in 2020 Design, Automation & Test in Europe Conference & Exhibition (DATE), IEEE, 2020, p. 216-221.
- [4] E. MASSA, G. LIMA, B. ANDERSSON et V. PETRUCCI, « Heterogeneous Quasi-Partitioned Scheduling », in 2021 IEEE Real-Time Systems Symposium (RTSS), 2021, p. 266-278. DOI : 10.1109/RTSS52674.2021.00033.