

Analyse flot-de-donnee cyclo-statique pour SC Java

Jean-Pierre.Talpin@inria.fr

Quiconque a pratiqué la programmation concurrente en reconnaîtra la difficulté. Résoudre synchronisation, causalité, ordonnancement, non-déterminisme dans les logiciels multi-tâches est extrêmement difficile. Assurer l'exactitude d'un programme par rapport à sa spécification et garantir un comportement déterministe sont pourtant nécessaires pour l'exécution d'un tel code en toute sécurité. Il est donc souhaitable de synthétiser un multi-tâche partir de spécifications formelles de haut-niveau, abstraites, en utilisant une approche de développement "correcte par construction".

Le programme de doctorat proposé prend la suite de résultats relatifs à l'ordonnancement affine abstrait de tâches Java dont les communications sont décrites par un graphe UCSDF (flot de données cyclo-statique ultimement périodique, permettant la synthèse automatique d'ordonnanceurs temps réel.

Nous souhaitons poursuivre ce travail en développant l'analyse du programme nécessaire à la construction du graphe UCSDF d'un ensemble de tâches SCJ donné (SCJ: Safety-Critical Java). Il s'agit d'une analyse quantitative des communications et des calculs de tâches Java d'une manière affinant leur topologie initiale (éventuellement construit à partir de la structure de l'ensemble des tâches d'un programme). L'objectif est d'obtenir une abstraction de cette topologie sous la forme d'un graphe UCSDF.

Le programme de thèse consistera à la fois de l'élaboration d'un cadre théorique, de comprendre les techniques existantes de représenter communiquer tâches Java en utilisant des automates, l'élaboration des abstractions de ces automates dans un modèle adéquat de graphiques de flux de données et / ou des relations affines, pour finalement réutiliser et éventuellement étendre résultats précédents sur la programmation abstraite affine.

Il se conclura par la réalisation d'un logiciel permettant, à partir d'un ensemble de tâches SCJ, synthétiser automatiquement un ordonnanceur des communications et des calculs (la méthode init du programme) à partir des outils existants et des prototypes d'analyse développés

La thèse comprend la validation expérimentale d'une preuve de concepts à travers un prototype fonctionnel. Il requiert français et anglais courant, et des connaissances et expériences, niveau Master, en programmation: analyse de programme et compilation, interprétation abstraite, théorie de la concurrence, la théorie de l'ordonnancement. Une expérience de programmation concurrente en Java (RTJ / SCJ) est un plus.

REFERENCES

"Timed behavioural modelling and affine scheduling of embedded software architectures in the AADL using Polychrony". L. Besnard, A. Bouakaz, T. Gautier, P. Le Guernic, Y. Ma, J.-P. Talpin, H. Yu. In Science of Computer Programming.

Elsevier, 2014

"Design of Safety-Critical Java Level 1 Applications Using Affine Abstract Clocks". A. Bouakaz and J.-P. Talpin. International Workshop on Software and Compilers for Embedded Systems (M-SCOPES'13). ACM, June 2013.

"Buffer minimization in earliest-deadline first scheduling of dataflow graphs". A. Bouakaz and J.-P. Talpin. Conference on Languages, Compilers and Tools for Embedded Systems (LCTES'13). ACM, June 2013