Proposition de recherche doctorale

Maîtrise de la couche hyperviseur dans une architecture multi-coeur COTS afin de prédire les pires temps d'exécution nécessaires à la certification

Mots clés :

- **Directeur de thèse** : LAURENT PAUTET
- **Co-encadrant(s)** :
- **Unité de recherche** : Laboratoire Traitement et Communication de l'Information
- **Ecole doctorale** : École Doctorale Informatique, Télécommunications, Électronique de Paris
- **Domaine scientifique principal**: Divers

Résumé du projet de recherche (Langue 1)

Contexte Le programme A380 a marqué le début d'une mutation de l'architecture des systèmes informa- tiques dans l'avionique. Ce programme a démontré la faisabilité et les avantages du passage d'une architecture distribuée reposant sur des calculateurs à fonction unique (architecture dite fédérée) vers une architecture où un seul calculateur hébergerait plusieurs fonctions. De là est né le concept dit d'IMA, signifiant Integrated Modular Avionique. Ce concept d'IMA a permis de rationaliser l'électronique de bord en ramenant de plusieurs di- zaines (environ 37) à quelques unités (7), les différents types de calculateurs nécessaires pour exécuter l'ensemble des applications embarquées dans un avion (carburant, freinage, moteur, électricité, cabine, climatisation, pilote automatique, « flight management system », etc.) Cette mutation a pu être observée chez les deux principaux constructeurs avec l'A380 côté AIRBUS et le B787 côté BOIENG. Cette convergence a donné naissance à un standard : l'IMA [1] dit de première génération ou IMA1G. Ce phénomène d'uniformisation des calculateurs s'amplifie avec le programme A350 dans lequel ces 7 types de calculateurs différents ont été réduits à 2 types pour supporter l'ensemble des fonctions avioniques. Une des finalités de l'IMA est de proposer une vision de l'architecture centrée sur des modules logiciels « indépendants » déployés sur une nombre réduit de calculateurs relativement puis- sant de type COTS, Pour atteindre cet objectif, il faut s'assurer que ces modules logiciels puis- sent partager les ressources matérielles tout en restant rester le plus indépendant possible du point de vue de leur propriétés fonctionnelles et non-fonctionnelles. La maitrise du comporte- ment de l'architecture logicielle requiert celle du matériel, ainsi qu'une bonne connaissance des moyens disponibles pour contrôler le matériel depuis le logiciel à l'exécution. Problématique Depuis plusieurs années, le marché des processeurs a évolué des architectures dites mono- cœurs vers des architectures dites multi-cœurs. Conjointement aux problèmes posés par le changement d'architecture logicielle dû à l'IMA, il est nécessaire de prendre en compte l'évolution de la chaine d'approvisionnement en proces- seurs. Au delà du gain potentiel évident en termes de puissance de calcul, les architectures multi-cœurs sont simplement plus présentes sur le marché que les architectures mono-cœurs. Il est possible qu'à moyen terme la production de mono-cœurs offrant des performances suffisan- tes soit tout simplement impossible. C'est ce constat qui motive de nombreux acteurs de l'industrie aéronautique ou automobile à mener des recherches sur l'impact de ces technologies sur les résultats acquis pour des architectures mono-cœurs. La maîtrise de ces évolutions d'architectures logicielles et matérielles ne pourra se faire que si les méthodes et technologies de génie logiciel associées sont elles mêmes maîtrisées. Par exemple, il est clair qu'il faudra assurer la conservation des acquis de l'IMA, en particulier concernant les propriétés d'isolation des modules logiciels. Les deux propriétés à conserver concernant l'approche IMA sont les propriétés de ségrégation des modules et la maîtrise des pire temps de réponses de chaque module. Ces propriétés sont assurées soit par construc- tion soit par analyse en amont de l'architecture lors de sa conception. Les supports d'exécution IMA tels que ceux compatibles avec le standard ARINC 653 doivent être configurés statiquement en déclarant les propriétés de chaque module logiciel a priori. C'est cette configuration qui permet en opération de maintenir l'isolation spatiale et temporelle entre modules malgré leur défaillance potentielle. Le cœur de cette configuration consiste à spécifier les caractéristiques temporelles des tâches constituant les modules logiciels indé- pendants (appelés partitions). Cette configuration logicielle passe par une estimation a priori du temps d'exécution de chaque tâche d'un module et de l'impact des dépendances entre tâches sur le pire temps de réponse du module logiciel. De nombreuses études se sont attaquées au problème de l'estimation des pires temps d'exécution sur des architectures multi-cœurs. Cependant, il est possible que l'obligation de séparation temporelle et spatiale modifie en profondeur les perfor- mances des processeurs. Il s'agit donc d'étudier ces problématiques d'estimation des caracté- ristiques temporelles d'une tâche dans le cas où le fonctionnement interne du processeur et du support logiciel ARINC déployé ont été adaptés pour assurer l'isolation des modules. Sujet de Thèse L'évolution conjointe de l'architecture logicielle et matérielle de l'informatique de bord d'un avion est une problématique forte en terme de recherche appliquée. Cette évolution transforme en profondeur la nature et les contraintes de réalisation de tels systèmes sur plusieurs points : • L'élargissement constant des domaines concernés par l'approche IMA. • L'augmentation du nombre et de la précision des contraintes imposées par le déploiement des applications logicielles concernant : o la puissance de calcul et la bande passante consommées o les fonctionnalités intégrées dans un même module o l'espace mémoire requis par le module o l'estimation des bornes de latences de propagation et pire temps de réponse • L'apparition de ressources partagées enfouies dans le matériel pouvant modifier la ma- nière dont l'isolation spatiale et temporelle entre modules est implémentée. Pour répondre aux défis posés par ces trois changements, nous, THALES et Telecom Paris- Tech, proposons d'étudier et mettre en œuvre, au cours d'une thèse, des méthodes de vérifica- tion et analyse reposant sur une modélisation conjointe de l'architecture matérielle et logicielle du système. L'objectif de ces analyses est de valider les propriétés d'isolation spatiale et temporelle, et d'estimer les pires temps d'exécution d'une tâche sur un tel système. Nous proposons de mener à bien cette étude à travers les étapes suivantes : 1. Recenser et modéliser un échantillon de processeurs multi-cœurs (de sorte à couvrir de manière raisonnable les différents types d'architectures

disponibles). La phase de modélisation devra mettre en évidence les composants du processeur susceptibles de poser problème au moment de la mise en œuvre des propriétés d'isolation 2. Proposer des critères de sélection et des scénarios d'utilisation des processeurs et de leurs cœurs de calcul permettant de maintenir au sein du matériel les propriétés d'isolation spatiale et temporelle. Cette étape vise à identifier les conditions d'usages permettant d'instaurer un niveau d'indépendance suffisant entre cœurs de calcul. 3. Analyser la couche « hyperviseur » fournie par les fabricants de processeurs permet- tant de contrôler le fonctionnement de ces derniers : INTEL – FREESCALE – ARM – etc. Puis, proposer une méthode pour contrôler les ressources partagées entre les cœurs pour se conformer aux scénarios d'usage validés en phase 2. 4. Estimer la perte de puissance de calcul résultant de la mise en place du niveau d'indépendance souhaité entre cœurs de calcul grâce à des modèles comportementaux liant la couche matérielle et logicielle,. 5. En fonction, des résultats des étapes 2 à 4 proposer un modèle d'ordonnancement des partitions sur les cœurs de calcul des matériels étudiés. Cela revient à déterminer le modèle d'ordonnancement étendu que pourra supporter l'exécutif ARINC 653 6. Proposer une méthode permettant d'inférer une borne du WCET d'une tâche incluent le temps consommé par la mise en œuvre au niveau matériel des propriétés d'isolation spatiale et temporelle. Le but final est de pouvoir fournir au moins une méthode permettant d'estimer l'impact de l'usage de multi-cœur comme support pour déployer un exécutif compatible ARINC sur les per- formances du support d'exécution en fonction du type d'isolation fournie entre modules. La ri- chesse des architectures et des scénarios d'usage des processeurs considérés pourra être limi- tée dans un premier temps pour pouvoir évaluer la validité de la démarche scientifique. Aecturale AADL qui permet la modélisation conjointe du matériel et du logiciel. En effet, Telecom ParisTech a su développer une activité ri- che autour de l'usage de tels formalismes pour faciliter ou automatiser certaines étapes de la conception et du déploiement de systèmes temps réels embarqués [3,4]. Pour chacun, l'intérêt de l'étape 1 et 2 est d'étudier le potentiel offert par une approche de type MDE pour guider et valider l'usage de multi-cœurs pour déployer une architecture logicielle de type IMA. L'un des enjeux clés sera la modélisation des interférences entre les cœurs de calcul engendrées par le partage de ressources physiques tels que les contrôleurs mémoires ou les bus de données. La figure ci-dessous représente une architecture matérielle clas- sique pour un multi-cœur. Il apparaît claire- ment que l'accès aux entrées-sorties des cœurs de calcul vers la mémoire devra être sévèrement contrôlée. A cause de la diversi- té des architectures mise à disposition, res- treindre l'étude à une seule architecture se- rait relative limitant. L'usage d'un langage générique de description de l'architecture matérielle, couplée à une mé- thode d'analyse spécifique au problème d'isolation spatiale et temporelle nous semble être une étape clés pour mener à bien les étapes 1 et 2 décrites dans le sujet. Cette approche est une alternative intéressante par rapport à des démarche ad'hoc vus que le challenges de ces archi- tectures réside dans le contrôle, à l'exécution, des ressources partagées du processeur [5]. Si il est possible d'utiliser les multi-cœurs dans des conditions où l'on peut instaurer un niveau d'indépendance satisfaisant entre cœurs de calcul, alors il faudra adapter le BSP et l'exécutif ARINC 653 (cf figure à gauche) pour tirer au mieux partie de ces résultats lors du déploiement de l'exécutif ARINC 653 sur le multi-cœurs. Ces résultats présentent un intérêt évident pour THALES mais permettront aussi à Telecom ParisTech de poursuivre son activité sur l'étude des motifs de conceptions utiles pour assembler et configurer automatiquement un support parti- tionné sur mesure [4]. Les dernières étapes proposées pour mener les travaux de thèses formeront le cœur des résul- tats en terme d'innovation. Leur finalité est de donner les clés pour évaluer les temps d'exécutions d'une application sur un support d'exécution IMA contraignant les multi-cœurs en selon le modèle d'ordonnancement en tenant compte des ressources partagées aux niveau ma- tériel [6]. Le modèle d'attribution des cœurs aux partitions pourra être largement inspirés de po- litiques existantes tels que les modes AMP et SMP usuels dans le cas non partitionné [7]. Références : [1] Integrated Modular Avionic ARINC6XX series, www.arinc.com/aviation.html [2] Brian Sutterfield, John A. Hoschette, Paul Anton, Future embedded real time processors jet fighter mission computers, , Digital Avionics Systems Conference, 2008 [3] Julien Delange, Laurent Pautet, Alain Plantec, Mickael Kerboeuf, Frank Singhoff, and Fabrice Kordon. 2009. Validate, simulate, and implement ARINC653 systems using the AADL. Ada Lett. 29, 3, 2009, p 31-44. [4] J. Delange, L. Pautet, Jérôme Hugues and Dionisio de Niz. A MDE-based Process for the Design, Im- plementation and Validation of Safety-Critical Systems. In 5th IEEE International workshop UML & AADL, 2010. [5] Nidhi Aggarwal, Parthasarathy Ranganathan, Norman P. Jouppi, James E. Smith, "Isolation in Commodity Multicore Processors," Computer, pp. 49-59, June, 2007 [6] Jun Yan and Wei Zhang, WCET Analysis for Multi-Core Processors with Shared L2 Instruction Caches. In Proceedings of the 2008 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS '08). IEEE Computer Society, 2008, p 80-89. [7] James H. Anderson, John M. Calandrino, and Umamaheswari C. Devi. Real-Time Scheduling on Multicore Platforms. In Proceedings of the 12th IEEE Real-Time and Embedded Technology and Appli- cations Symposium (RTAS '06). IEEE Computer Society, 2006, 179-190.

Résumé du projet de recherche (Langue 2)

Depuis plusieurs années, le marché des processeurs a évolué des architectures dites mono- cœurs vers des architectures dites multi-cœurs. Conjointement aux problèmes posés par le changement d'architecture logicielle dû à l'IMA, il est nécessaire de prendre en compte l'évolution de la chaine d'approvisionnement en proces- seurs. Au delà du gain potentiel évident en termes de puissance de calcul, les architectures multi-cœurs sont simplement plus présentes sur le marché que les architectures mono-cœurs. Il est possible qu'à moyen terme la production de mono-cœurs offrant des performances suffisan- tes soit tout simplement impossible. C'est ce constat qui motive de nombreux acteurs de l'industrie aéronautique ou automobile à mener des recherches sur l'impact de ces technologies sur les résultats acquis pour des architectures mono-cœurs. La maîtrise de ces évolutions d'architectures logicielles et matérielles ne pourra se faire que si les méthodes et technologies de génie logiciel associées sont elles mêmes maîtrisées. Par exemple, il est clair qu'il faudra assurer la conservation des acquis de l'IMA, en particulier concernant les propriétés d'isolation des modules logiciels. Les deux propriétés à conserver concernant l'approche IMA sont les propriétés de ségrégation des modules et la maîtrise des pire temps de réponses de chaque module. Ces propriétés sont assurées soit par construc- tion soit par analyse en amont de l'architecture lors de sa conception. Les supports d'exécution IMA tels que ceux compatibles avec le standard ARINC 653 doivent être configurés statiquement en déclarant les propriétés de chaque module logiciel a priori. C'est cette configuration qui permet en opération de maintenir l'isolation spatiale et temporelle entre modules malgré leur défaillance potentielle. Le cœur de cette configuration consiste à spécifier les caractéristiques temporelles des tâches constituant les modules logiciels indé- pendants (appelés partitions). Cette configuration logicielle passe par une estimation a priori du temps d'exécution de chaque tâche d'un module et de l'impact des dépendances entre tâches sur le pire temps de réponse du module logiciel. De nombreuses études se sont attaquées au problème de l'estimation des pires temps d'exécution sur des architectures multi-cœurs. Cependant, il est possible que l'obligation de séparation temporelle et spatiale modifie en profondeur les perfor- mances des processeurs. Il s'agit donc d'étudier ces problématiques d'estimation des caracté- ristiques temporelles d'une tâche dans le cas où le fonctionnement interne du processeur et du support logiciel ARINC déployé ont été adaptés pour assurer l'isolation des modules.