

## Offre de thèse à l'INSA de Lyon

**Titre :** Gestion dynamique de la mémoire non-volatile embarquée

**Lieu :** Laboratoire CITI, INSA-Lyon

**Partenaires :** société eVaderis (Grenoble) et laboratoire Vérimag (Grenoble) <sup>1</sup>

**Financement :** Allocation doctorale (36 mois) du conseil régional Rhône-Alpes

**Date de début :** 1<sup>er</sup> octobre 2016

**Contact :** Guillaume Salagnac

04 72 43 64 13

guillaume.salagnac@insa-lyon.fr

### Résumé

Cette thèse porte sur la conception et l'étude de nouveaux mécanismes logiciels destinés à améliorer les performances des objets connectés de prochaine génération, basés sur des puces à mémoire non-volatile (NVRAM). La problématique scientifique est celle de la gestion de la mémoire dynamique, c'est à dire l'allocation et le placement des zones mémoire où seront stockées les données du calcul. L'entreprise eVaderis, impliquée dans le projet, conçoit des nouveaux types de puces pour lesquelles les techniques actuelles de gestion de la mémoire sont insuffisantes et doivent être repensées.

### Recrutement

**Profil recherché :** diplôme d'ingénieur ou de Master 2 en informatique (ou équivalent)

#### Compétences requises :

logiciel : programmation de bas niveau et/ou embarquée, langages C et/ou ASM

matériel : architecture des systèmes sur puce (microprocesseur, hiérarchie mémoire)

langues : anglais courant, aussi bien en lecture et rédaction qu'à l'oral

#### Compétences souhaitées :

logiciel : simulation à événements discrets

environnement de travail : Linux, ligne de commande, langage(s) de script

qualités personnelles : travail en autonomie, capacité d'organisation, initiative, curiosité

qualités relationnelles : rigueur, bonnes capacités de communication

**Candidature :** envoyer par e-mail un dossier en PDF avec

- un curriculum vitae détaillé
- une lettre de motivation et/ou votre dernier rapport de stage
- vos relevés de notes (si possible avec classement) des deux dernières années
- 2 ou 3 lettres de recommandation, ou les coordonnées de personnes à contacter

### Contexte scientifique

L'informatique ambiante est en passe de devenir une réalité. Les progrès récents dans les technologies de semi-conducteurs permettent de concevoir et fabriquer des puces électroniques de toute petite taille et de basse consommation énergétique. Une innovation clé de ce domaine est la mise au point des nouvelles mémoires non-volatiles embarquées (NVRAM), qui promettent de réduire considérablement la consommation des systèmes embarqués. Mais elles posent aussi de nouveaux problèmes de programmation logicielle [BCGL11].

Un système embarqué s'articule classiquement sur une hiérarchie mémoire composée de mémoire vive d'une part, et de stockage d'autre part. La mémoire vive (typiquement de type «SRAM» ou «eDRAM») est rapide, mais perd son contenu lorsque l'alimentation est coupée. À l'inverse, les mémoires non-volatiles de type «flash NOR» et/ou «EEPROM» sont lentes et donc sont limitées à un usage en lecture seule ou en stockage. Plus récemment cependant, de nouveaux types de mémoires dites «NVRAM» (pour *non-volatile*

1. La thèse se déroulera à Lyon, mais des déplacements ponctuels à Grenoble sont à prévoir tout au long de la thèse.

RAM) sont en train d'apparaître qui permettent de réaliser, dans une même technologie, les deux fonctions de mémoire vive et de stockage. Ainsi, il devient possible de concevoir de nouvelles architectures de systèmes embarqués avec une consommation énergétique largement réduite [KKSM13]. C'est le domaine d'expertise de l'entreprise eVaderis, impliquée dans ce projet, qui conçoit [LJB+15 ; LBJ+16] des systèmes embarqués très basse consommation à base de mémoire magnétorésistive (STTRAM).

Pour les applications de type «intelligence ambiante», l'émergence des NVRAM permet d'envisager un fonctionnement «*normally-off*», où l'alimentation du système est coupée la plupart du temps afin d'économiser l'énergie. La capacité d'allumage et d'extinction quasi-instantanés, avec sauvegarde du contexte logiciel [AAMS14] permet d'adapter finement le taux d'activité du système, et donc sa consommation, aux besoins applicatifs.

Cependant, il n'est pas pertinent de remplacer naïvement l'ensemble de la hiérarchie mémoire par de la NVRAM [RL14]. Des contraintes matérielles, comme par exemple un nombre limité de cycles d'écriture ou une forte latence d'accès, sont inhérentes à chaque technologie. L'architecture développée par eVaderis permet de combiner dans une même puce plusieurs régions de mémoire aux caractéristiques différentes, par exemple avec un compromis différent entre vitesse d'écriture et densité de stockage. Il faut donc adapter en conséquence le logiciel [MV15] pour éviter le contre-emploi. En particulier, le programme devra choisir finement, au cours de l'exécution, dans quelle région allouer chaque donnée. Mais le développeur d'application n'a pas vocation à se préoccuper de tels détails de bas-niveau, et voudrait par ailleurs pouvoir réutiliser des briques logicielles pré-existantes, disponibles sur étagère.

L'objectif de cette thèse est d'apporter une réponse à ces problématiques nouvelles posées par la gestion dynamique de la NVRAM. Notre approche, qui est détaillée plus amplement dans la description du sujet de thèse, consiste à proposer à la fois des composants logiciels spécifiques pour le système d'exploitation, ainsi que des composants matériels nouveaux à intégrer dans l'architecture visée.

## Objectifs de la thèse

L'objectif de ce travail est d'apporter une solution globale au problème de la gestion de la mémoire dynamique pour les systèmes embarqués à mémoire non-volatile (NVRAM). La problématique consiste à faire s'exécuter la (ou les) application(s) en bonne intelligence avec la plate-forme matérielle sous-jacente. En particulier, utiliser de la NVRAM suppose de choisir finement, pour chaque donnée allouée, de son placement dans l'une ou l'autre des régions mémoires du système, voire de la déplacer au cours de l'exécution. Le développeur d'application n'est pas bien placé pour décider de ces questions de bas-niveau. Au contraire, notre approche consiste à proposer une solution automatique, où les (dé)placements sont gérés par une couche logicielle ad hoc du système d'exploitation. La solution visée comprendra ainsi un allocateur dynamique (logiciel) ainsi qu'un bloc matériel pour recenser et caractériser les accès mémoire (lectures, écritures) au cours de l'exécution.

Les différentes étapes du projet envisagées sont les suivantes. Dans un premier temps, le doctorant procédera, avec lien étroit avec l'entreprise eVaderis, à une sélection dans la littérature et/ou au développement d'une suite de programmes témoin représentatifs des applications visées : multimédia portable, traitement embarqué de données volumineuses. Cette concertation aura aussi comme objectif d'identifier les modèles possibles pour mettre les fonction de gestion mémoire à disposition des programmeurs d'application (bibliothèques partagées, kernel, OS). En effet, les décisions prises dans ce domaine peuvent influencer l'architecture logicielle de la solution cible. Ensuite, notamment grâce à l'expertise de Verimag en termes de simulation de plate-forme, il faudra mettre sur pied simulateur léger (émulateur de jeu d'instructions) permettant d'exécuter ces programmes et analyser leur comportement mémoire. La tâche suivante consistera à étudier les traces d'accès mémoires ainsi obtenues : quelles allocations, dans quelles tailles, avec quels types d'accès (lecture/écriture) à quelle fréquence, etc. Certaines données seront «chaudes» (accédées et modifiées souvent) ou au contraire «froides», et auront des durées de vie variées. Cette étude sur la forme et la démographie des structures de données permettra une meilleure compréhension du comportement mémoire des applications embarquées.

La deuxième phase consiste à proposer des politiques de placement/déplacement «omniscientes», qui améliorent «optimalement» les performances du logiciel. Par exemple, si on sait qu'un site d'allocation particulier produit seulement des objets très éphémères, alors il faudra placer ceux-ci en RAM. Respectivement, les données froides se prêteront mieux à une allocation dans la STTRAM «dense», etc. Par la suite, afin d'obtenir des mécanismes implémentables en pratique, il faudra proposer des heuristiques non-omniscientes de placement/déplacement, qui auront donc de moins bonnes performances mais qui sont implémentables, sous la forme d'un gestionnaire de mémoire dynamique ad hoc. Typiquement, de telles heuristiques auront besoin d'informations dynamiques sur le comportement de l'application, donc on s'intéressera alors à proposer des modules matériels (de type «*Performance Monitoring Counter*» PMC) pour mesurer à bas coût, pendant l'exécution, les bonnes informations nécessaires aux heuristiques de placement. L'évaluation «hors-ligne» (i.e. en rejouant les traces) des bénéfices obtenus sur les applications témoin sera un jalon intéressant. Les solutions proposées devront prendre en compte les contraintes des architectures matérielles développées par eVaderis. Ceci étant, ces architectures pourront être amenées à évoluer si un impact significatif sur les performances peut être identifié.

La troisième phase porte sur l'implémentation d'un simulateur plus détaillé, incluant le comportement mémoire à grain fin (e.g. latence d'écriture) ainsi que les bons PMC. Cette plate-forme nous permettra de faire une validation empirique globale du travail, cette fois-ci en exécutant nos applications au-dessus de notre OS (allocateur) maison, et de la plate-forme simulée NVRAM+PMC. L'avantage de travailler ainsi purement en simulation est double : non seulement la puce matérielle visée n'est pas encore disponible aujourd'hui, mais aussi et surtout nous comptons investiguer différentes modifications de son architecture (co-développement). La simulation est donc le meilleur moyen d'explorer l'espace de conception assez rapidement pour évaluer et valider la pertinence de nos propositions.

## Références

- [AAMS14] Fayçal AIT AOUDIA, Kevin MARQUET et Guillaume SALAGNAC. « Incremental checkpointing of program state to NVRAM for transiently-powered systems ». In *ReCoSoC 2014 : 7th International Workshop on Reconfigurable Communication-centric Systems-on-Chip*. 2014.
- [BCGL11] Katelin BAILEY, Luis CEZE, Steven D. GRIBBLE et Henry M. LEVY. « Operating system implications of fast, cheap, non-volatile memory ». In *HotOS 2011 : 13th USENIX conference on Hot topics in Operating Systems*. 2011.
- [KKSM13] Emre KULTURSAY, Mahmut KANDEMIR, Anand SIVASUBRAMANIAM et Onur MUTLU. « Evaluating STT-RAM as an energy-efficient main memory alternative ». In *ISPASS'13 : IEEE International Symposium on Performance Analysis of Systems and Software*. IEEE. 2013.
- [LBJ+16] Christophe LAYER, Laurent BECKER, Kotb JABEUR, Sylvain CLAIREUX, Bernard DIENY, Guillaume PRENAT, Gregory Di PENDINA, Stephane GROS, Pierre PAOLI, Virgile JAVERLIAC et al. « Reducing System Power Consumption Using Check-Pointing on Nonvolatile Embedded Magnetic Random Access Memories ». In *ACM Journal on Emerging Technologies in Computing Systems (JETC)* 12.4 (2016).
- [LJB+15] Christophe LAYER, Kotb JABEUR, Laurent BECKER, Bernard DIENY, Stephane GROS, Pierre PAOLI, Virgile JAVERLIAC et Fabrice BERNARD-GRANGER. « Hybrid STT/CMOS Design of an Interrupt based Instant On/Off Mechanism for Low-Power SoC ». In *ISVLSI 2015 : IEEE Computer Society Annual Symposium on VLSI*. 2015.
- [MV15] Sparsh MITTAL et Jeffrey VETTER. « A Survey of Software Techniques for Using Non-Volatile Memories for Storage and Main Memory Systems ». In *IEEE Transactions On Parallel And Distributing Systems* 27.5 (2015).
- [RL14] Benjamin RANSFORD et Brandon LUCIA. « Nonvolatile Memory is a Broken Time Machine ». In *MSPC 2014 : ACM SIGPLAN Workshop on Memory Systems Performance and Correctness*. 2014.