Haut Conseil de l'Évaluation de la Recherche et de l'Enseignement Supérieur

DOCUMENT D'AUTOÉVALUATION

Équipe ALSOC

Campagne d'évaluation 2023-2024 — Vague D

Table des matières

1 INFORMATIONS GÉNÉRALES SUR L'ÉQUIPE ALSOC

Nom de l'équipe : Architecture et Logiciels pour Systèmes Embarqués sur Puce (ALSOC)

Responsable de l'équipe Alix Munier Kordon

	2017	2018	2019	2020	2021	2022
PR	4	4	4	4	3	3
MCF HDR	1	2	2	2	2	2
MCF	7	6	6	6	6	7
DR	0	0	0	0	0	0
CR HDR	0	0	0	0	0	0
CR	1	1	1	1	0	0
Total permanents	13	13	13	13	11	12
Émérites	0	0	0	0	1	1
Doctorants	7	8	9	7	6	6
Ingénieurs CDD ou hors tutelles	0	0	1	1	1	0
Post-doc, ATER, etc.	1	2	0	1	1	0
Stagiaires	6	6	6	3	4	12
Total non permanents	14	16	16	12	12	18
Total avec émérites	**27**	**29**	**29**	**25**	**24**	**31**
Equivalent temps plein recherche	7.0	7.0	7.0	7.0	5.5	6.0

TABLE 1 – Personnels ALSOC sur la période 2017-2022 (au 1er juillet de chaque année)

1.1 Les thématiques scientifiques et leurs enjeux

Les activités de recherche de l'équipe portent sur le développement de méthodes et d'outils pour la conception et l'utilisation de systèmes multi-processeurs. Elles s'articulent autour des trois thématiques suivantes : *Optimisation*, *Architecture* et *Sécurité et durcissement de code*.

Optimisation : méthodes, algorithmes et outils pour l'optimisation de systèmes embarqués

On considère l'optimisation de l'exécution d'une application sur un système embarqué. Il s'agit de prendre en compte à la fois la spécificité des applications traitées et les caractéristiques de l'architecture cible.

Étude des Synchronous Dataflow graph

L'étude des Synchronous DataFlow Graphs (SDF en abrégé) reste une activité centrale dans l'équipe. Ce formalisme, introduit par Lee et Messerschmitt [4] permet de modéliser des applications constituées de processus communiquant de manière prédictive et régulière. Ainsi, ce formalisme est particulièrement adapté à des applications de type traitement du son ou de l'image.

Nous avons co-encadré une thèse avec Jean-François Nézan et Karol Desnos (Institut d'Electronique et des Technologies du numéRique, IETR) sur l'extension de méthodes d'évaluation du débit optimum d'une application pour des SDF hiérarchiques. Ce modèle de SDF développé par l'IETR permet de modéliser plus simplement des applications composées de sous-programmes communicants [1]. Les méthodes généralement utilisées pour évaluer le débit expansent l'ensemble des contraintes du problème, ce qui mène à une explosion combinatoire qui limite les outils à des cas d'école. Nous avons développé à l'occasion de cette thèse des méthodes efficaces pour évaluer des bornes du débit maximum d'une application qui évite cet explosion par le développement d'outils mathématiques qui isolent des stratégies simples et faciles à évaluer [Deroui et al., 2017a].

Nous avons également étudié la structure des communications dans le cadre d'une application temps-réel embarqué qui vérifie le standard AUTOSAR. Une application est ici décrite par un ensemble de tâches à exécuter dans des fenêtres de temps périodiques de périodes différentes selon le modèle de Liu et Layland [5]. Les communications sont réalisées par l'utilisation de mémoires partagées ; les lectures et écritures des données échangées sont effectuées aux dates de début et fin des fenêtres de temps. Le problème est l'évaluation de la latence d'une application décrite par ce formalisme. Les méthodes existantes se limitent à expanser l'ensemble des contraintes du problème, et ne peuvent traiter qu'un ensemble de tâches qui communiquent sur la forme d'un chemin. Nous avons montré que les communications peuvent se modéliser dans un formalisme proche des SDF, et nous en avons déduit des algorithmes efficaces et performants pour évaluer la latence du système dans le cas d'un graphe

quelconque [Munier Kordon and Tang, 2020]. A notre connaissance, ce sont les premiers outils généraux qui permettent d'effectuer ce calcul.

Étude de Problèmes d'ordonnancement avec contraintes de précédence et de ressource

Nous avons également étudié plusieurs problèmes d'ordonnancement d'applications sur des architectures fixées.

A l'occasion d'une thèse co-encadré avec Lilia Zaourar (CEA-LIST), nous avons travaillé sur le problème de l'ordonnancement d'un graphe de tâches communicantes sur une architecture hétérogène de type GPU-CPU. Nous considérons ici un temps de communication quand deux tâches communicantes sont sur des processeurs de type différent. Nous avons montré que la recherche d'une ordonnancement réalisable de durée minimale est un problème NP-complet, même pour un nombre illimité de machines [Aba et al., 2019]. Ce problème est un problème frontière, et toute hypothèse supplémentaire sur les durées des tâches ou l'absence de temps de communication permet d'obtenir un problème polynomial.

Enfin, nous travaillons sur l'existence d'algorithmes de complexité paramétrée pour des problèmes d'ordonnancement classiques avec fenêtres de temps. L'étude de la complexité paramétrée des problèmes d'ordonnancement avec contraintes de ressource est un sujet de recherche assez récent qui présente de nombreuses questions ouvertes [6]. Le point clé ici est de trouver les paramètres appropriés, qui permettent de développer des algorithmes de complexité Fixed-Parameter Tractable (FPT en abrégé). Dans l'article présenté dans le portfolio 4 [Munier Kordon, 2020] nous avons identifié un nouveau paramètre, le *pathwidth* qui correspond au nombre de tâches pouvant être exécutées simultanément. Nous avons alors développé un algorithme exact FPT pour résoudre l'existence d'un ordonnancement pour des tâches de durée unitaire sur m machines identiques avec fenêtres de temps et contraintes de précédence. A notre connaissance, il s'agit du premier résultat positif pour ce problème. En collaboration avec des membres de l'équipe RO nous poursuivons l'étude de ce problème en y rajoutant des contraintes de latence [Mallem et al., 2022a].

Implantation d'algorithmes d'étiquetage en composantes connexes et application à des algorithmes de vision

L'étiquetage en composantes connexes (ECC en abrégé) d'images binaires est une étape très importante pour de nombreuses applications liées au traitement de l'image. Cette étape consiste à attribuer une étiquette unique à chaque région formée de pixels connexes. Cela permet de passer d'une représentation binaire d'une image à une liste d'objets. L'analyse en composantes connexes (ACC en abrégé) permet d'extraire des caractéristiques de chacune de ces régions, comme le rectangle englobant ou le centre d'inertie et plus généralement les premiers moments.

De part sa nature, l'ECC est un traitement qui dépend des données. Il n'est pas simple de trouver le facteur limitant de ce type d'algorithme. Cela dépend de plusieurs critères : l'implémentation, l'architecture cible et les données d'entrée. Par exemple, sur certaines architectures récentes, il a été montré que l'algorithme n'était ni limité par la bande passante mémoire RAM ni par la puissance crête du processeur. Une analyse fine des performances du code a montré que les instructions de branchement (le conditions dans le code) jouaient un rôle très important. De plus, la parallélisation est un enjeu important à la fois pour les machines modernes (processeurs SIMD multicœurs, et GPU) et pour les systèmes embarqués.

L'équipe conçoit et implémente des algorithmes d'ECC/ACC efficaces pour CPU et GPU. Ces algorithmes sont basés sur la méthode *Light Speed Labeling* (LSL) qui a pour particularité de manipuler des segments contrairement à la majorité des algorithmes d'étiquetage qui manipulent des pixels. Plusieurs avancées majeures ont été réalisées.

Nous avons encadré une thèse financée par le CERN sur la conception du premier algorithme manipulant des sous-segments sur GPU. Dans la continuité et à l'occasion d'un post-doc dans l'équipe, nous avons proposé un nouvel algorithme : le *FSL-GPU*. Ce dernier est basé sur un vote efficace appliqué à l'étiquetage et l'analyse en composantes connexes (cela est présenté plus en détails dans le portfolio 2). Nous avons également proposé une implémentation sur CPU : le *FSL-CPU*. Cette implémentation tire partie des instructions SIMD pour accélérer le LSL [Lemaitre et al., 2021a]. Enfin, à l'occasion d'une thèse co-encadrée avec un membre de l'équipe DELYS, nous avons développé le *LSL3D-ECC*, une version du LSL pour des scènes en 3 dimensions constituées de voxels. De nombreuses optimisations, liées à la troisième dimension, ont été testées et validées. Cette méthode est entre 1.5$\times$ et 2.5 $\times$ plus rapide que l'état de l'art en ECC et de 2.4$\times$ à 4.4$\times$ plus rapide en ACC [Maurice et al., 2022b].

Enfin, nous avons développé à l'occasion d'une thèse co-encadrée dans l'équipe DLP (*DistanceLess Propagation*), un algorithme pixel direct (en deux itérations) sur GPU [Cabaret et al., 2017]. Une approche similaire a été développée au préalable par [3].

Arithmétique entière et flottante et application à des algorithmes de flot optique

L'équipe a travaillé sur la combinaison d'optimisations de haut niveau (pipeline ou fusion d'opérateurs avec des transformations liées à l'implantation mémoire, non accessibles aux compilateurs) avec des optimisations bas niveau. Une thèse encadrée par l'équipe a été financée dans le cadre de LHCb (Expérience du LHC sur le quark beauté) du CERN. Il s'agissait d'optimiser l'implantation de la factorisation de petites matrices et l'algorithme de Kalman pour faire de la trajectographie temps-réel de particules. La maîtrise fine de la micro-architecture et des techniques d'optimisation ont permis de concevoir des algorithmes 16 fois plus rapides que l'état de l'art (bibliothèque Intel MKL). Les techniques d'optimisation utilisées étaient principalement basées sur le déroulage de boucles avec ré-ordonnancement des instructions (unroll & jam) et l'utilisation des instructions SIMD (SSE, AVX, AVX-512, Neon et SVE) [Lemaitre et al., 2017]. Cela a été réalisé pour des nombres flottants simples et demi précision (F_{32} et F_{16}). Une autre thèse a été financée par le CEA sur l'optimisation d'un code HPC de calcul d'éléments finis spectraux pour le contrôle non destructif.

Dans le cas de systèmes embarqués pour des applications liées à la vision, les transformations de haut niveau ont un impact sur la consommation. Elles sont donc systématiquement appliquées dans les travaux de recherche décrits ci-dessous. Nous avons étudié et optimisé deux algorithmes de flot optique : TVL1 et Horn & Shunck. Ces algorithmes permettent d'estimer un vecteur vitesse pour chaque pixel entre deux images. Ils sont basés sur des motifs réguliers d'accès à la mémoire (stencils). Leur complexité vient du fait qu'ils sont itératifs et manipulent des pyramides d'images (chaque étage de la pyramide représentant la même image à une résolution différente). Le défi est de trouver des compromis entre vitesse d'exécution, qualité numérique de l'estimation de vitesse et consommation énergétique.

Pour notre étude, un banc de mesure expérimental a été assemblé sur fonds propres. De plus, un ingénieur de recherche du laboratoire a conçu une carte permettant, à une fréquence de 5 KHz, de mesurer l'intensité et la tension en entrée d'une carte embarquée. Le code source est instrumenté avec des balises elles mêmes connectés aux broches GPIO de la carte. Des scripts Python permettent d'automatiser la collecte, le traitement et la génération de courbes. Un travail de recherche d'optima locaux a été mis en place sur la carte embarquée Nvidia Jetson. Cette dernière dispose de dizaines de fréquences de fonctionnement possibles pour le CPU, le GPU et la mémoire globale (RAM).

Un premier doctorant en CIFRE DGA encadré par deux membres de l'équipe a travaillé sur l'optimisation de l'algorithme de flot optique TVL1 sur CPU pour du débruitage vidéo temps-réel. Une exploration de l'espace des paramètres pour trouver des compromis entre vitesse et qualité du flot optique sur CPU a été réalisée dans ce contexte [Petreto et al., 2018c]. Ses résultats ont permis de proposer un algorithme temps réel de débruitage vidéo pour des scènes avec une très faible luminosité. Ce travail a été poursuivi par un second doctorant également en CIFRE DGA encadré dans les mêmes conditions et qui a optimisé l'algorithme de flot optique TVL1 sur GPU.

Un autre doctorant financé par une bourse de la région IDF a étudié le déploiement hybride d'un algorithme de flot optique sur CPU et GPU. Cet algorithme est utilisé pour la détection de météores et de débris spatiaux depuis un nano-satellite. La combinaison de transformations de haut niveau avec des compromis sur la qualité fait que, sur une plateforme embarquée Jetson, le code CPU va à la même vitesse que le code GPU [Vaubaillon et al., 2022]. Enfin, nous avons travaillé en collaboration avec l'équipe CIAN sur l'implantation optimisée d'algorithme de flot optique sur FPGA.

Le projet METEORIX soutenu par Sorbonne Université a pour but de détecter les météores dans une séquence vidéo prise depuis l'espace. Ce projet nous a permis de mettre en œuvre nos recherches sur l'arithmétique SIMD, le flot optique, l'étiquetage en composantes connexes et le déploiement automatisé de la chaîne de traitement. Notre participation est décrite dans le portfolio 3.

Un projet à code source ouvert a récemment était mis à disposition de la communauté. *Fast Meteor Detection Toolbox* ou FMDT [1] sont disponibles sur GitHub. FMDT intègre une chaîne de traitement prévue pour des caméras embarquées dans des avions ou dans des ballons atmosphériques. La chaîne a comme particularité de traiter des séquences vidéos pointant vers l'espace. De plus, elle est robuste aux mouvements de la caméra grâce à un algorithme de compensation du mouvement. Cette chaîne est développée en collaboration avec l'observatoire de Paris (IMCCE) avec d'important enjeux scientifiques : être capable de détecter automatiquement (et à bas coût) les météores permet aux astronomes de mieux comprendre la constitution de la terre et du système solaire. L'outil a permis de détecter 100% des météores lors d'une mission menée par des astronomes en Australie [Vaubaillon et al., 2022]. La chaîne de traitement a été implémentée sous la forme d'un graphe *Synchronous DataFlow* puis elle a été parallélisée en multi-thread sur CPU. Les résultats montrent que la chaîne satisfait les contraintes temps réel sur plusieurs cartes embarquées (dont la Raspberry Pi 4). L'ambition de FMDT est de regrouper un ensemble

1. `https://github.com/alsoc/fmdt/`.

d'outils de l'équipe en libre accès. Cela permet de faciliter la reproductibilité des résultats scientifiques et de favoriser les partenariats avec d'autres Universités ou industriels.

Architecture : Conception d'architectures parallèle et leur OS

Conception d'une architecture multi-coeurs et de son OS

En 2017, le CEA LETI a conçu un ordinateur autour d'un processeur TSAR. TSAR est une architecture de processeur manycore conçue dans l'équipe ALSOC avec la bibliothèque de composants SystemC SOClib, précis au cycle et au bit. L'équipe a fourni la description synthétisable en langage VHDL qui a permis au CEA LETI de concevoir deux versions de TSAR, une à 16 cœurs nommée TSARlet et l'autre à 96 cœurs. Les circuits ont été fabriqués par STMicroelectronics en technologie SOI 28nm. Nous avons pu disposer d'un ordinateur avec TSARlet en novembre 2017 et, en 2018, un ingénieur de recherche du LIP6, a pu porter netBSD pour démontrer son bon fonctionnement.

De 2017 à 2021, nous avons également entrepris la réécriture complète du système d'exploitation ALMOS. ALMOS est un système d'exploitation généraliste conçu spécifiquement pour TSAR afin d'étudier les problèmes de passage à l'échelle des services d'un noyau généraliste sur une machine manycore. Il est issu des travaux de thèse d'un ancien doctorant de l'équipe sur le système de fichiers. Le but de cette réécriture est de rendre totalement explicite le partitionnement du noyau en une constellation de noyaux communicants, se partageant par distribution et/ou réplication les structures internes. Cette nouvelle version nommée ALMOS-MKH [2] (Multi-Kernel-Hybride) est publique et utilisée en enseignement en Master 2.

D'autre part, avec le succès de l'assembleur RISC-V, nous avons décidé de faire évoluer SOClib pour simuler cette classe de processeurs. Nous avons ainsi intégré un premier modèle SOClib fourni par le laboratoire TIMA. Nous avons commencé la transition des enseignements d'architecture de Master du processeur MIPS au processeur RISC-V en codant une version du RISC-V pipeliné à 5 étages (RiVer), avec une implémentation sur FPGA et commencé le portage du système d'exploitation léger kO6. En collaboration avec des membres des équipes CIAN et QI, nous avons monté de nouveaux TP [3] de master 2 autour d'un modèle simplifié du RiVer.

Prototypage virtuel

Les travaux autour du prototypage virtuel des systèmes embarqués sur puce se déclinent en deux axes : l'exploration de l'espace de conception multi-niveaux et les activités autour de SystemC AMS. Ces travaux ont tous donné lieux à des implémentations en forme de logiciel libre dans l'outil TTool [4] de Télécom Paris au développement duquel ALSOC participe activement depuis 2013.

De 2017 à 2022, nous avons collaboré avec Ludovic Aprille et Letitia Li (Télécom Paris, équipe LabSoc) sur une méthode de conception d'exploration multi-niveaux de systèmes embarqués sur puce [Genius et al., 2017]. Nous y exposons notre méthode, qui va du partitionnement matériel-logiciel au niveau SysML jusqu'à la génération de code pour la simulation d'un système entier (full-system simulation) autour de SOClib et MutekH. Ce travail a également donné lieu à un chapitre de livre. Nous avons enrichi ce travail par un modèle de latences [Li et al., 2018].

Notre collaboration avec Rodrigo Cortés Porto (Université de Kaiserslautern) et Ludovic Aprille (Télécom Paris, équipe LabSoc) a contribué à faire des avancées majeures dans la recherche des problèmes de causalité dans la simulation des systèmes cyber physiques [Cortés Porto et al., 2021]. Les résultats de ce travail ont ensuite été utilisés dans d'autres travaux, en particulier dans le domaine des systèmes bio-médicaux. Cet article est présenté dans portfolio 5.

Nous avons enfin participé au projet EchOpen, géré pour le LIP6 par des membres de l'équipe CIAN, sur la conception d'un dispositif échographique portable destiné aux pays en développement. Ce projet était l'élément déclencheur de l'extension SystemC AMS de TTool : le but ici est de modéliser le système analogique/numérique mixte [Genius et al., 2020b]. Il s'agissait également de la première application en grandeur réelle modélisée avec cet outil. Elle a ainsi à l'origine divers extensions pour assurer le passage à l'échelle de l'outils.

2. https://www-soc.lip6.fr/trac/almos-mkh.
3. https://github.com/lovisXII/RiVer_SoC.
4. https://ttool.telecom-paris.fr.

Sécurité et durcissement de code : méthode, algorithme et outils

Les activités d'analyse de propriétés de sécurité de programmes assembleurs ou binairse et de renforcement de code menées dans l'équipe depuis 2010 environ se sont fortement développées sur la période 2017-2022. L'équipe s'est principalement concentrée sur l'analyse de sécurité de codes binaires soumis à des attaques physiques ou logicielles et la conception et déploiement de protections, mettant en avant les interactions entre le programme et l'architecture d'exécution sous-jacente. En novembre 2017, un membre de l'équipe a soutenu son HDR, intitulée "Sécurité et performance des applications : analyses et optimisations multi-niveaux".

Analyse et renforcement de programmes soumis à des attaques en fautes

Dans des travaux publiés ultérieurement, nous avions proposé la modélisation de fautes, induites par une impulsion électromagnétique contrôlée, au niveau d'un programme assembleur. Nous avions également posé les bases d'une méthode d'analyse des programmes fautés, permettant de renforcer la robustesse de ces programmes en localisant les vulnérabilités dans le programme et en y ajoutant des instructions redondantes. Ces travaux se sont poursuivis sur la période 2017-2022, dans le cadre du projet collaboratif ANR PROSECCO (2015-2020), d'une thèse CIFRE (2016-2019) puis actuellement du projet ANR COFFI (2019-2023). Dans ce contexte, plusieurs thèses soutenues à Sorbonne Université ont été (co-)dirigées par des membres de l'équipe.

Un premier doctorant co-encadré par plusieurs membres de l'équipe a soutenu sa thèse en 2020 sur la quantification de la sécurité des applications en présence d'attaques physiques et la détection de chemins d'attaques. Dans ces travaux, le modèle de fautes initialement considéré (de type NOP) a été étendu à des modèles corrompant les instructions ou les registres généraux du processeur. Une méthode originale de localisation de vulnérabilités, basée sur une approche d'equivalence-checking, a été proposée, ainsi que plusieurs métriques permettant de caractériser le degré de robustesse d'un code soumis à une ou plusieurs fautes. Ces éléments ont été implantés dans l'outil RobustB, qui, étant donné un code binaire, une région à analyser et une propriété de sécurité à vérifier, établit un diagnostic de robustesse du code. Cela a permis d'analyser différents programmes, et comparer différentes implémentations d'une même fonction [Bréjon et al., 2019].

Un second doctorant, co-encadré par un membre de l'équipe, Albert Cohen (Google AI) et Arnaud de Grandmaison (Arm), a soutenu sa thèse en 2021 sur l'expression et la préservation de propriétés de sécurité d'un code tout au long du flot de compilation. Dans cette thèse, nous avons étudié l'expression de propriétés d'état dans le code source et leur préservation tout au long du flot de compilation, avec optimisation. La solution proposée a été implantée dans l'infrastructure Clang/LLVM et a permis à RobustB d'extraire automatiquement la propriété de sécurité à vérifier. La solution utilisée pour permettre le maintien des éléments en jeu dans la propriété repose sur l'expression d'observations opaques pour le compilateur et de dépendances entre les observations. Nous avons montré que cette expression d'observations permet de préserver des protections de code mises dans le code source, malgré les optimisations du compilateur. Nous avons proposé une version plus efficace, évitant d'ajouter des effets de bord coûteux [Vu et al., 2021]. Ces travaux ont donné lieu à un développement logiciel conséquent, aujourd'hui utilisé dans 2 laboratoires (LCIS et Verimag).

Un troisième doctorant a soutenu sa thèse en 2022, effectuée dans le cadre du projet COFFI, sur les protections assurant l'intégrité du code, du flot de contrôle et des signaux de contrôle dans le processeur. Seules des protections mixtes logicielles-matérielles peuvent assurer ces propriétés (le code n'a pas accès aux signaux internes du processeur). Une solution complète incluant un flot de compilation et un processeur avec modules de protection a été réalisée [Chamelot et al., 2022].

En sus de ces travaux menés dans le cadre de projets, une thèse CIFRE INVIA/Thales co-dirigée avec l'INRIA a étudié la protection d'applications soumises à des attaques en faute à la compilation. Il en a résulté un schéma de protection de boucles, mis en œuvre dans LLVM [Proy et al., 2017] mais aussi une étude des effets de l'injection de fautes sur un processeur complexe (ARES 2019). Les travaux sur ces problématiques d'attaques en faute et d'analyse de vulnérabilité se poursuivent dans la cadre d'une thèse co-encadrée par l'INRIA et Sorbonne Université. Ces travaux visent la mise au point d'une solution purement statique dans l'objectif de prendre en compte des fautes multiples. Une collaboration avec le CEA étudie aussi l'analyse conjointe logiciel/matériel en présence de fautes.

Analyse de programmes soumis à des attaques par canaux auxiliaires

En 2017, nous avons mené une étude sur l'efficacité des méthodes formelles à base de SMT pour l'évaluation de la robustesse de programmes assembleurs sujets à des fuites d'information, dans la continuité des travaux de Eldib et al. [2]. Si ces travaux ont montré la nécessité de mener des analyses au niveau de programmes as-

sembleurs (les compilateurs pouvant réorganiser les codes et rendre visibles des informations qui étaient cachées dans le programme source), les performances obtenues nous ont convaincu que l'approche SMT ne pouvait pas s'appliquer à des programmes de taille significative.

Un premier doctorant co-encadré par plusieurs membres de l'équipe, a soutenu sa thèse en 2021 sur l'analyse de la robustesse et de la sécurisation de codes assembleurs soumis à des attaques par canaux cachés. Nous avons proposé une méthode symbolique originale, basée sur une inférence de type, permettant de caractériser les dépendances d'informations entre différentes variables apparaissant dans les expressions manipulées par un programme assembleur. Cette méthode s'applique à des programmes présentant un flot d'exécution unique, et manipulant des données secrètes masquées (comme par exemple les algorithmes de chiffrement AES). Différents niveaux d'analyse ont été proposés : dépendance au niveau d'un mot opérande, ou identification des différents bits d'un mot opérande. La démarche a été implantée dans l'outil ARISTI-2, et a permis d'analyser des programmes de chiffrement AES masqués complexes [Ben El Ouahma et al., 2019].

Deux projets ANR ont porté cette thématique :

- PROSECCO (2015-2020), durant lequel une thèse co-encadrée par le CEA-LIST et un membre de l'équipe sur le déploiement de protections à la compilation (masquage et polymorphisme) a éé réalisée. Ces travaux ont été décrits dans [Belleville et al., 2020], qui contient notamment la vérification d'un AES masqué avec ARISTI.
- IDROMEL (2021-2025) concerne la prise en compte de la micro-architecture dans l'analyse des fuites.

L'outil ARMISTICE, présenté dans le portfolio 1, est une ré-implantation des concepts initiés précédemment et étendus à des modèles de fuite prenant en considération une implantation micro-architecturale du jeu d'instructions d'un processeur, permettant ainsi de considérer des fuites au travers du chemin de données interne du processeur, qui ne sont pas capturées par le niveau ISA (Instruction Set Architecture).

Sécurisation de plates-formes d'exécution RISC-V intégrant des périphériques malveillants

La sécurisation de plates-formes d'exécution, vis à vis de malveillances logicielles, est également une problématique abordée par l'équipe. Une collaboration avec la société TrustedLabs avait eu cours entre 2014 et 2018 dans le cadre d'une thèse CIFRE (malheureusement non soutenue), sur l'analyse sécuritaire de l'environnement d'exécution sécurisé TrustZone par une approche de simulation concolique (combinaison de simulation symbolique et concrétisation de variables en certains points de la simulation).

La problématique de l'insertion de périphériques, potentiellement malveillants, ayant accès à la mémoire au travers d'un espace virtuel, a été étudiée en collaboration avec Thales dans le cadre d'une seconde thèse CIFRE d'avril 2019 à mars 2023. Cette thèse porte sur la sécurisation des accès aux périphériques et depuis les périphériques dans une architecture multicœurs RISC-V utilisée pour la virtualisation, co-encadré par Daniel Gracia-Perez (Thales) et deux membres de l'équipe. Dans ce travail, nous avons mis en évidence la possibilité de contourner les propriétés de compartimentage des informations mis en place par le mécanisme de mémoire virtuelle, en introduisant un canal caché utilisant une ressource partagée par les périphériques malicieux au sein du composant IOMMU. Nous avons ensuite proposé une modification de ce composant, basée sur un mécanisme de réservation, réduisant les capacités du canal caché, et avons proposé une implantation FPGA du dispositif, montrant la faisabilité de l'approche. La solution retenue a fait l'objet d'un dépôt de brevet en 2021, et Thales a proposé, dans le cadre du consortium RISC-V, une spécification fonctionnelle du composant IOMMU reprenant les principes développés dans ces travaux.

2 INTRODUCTION DU PORTFOLIO

Cette section identifie les éléments de portfolio présentés par l'équipe ALSOC. Chaque élément disposant de sa propre fiche explicative, nous nous contentons ici d'en donner une liste simple :

- *Élément 1 (publication) :* ARMISTICE : Microarchitectural Leakage Modeling for Masked Software Formal Verification, A. de Granmaison, K. Heydemann, Q. Meunier, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2022, 41 (11), pp.3733-3744. ARMISTICE est un outil de vérification formelle pour programmes avec un modèle de fuite en prenant en compte la micro-architecture du processeur ;
- *Élément 2 (publication) :* Taming Voting Algorithms on Gpus for an Efficient Connected Component Analysis Algorithm, F. Lemaitre, A. Hennequin, L. Lacassagne, International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2021. Cette publication présente un nouvel algorithme de vote et l'applique au cas de l'étiquetage et de l'analyse en composantes connexes sur GPU capable de passer à l'échelle sur des architectures de milliers de coeurs, lorsque l'approche classique par *lock-free* et *atomic* devient inefficace ;
- *Élément 3 (Projet ou collaboration) :* MÉTÉORIX est un projet multi-laboratoires de conception d'un nano-satellite au sein de Sorbonne Université. ALSOC est en charge de la conception de la charge utile qui vise à la détection temps-réel de météors et de débris spatiaux avec de fortes contraintes énergétique. Météorix est donc pour ALSOC une plateforme expérimentale d'intégration d'algorithmes et de test d'architectures CPU, GPU, FPGA et hybrides ;
- *Élément 4 (publication) :* A Fixed-Parameter Algorithm for Scheduling Unit Dependent Tasks on Parallel Machines with Time Windows, A. Munier Kordon, Discrete Applied Mathematics, 2020. Cet article présente un algorithme de complexité paramétrée original pour un problème d'ordonnancement classique avec fenêtres de temps ;
- *Élément 5 (publication) :* Handling Causality and Schedulability when Designing and Prototyping Cyber-physical Systems, R. Cortés Porto, D. Genius, L. Apvrille, Software and Systems Modeling, 2021, 20 (1). Cet article présente une méthode itérative pour résoudre les problèmes de causalité en recalculant des ordonnancements des modules TDF tout en proposant des délais adaptés.

3 AUTOÉVALUATION DU BILAN

3.1 Autoévaluation de l'équipe

Domaine 2. Attractivité

Référence 1. L'unité est attractive par son rayonnement scientifique et s'insère dans l'espace européen de la recherche.

Séminaires Invités.

- Séminaire invité au Collège de France le 7 avril 2022 dans le cadre du cours de Xavier Leroy, professeur titulaire de la chaire intitulée "Sciences du Logiciel". Le cours de cette année porte sur la sécurité du logiciel ;
- Exposé invité au Colloque "Architecture : hier, aujourd'hui, demain ". Toulouse, le 3 juillet 2018 ;
- Exposé invité au Scheduling Seminar dématérialisé le 28 septembre 2022 sur "Synchronous Dataflow : a survival guide " ;
- Tutoriel à IEEE HOST en 2018 ;
- Tutoriel à FDL 2019.

Program Chair et comité de programme.

- Présidence du track Architecture à la conférence COMPAS 2019 ;
- Co-program chair du workshop Security Proofs for Embedded Systems, joint à la conférence CHES, en 2019 ;
- Co-program chair du Workshop Cryptograghy and Security in Computing Systems, joint à la conférence HIPEAC, en 2019.

D'autre part, des membres de l'équipe ont participé à de nombreux comités de programme : Workshop Security of Software/Hardware Interface (SILM), joint à la conférence Euro S&P, depuis 2020 ; Conférence COMPAS, 2020 ; Workshop Software Protection (SPRO), joint à la conférence ACM CCS, 2019 ; Workshop Cryptograghy and Security in Computing Systems (CS2), joint à la conférence HIPEAC, 2018 ; Workshop Security Proofs for Embedded Systems (PROOFS), joint à la conférence CHES, 2017 ; Workshop on the Reliability of Intelligent Environments (WoRIE), 2017 ; DSD 2019 et 2020.

Relations avec les GDR du CNRS. Les membres de l'équipe sont activement impliqués dans les activités d'organisation des deux Groupements de Recherche (GDR) du CNRS.

Pour le GDR Sécurité :

- Co-organisation de la journée de travail Méthodes Formelles et sécurité en janvier 2020 ;
- Membre du comité d'organisation des journées sur les attaques par injection de fautes JAIF.

Pour le GDR RO :

- Co-responsabilité du groupe de travail optimisation pour la conception des systèmes intégrés (une réunion par an au LIP6 et une réunion hors Ile-de-France).
- Co-animation de l'axe Ordonnancement, Planification et Ordonnancement depuis 2020 ;
- Co-organisation de l'Ecole Jeunes Chercheur 2021 du GDR au LIP6 sur le thème Ordonnancement, Planification et Ordonnancement.

Activités Editoriales.

- Membre du comité éditorial de la revue IEEE Transaction on Dependable and Secure Computing d'avril 2019 à mai 2021 ;
- Membre du bureau éditorial de la revue 4OR depuis sa création en 2003 ;
- Membre du bureau éditorial de la revue Journal of Scheduling depuis 2020.

Prix et distinctions.

- Prix du meilleur papier à la conférence MODELSWARD 2017.

Autres.

- Expertise HCERES pour l'évaluation du LIMOS et du LIP en 2020, et du Lab-STICC et de l'IRISA en 2021 ;
- Membre de commission d'expertise de déroulement de projet FILOG-2 combinant architectures hybrides et sécurité.

Référence 2. L'unité est attractive par la qualité de sa politique d'accompagnement des personnels.

L'équipe accompagne au mieux les stagiaires, doctorants, post-doc et ingénieurs intégrés dans ses projets de recherche, ainsi que les nouveaux enseignants-chercheurs :

- Soutien sur ressources propres pour permettre aux doctorants de terminer leur thèse dans de bonnes conditions ;
- Incitation à la publication ou au dépôt de brevet. Les doctorants apparaissent en premier auteur et vont présenter leurs travaux dans des journées et conférences ;
- Incitation à participer aux écoles d'été et journées thématiques des GDR du domaine.

Référence 3. L'unité est attractive par la reconnaissance de ses succès à des appels à projets compétitifs.

L'équipe coordonne ou participe à plusieurs projets de recherche :

- Responsabilité du projet ANR COFFI 2019-2023 pour le LIP6, avec l'EMSE, CEA, INVIA/Thales ;
- Coordination du projet ANR PROSECCO 2015-2020 avec le CEA LIST (SAS et DACLE) ;
- Responsabilité du projet ANR IDROMEL 2021-2025 pour le LIP6, avec le CEA, l'IRISA, l'IRIT et la société ARM ;
- Participation au projet AP-HP EchOpen en collaboration avec des membres de l'équipe CIAN.

Référence 4. L'unité est attractive par la qualité de ses équipements et de ses compétences techniques.

L'équipe dispose de savoir-faire utilisés par nos communautés :

- La version de l'infrastructure de compilation LLVM développée par un doctorant de l'équipe est utilisée actuellement par Marie Laure Potet UGA, Verimag et un doctorant de LCIS de Laure Gonnord. Cette version inclut une partie des développements réalisées par un autre doctorant de l'équipe sur le déploiement de protection à la compilation. Disponible sur le gitlab du LIP6 ;
- L'infrastructure développée avec Arm dans le cadre du projet IDROMEL est utilisée par tous les partenaires du projet (CEA, IRIT, IRISA) ainsi que deux doctorants en CIFRE au LIP6 dans l'équipe ALMASTY (CryptoExpert et Thales) ;
- L'équipe a participé au développement de l'outil TTool, en fournissant une extension destinée au prototypage des systèmes analogiques-numériques mixtes (TTool-AMS) ;

Domaine 3. Production scientifique

	2017	2018	2019	2020	2021	2022
Articles (revues)	0.57	0.42	0.42	0.71	0.54	0.50
Communications (conférences)	1.57	2.42	2.42	1.85	2.72	1.83

TABLE 2 – Publications par ETPR par an entre 2017 et 2022

Référence 1. La production scientifique de l'unité satisfait à des critères de qualité.

L'équipe privilégie la qualité des résultats publiés à la quantité. Les conférences sont choisies avec des difficultés progressives de sorte à permettre aux doctorants d'avoir une ou deux publications moyennement sélectives, avant de tenter des conférences plus sélectives.

Référence 2. La production scientifique de l'unité est proportionnée à son potentiel de recherche et correctement répartie entre ses personnels.

L'activité de publication est mal répartie sur l'ensemble des membres de l'équipe. Durant cette période, plusieurs membres de l'équipe ont arrêté ou ralenti considérablement leur activité de recherche du fait de leurs responsabilités administratives importantes au sein de Sorbonne Université. Citons :

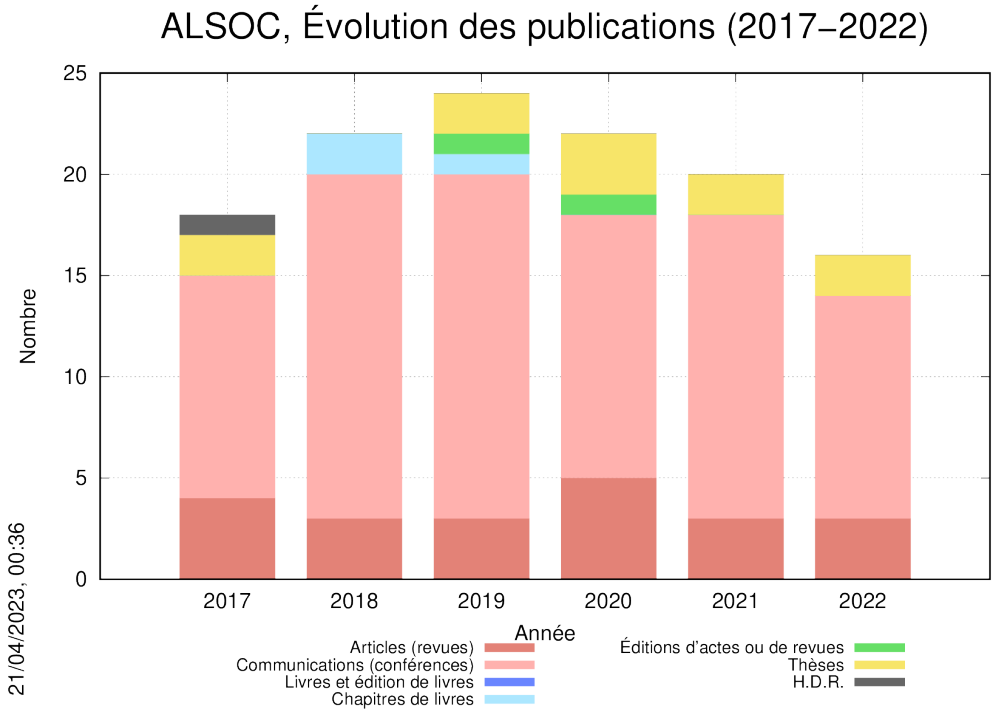


FIGURE 1 – Évolution des publications entre 2017 et 2022

- Présidence de l'Université depuis 2021 (précédemment, vice-présidence de l'Université chargée de l'insertion de la formation et de l'insertion professionnelle de 2016 à 2017, puis vice-présidence Recherche, Innovation et Science ouverte de 2018 à 2021) ;
- Direction de l'UFR d'Ingénierie de décembre 2017 à décembre 2022 ;
- Responsabilité de la licence d'Informatique sur la période évaluée.

D'autre part, plusieurs membres de l'équipe n'ont plus aucune activité de recherche, ni aucun investissement dans le collectif.

Référence 3. La production scientifique de l'unité respecte les principes de l'intégrité scientifique, de l'éthique et de la science ouverte. Elle est conforme aux directives applicables dans ce domaine.

L'équipe développe une politique active de science ouverte : nos publications sont intégrées dans HAL et plus récemment IACR (thématique sécurité) ; les développements logiciels sont déposés sur GitHub avec la création d'une organisation ALSOC sur GitHub [5].

Domaine 4. Inscription des activités de recherche dans la société

Référence 1. L'unité se distingue par la qualité et la quantité de ses interactions avec le monde non-académique.

L'équipe développe de nombreuses interactions suivies avec le monde non-académique, au travers de collaborations avec différentes entreprises au sein de projets (ANR notamment), et de thèses CIFRE : CEA, ARM, Thalès, DGA, STMicroelectronics, Ministère des Armées.

Référence 2. L'unité développe des produits à destination du monde culturel, économique et social.

L'équipe contribue à des dépôts de brevets :

5. https://github.com/alsoc/.

- R. Ducousso, D. Gracia-Perez, E. Encrenaz, Q. Meunier : “Système sur Puce comprenant au moins une IOMMU sécurisée”, dépot num. 2114333, décembre 2021 par Thalès, Sorbonne Université et CNRS.
- D. Couroussé, K. Heydemann, T. Barry : “Procédé d’exécution d’un code machine d’une fonction sécurisée”. FR20170053175, WO2018FR50678, 2017.

D’autre part, les travaux de thèse d’un doctorant de l’équipe ont conduit à une proposition de spécification d’une IOMMU au sein du consortium RISC-V (proposition portée par Thalès).

Référence 3. L’unité partage ses connaissances avec le grand public et intervient dans des débats de société.

- Plusieurs membres de l’équipe se sont investis dans les enseignements d’algorithmique et d’architecture dans les vagues du DIU “Enseigner l’Informatique au Lycée” (2019-2021) qui a contribué à la formation d’une centaine d’enseignants, intervenant en lycée, dans la nouvelle spécialité NSI ;
- Plusieurs membres de l’équipe sont actifs dans l’enseignement de l’architecture dans la préparation à l’agrégation d’informatique qui s’est montée en 2021 ;
- Présidence du jury de l’agrégation des Sciences de l’Ingénieur, Option Ingénierie Informatique depuis 2021, participation à ce jury depuis 2019 ;
- Membre du jury de l’agrégation d’Informatique depuis 2021 ;
- Membre du jury du CAPES NSI en 2020-2021 ;
- Participation à la commission d’élaboration du programme de l’agrégation d’Informatique en 2021.

4 RÉFÉRENCES BIBLIOGRAPHIQUES EXTERNES

[1] Karol Desnos, Maxime Pelcat, Jean-François Nezan, and Slaheddine Aridhi. Memory bounds for the distributed execution of a hierarchical synchronous data-flow graph. In *2012 International Conference on Embedded Computer Systems : Architectures, Modeling, and Simulation, SAMOS XII, Samos, Greece, July 16-19, 2012*, pages 160–167, 2012.

[2] Hassan Eldib, Chao Wang, and Patrick Schaumont. Smt-based verification of software countermeasures against side-channel attacks. In Erika Ábrahám and Klaus Havelund, editors, *Tools and Algorithms for the Construction and Analysis of Systems*, pages 62–77, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.

[3] Y. Komura. Gpu-based cluster-labeling algorithm without the use of conventional iteration : application to swendsen-wang multi-cluster spin flip algorithm. *Computer Physics Communications*, pages 54–58, 2015.

[4] Edward A. Lee and David G. Messerschmitt. Synchronous data flow. *Proceeding of the IEEE*, vol. 75(no. 9) :pp. 1235–1245, 1987.

[5] C. L. Liu and James W. Layland. Scheduling algorithms for multiprogramming in a hard-real-time environment. *J. ACM*, 20(1) :46–61, 1973.

[6] Matthias Mnich and René van Bevern. Parameterized complexity of machine scheduling : 15 open problems. *Computers and Operations Research*, 100 :254 – 261, 2018.

5 RÉFÉRENCES BIBLIOGRAPHIQUES SIGNIFICATIVES DE ALSOC

[Aba et al., 2019] Aba, M. A., Munier Kordon, A., and Pallez, G. (2019). Scheduling on Two Unbounded Resources with Communication Costs. In *Euro-Par - European Conference on Parallel Processing*, Gottingen, Germany.

[Belleville et al., 2020] Belleville, N., Couroussé, D., Heydemann, K., Meunier, Q., and Ben El Ouahma, I. (2020). Maskara : Compilation of a Masking Countermeasure with Optimised Polynomial Interpolation. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 39(11) :1–1.

[Ben El Ouahma et al., 2019] Ben El Ouahma, I., Meunier, Q., Heydemann, K., and Encrenaz, E. (2019). Side-channel robustness analysis of masked assembly codes using a symbolic approach. *Journal of Cryptographic Engineering*, pages 1–12.

[Bréjon et al., 2019] Bréjon, J.-B., Heydemann, K., Encrenaz, E., Meunier, Q. L., and Vu, S. T. (2019). Fault attack vulnerability assessment of binary code. In *Cryptography and Security in Computing Systems (CS2'19)*, pages 13–18, Valencia, Spain. ACM.

[Cabaret et al., 2017] Cabaret, L., Lacassagne, L., and Etiemble, D. (2017). Distanceless Label Propagation : an Efficient Direct Connected Component Labeling Algorithm for GPUs. In *IPTA2017 - International Conference on Image Processing Theory, Tools and Applications*, Montreal, Canada.

[Chamelot et al., 2022] Chamelot, T., Courousse, D., and Heydemann, K. (2022). SCI-FI : Control Signal, Code, and Control Flow Integrity against Fault Injection Attacks. In *2022 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 556–559, Antwerp, Belgium. IEEE.

[Cortés Porto et al., 2021] Cortés Porto, R., Genius, D., and Apvrille, L. (2021). Handling causality and schedulability when designing and prototyping cyber-physical systems. *Software and Systems Modeling*, 20(1).

[Deroui et al., 2017a] Deroui, H., Desnos, K., Nezan, J.-F., and Munier-Kordon, A. (2017a). Relaxed Subgraph Execution Model for the Throughput Evaluation of IBSDF Graphs. In *International Conference on Embedded Computer Systems : Architectures, Modeling, and Simulation (SAMOS)*, SAMOS, Greece.

[Genius et al., 2020b] Genius, D., Bournias, I., Apvrille, L., and Chotin, R. (2020b). Model-Based Virtual Prototyping of CPS : Application to Bio-Medical Devices. In *MODELSWARD 2020 : Model-Driven Engineering and Software Development*, volume 1361 of *CCIS*, pages 74–96, Valletta, Malta. Springer, Cham.

[Genius et al., 2017] Genius, D., Li, L., and Apvrille, L. (2017). Model-Driven Performance Evaluation and Formal Verification for Multi-level Embedded System Design. In *5th International Conference on Model-Driven Engineering and Software Development (MODELSWARD 2017)*, Porto, Portugal. INSTICC. (best paper award).

[Lemaitre et al., 2017] Lemaitre, F., Couturier, B., and Lacassagne, L. (2017). Cholesky Factorization on SIMD multi-core architectures. *Journal of Systems Architecture*.

[Lemaitre et al., 2021a] Lemaitre, F., Hennequin, A., and Lacassagne, L. (2021a). Taming Voting Algorithms on Gpus for an Efficient Connected Component Analysis Algorithm. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7903–7907, Toronto, Canada. IEEE.

[Li et al., 2018] Li, L. W., Genius, D., and Apvrille, L. (2018). Formal and Virtual Multi-level Design Space Exploration. In *International Conference on Model-Driven Engineering and Software Development.*, pages 47–71. Springer.

[Mallem et al., 2022a] Mallem, M., Hanen, C. C., and Munier Kordon, A. (2022a). Parameterized complexity of a parallel machine scheduling problem. In *International Symposium on Parameterized and Exact Computation (IPEC)*, Postdam, Germany.

[Maurice et al., 2022b] Maurice, N., Lemaitre, F., Sopena, J., and Lacassagne, L. (2022b). LSL3D : a run-based Connected Component Labeling algorithm for 3D volumes. In *Binary is the new Black and White workshop @ IEEE ICIAP 2022*, Lecce, Italy.

[Munier Kordon, 2020] Munier Kordon, A. (2020). A Fixed-Parameter Algorithm for Scheduling Unit Dependent Tasks on Parallel Machines with Time Windows. *Discrete Applied Mathematics*.

[Munier Kordon and Tang, 2020] Munier Kordon, A. and Tang, N. (2020). Evaluation of the Age Latency of a Real-Time Communicating System using the LET paradigm. In Voelp, M., editor, *ECRTS 2020*, volume 165 of *Leibniz International Proceedings in Informatics (LIPIcs)*, Modena, Italy. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[Petreto et al., 2018c] Petreto, A., Hennequin, A., Koehler, T., Romera, T., Fargeix, Y., Gaillard, B., Bouyer, M., Meunier, Q., and Lacassagne, L. (2018c). Energy and Execution Time Comparison of Optical Flow Algorithms on SIMD and GPU Architectures. In *Conference on Design and Architectures for Signal and Image Processing (Dasip 2018)*, Porto, Portugal.

[Proy et al., 2017] Proy, J., Heydemann, K., Berzati, A., and Cohen, A. (2017). Compiler-Assisted Loop Hardening Against Fault Attacks. *ACM Transactions on Architecture and Code Optimization*, 14(4) :36.

[Vaubaillon et al., 2022] Vaubaillon, J., Loir, C., Millet, M., Ciocan, C., Kandeepan, M., Cassagne, A., Lacassagne, L., da Fonseca, P., Zander, F., Buttsworth, D., Loehle, S., Tóth, J., Gray-Owen, S. D., Moingeon, A., and Rambaux, N. (2022). A 2022 τ-Herculids meteor cluster. In *Internationational Meteor Conference (IMC) 2022*, Poroszlo, Hungary. Research Centre for Astronomy and Earth Sciences (CSFK) and Konkoly Thege Astronomical Institut.

[Vu et al., 2021] Vu, S. T., Cohen, A., de Grandmaison, A., Guillon, C., and Heydemann, K. (2021). Reconciling optimization with secure compilation. *Proceedings of the ACM on Programming Languages*, 5(OOPSLA) :1–30.

A ANNEXE — MEMBRES PERMANENTS AU 31/12/2022

La table ci dessous liste les membres permanents de l'équipe ALSOC.

NOM	Prénom	Corps	Employeur
BRAUNSTEIN	Cécile	MCF	Sorbonne Université
CASSAGNE	Adrien	MCF	Sorbonne Université
DESBARBIEUX	Jean-Lou	MCF	Sorbonne Université
DRACH-TEMAM	Nathalie	PR	Sorbonne Université
ENCRENAZ	Emmanuelle	MCF (HDR)	Sorbonne Université
GENIUS	Daniela	MCF	Sorbonne Université
HEYDEMANN	Karine	MCF (HDR)	Sorbonne Université
LACASSAGNE	Lionel	PR	Sorbonne Université
MARCHETTI	Olivier	MCF	Sorbonne Université
MEUNIER	Quentin	MCF	Sorbonne Université
MUNIER	Alix	PR	Sorbonne Université
WAJSBÜRT	Franck	MCF	Sorbonne Université

ÉLÉMENT DE PORTFOLIO 01

Publication

1 DÉFINITION DE CET ÉLÉMENT

Titre de l'élément : ARMISTICE : Modélisation de fuites micro-architecturales pour la vérification formelle de programmes masqués [1]

URL de l'élément : `https://hal.sorbonne-universite.fr/hal-03954892`

2 MOTIVATIONS DU CHOIX DE CET ÉLÉMENT

Armistice constitue une avancée significative dans le domaine de la vérification de masquage, car c'est le premier travail qui vérifie de manière formelle les programmes avec un modèle de fuite en prenant en compte la micro-architecture du processeur. Ce travail combine des aspects à la fois théoriques et pratiques, allant de la simulation de code sur un coeur à la vérification formelle d'expressions symboliques, en passant par la conception de programmes de tests pour la caractérisation physique des fuites sur un processeur.

3 PRÉSENTATION DE CET ÉLÉMENT

Les attaques par canal auxiliaire (SCA) exploitent les mesures physiques, comme la consommation d'énergie ou les émissions électromagnétiques (EM), pendant l'exécution d'une application pour récupérer des données secrètes. Elles permettent de casser des implémentations d'algorithmes cryptographiques prouvés algorithmiquement sûrs.

Une contre-mesure à ces attaques est le masquage, qui vise à coder une donnée secrète en $d+1$ parties appelées *shares*, de telle sorte que toute combinaison de moins de $d+1$ parts soit statistiquement indépendante du secret. Cela empêche théoriquement les SCA, car la consommation d'énergie et les émissions EM sont directement liées aux valeurs manipulées par le programme.

La mise en œuvre d'un schéma de masquage au niveau logiciel sans démasquer les secrets et la détection manuelle de ce démasquage sont loin d'être des tâches simples. Par conséquent, certaines techniques et outils de vérification ont été récemment proposés pour aider les concepteurs à détecter les failles dans leurs implémentations.

Avec Armistice, nous montrons que les outils et modèles de fuite actuels sont insuffisants pour garantir une sécurité pratique. À titre d'exemple, la figure 1 montre un code du AND masqué selon un schéma classique (appelé ISW), ainsi que le code assembleur correspondant généré. Ce dernier ne comporte pas de fuite avec un modèle prenant en compte les registres généraux du processeur (vue ISA). La partie droite de la figure illustre que malgré cela, la consommation est fortement corrélée avec les valeurs secrètes (entrées et sortie). Cela est dû au fait que le modèle de fuite considéré n'est pas assez proche du matériel.

Dans Armistice, nous avons analysé le code Verilog d'un processeur Arm Cortex-M3 afin d'en extraire un chemin de données et d'en faire un modèle qui serve de base à la modélisation de la consommation. Dans ce but, nous avons réalisé de nombreux "test vectors" de fuite, dont le but était triple : 1) Confirmer que les valeurs qui transitent dans les éléments matériels modélisés du cœur ont un effet visible sur la consommation ; 2) Déduire un modèle matériel de la mémoire pour laquelle nous n'avions pas accès au RTL ; 3) Déterminer pour chaque composant matériel modélisé le nombre de traces requises pour caractériser une fuite.

Tous ces tests vectors et les résultats associés sont disponibles à l'URL suivante : `https://www-soc.lip6.fr/armistice/`

Une fois le modèle du processeur implanté, nous l'avons associé à un outil de vérification d'expressions symboliques (figure 2). Le modèle du cœur simule l'exécution du code assembleur cycle par cycle, générant ainsi des expressions symboliques masquées dans les différents éléments modélisés du cœur (registres, bus). Ces expressions sont ensuite vérifiées par l'outil LeakageVerif [2], qui analyse ces expressions à la recherche de fuite secrète. En cas de fuite, l'outil est donc capable de déterminer le cycle de la fuite, l'élément matériel impliqué, ainsi que

Listing 1 – ET logique selon le schéma ISW

```
// Inputs: - Secrets a = a0 ^ a1, b = b0 ^ b1
//         - Mask m
// Output: - Secret c = c0 ^ c1
aux0 = m ^ (a0 & b1);
aux1 = aux0 ^ (a1 & b0);
c0 = (a0 & b0) ^ m;
c1 = (a1 & b1) ^ aux1;
```

Listing 2 – Code assembleur produit par GCC

```
; r0:a0, r1:b0, r2:a1, r3:b1, r6:c[] r7:m
and.w    r4, r0, r3    ; a0 & b1
eors     r4, r7        ; aux0 = (a0 & b1) ^ m
and.w    r5, r2, r1    ; a1 & b0
ands     r0, r1        ; a0 & b0
ands     r3, r2        ; b1 & a1
eors     r4, r5        ; aux1 = aux0 ^ (a1 & b0)
eors     r0, r7        ; c0 = (a0 & b0) ^ m
eors     r4, r3        ; c1 = aux1 ^(a1 & b1)
str r0, [r6, #0]
str r4, [r6, #4]
```

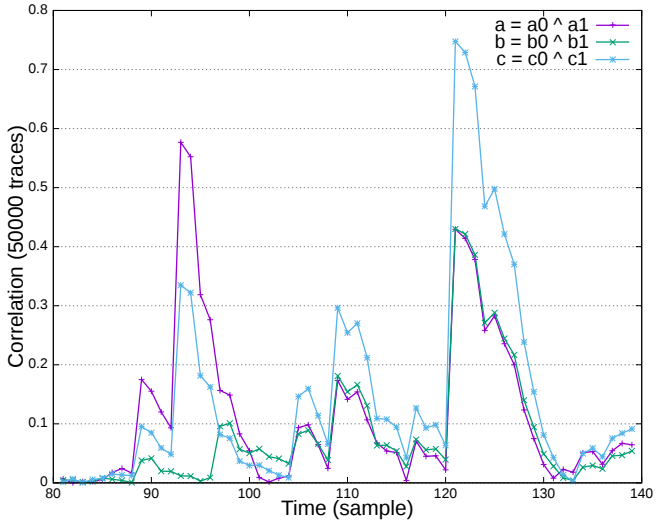
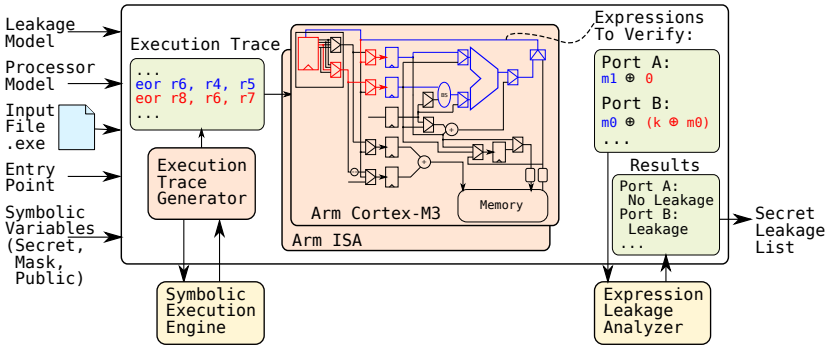


FIGURE 1 – ET logique masqué selon le schéma ISW et prenant en compte les registres généraux du processeur ; Consommation associée à l'exécution du code



Symbolic Expressions in Hardware Components

ISA Model

Initial	
r4	m1
r5	m0
r6	0
r7	k⊕m0
r8	0

Final	
r6	m0⊕m1
r8	k⊕m1

Subset of the Cortex-M3 Model

Cycle	0	1	2	3	4
r4	m1	m1	m1	m1	m1
r5	m0	m0	m0	m0	m0
r6	0	0	0	m0⊕m1	m0⊕m1
r7	k⊕m0	k⊕m0	k⊕m0	k⊕m0	k⊕m0
r8	0	0	0	0	k⊕m1
Port A	-	m1	0	...	...
Port B	-	m0	k⊕m0	...	...

Leakage in transition on Port B

FIGURE 2 – Framework de vérification Armistice

l'expression symbolique concernée. Ces informations permettent de corriger le code afin de supprimer les fuites. De nombreux codes ont été vérifiés avec Armistice, tous exhibant des fuites au niveau micro-architectural, et la suppression des fuites par ajout d'instructions assembleur spécialement conçues d'après la sortie de l'outil a été faite pour une des applications.

4 RÉFÉRENCES BIBLIOGRAPHIQUES

[1] Arnaud De Grandmaison, Karine Heydemann, and Quentin L. Meunier. ARMISTICE : Microarchitectural Leakage Modeling for Masked Software Formal Verification. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 41(11) :3733–3744, November 2022.

[2] Quentin L Meunier, Etienne Pons, and Karine Heydemann. Leakageverif : Efficient and scalable formal verification of leakage in symbolic expressions. *IEEE Transactions on Software Engineering*, 2023.

ÉLÉMENT DE PORTFOLIO 02

Publication

1 DÉFINITION DE CET ÉLÉMENT

Titre de l'élément : Taming Voting Algorithms on GPUs for an Efficient Connected Component Analysis Algorithm [6]

URL de l'élément : `https://hal.archives-ouvertes.fr/hal-03330414`

2 MOTIVATIONS DU CHOIX DE CET ÉLÉMENT

Avec l'augmentation sans cesse croissante du nombre de cœurs des processeurs GPU, même la programmation *lock-free* à base d'instructions `atomic` est devenue inefficace. Cet article présente un algorithme de vote astucieux appliqué à l'étiquetage (ECC) et l'analyse en composantes connexes (ACC) sur GPU.

3 HISTORIQUE DES ALGORITHMES D'ÉTIQUETAGE ET D'ANALYSE EN COMPOSANTES CONNEXES

Les algorithmes d'ECC font partie des problèmes mal posés. Ils associent une étiquette unique à chaque groupe de pixels connexes d'une image binaire via un opérateur de voisinage qui détecte l'adjacence entre pixels. Les algorithmes modernes d'ECC et d'ACC dérivent tous de deux algorithmes pionniers : celui de Rosenfeld [8] qui est direct (en deux passes) mais qui a besoin d'une table d'équivalence et celui de Haralick [3] sans table d'équivalence, mais qui est itératif. La table d'équivalence est une structure Union-Find. Assigner une étiquette à une composante connexe revient donc à réaliser la fermeture transitive du graphe associé. Ce qui est rapide car la table d'équivalence possède en plus une relation d'ordre.

Si rapidement les algorithmes pour CPU se sont basés sur Rosenfeld, les premiers algorithmes pour GPU étaient basés sur Haralick car plus simples à mettre en œuvre. De plus la mémoire Shared permettait de faire des paquets d'itérations rapides. Mais rapide ne signifie pas efficace. Les principales étapes sont les suivantes :

- 1) Avoir l'idée de "plonger" la table d'équivalence dans l'image (aussi appelée notation linéaire) et 2) inventer un nouvel algorithme union-find *lock-free* (itération d'`atomic_min` jusqu'à a la stabilité). Cela fut co-inventé par Cabaret [2] et Komura [5].
- Playne et Hawick [7] ont inventé un algorithme énumérant les cas amenant à une équivalence entre étiquettes de ceux ne nécessitant que la propagation de l'étiquette courante. Cet algorithme générait beaucoup moins d'accès mémoire et était plus rapide. Les algorithmes d'ECC sur GPU venaient de changer de *paradigme* : il valait mieux faire beaucoup de tests – et donc de provoquer beaucoup de divergences de threads au sein d'un warp – que de faire des accès mémoire, même si ces cas sont rares et que les instructions atomic sont rapides.
- La seconde amélioration vient d'un doctorant de l'équipe [4] qui s'est inspiré de l'approche segment du LSL. Son algorithme HA manipule des sous-segments de la taille d'un warp (32). Tous les threads d'un warp déterminent leur position dans les sous-segments courants grâce à des intrinsics matériels. Et seul le premier thread de chaque sous-segment réalise un accès mémoire pour mettre à jour la table d'équivalence (ECC) ou pour voter (ACC). Le calcul des descripteurs de segment est aussi fortement accéléré par cette première approche segment.
- Alegretti et al. [1] ont aussi porté leurs algorithmes à base d'arbre de décision sur GPU. Et bien que l'arbre soit très grand, il est performant. Par contre il ne fait que de l'ECC et pas d'ACC.

La dernière évolution est le FLSL-GPU.

4 PRÉSENTATION DU FLSL-GPU

Lorsqu'on regarde l'évolution de la performance de ces différents algorithmes, on observe sur des images aléatoires que le pire cas empirique se situe au seuil de percolation (40% dans le cas 8-connexe) et que ce pire cas dégénère avec l'augmentation du nombre de coeurs. Il était à peine observable lors du passage d'une Jetson TX2 (256 CUDA cores) à une Jetson AGX (512 CUDA cores), mais était très problématique sur les grosses cartes massivement parallèles.

Le FLSL-GPU résout ce problème grâce à deux nouvelles avancées :

1. le traitement de segments complets (comme le LSL pour CPU) et,
2. la détection de conflits (CD), lorsque plusieurs threads veulent mettre à jour la même étiquette.

Ainsi, l'algorithme naif est en permanence inefficace avec 0.9 Gpixel/s. HA passe d'un débit de 4.22 Gpixel/s pour une granularité $g = 1$ à 25.8 Gpixel/s pour $g = 16$. Le FLSL+CD passe d'un débit de 24.5 à 170 Gpixel/s pour $g = 16$. LE FLSL-GPU est ainsi de $\times 4$ à $\times 10$ plus rapide que HA en ACC. A notre connaissance, il n'y a pas d'algorithme d'ACC sur GPU à part ceux de l'équipe.

5 RÉFÉRENCES BIBLIOGRAPHIQUES

[1] Stefano Allegretti, Federico Bolelli, and Costantino Grana. Optimized Block-Based Algorithms to Label Connected Components on GPUs. *IEEE Transactions on Parallel and Distributed Systems*, 2019.

[2] L. Cabaret, L. Lacassagne, and D. Etiemble. Distanceless label propagation : an efficient direct connected component labeling algorithm for GPUs. In *International GPU Technical Conference (GTC)*, 2017.

[3] R.M. Haralick. Some neighborhood operations. In *Real-Time Parallel Computing Image Analysis*, pages 11–35. Plenum Press, 1981.

[4] A. Hennequin and L. Lacassagne. A new direct connected component labeling and analysis algorithm for GPUs. In *GPU Technology Conference (GTC)*, 2019.

[5] Y. Komura. Gpu-based cluster-labeling algorithm without the use of conventional iteration : application to swendsen-wang multi-cluster spin flip algorithm. *Computer Physics Communications*, pages 54–58, 2015.

[6] F. Lemaitre, A. Hennequin, and L. Lacassagne. Taming voting algorithms on GPUs for an efficient Connected Component Analysis Algorithm. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021.

[7] D. P. Playne and K. Hawick. A new algorithm for parallel connected-component labelling on GPUs. *IEEE Transactions on Parallel and Distributed Systems*, 2018.

[8] A. Rosenfeld and J.L. Platz. Sequential operator in digital pictures processing. *Journal of ACM*, 13,4 :471–494, 1966.

ÉLÉMENT DE PORTFOLIO 03

Projet ou collaboration

1 DÉFINITION DE CET ÉLÉMENT

Titre de l'élément : Projet multi-laboratoires METEORIX

URL de l'élément : `http://www.nanosat.upmc.fr/fr/projet-nanosatellite-meteorix.html`

2 MOTIVATIONS DU CHOIX DE CET ÉLÉMENT

METEORIX est le projet de nano-satellite soutenu par Sorbonne Université et 3 UFR dont le but est de détecter des météores et des débris spatiaux par traitement d'images en temps réel depuis un satellite. A ce jour, c'est le seul projet de ce type. Par extension METEORIX désigne – pour le LIP6 – la conception d'une charge utile embarquable, dans un satellite, un ballon sonde ou un avion d'observation. Ces travaux se font avec l'équipe CIAN du LIP6 et l'Institut de Mécanique Céleste et du Calcul des Ephémérides (IMCCE) à l'Observatoire de Paris.

3 PRÉSENTATION DE CET ÉLÉMENT

Ce projet comporte à la fois des objectifs scientifiques astronomiques – déterminer le flux de météores et de débris spatiaux qui entrent dans l'atmosphère terrestre – un objectif pédagogique – impliquer les étudiants dans la conception, la réalisation et l'exploitation d'une mission spatiale via le programme Nanolab Academy du CNES – mais aussi un objectif scientifique majeur : concevoir différentes chaînes de traitement adaptables en complexité pour répondre à différentes contraintes de vitesse de traitement et de consommation (dépendant du porteur). Ce projet permet d'utiliser nos recherches sur arithmétique SIMD, le flot optique, l'étiquetage en composantes connexes et le déploiement automatisé de la chaîne de traitement.

Cela a donné lieu à plusieurs chaînes de traitements :

- Projet 0 : le projet Meteor de l'Université de Chiba (Japon) a déployé à bord de l'ISS une caméra et un PC pour faire la détection, mais c'est un échec : les vidéos sont descendues sur Terre et traitées manuellement par des volontaires.
- Projet 1 : conception d'une chaîne de traitement à base de flot optique (les algorithmes à base de fond fixe ne peuvent être utilisés car, la caméra et le fond sont en mouvement) et d'étiquetage en composantes connexes ainsi qu'un banc de validation automatique pour tester différentes combinaisons et paramétrisation d'algorithmes. C'est un succès avec plus de 95% de météores détectés sur l'ensemble des vidéos.
- Projet 2 : conception d'une seconde chaine de traitement pour traiter les vidéos d'un ballon sonde espagnol (Géminids 2016). La difficulté est que le plateforme n'est pas stabilisée : au roulis et au tangage s'ajoute parfois un angle de lacet très fort. Succès avec 85% de détection sur 31,400 frames.
- Projet 3 : adaptation du projet 2 pour les tau-Herculedids 2022. Projet international : 4 pays embarquent à bord d'un jet privé une dizaine de caméras, certaines stabilisées, d'autres pas. Résultats encourageants : 68% et 78% de détection sur 17,200 et 52,500 frames des séquences françaises et australiennes. Les contraintes sont de plus respectées : 6.5 W pour une Jetson Nano et 111 fps, 3 W pour une Raspberry Pi4 et 44 fps. Mais plus important, cela a permis la détection d'un cluster de 34 météores où la machine a fait mieux que l'humain entraîné.

Cela a donné lieu a un dépot open-source git (version non optimisée) et la création d'un framework de test FMDT (*Fast Meteor Detection Toolbox*).

Ces travaux impliquent la participation de 5 doctorants et anciens doctorants de l'équipe sur le flot optique F_{32} et F_{16}, FLSL et arithmétique, LSL et flot optique F_{32}. De plus, deux stagiaires de fin d'étude travaillent sur FMDT. Enfin, deux membres permanents de l'équipe supervisent ces travaux.

Bilan :

- 4 revues internationales, 3 conférences internationales, 1 conférence nationale, 1 poster [1–10]

- 1 code open-source disponible `https://github.com/alsoc/fmdt`
- 6 étudiants de Master2, 4 groupes de 5 élèves ingénieurs Polytech I4.

4 RÉFÉRENCES BIBLIOGRAPHIQUES

[1] I. Bournias, R. Chotin, and L. Lacassagne. Using HLS for designing a parametric optical flow hierarchical algorithm in FPGAs. In *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2022.

[2] F. Colas et al. Fripon : a worldwide network to track incoming meteoroids. *Astronomy and Astrophysics (A & A)*, 644 :1–23, 2020.

[3] N. Rambaux et al. Meteorix : a cubesat mission dedicated to the detection of meteors. In *42nd Assembly of Committee on Space Research (COSPAR)*, 2018.

[4] N. Rambaux et al. Meteorix camera tests for space-based meteor observations. *WGN, Journal of the International Meteor Organization (IMO)*, 49(5) :1–3, 2021.

[5] F. Lemaitre, A. Hennequin, and L. Lacassagne. How to speed Connected Component Labeling up with SIMD RLE algorithms. In *ACM Workshop on Programming Models for SIMD/Vector Processing (PPoPP)*, pages 1–8, 2020.

[6] M. Millet, N. Rambaux, A. Cassagne, M. Bouyer, A. Petreto, and L. Lacassagne. High performance computer vision application for Meteor detection from a cubesat. In *44th Assembly of Committee on Space Research (COSPAR)*, 2022.

[7] M. Millet, N. Rambaux, A. Petreto, F. Lemaitre, and L. Lacassagne. Meteorix — a new processing chain for real-time detection and tracking of meteors from space. *WGN, Journal of the International Meteor Organization (IMO)*, 49(6) :1–5, 2022.

[8] A. Petreto, A. Hennequin, , T. Koehler, T. Romera, Y. Fargeaix, B. Gaillard, M. Bouyer, Q. L. Meunier, and L. Lacassagne. Energy and execution time comparison of optical flow algorithms on SIMD and GPU architectures. In *IEEE International Conference on Design and Architectures for Signal and Image Processing (DASIP)*, pages 1–6, 2018.

[9] T. Romera, A. Petreto, F. Lemaitre, M. Bouyer, Q. Meunier, L. Lacassagne, and D. Etiemble. Optical flow algorithms optimized for speed, energy and accuracy on embedded GPUs. *Journal of Real-Time Image Processing (JRTIP)*, 20,2(32) :1–12, 2023.

[10] J. Vaubaillon, Ch. Loir, C. Ciocan, M. Kandeepan, M. Millet, A. Cassagne, L. Lacassagne, P. Da Fonseca, F. Zander, D. Buttsworth, S. Loehle, J. Tóth, A. Moingeon, and N. Rambaux. A 2022 tau-herculid meteor cluster from an airborne experiment : automated detection, characterization, and consequences for meteoroids. *Astronomy and Astrophysics(A&A)*, 2023.

ÉLÉMENT DE PORTFOLIO 04

Publication

1 DÉFINITION DE CET ÉLÉMENT

Titre de l'élément : A fixed-parameter algorithm for scheduling unit dependent tasks on parallel machines with time windows [3]

URL de l'élément : `https://hal.science/hal-03041735`

2 MOTIVATIONS DU CHOIX DE CET ÉLÉMENT

L'analyse de la complexité paramétrée d'un problème d'optimisation combinatoire permet d'aller un peu plus loin dans l'étude de la frontière entre problèmes NP-complets et polynomiaux [1, 2]. Pour un paramètre k fixé et une instance de taille n, un problème combinatoire est dans la classe FPT (pour Fixed-Parameter Tractable) si il existe un algorithme de complexité en temps $\mathcal{O}(f(k) \times \text{poly}(n))$, où f est une fonction calculable. Ainsi, si le paramètre est borné par une valeur constante, l'algorithme obtenu est de complexité polynomiale.

L'étude de la complexité paramétrée des problèmes d'ordonnancement avec contraintes de ressource est un sujet de recherche très ouvert avec de nombreux défis. A titre d'exemple, Mnich et Van Bevern [5] ont identifié quinze challenges en ordonnancement. Le point clé ici est d'identifier les paramètres appropriés, qui permettent de développer des algorithmes de complexité FPT. La majorité des résultats pour des problèmes d'ordonnancement avec contraintes de précédence et machines limitées concluent à la non existence d'algorithme FPT pour les paramètres considérés.

Dans l'article présenté ici, nous avons identifié un nouveau paramètre, le *pathwidth* noté μ associé à un problème d'ordonnancement avec des fenêtres de temps ; il correspond au nombre de tâches pouvant être exécutées simultanément. Un algorithme exact FPT est alors présenté pour résoudre l'existence d'un ordonnancement pour des tâches de durée unitaire sur m machines identiques avec fenêtres de temps et contraintes de précédence.

3 PRÉSENTATION DE CET ÉLÉMENT

Nous présentons à la suite les principes généraux de notre algorithme et les premières extensions de ce travail.

3.1 Présentation de l'algorithme FPT

L'algorithme présenté est un schéma de programmation dynamique original. Des propriétés de dominance basées sur la structure des intervalles permettent de réduire l'espace des solutions réalisables. A chaque instant t entier est associé un ensemble d'états $\mathcal{V}_t$. Un état est un ensemble de tâches qui sont effectuées dans l'intervalle $[0, t]$. On défini un arc entre deux états $E_1 \in \mathcal{V}_t$ et $E_2 \in \mathcal{V}_{t+1}$ si on peut construire un ordonnancement réalisable des tâches de E_2 dans $[0, t+1]$ en ordonnant celles de E_1 dans $[0, t]$.

On construit ainsi un graphe d'états, et on démontre que les chemins maximaux coïncident avec les ordonnancements réalisables. On montre que l'algorithme de construction de ce graphe est un algorithme FPT de paramètre μ.

3.2 Extensions

Ce résultat a permis d'ouvrir de nombreuses perspectives sur le développement d'algorithmes FPT paramétrés par le pathwidth pour des problèmes d'ordonnancement avec fenêtres de temps.

Ainsi, nous avons montré que cette approche pouvait s'étendre en présence de temps de communications unitaires. Tout arc (i, j) du graphe de précédence est associé à un transfert de données entre les deux tâches. Si elles sont effectuées sur des machines différentes, un temps de communication supplémentaire doit être considéré. Avec un codage des états qui intègre également les tâches exécutées à l'instant précédent $[t-1, t]$ pour les états

de $\mathcal{V}_t$, nous avons développé un algorithme FPT paramétré par la pathwidth pour le cas d'un nombre illimité de processeurs [7] puis étendu pour un nombre limité de processeurs [6].

Nous avons également étudié l'existence d'algorithme FPT paramétré par le pathwidth pour des problèmes d'ordonnancement avec des contraintes de latence. Dans ce cas, tout arc (i, j) est associé à une valeur $\ell_{ij} \geq 0$ qui correspond à un temps minimum, maximum ou exact entre la fin de l'exécution de i et le démarrage de j. Cette étude est effectuée en collaboration avec des membres de l'équipe RO. Nous avons identifié comme second paramètre le delai maximum $\ell_{\max}$. Nous avons ainsi démontré que pour des tâches de durée unitaire, m machines, les problèmes étaient para-NP-complets si on considère comme paramètre $\ell_{\max}$ ou μ séparément, et qu'un FPT existe si on considère les deux paramètres simultanément [4].

Enfin, ce résultat nous a permis de proposer avec un membre de l'équipe RO le projet Sorbonne Universités émergences EASI porté par Antoine Jouglet (Heudiasyc) sur l'étude et l'implantation d'un algorithme paramétré pour un problème d'ordonnancement plus général. Ce projet permet de financer un post-doc commun entre les deux laboratoires pendant une année.

4 RÉFÉRENCES BIBLIOGRAPHIQUES

[1] Marek Cygan, Fedor V. Fomin, Łukasz Kowalik, Daniel Lokshtanov, Daniel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer Publishing Company, Incorporated, 1st edition, 2015.

[2] Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Springer-Verlag London, 1st edition, 2013.

[3] Alix Munier Kordon. A Fixed-Parameter Algorithm for Scheduling Unit Dependent Tasks on Parallel Machines with Time Windows. *Discrete Applied Mathematics*, December 2020.

[4] Maher Mallem, Claire C. Hanen, and Alix Munier Kordon. Parameterized complexity of a parallel machine scheduling problem. In *International Symposium on Parameterized and Exact Computation (IPEC)*, Postdam, Germany, September 2022.

[5] Matthias Mnich and René van Bevern. Parameterized complexity of machine scheduling : 15 open problems. *Computers and Operations Research*, 100 :254 – 261, 2018.

[6] Alix Munier-Kordon and Ning Tang. A fixed-parameter algorithm for a unit-execution-time unit-communication-time tasks scheduling problem with a limited number of identical processors. *RAIRO - Operations Research*, 56(5) :3777–3788, September 2022.

[7] Ning Tang and Alix Munier Kordon. A Fixed-Parameter Algorithm for Scheduling Unit dependent Tasks with Unit Communication Delays. In *Euro-Par 2021 - 27th International European Conference on Parallel and Distributed Computing*, volume 12820 of *Lecture Notes in Computer Science*, pages 105–119, Lisbon, Portugal, August 2021. Springer, Cham.

ÉLÉMENT DE PORTFOLIO 05

Publication

1 DÉFINITION DE CET ÉLÉMENT

Titre de l'élément : Handling causality and schedulability when designing and prototyping cyber-physical systems [1]

URL de l'élément : `https://hal.sorbonne-universite.fr/hal-03159402`

2 MOTIVATIONS DU CHOIX DE CET ÉLÉMENT

La détection des problèmes de causalité entre les parties numériques et analogiques d'un modèle SystemC/SystemC AMS destiné à la simulation conjointe, directement à partir d'une représentation haut niveau (SysML) et avant même la génération de code est une avancée importante dans le domaine de prototypage virtuel des systèmes cyber-physiques. De plus, notre approche permet de gérer des situations ne pouvant à cette date être gérées pas d'autres approches. L'algorithme a été intégré à un logiciel libre et élargit significativement le domaine d'application de celui-ci.

3 PRÉSENTATION DE CET ÉLÉMENT

La conception des systèmes cyber-physiques, trouvant leur application par exemple en robotique, dans les domaines médicales et des transports, doit se baser sur une combinaison de techniques de modélisation et de simulation de domaines différents.

Notre travail a ses racines dans une collaboration entre les équipes ALSOC et CIAN du LIP6 d'une part (projets BeyondDreams et HInception dans les années 2010), et la collaboration de membres d'ALSOC avec l'équipe LabSoC de Télécom Paris depuis dix ans. La première collaboration a permis d'obtenir des résultats sur l'analyse pré-simulation de systèmes mixtes analogiques-numériques ; la seconde constitue une contribution significative du LIP6 à un logiciel libre de vérification et de prototypage virtuel de Télécom Paris. Le but ici est de développer un simulateur de systèmes complets et ainsi permettre à des applications embarquées de taille significative de fonctionner sur un modèle SystemC de multiprocesseur sur puce en forme de tâches Posix, supportées par un micro noyau.

SystemC AMS est un langage de description matérielle basé sur C++ qui permet de décrire et de simuler des systèmes mixtes analogiques/numériques. Il prédéfinit plusieurs modèles de calcul, dont le modèle de flot de données temporisées (TDF, *timed data flow*) et des systèmes à évènements discrets.

Un système TDF est composé de clusters, qui contiennent à leur tour des modules. Les modules d'un cluster sont connectés entre eux via leurs ports par des signaux. Les modules TDF communiquent à travers des ports : un nombre fixe d'échantillons, appelé *rate*, sont pris à des intervalles prédéfinis, appelés *time steps*. Un cluster est considéré ordonnançable si un ordre d'exécution entre ses modules peut être établi afin que les time step et rate soient consistants.

Le modules TDF communiquent avec des modules à flot de données à travers des ports de conversion. Lors d'une simulation, la partie TDF avance donc à un rythme imposé, tandis que la partie à évènement discret ne peut avancer que lorsqu'un port de conversion est accédé. Ceci entraîne des problèmes de synchronisation temporelle, appelé des problèmes de *causalité*. Nous avons combiné le simulateur de systèmes complets de TTool et basé sur SystemC avec le simulateur SystemC AMS.

Les problèmes de causalité sont habituellement détectés uniquement au moment de la simulation - s'ils ne passent pas inaperçus et causent de dysfonctionnements graves à l'exécution. La nouveauté de notre contribution est que toutes les parties du système à simuler, numériques et analogiques/mixtes peuvent être modélisées par des diagrammes SysML ; un ordonnancement consistant et la causalité temporelle sont ensuite déterminés avant même la génération de code à partir de ces diagrammes. De plus, l'outil permet la conception de logiciel embarqué vérifié et la simulation d'un système matériel/logiciel/system d'exploitation.

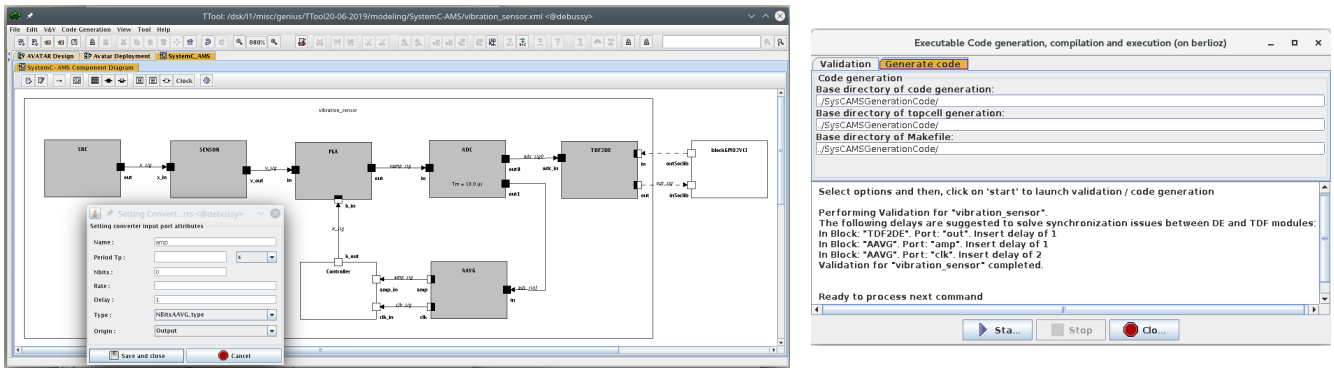


FIGURE 1 – Canevas SystemC AMS et insertion de délais

L'algorithme publié en [1] résout les problèmes de causalité en recalculant des ordonnancements des modules TDF de manière itérative en proposant des délais adaptés jusqu'à résolution de tous les problèmes de causalité. Il insère également des délais pour résoudre des problèmes de boucles de rétroaction.

L'algorithme a été intégré à un outil de prototypage virtuel et d'exploration de l'espace de conception existant TTool de Télécom Paris `https://ttool.telecom-paris.fr`. Le volet TTool-AMS contient plusieurs canevas destinés à la conception de plusieurs modèles de calcul utilisés en SystemC AMS, propose un ordonnancement consistant et respectant la causalité, puis génère du code SystemC pour la partie numérique, et du code SystemC AMS pour la partie analogique/mixte `https://www-soc.lip6.fr/trac/ttool-ams`.

La figure 1 montre à gauche la fenêtre de conception SystemC AMS et un canevas de conception TDF ; à droite on observe la validation de l'ordonnancement après l'insertion de délais. L'exemple choisi est un capteur de vibrations, contenant un contrôleur à évènements discrets connecté à des modules TDF via des ports de conversion. On note la présence d'une boucle de rétroaction entre quatre modules. Le capteur est connecté à une plate-forme numérique contenant un micro-noyau et l'application logicielle. Ces derniers sont représentés sous la forme d'un bloc à évènements discrets à droite de l'image. L'insertion de délais devient nécessaire dans les deux cas (Figure de droite).

4 RÉFÉRENCES BIBLIOGRAPHIQUES

[1] Cortés Porto, Rodrigo, Genius, Daniela, and Apvrille, Ludovic. Handling causality and schedulability when designing and prototyping cyber-physical systems. *Software and Systems Modeling*, pages 1–17, 2021.