Ecole Nationale Supérieure de Mécanique et d'Aérotechnique
Téléport 2 - 1 avenue Clément Ader
BP 40109
86961 Futuroscope Chasseneuil Cedex
Tél : +33 (0)5 49 49 80 80

Sérialisation en temps logique de multiples simulations avioniques temps réel

Laboratoire : LIAS (https://www.lias-lab.fr)
Établissement : ISAE-ENSMA
Directeur de thèse : Emmanuel Grolleau
Référent industriel : Olivier Fourcade
Encadrement : Mickaël Baron, Henri Bauer, Antoine Bertout, Frédéric Ridouard
(l'encadrant souligné est le référent. : baron@ensma.fr)
Financement : CIFRE Airbus

1 Candidature recherchée

Le ou la candidat(e) doit être diplômé(e) d'une école d'ingénieur ou de Master, avec de solides compétences en informatique, programmation et architecture. Des connaissances en aéronautique sont un plus.

Envoyer le dossier par mail à baron@ensma.fr avant le 28 avril 2025 à 19h. Il doit contenir :

- Lettre de motivation ;
- CV mentionnant explicitement le parcours académique ;
- Feuilles de notes des années universitaires mentionnant, si possible, le classement ;
- Tout élément jugé utile pour mettre la candidature en valeur.

2 Contexte et problématique industrielle

La simulation joue un rôle primordial dans le processus de conception, d'intégration, de vérification et de validation des systèmes critiques, en particulier en avionique. Plusieurs niveaux de simulation sont possibles pour l'avionique. Le premier niveau (***SIL*** – *Software-in-the-Loop*), purement logiciel et logique, est de type banc de test virtuel (***VTB*** – *Virtual Test Bench*). Il permet de tester de nouvelles fonctionnalités innovantes dans des conditions nominales. Le ***second niveau VTB*** prend en compte les différentes redondances du système et permet de tester le comportement du système face à des situations non nominales (combinaisons de pannes, conditions extérieures éprouvantes pour l'avion, perturbations, pilotage particulier, etc.). Il permet également de rejouer des conditions de panne afin d'analyser les causes et éventuellement de détecter des bugs. Enfin, la simulation ***HIL*** (*Hardware in the Loop*) exécute les sous-systèmes sur l'architecture réelle cible (calculateurs, réseaux, entrées, sorties sur actionneurs réelles), et est de type banc de test hybride (***HTB*** - *Hybrid Test Bench*) ou physique (***PTB*** – *Physical Test Bench*). Un PTB est bien souvent unique pour un programme d'avion (Avion 0 – *Iron Bird*). Il peut satisfaire certains tests de certification avion, mais les coûts sont importants, de l'ordre de centaines de millions d'euros pour le développement (d'où son unicité) et plusieurs milliers d'euros par heure d'utilisation. Son fonctionnement temps réel et son coût de fonctionnement limite son utilisation pour simuler toutes les situations anormales envisageables.

Le second niveau VTB, focus de la thèse, est celui qui peut être utilisé pour tester plus largement, quasi exhaustivement, certaines séquences d'événements. Ici exhaustif fait référence

à l'état interne des calculateurs vis-à-vis de l'ordre des échanges d'information dans un système avion qui est conçu de façon asynchrone. Aujourd'hui, tous les niveaux de simulations s'exécutent en temps réel. Ainsi, la génération de plusieurs dizaines ou centaines de milliers de simulations de second niveau VTB est exclue. Pourtant, ces simulations permettraient d'anticiper dans le cycle de conception de l'avion la détection d'erreurs (et donc de réduire leur coût de correction). Elles faciliteraient également l'entraînement de modèles d'apprentissage automatique qui pourrait permettre à moyen terme l'émergence de l'utilisation de ces techniques dans le cockpit.

3 Problématiques scientifiques

FIGURE 1 – Moyens de simulation et Iron Bird chez Airbus.

Aujourd'hui, le lancement d'une simulation VTB de second niveau s'effectue par un lancement en parallèle des fonctions nécessaires (p.ex. affichage graphique, fonctions testées, etc.) sur différents cœurs de calculs. Cependant, les taux d'utilisation de ses cœurs sont très variables. Par exemple, pour un ou deux cœurs de CPU chargés à 100 %, les autres peuvent ne pas dépasser les 50 %. La problématique est que l'ensemble de ces cœurs reste utilisé (et donc indisponible) pour toute la durée de la simulation. Ainsi, afin de maximiser le nombre de simulations VTB, l'objectif est de sérialiser une simulation pour minimiser le nombre de cœurs (l'idéal étant un unique cœur) utilisés, quitte à en augmenter la durée. Tous les cœurs utilisés par une simulation doivent être chargés au maximum (optimalement à 100 %) et les autres cœurs ainsi libérés seront utilisés pour pouvoir lancer d'autres simulations en parallèle. La plateforme de simulation de second niveau VTB doit pouvoir être aussi bien locale que dans un *cloud*. Différents défis sont à relever comme énoncé ci-dessous.

- **Reversibilité :** La représentativité d'une architecture sérialisée doit être la même que celle de l'architecture initiale. En particulier le délai dans la transmission des informations entre modules de simulation doit être pris en compte.
- **Propriété Intellectuelle :** différents modules de simulation sont fournis sous forme de boîte noire, dont seuls certains paramètres (interfaces d'entrées/sorties, fréquences internes et externes) sont connus. Le contexte est donc celui d'une co-simulation [7]. La méthode pourra produire des exigences sur ces modules de simulation pour être intégrées dans les boîtes noires ultérieurement.
- **Hétérogénéité :** divers modules de simulation peuvent requérir divers environnements aussi bien logiciels (système d'exploitation, bibliothèques et versions spécifiques) que matériels (GPU, processeurs particuliers).
- **Complexité :** une simulation est constituée de plusieurs dizaines, voire centaines de

modules de simulation, s'échangeant des centaines de milliers de variables sous des rythmes périodiques. Il convient donc de proposer des solutions de complexité polynomiale.

— **Distribution de fonctions multi périodiques communicantes :** quelle que soit la plateforme, *cloud* privé, publique ou plateformes dédiées, les ressources doivent être utilisées au mieux de leurs possibilités. Un scénario peut être vu comme l'exécution de traitements périodiques durant un temps logique donné (exemple : 60 secondes de simulation nécessitent l'exécution en périodique logique de 60×50 instances d'un module de simulation s'exécutant à 20 Hz). Il s'agit donc de placer « au mieux » plusieurs scénarios sur une plateforme hétérogène (si possible de façon optimale au regard du *makespan*, c-à-d de la durée totale nécessaire à l'exécution d'un – grand – nombre de scenarios donnés).

4 Pistes scientifiques

Plusieurs pistes sont ouvertes par l'avènement concomitant de plusieurs théories et technologies.

— La norme EUROCAE ED-247A [5] et la future norme ED-247B [6] permettent d'uniformiser les protocoles avioniques en maintenant les définitions des propriétés temporelles internes des modules de simulation, afin d'uniformiser l'intégration des modèles d'une co-simulation. De plus, les modules de simulation utilisés en avionique sont généralement peu couplés, et permettent de simuler à l'échelle d'une application s'exécutant périodiquement sur un équipement ;

— L'utilisation de conteneurs devrait permettre de garantir une indépendance de l'environnement de simulation par rapport à l'hétérogénéité [11] ;

— Des travaux du laboratoire permettent, de façon statique et sous réserve d'un temps d'exécution connu, de créer des ordonnancements optimaux de systèmes de tâches périodiques sur plateformes hétérogènes en temps polynomial [4, 2, 3]. Il est donc possible de calculer en temps polynomial, sous des hypothèses de durées de préemption et de migration nulles, l'utilisation maximale qui peut être faite d'une plateforme hétérogène par des traitements périodiques. Ces travaux se sont limités aux tâches indépendantes, et il conviendra de les étendre au cas de précédences multi périodiques, en se basant sur un modèle de (***SPC*** – *Semaphore Precedence Constraints*) proposé par le laboratoire [9]. Ce modèle permettra d'exprimer les décalages possibles entre les différents modules de simulations lors de l'occurrence d'événements.

5 Travaux de la thèse

Le doctorant suivra les différentes étapes de la thèse :

— État de l'art sur la co-simulation distribuée, la simulation dans le cloud, l'ordonnancement multiprocesseur hétérogène, et les précédences multi périodiques, normes et outils autour de la simulation distribuée (ED 247, HLA, FMI, Ptolemy, etc.) ;

— Prise en main de l'environnement de simulation de second niveau VTB Airbus - Une application de simulation de référence sera fournie par l'industriel ;

— Prise en main des technologies d'orchestration de conteneurs (Docker, Kubernetes, Argo, etc.) et évaluation des délais type de création, destruction, communications entre conteneurs. À noter qu'un stage de travaux préliminaires sur ce point a eu lieu ;

- Modélisation de problèmes d'ordonnancement de modules de simulation asynchrones communiquant de façon multi périodique. Ici, les hypothèses sont clés dans la complexité du problème. Ainsi, il est probable que la prise en compte de délais liés à la préemption, ou la migration de modules rende le problème NP-difficile au sens fort [10]. Plusieurs modèles seront proposés, du moins réaliste négligeant certains délais (ordonnancement optimal en temps polynomial), à l'insertion d'hypothèses plus réalistes permettant de proposer des heuristiques sous-optimales, mais dont la qualité pourrait être comparée à l'optimale par des méthodes de type ratio de compétitivité [8, 1]. Dans cette modélisation, on identifiera les ordonnanceurs statiques permettant à chaque module de ne pas migrer durant une simulation, par rapport aux ordonnanceurs permettant la migration afin de maximiser l'utilisation des ressources ;
- Mise en œuvre de la simulation telle qu'obtenue par méthode d'ordonnancement en s'appuyant sur l'aide technique d'un stage ;
- Choix expérimentaux des paramètres les plus simples (prise en compte des délais de communication, des durées moyennes d'exécution, etc.) permettant d'obtenir une simulation sérialisée représentative de la simulation PTB.

Références

[1] S. Baruah et al. « On the competitiveness of on-line real-time task scheduling ». In : *Real-Time Systems* 4.2 (juin 1992), p. 125-144. ISSN : 09226443. DOI : 10.1007/BF00365406/METRICS. URL : https://link.springer.com/article/10.1007/BF00365406.

[2] A. Bertout et al. « Template schedule construction for global real-time scheduling on unrelated multiprocessor platforms ». In : *Proceedings of the 2020 Design, Automation and Test in Europe Conference and Exhibition, DATE 2020*. 2020. ISBN : 9783981926347. DOI : 10.23919/DATE48585.2020.9116409.

[3] A. Bertout et al. « Workload assignment for global real-time scheduling on unrelated clustered platforms ». In : *Real-Time Systems* 58.1 (2022). ISSN : 15731383. DOI : 10.1007/s11241-021-09369-0.

[4] A. Bertout et al. « Workload assignment for global real-time scheduling on unrelated multicore platforms ». In : *ACM International Conference Proceeding Series*. 2020. ISBN : 9781450375931. DOI : 10.1145/3394810.3394823.

[5] EUROCAE. *ED-247A - Technical Standard for Virtual Interoperable Simulation for Tests of Aircraft Systems in Virtual or Hybrid Bench*. 2020. URL : https://eurocae.net/news/posts/2020/april/ed-247a-technical-standard-for-virtual-interoperable-simulation-for-tests-of-aircraft-systems-in-virtual-or-hybrid-bench/.

[6] EUROCAE. *ED-247B Technical Standard of Virtual Interoperable Ssimulation for Tests of Aircraft Systems in Virtual or Hybrid Bench*. 2023.

[7] Cláudio Gomes et al. « Co-simulation : State of the art ». In : *arxiv.org* (2017), p. 1-157. arXiv : 1702.00686v1. URL : https://arxiv.org/abs/1702.00686.

[8] J Goossens et P Richard. « Optimal scheduling of periodic gang tasks ». In : *Leibniz transactions on embedded Computers* 3.4 (2016). DOI : 10.4230/LITES-v003-i001-a004. URL : https://dipot.ulb.ac.be/dspace/bitstream/2013/222584/3/62-287-2-PB.pdf.

[9] T.-D. Nguyen et al. « Design and analysis of semaphore precedence constraints : A model-based approach for deterministic communications ». In : *Proceedings of the 2018 Design, Automation and Test in Europe Conference and Exhibition, DATE 2018*. T. 2018-Janua. 2018. ISBN : 9783981926316. DOI : 10.23919/DATE.2018.8342008.

[10] Guillaume Phavorin et al. « Scheduling with preemption delays : Anomalies and issues ». In : *ACM International Conference Proceeding Series* 04-06-November-2015 (nov. 2015), p. 109-118. DOI : 10.1145/2834848.2834853. URL : https://dl.acm.org/doi/10.1145/2834848.2834853.

[11] K Rehman, O Kipouridis et S Karnouskos. « A cloud-based development environment using HLA and kubernetes for the co-simulation of a corporate electric vehicle fleet ». In : *IEEE/SICE International Symposium on System Integration (SII)*. 2019, p. 47-54.