PRIX DE L'INGENIERIE DU FUTUR

THÈME 2025 : INNOVER POUR S'ADAPTER AU CHANGEMENT CLIMATIQUE

Membres de l'équipe :

- Jean-Luc Gauvrit
- Capucine Gaudron

ApeBase

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

Inference-HW : accélérer l'Intelligence Artificielle de manière durable

Mémoire

Travail de :

Jean-Luc Gauvrit

gauvritj37@gmail.com

Et :

Capucine Gaudron

capucinegaudron@hotmail.com

Dépot Github du projet :
https://github.com/JLucGauvrit/Inference-HW

Septembre 2025

1 Introduction

1.1 Contexte et enjeux

L'essor de l'Intelligence Artificielle ouvre des perspectives immenses ; mais cela s'accompagne d'un coût énergétique considérable. Alimenter de vastes centres de données, refroidir les infrastructures et multiplier les serveurs entraînent une explosion de l'empreinte carbone numérique. D'autant plus que chaque génération de modèles de langage (LLM : Large Language Model) est plus lourde et plus exigeante que la précédente.

Comme le souligne le professeur Christoforos Kachris (2025) dans son article « *A Survey on Hardware Accelerators for Large Language Models* », l'innovation matérielle est devenue incontournable pour concilier performance et sobriété énergétique. La Figure 1 illustre le positionnement des différentes architectures matérielles sur un spectre allant de la flexibilité logicielle à l'efficacité énergétique (PPA[1]).

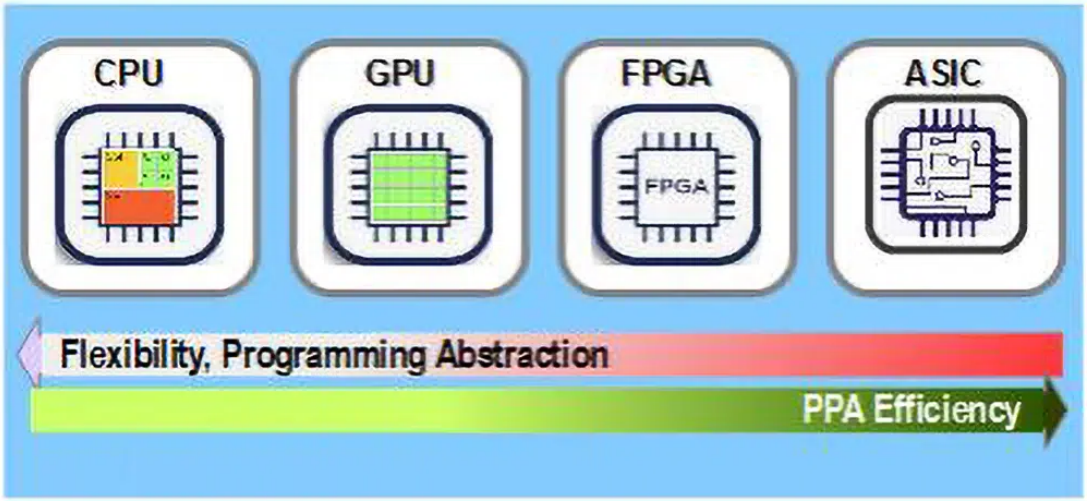

FIGURE 1 – Comparaison des architectures matérielles selon le compromis entre flexibilité de programmation et efficacité énergétique (PPA)

Pour mieux comprendre ce panorama, rappelons brièvement les quatre principales architectures matérielles utilisées en intelligence artificielle. Le **CPU** (Central Processing Unit) est le processeur classique d'un ordinateur, très flexible mais peu efficace pour les calculs massivement parallèles. Le **GPU** (Graphics Processing Unit), initialement conçu pour le rendu graphique, est devenu la norme pour l'entraînement et l'inférence[2] de modèles d'IA grâce à sa capacité à exécuter en parallèle des milliers d'opérations. Mais il demande une consommation énergétique très élevée. Le **FPGA** (Field-Programmable Gate Array) est un circuit reconfigurable, qui permet de définir après fabrication une architecture matérielle spécialisée. Il offre un bon compromis entre flexibilité et efficacité énergétique. Enfin, l'**ASIC** (Application-Specific Integrated Circuit) est un circuit dédié à une tâche précise, extrêmement performant et sobre en énergie, mais rigide et coûteux à concevoir.

Dans ce paysage, le choix du FPGA s'impose comme un compromis stratégique. Contrairement aux CPU et GPU, les FPGA permettent d'exécuter un modèle d'IA de manière spécialisée, tout en restant beaucoup plus sobres en énergie. Ils ne requièrent pas la lourde conception matérielle propre aux ASIC. Les FPGA offrent ainsi une plateforme unique pour prototyper rapidement, explorer les compromis entre performance et consommation, et déployer des architectures adaptées à des environnements contraints (edge computing, objets connectés, systèmes embarqués) où chaque watt compte.

Inference-HW est comme un atelier clé en main pour développer des accélérateurs matériels pour l'IA. Au lieu de devoir monter manuellement chaque outil (compilateurs, bibliothèques, configurations),

1. PPA : *Power, Performance, Area*. Cet indicateur est couramment utilisé en conception matérielle pour évaluer un circuit intégré. Optimiser le PPA signifie trouver le meilleur compromis entre efficacité énergétique, rapidité et coût matériel.

2. L'inférence est le processus d'application d'un modèle d'intelligence artificielle. Contrairement à l'entraînement, qui consiste à apprendre les paramètres du modèle à partir de données existantes, l'inférence utilise ces paramètres pour générer des résultats sur des données invisibles lors de l'entraînement.

l'utilisateur dispose d'une interface unique qui automatise tout le processus :

1. Test sur ordinateur : Vérifier que le code fonctionne correctement.
2. Génération pour FPGA : Transformer le code en instructions optimisées pour la carte FPGA.
3. Déploiement : Créer une image prête à l'emploi pour la carte FPGA, comme un fichier ISO pour une carte SD.

1.2 Présentation des porteurs du projet

Nous sommes **Capucine Gaudron** et **Jean-Luc Gauvrit**, étudiants ingénieurs à l'IMT Atlantique, engagés dans la recherche d'une IA plus responsable. Avec notre structure **ApeBase**, nous nous consacrons à l'optimisation énergétique des modèles d'IA, convaincus que les innovations techniques doivent être au service de la durabilité et de la réponse environnementale.

2 Méthodologie : l'outil Inference-HW

2.1 Environnement Docker

Pour gérer la complexité du développement et du déploiement d'un LLM sur FPGA, nous avons mis en place un environnement nommé **Inference-HW**. Cet outil repose sur Docker[3], et vise à isoler chaque étape du processus : préparation du code, compilation pour l'ordinateur ou FPGA, création d'une image PetaLinux et déploiement sur la carte ZCU106.

L'architecture est organisée en trois services principaux :

— **UI (Interface Web)** : une application python, permettant à l'utilisateur de piloter la compilation et le déploiement depuis un navigateur. Elle établit la connexion avec la carte FPGA.

— **Build-CPU** : un conteneur dédié à la compilation native C++ pour tester son code sur l'ordinateur. Il permet de valider et tester rapidement le code source de l'inférence, sans dépendre de la plateforme FPGA.

— **Build-FPGA** : un conteneur qui embarque l'environnement de compilation spécifique aux cartes ZCU106. Il permet de transformer le code C++ en programmes exécutables sur le processeur embarqué de la carte (ARM aarch64) et de préparer les modules matériels (*kernels*) destinés à être déployés sur le FPGA.

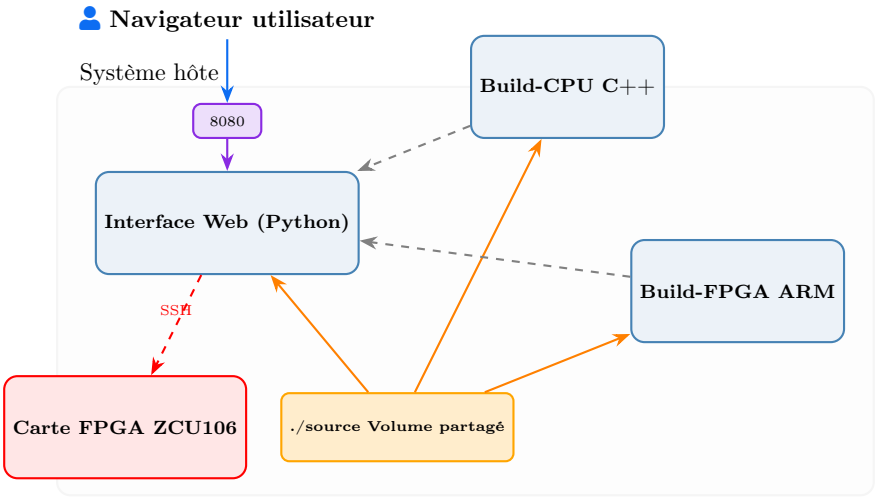

FIGURE 2 – Architecture de développement et de déploiement basée sur Docker pour l'inférence accélérée par FPGA

3. Docker est une technologie qui permet d'emballer une application et ses dépendances dans un conteneur virtuel. Ce conteneur agit comme une boîte isolée, garantissant que l'application fonctionnera de la même manière sur n'importe quel ordinateur ou serveur. Cela simplifie le déploiement et améliore la portabilité des environnements de développement et de production.

La Figure 2 illustre cette architecture, où l'utilisateur interagit via une interface web unique, tandis que la complexité du build et du déploiement est encapsulée dans des services spécialisés. Ce découpage modulaire simplifie la collaboration et rend le flux de développement reproductible.

2.2 Fonctionnement de l'outil Inference-HW

L'outil **Inference-HW** a été conçu pour masquer la complexité technique du développement FPGA derrière une interface simple et accessible. L'utilisateur n'interagit pas directement avec les chaînes de compilation ou les outils matériels, mais passe par une interface web unifiée (Figure 3).

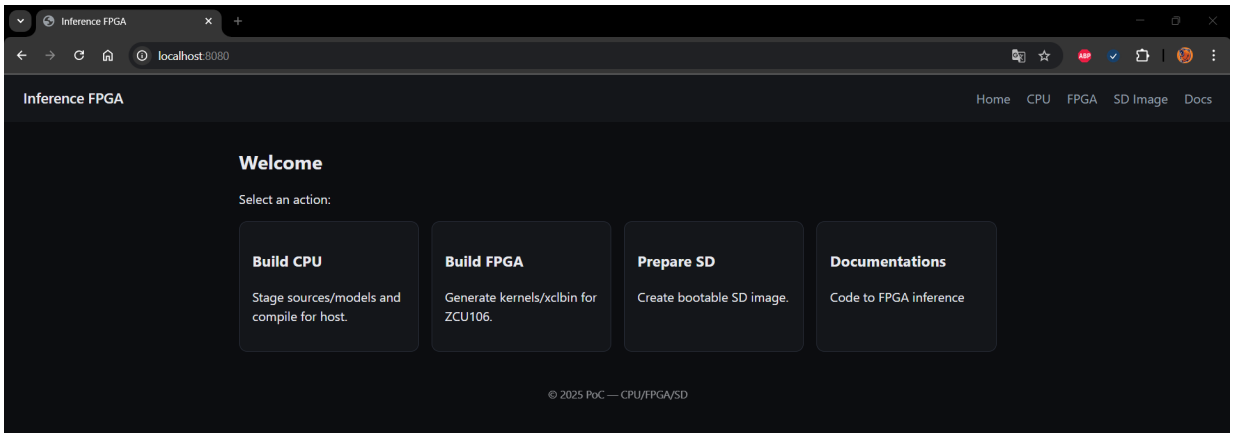

FIGURE 3 – Interface web d'**Inference-HW** : un point d'entrée unique pour piloter la compilation CPU, la génération FPGA et la création d'une image SD de déploiement

Depuis cette interface, il peut lancer trois actions principales : compiler et tester le modèle sur l'ordinateur, générer les kernels et binaires pour le FPGA, et préparer une image SD amorçable intégrant l'ensemble des composants nécessaires au déploiement sur le FPGA.

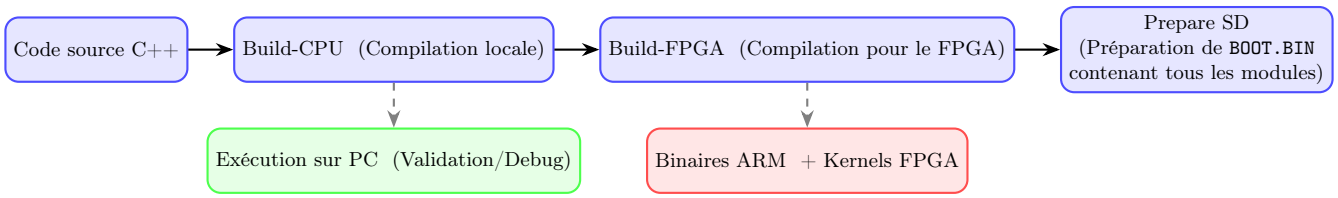

FIGURE 4 – Chaîne de développement : du code source C++ validé sur CPU, jusqu'à la génération d'une image SD amorçable contenant binaires et kernels pour la carte FPGA ZCU106

Cette organisation modulaire et automatisée réduit considérablement les délais entre chaque itération. Là où une chaîne classique nécessiterait des manipulations manuelles lourdes (installation d'outils, configuration de compilateurs, gestion de dépendances), **Inference-HW** encapsule tout dans des conteneurs. L'ingénieur peut ainsi se concentrer sur la logique du modèle plutôt que sur les aspects bas niveau du déploiement matériel. Ce gain de temps et de clarté favorise des cycles de prototypage rapides, indispensables pour suivre le rythme effréné d'évolution des modèles d'IA.

3 Résultats

Afin d'évaluer notre chaîne de compilation et la pertinence de l'accélération matérielle, nous avons comparé les performances d'inférence du modèle *TinyLLaMA* sur CPU et sur FPGA (ZCU106).

Nos mesures montrent que dans l'état actuel, l'inférence sur ordinateur est encore environ trois fois plus rapide que sur FPGA. Ce résultat s'explique par plusieurs facteurs :

— les optimisations logicielles matures du CPU (compilation vectorisée, pipelines optimisés) ;

— la relative immaturité de notre code C++, qui n'exploitent pas encore pleinement le parallélisme et la bande passante mémoire du FPGA ;
— les surcoûts liés aux transferts de données entre le processeur ARM et la logique programmable ;

Malgré cette différence de vitesse, nos résultats expérimentaux sont très encourageants du point de vue énergétique. L'implémentation FPGA permet une réduction de consommation comprise entre **70% et 90%** par rapport à l'exécution sur serveur ou ordinateur.

4 Impact et perspectives

4.1 Inference-HW au service du futur de l'IA

Cette réduction de la consommation d'énergie ouvre des perspectives majeures. Concrètement, cela signifie qu'une tâche d'inférence qui nécessiterait 100 watts sur un GPU classique peut être exécutée pour seulement 10 à 30 watts sur FPGA. À l'échelle d'un serveur, d'un cluster ou d'un centre de données, l'impact devient considérable : moins de chaleur à dissiper, moins de besoins en refroidissement actif, et donc une réduction indirecte supplémentaire de la consommation électrique.

Dans un contexte où l'empreinte carbone du numérique ne cesse de croître, un tel gain peut transformer la viabilité des grands modèles de langage. Par exemple, un modèle déployé sur plusieurs milliers de machines pourrait voir sa facture énergétique et ses émissions associées divisées par un facteur allant de trois à dix. Ces économies rendent envisageables des usages aujourd'hui limités par les contraintes énergétiques, comme l'IA embarquée dans des objets connectés.

Au-delà de l'énergie, l'impact est également matériel. Réduire le besoin en GPU signifie produire moins de cartes électroniques, donc consommer moins de matières premières critiques comme le cuivre, le silicium, ou encore les terres rares nécessaires à la fabrication des semi-conducteurs. À grande échelle, cette approche permet de limiter l'empreinte environnementale de la chaîne de production elle-même, en réduisant l'extraction minière, la consommation d'eau et les déchets électroniques.

Ainsi, **Inference-HW** contribue non seulement à diminuer la consommation d'énergie à l'usage, mais aussi à allonger la durée de vie et à réduire le volume du matériel nécessaire. C'est une démarche complète de sobriété numérique, qui associe performance, efficacité énergétique et réduction de l'impact en ressources naturelles.

4.2 Améliorations

À court terme, l'outil pourra être enrichi par :
— l'optimisation automatique des **kernels HLS** (directives, quantization, sparsity) afin d'améliorer la vitesse d'inférence sur FPGA ;
— la prise en charge d'autres plateformes matérielles (cartes FPGA de nouvelle génération, ASIC spécialisés, NPU embarqués) ;
— l'intégration de métriques de consommation énergétique directement dans le flux de développement, pour guider les choix d'architecture.

À long terme, nous envisageons que ce type d'outil contribue à un écosystème matériel plus sobre, où chaque modèle IA pourra être adapté à son environnement d'exécution de manière efficace et responsable. **Inference-HW** illustre ainsi comment l'ingénierie peut innover pour s'adapter aux défis écologiques et technologiques de demain.

5 Conclusion

Le projet **Inference-HW** répond directement aux enjeux du concours en proposant une solution innovante et adaptable pour réduire l'impact environnemental de l'intelligence artificielle. En facilitant la conception et le déploiement d'accélérateurs matériels sobres en énergie, nous contribuons à une ingénierie plus durable et résiliente. Au-delà de notre preuve de concept, l'intérêt suscité, notamment par plusieurs doctorants qui nous ont contactés via LinkedIn, confirme le potentiel de cette démarche pour bâtir une ingénierie plus durable et résiliente.