Sujet de stage
Ajustement automatique de la précision de calcul sur FPGA

Niveau du stage :

Master 2e année ou fin d'études d'ingénieur

Laboratoire d'accueil :

Laboratoire LIP6, Sorbonne Université, 4 place Jussieu, Paris 5e

Durée :

4 à 6 mois

Mots clés :

architectures reconfigurables, FPGA, autotuning de précision, consommation d'énergie, validation numérique

Contexte et objectifs :

Les architectures reconfigurables comme les FPGA (Field-Programmable Gate Array) sont utilisées pour les applications embarquées et sont aussi de plus en plus considérées pour le calcul scientifique. Parmi les particularités des FPGA, on peut mentionner leur faible consommation d'énergie, leur adaptabilité et le fait de pouvoir utiliser, pour chaque variable d'une application, un format numérique différent, choisi en précision arbitraire. En effet, dans une application en virgule flottante sur FPGA, chaque variable peut avoir sa propre taille de mantisse, indépendante des tailles proposées par la norme IEEE. Comme cette taille est très fortement liée à l'énergie et aux ressources consommées, il est crucial d'utiliser pour chaque variable une précision minimale, tout en garantisant la qualité numérique des résultats. Le but du stage est, en tenant compte de ce compromis *énergie consommée/ressources/débit/précision*, de pouvoir déterminer sur les architectures reconfigurables comme les FPGA une précision adéquate pour chaque variable ou partie d'un code.

Le LIP6 développe des logiciels permettant d'estimer la propagation d'erreur d'arrondi : CADNA [1] pour les codes utilisant les formats de la norme IEEE (précision *half*, simple, double ou quadruple), SAM [2] pour les codes en précision arbitraire. À partir d'un code en C/C++ et d'une précision souhaitée sur les résultats, le logiciel PROMISE [3] [2, 3] modifie automatiquement le type des variables et fournit une version du code mêlant les précisions *half*, simple ou double. PROMISE utilise CADNA pour produire un résultat de référence validé en précision élevé qui est ensuite comparé à ceux obtenus en dégradant la précision de certaines variables. PROMISE n'effectue pas une recherche exhaustive des différentes configurations de types possibles. En effet, celle-ci serait trop coûteuse. Pour déterminer quelles variables peuvent passer dans une précision plus faible sans dégrader la qualité numérique des résultats, PROMISE utilise l'algorithme de Delta Debug [6] d'une complexité moyenne en $O(n \log n)$ pour n variables. À partir d'un code en double précision, l'algorithme de Delta Debug peut être utilisé une première fois pour déterminer quelles variables peuvent être déclarées en simple précision, puis une seconde fois pour déterminer lesquelles peuvent l'être en précision *half*.

Des travaux ont été menés au LIP6 afin d'optimiser les performances d'algorithmes de flot optique sur FPGA [1]. Différentes implémentations ont été effectuées, en précision simple et *half* ainsi qu'en précision *custom* F_{13} [5] et mettent en évidence l'impact de la précision sur la qualité des résultats ainsi que sur l'énergie consommée [4], le débit et les ressources utilisées. Dans ce stage, il s'agit de déterminer, pour chaque variable ou partie d'un code sur FPGA, le format numérique adéquat, pas nécessairement choisi parmi ceux de la norme IEEE. Les codes visés pourront être

1. `http://cadna.lip6.fr`
2. `http://www-pequan.lip6.fr/~jezequel/SAM`
3. `http://promise.lip6.fr`

les implémentations d'algorithmes de flot optique effectuées au LIP6 ou des réseaux de neurones, qui ont en commun d'être composés d'un grand nombre de convolutions/stencils.

Description du stage :

Plusieurs pistes seront explorées dans ce stage.

- Dans un premier temps, la bibliothèque SAM pourra être utilisée dans les codes visés afin de déterminer l'impact de différents formats numériques sur la qualité des résultats.
- L'ajustement automatique des formats numériques en précision arbitraire est une thématique novatrice. Pour cela, des algorithmes seront proposés puis inclus dans une nouvelle version de PROMISE. Un algorithme possible fondé sur le Delta Debug est le calcul d'un résultat de référence avec CADNA en précision élevée, puis l'exécution de différentes versions du code en précision arbitraire afin de déterminer une configuration de types adéquate. Ces exécutions en précision arbitraire pourront être effectuées sur CPU grâce à la bibliothèque MPFR [4]. Etant donnés le nombre de formats numériques possibles et le nombre de variables des codes, des stratégies seront proposées afin que la recherche d'une configuration de types adéquate reste raisonnable en temps. À cet effet, le nombre de formats testés devra être restreint. Une possibilité pour améliorer les performances de la recherche de types sera aussi la restriction du nombre de variables en donnant le même type à certaines variables ou parties du code.
- Une fois la précision déterminée au niveau logiciel, des implantations sur FPGA de l'algorithme, en utilisant des outils de synthèse de haut niveau de type OpenCL [5], seront déployées. Ces implantations permettront d'étudier l'impact de la précision sur l'énergie consommée, le débit et l'utilisation des ressources matérielles.

Financement :

Sur la base des indemnités de stage CNRS (27,30 euros/jour ouvré et remboursement d'une partie des frais de transport)

Dépôt des candidatures :

Les candidatures doivent être adressées par courriel à Roselyne Chotin (`Roselyne.Chotin@lip6.fr`) et Fabienne Jézéquel (`Fabienne.Jezequel@lip6.fr`) sous forme d'un CV, d'une lettre de motivation détaillant les compétences et des relevés de notes des années précédentes. Les coordonnées de deux personnes capables de juger les capacités du candidat seront jointes à la candidature.

Références :

[1] Ilias Bournias, Roselyne Chotin, and Lionel Lacassagne. Using HLS for Designing a Parametric Optical Flow Hierarchical Algorithm in FPGAs. In *IEEE International Symposium on Circuits and Systems (ISCAS 2022)*, Austin, TX, United States, May 2022.

[2] S. Graillat, F. Jézéquel, R. Picot, F. Févotte, and B. Lathuilière. Auto-tuning for floating-point precision with discrete stochastic arithmetic. *Journal of Computational Science*, 36 :101017, 2019.

[3] F. Jézéquel, S. sadat Hoseininasab, and T. Hilaire. Numerical validation of half precision simulations. In *1st Workshop on Code Quality and Security (CQS 2021) in conjunction with WorldCIST'21 (9th World Conference on Information Systems and Technologies)*, Terceira Island, Azores, Portugal, 2021.

[4] 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.

4. `http://mpfr.org`
5. `https://www.khronos.org/opencl/`

[5] S. Piskorski, L. Lacassagne, S. Bouaziz, and D. Etiemble. Customizing CPU instructions for embedded vision systems. In *Computer Architecture, Machine Perception and Sensors (CAMPS)*, pages 59–64. IEEE, 2006.

[6] Andreas Zeller and Ralf Hildebrandt. Simplifying and isolating failure-inducing input. *IEEE Trans. Softw. Eng.*, 28(2) :183–200, February 2002.