Performance des algorithmes numériques précis

Philippe LANGLOIS, DALI (UPVD) - LIRMM (CNRS - UM2)

Bernard GOOSSENS, DALI (UPVD) - LIRMM (CNRS - UM2)

David PARELLO, DALI (UPVD) - LIRMM (CNRS - UM2)

La publication d'un nouvel algorithme numérique est (souvent) assortie de mesures qui exhibent l'amélioration qu'il apporte à l'état de l'art en terme de précision du résultat ou de complexité, en temps ou en espace, voire des deux.

Un nombre important de nouveaux algorithmes de sommation précise et validée en arithmétique flottante ont été récemment introduits [6, 9, 10, 8]. Ces algorithmes utilisent uniquement l'arithmétique flottante IEEE-754 pour calculer des sommes, et donc des produits scalaires, de précision optimale (indépendante du conditionnement) ou k-fois plus plus précises que l'algorithme naïf. Les propriétés numériques de ces algorithmes sont prouvées par leurs auteurs. Les performances en terme de temps d'exécution deviennent donc le paramètre essentiel pour juger des améliorations apportées par chaque nouvelle version.
La pratique classique consiste à fournir une analyse théorique de la complexité en nombre d'opérations flottantes, complétée d'un ensemble de mesures expérimentales des temps de calcul sur un ensemble représentatif d'environnements de calcul.
Cette pratique est obsolète. Nos environnements de calcul ne respectent pas le modèle théorique RAM et présentent une dynamique non déterministe et non reproductible, conséquences du parallélisme, des caches, des prédictions ...

> Measuring the computing time of summation algorithms in a high-level language on today's architectures is more of a hazard than scientific research. S.M. Rump [8].

Nous présenterons nos résultats et développements en vue de l'obtention d'une *analyse de performance validée*, en particulier applicable aux coeurs des codes de calcul. Une première analyse manuelle du parallélisme d'instruction (ILP) permet d'expliquer les écarts constatés entre complexité et mesures expérimentales [5]. L'automatisation de cette analyse est implantée dans l'outil PerPI (Performance and Parallélisme d'Instruction) [1]. PerPI est basé sur Pin [7] et simule la machine idéale de Hennessy-Patterson [3]. PerPI permet de mesurer et d'observer l'ILP des algorithmes de façon largement indépendante de l'implémentation (matériel et compilateur) [2]. Il permet une *analyse reproductible du potentiel de performance* d'un algorithme numérique.
Nous illustrerons comment PerPI peut aider le programmeur à développer de meilleurs algorithmes à l'aide d'une analyse détaillée de ces algorithmes de sommation précise.

Références

[1] Bernard Goossens, Philippe Langlois, and David Parello. Processor simulation: a new way for the performance analysis of numerical algorithms. In B. M. Brown, E. Kaltofen, S. Oishi, and S. M. Rump, editors, *Computer-assisted Proofs – tools, methods and applications*, Dagstuhl Seminar 9471, November 2009.

[2] Bernard Goossens, Philippe Langlois, David Parello, and Eric Petit. Performance evaluation of core numerical algorithms: A tool to measure instruction level parallelism. In *Para 2010 – State of the Art in Scientific and Parallel Computing*, volume EA-119, pages 1–4. University of Iceland, Reykjavik, June 2010.

[3] John L. Hennessy and David A. Patterson. *Computer Architecture – A Quantitative Approach.* Morgan Kaufmann, 2nd edition, 2003.

[4] Philippe Langlois. Compensated algorithms in floating point arithmetic. In *12th GAMM - IMACS International Symposium on Scientific Computing, Computer Arithmetic, and Validated Numerics, Duisburg, Germany*, September 2006. (Invited plenary speaker).

[5] Philippe Langlois and Nicolas Louvet. More instruction level parallelism explains the actual efficiency of compensated algorithms. Technical report, DALI Research Team, 2007. http://hal.archives-ouvertes.fr/hal-00165020.

[6] Takeshi Ogita, Siegfried M. Rump, and Shin'ichi Oishi. Accurate sum and dot product. *SIAM J. Sci. Comput.*, 26(6):1955–1988, 2005.

[7] Pin. http:www.pintool.org.

[8] Siegfried M. Rump. Ultimately fast accurate summation. *SIAM J. Sci. Comput.*, 31(5):3466–3502, 2009.

[9] Siegfried M. Rump, Takeshi Ogita, and Shin'ichi Oishi. Accurate floating-point summation – part I: Faithful rounding. *SIAM J. Sci. Comput.*, 31(1):189–224, 2008.

[10] Siegfried M. Rump, Takeshi Ogita, and Shin'ichi Oishi. Fast high precision summation. pages 1–23, July 2008. (Submitted).

Philippe LANGLOIS, Equipe-Projet DALI - Université de Perpignan Via Domitia
Laboratoire LIRMM - UMR 5506 : CNRS - Université Montpellier 2
langlois@univ-perp.fr