biomanycores.org: a repository of interoperable open-source code for many-core bioinformatics

Jean Frédéric Berthelot[1], Charles Deltel[2], Mathieu Giraud[1,3], Stéphane Janot[1,3], Laetita Jourdan[1,3], Dominique Lavenier[2,4], Hélène Touzet[1,3] and Jean-Stéphane Varré[1,3]

[1] INRIA Lille - Nord Europe, 40 avenue Halley, 59650 Villeneuve d'Ascq, France
[2] INRIA Rennes - Bretagne Atlantique, Campus universitaire de Beaulieu, 35042 Rennes Cedex, France
firstname.lastname@inria.fr
[3] LIFL, UMR 8022 CNRS, Bât M3, 59655 Villeneuve d'Ascq Cedex, France
firstname.lastname@lifl.fr
[4] IRISA, UMR 6074 CNRS, Campus universitaire de Beaulieu, 35042 Rennes Cedex, France
firstname.lastname@irisa.fr

Abstract *biomanycores is a repository of open-source parallel bioinformatics code in OpenCL (and, temporarily, in CUDA). We would like to bridge the gap between research in high-performance-computing and platforms of usual bioinformaticians and biologists through Bio* frameworks.*

Keywords manycore, GPU, CUDA, OpenCL

biomanycores.org : un portail de codes libres interopérables pour la bio-informatique haute performance

Résumé *biomanycores.org est un portail pour la diffusion de codes libres en bio-informatique pour processeurs massivement multi-coeurs. Biomanycores propose des interfaces de ces programmes à des projets Bio*, tels que BioPerl, Biopython et BioJava.*

Mots-clés processeurs massivements multi-coeurs, cartes graphiques, CUDA, OpenCL

1 Contexte

Les architectures massivement multi-cœurs, notamment les cartes graphiques (GPU), permettent un fort parallélisme à faible coût et sont utilisées pour du calcul haute performance. Depuis 2005, c'est par l'intermédiaire de primitives graphiques détournées que les premiers développements en bio-informatique ont vu le jour [3,8]. Mais depuis 2007, avec l'apparition du langage CUDA [2], de nombreuses applications ont été développées. On compte aujourd'hui plus d'une quinzaine d'articles provenant d'une dizaine d'équipes, que cela soit en France ou dans le monde [5,12,13,10,7,14,9] (revue dans [17]). Défini en 2009, le standard OpenCL [1] améliore la portabilité des applications multi-cœurs, en permettant de développer à la fois pour des CPU multi-coeurs et des GPU de différents constructeurs.

Selon les applications, les programmes CUDA ou OpenCL proposent des accélérations de 5× à 50× par rapport à un processeur mono-cœur. Cependant, la valorisation de ces résultats de recherche reste faible. La raison est triple. Les outils développés sont rarement aboutis et restent à l'état de prototype. Ces travaux ont peu de visibilité du fait de leur nouveauté et du changement de culture que cela suppose. Enfin, ils ne proposent pas d'intégration aisée dans les frameworks d'analyse bio-informatique couramment utilisés, et demandent donc aux éventuels utilisateurs un surcroît de travail important. Biomanycores a pour objectif de permettre la diffusion et l'utilisation effective de ces applications.

2 Biomanycores

Biomanycores (http://www.biomanycores.org) est une collection d'applications bioinformatiques pour architectures masssivement multi-cœurs, conçue pour faire le lien entre la recherche en calcul haute-performance et le quotidien des biologistes et des bio-informaticiens.

Biomanycores a pour objectif d'offrir les services suivants : collecte des codes sources disponibles, développement d'interfaces aux frameworks Bio* pour l'interopérabilité et l'intégration de ces méthodes, définition et mise à disposition de données de benchmarks construits autour de données biologiques, afin de permettre l'évaluation des outils parallèles par l'utilisateur. Biomanycores contient des interfaces à Biojava [6], Bioperl [15], et Biopython [4]. Le langage de référence sera à terme OpenCL, mais actuellement, nous incluons des projets CUDA.

3 État du projet et développements futurs

Biomanycores est soutenu par une ADT (action de développement technologique) INRIA 2010–2012. Depuis novembre 2010, un ingénieur est à temps plein sur le projet pour réfléchir à l'architecture globale et intégrer de nouvelles applications. Nous proposons pour l'instant 5 applications : comparaison de séquences (Smith-Waterman) [10], recherche de modèles de Markov (HMMER), recherche de matrices poids-positions [5], prédiction de structures secondaires d'ARN (RNAfold) [16], et détection de pseudo-noeuds (pKnotsRG) [11]. D'ici fin 2011, 5 nouvelles applications seront intégrées.

Nous souhaitons ouvrir Biomanycores autant que possible aux différentes applications produites par la communauté. Les équipes qui souhaiteraient voir intégrer leurs développements CUDA ou OpenCL dans Biomanycores peuvent se faire connaître auprès de `contact@biomanycores.org`.

Références

[1] The Khronos Group, OpenCL 1.0 specification, 2008.

[2] Nvidia CUDA programming guide 2.0, 2008.

[3] M. Charalambous, P. Trancoso, and A. Stamatakis. Initial experiences porting a bioinformatics application to a graphics processor. *Adv. in Informatics*, pages 415–425, 2005.

[4] P. J. A. Cock, T. Antao, J. T. Chang, and al. Biopython : freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics*, page btp163, 2009.

[5] M. Giraud and J.-S. Varré. Parallel position weight matrices algorithms. *Parallel Computing*, 2010.

[6] R. C. G. Holland, T. A. Down, M. Pocock, and al. BioJava : an open-source framework for bioinformatics. *Bioinformatics*, 24(18) :2096–2097, 2008.

[7] L. Ligowski and W. Rudnicki. An efficient implementation of Smith-Waterman algorithm on GPU using CUDA, for massively parallel scanning of sequence databases. In *HiCOMB 2009*, 2009.

[8] W. Liu, B. Schmidt, G. Voss, and W. Müller-Wittig. GPU-ClustalW : using graphics hardware to accelerate multiple sequence alignment. In *High Performance Computing (HiPC 2006), LNCS 4297*, pages 363–374, 2006.

[9] Y. Liu, B. Schmidt, and D. Maskell. Parallel reconstruction of neighbor-joining trees for large multiple sequence alignments using CUDA. In *HiCOMB 2009*, 2009.

[10] S. A Manavski and G. Valle. CUDA compatible GPU cards as efficient hardware accelerators for Smith-Waterman sequence alignment. *BMC Bioinformatics*, 9 Suppl 2 :S10, 2008.

[11] J. Reeder, P. Steffen, and R. Giegerich. pknotsRG : RNA pseudoknot folding including near-optimal structures and sliding windows. *Nucl. Acids Res.*, 35(S2) :W320–324, 2007.

[12] G. Rizk and D. Lavenier. GPU accelerated RNA folding algorithm. In *Using Emerging Parallel Architectures for Computational Science (ICCS 2009)*, 2009.

[13] M. C Schatz, C. Trapnell, A. L. Delcher, and A. Varshney. High-throughput sequence alignment using graphics processing units. *BMC Bioinformatics*, 8 :474, 2007.

[14] H. Shi, B. Schmidt, W. Liu, and W. Mueller-Wittig. Accelerating error correction in high-throughput short-read DNA sequencing data with CUDA. In *HiCOMB 2009*, 2009.

[15] J. E. Stajich, D. Block, K. Boulez, and al. The Bioperl toolkit : Perl modules for the life sciences. *Genome Research*, 12(10) :1611–1618, 2002.

[16] P. Steffen, R. Giegerich, and M. Giraud. Gpu parallelization of algebraic dynamic programming. In *Parallel Processing and Applied Mathematics / Parallel Biocomputing Conference (PPAM / PBC 09)*, 2009.

[17] J.-S. Varré, B. Schmidt, S. Janot, and M. Giraud. *Advances in Genomic Sequence Analysis and Pattern Discovery*, chapter Manycore high-performance computing in bioinformatics. Number 978-981-4327-72-5. World Scientific Publishing Company, 2011.