Biomanycores, open-source parallel code for many-core bioinformatics

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

[1] LIFL, UMR CNRS 8022, Université Lille 1 and INRIA Lille, France

[2] IRISA, UMR CNRS 6074, Université Rennes 1, ENS Cachan, and INRIA Rennes, France

contact@biomanycores.org

URL: http://www.biomanycores.org
Licenses: Various open-source licenses

Graphics processing units (GPUs) enable efficient parallel processing at a very low cost, and are a first step towards the generalization of massively many-core architectures. Since CUDA (in 2007) and the standard OpenCL (2009), many GPU bioinformatics applications have been developed, from sequence alignment to proteomics or phylogenetics (review in [4]).

Biomanycores is a collection of bioinformatics tools, designed to bridge the gap between researches in OpenCL/CUDA high-performance computing on GPU and other "manycore processors" and usual bioinformaticians and biologists.

The main goal is to gather parallel programs and interface them with the BioJava [2], BioPerl [3] and Biopython [1] frameworks. We will also provide benchmarks to show in which cases it is worth using parallel versions of the programs.

Biomanycores was presented at BOSC 2009. Since November 2010, a developer is working full-time on this project, redesigning, extending and documenting Biomanycores. The release 1.1104 of Biomanycores (April 15) includes applications for sequence alignment, sequence processing, and RNA folding tools. Next releases are likely to include tools for proteomics, phylogenetics and genome-wide association studies.

Biomanycores design makes it easy to alter existing pipelines and scripts, in order to use a parallel application instead of the standard one. In some cases, this leads to large improvements. For example, the complexity of the brute-force Position Weight Matrix (PWM) scan algorithm is in $O(nm)$, where n is the sequence length and m the matrix size. PWM performance can be measured in millions of operations per second, one operation being the scoring of one sequence character against one matrix column. While the native Biopython `search_pwm` method peaks at 2 Mop/s, the Biomanycores GPU version (TFM-CUDA) peaks at more than 2 Gop/s, 1000 times faster, even once included the overhead due to the Python interpreter.

In the coming months, we plan to integrate new applications into Biomanycores. Moreover, we wish to receive critical feedback from BioJava, BioPerl and Biopython communities and users in order to improve our interfaces, and discuss further integration into these frameworks.

People who are developing CUDA or OpenCL bioinformatics code (available under a free license) are welcome to get involved in the project – we are willing to help them to integrate their code. Please contact us at contact@biomanycores.org.

References

[1] 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.

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

[3] 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.

[4] 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. World Scientific, 2011.