GPU Accelerated Implementation of NCI Calculations Using Promolecular Density

Gaëtan Rubez,[a,b,c] Jean-Matthieu Etancelin,[b] Jean-Charles Boisson,[b] and Eric Hénon*[c]

The NCI approach is a modern tool to reveal chemical noncovalent interactions. It is particularly attractive to describe ligand–protein binding. A custom implementation for NCI using promolecular density is presented. It is designed to leverage the computational power of NVIDIA graphics processing unit (GPU) accelerators through the CUDA programming model. The code performances of three versions are examined on a test set of 144 systems. NCI calculations are particularly well suited to the GPU architecture, which reduces drastically the computational time. On a single compute node, the dual-GPU version leads to a 39-fold improvement for the biggest instance compared to the optimal OpenMP parallel run (C code, icc compiler) with 16 CPU cores. Energy consumption measurements carried out on both CPU and GPU NCI tests show that the GPU approach provides substantial energy savings. © 2017 Wiley Periodicals, Inc.

DOI: 10.1002/jcc.24786

Introduction

Recent years have seen the growing use of graphics processing units (GPUs) in scientific computing. In this article, we address the potential of this technology to speedup noncovalent interactions (NCI) calculations through the NCI approach. NCI play a key role in chemistry. They are responsible for many properties of condensed phases, including the 3-dimensional arrangement that the biological polymers adopt (DNA double helix, proteins). Such interactions are also of importance in ligand–protein biomolecular recognition in the field of drug-design, a field that some of the present authors are interested in. A new method for revealing NCI[1] has been relatively recently published. It is based on the electron density topology and it enables the identification and visualization of NCI, such as hydrogen bonds, steric repulsions, and van der Waals interactions, both from experimental and theoretical results.[2,3] J. Contreras, one of the authors of this method, has written a program, so-called NCIPLOT,[4] which generates, from either quantum mechanical electron density or from promolecular densities,[5,6] the data needed to visualize NCI areas in real space. This program is now being used worldwide. The growing number of citations of the program published in 2011 (431) is indicative of the potential of this tool in chemistry.

The implementation of the NCI approach involves the use of a three-dimensional regular grid built to encompass the chemical system to be studied. Depending on the chosen grid-step and/or the size of the considered system (protein–protein interface is very large for instance) high performance resources can be required. Then, reducing the computational time is an important issue. The point is that a NCI calculation (see computational details section) can be divided into independent sub processes. Accordingly, it offers a great challenge for efficient parallelization. The authors of this approach have written a parallelized Fortran version for shared-memory CPU architectures using the OpenMP library. But nowadays the use of graphics processing units (GPU) for scientific computations represents a stimulating alternative for the acceleration of scientific computing software. Over the past few years the general-purpose computing on GPUs (GPGPU) for chemical and biochemical simulations has drawn more and more attention. For instance, GPU acceleration was implemented in molecular dynamics codes[7–9] but also in quantum chemistry[10–13] and molecular docking.[14] It is known that the GPU architecture can be highly beneficial, especially for computationally intensive codes based on easily parallelizable schemes. That is the reason why we have written a new NCI code employing the promolecular density that leverages the massively parallel GPU technology using the CUDA parallel computing platform. To our knowledge, it is the first GPU version of the NCI approach. Since the GPU technology has its own constraints, algorithms were reformulated to map onto the GPU architecture and to spend almost all of the execution time on the GPU (CPU resources only drive the GPU computations and performs I/O). Development and tests have been

[a] Gaëtan Rubez, Xavier Vigouroux
Parallel Computing division, ATOS Company, 1 rue de Provence, Echirolles 38130, France

[b] Gaëtan Rubez, Jean-Matthieu Etancelin, Michael Krajecki, Jean-Charles Boisson
Department of Computer Science, CReSTIC (Centre de Recherche en STIC) EA3804, University of Reims Champagne-Ardenne, Moulin de la Housse, Reims 51687, France

[c] Gaëtan Rubez, Eric Hénon
Department of Chemistry, ICMR, UMR CNRS 7312, University of Reims Champagne-Ardenne, Moulin de la Housse, Reims 51687, France
E-mail: eric.henon@univ-reims.fr
Contract grant sponsor: French foundation of Technological Research (PhD thesis of G. Rubez) agreement between the ATOS Company and the University of Reims Champagne-Ardenne; Contract grant number: CIFRE1782015

© 2017 Wiley Periodicals, Inc.