Computer Physics Communications 184 (2013) 2785–2793

Contents lists available at ScienceDirect

Computer Physics Communications

journal homepage: www.elsevier.com/locate/cpc

Implementing molecular dynamics on hybrid high performance computers—Three-body potentials

W. Michael Brown [a,*], Masako Yamada [b]

[a] National Center for Computational Sciences, Oak Ridge National Laboratory, Oak Ridge, TN, USA
[b] GE Global Research, 1 Research Circle K1-3A17A, Niskayuna, NY, USA

ARTICLE INFO

Article history:
Received 28 May 2013
Received in revised form
2 August 2013
Accepted 5 August 2013
Available online 13 August 2013

Keywords:
Molecular dynamics
3-body
GPU
Coprocessor
Accelerator
Stillinger–Weber

ABSTRACT

The use of coprocessors or accelerators such as graphics processing units (GPUs) has become popular in scientific computing applications due to their low cost, impressive floating-point capabilities, high memory bandwidth, and low electrical power requirements. Hybrid high-performance computers, defined as machines with nodes containing more than one type of floating-point processor (e.g. CPU and GPU), are now becoming more prevalent due to these advantages. Although there has been extensive research into methods to use accelerators efficiently to improve the performance of molecular dynamics (MD) codes employing pairwise potential energy models, little is reported in the literature for models that include many-body effects. 3-body terms are required for many popular potentials such as MEAM, Tersoff, REBO, AIREBO, Stillinger–Weber, Bond-Order Potentials, and others. Because the per-atom simulation times are much higher for models incorporating 3-body terms, there is a clear need for efficient algorithms usable on hybrid high performance computers. Here, we report a shared-memory force-decomposition for 3-body potentials that avoids memory conflicts to allow for a deterministic code with substantial performance improvements on hybrid machines. We describe modifications necessary for use in distributed memory MD codes and show results for the simulation of water with Stillinger–Weber on the hybrid Titan supercomputer. We compare performance of the 3-body model to the SPC/E water model when using accelerators. Finally, we demonstrate that our approach can attain a speedup of 5.1 with acceleration on Titan for production simulations to study water droplet freezing on a surface.

1. Introduction

Issues with power consumption, heat dissipation, and high memory access latencies have made heterogeneous architectures a popular idea for increasing parallelism with electrical power and cost efficiency. Basic heterogeneous architectures include hybrid systems that combine a traditional CPU with a coprocessor or accelerator such as a graphics processing unit (GPU), digital signal processor, field-programmable gate array, or other many-core chip. These architectures are becoming more popular in high-performance computers due to significant advantages in the performance to electrical power ratio; for example, the upgrade from the CPU-only Jaguar Cray XT5 at Oak Ridge National Laboratory to the hybrid Titan Cray XK7 resulted in ten times the observed performance while requiring only 19% more electrical power [1]. Not only was Titan the fastest ranked supercomputer at the time of writing, it was also ranked number 3 in terms of power efficiency [2,3].

In order to make effective use of hybrid machines, changes to the models, algorithms, and/or code are typically required. For the latter, changes are often required in order to (1) efficiently use shared memory parallelism, (2) increase concurrency with fine-grain parallelism, and (3) improve data locality, often with explicit code to improve hierarchical memory use. There has been extensive research along these lines to demonstrate significant performance improvements for molecular dynamics on hybrid machines [4,5]. Most of this work has been focused on pairwise potentials. Although these potentials are commonly employed in the simulation of polymers and biomolecules, many materials such as metals, covalent solids, and carbon nanotubes, as well as chemical reactions, are typically simulated with potential energy models that incorporate many-body effects. These potentials typically have a much higher computational cost per atom when compared to pairwise potentials. The simulation of materials with many-body potentials has been described in the context of the "Law of Constancy of Pain" [6]—the trend in the development of new many-body potentials has been to use increased CPU speeds and

[*] Corresponding author.
E-mail addresses: brownw@ornl.gov (W.M. Brown), yamada@ge.com (M. Yamada).

0010-4655/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.cpc.2013.08.002

core counts not for faster simulations, but to simulate with more complex models that have improved accuracy and transferability.

For these reasons, it is clearly desirable to develop algorithms and code for simulation of many-body potentials on accelerators [6,7]. Despite their importance, very little has been reported in the literature describing methods or performance gains from acceleration of many-body potentials. In part, this could be due to the increased complexity of these models—the models require multiple and/or nested loops that increase data dependencies, require changes to the standard neighbor list used in pairwise models, and can require additional communications in parallel codes [6]. Implementations of the embedded atom method (EAM) [8], for use on accelerators and coprocessors have been described that led to significant performance improvements [9,10]. The EAM potential incorporates the energy from embedding an atom into the electron density produced by its neighbors. In this sense, the EAM potential is many-body because the electron charge density at each neighboring atom position must be calculated with a loop over surrounding atoms within some cutoff. However, this model is somewhat unique among many-body potentials in that it can still be computed using only pairwise summations. While this requires additional interprocess communications during the force computation, we have shown that parallel implementations on hybrid machines can maintain significant performance improvements up to the entire 900 nodes available at the time of study [10].

For other many-body potentials, the data dependencies are more complex. 3-body interactions are commonly used and require terms calculated for every triplet of atoms in addition to every pair. 3-body terms are required for many popular potentials such as MEAM [11], Tersoff [12], REBO [13], AIREBO [14], Stillinger–Weber [15], Bond-Order Potentials [16], and others. Although the nested loops required for 3-body terms are simple to implement for serial calculations, their implementation for many-core accelerators/coprocessors results in some complications. The problems arise because non-uniform memory access and limited per-core memory typically favor shared-memory atom- or force-decomposition for parallelism. The naïve implementations of these decompositions result in data dependencies; the evaluation of each energy term in the summation is used to update the force of three different atoms. Therefore, naïve implementations require the use of atomic operations to prevent memory collisions—erroneous results caused by simultaneous update of the same location in memory from multiple threads. Atomic operations are generally undesirable because of the high latencies and because they introduce randomness into the code. Many experienced developers will prefer deterministic code whenever possible to make debugging feasible on high performance computers that are constantly changing out hardware and often changing the software stack.

Alternative implementations can alter the force computation loops to avoid data dependencies in exchange for increased computation. These implementations can reduce global memory access and allow for deterministic code, but the potential performance gains become limited due to substantial increases in the amount of floating point operations required. Although an elegant approach for implementing the Stillinger–Weber potential on GPUs has been described with impressive performance [17], the approach is only applicable for simulations of solid crystals where the neighbors of any given atom do not change. Therefore, the approach is not applicable to many problems such as vacancy diffusion or the simulation of liquids.

In this paper, we present a simple approach for computing 3-body interactions using an atom or force decomposition in shared memory. The approach avoids data dependencies allowing for a deterministic code. We present the changes necessary for implementation in parallel molecular dynamics codes using a spatial decomposition. We provide benchmark results on a hybrid Cray XK7 supercomputer for a 3-body implementation building on our previous work in the LAMMPS molecular dynamics package [18,19,10]. We evaluate performance using the mW water model. The mW water model is comprised of a single effective particle that preferentially forms four tetrahedral bonds. The model has no explicit charges, and hence no hydrogen-bonds or long-range electrostatic terms, but it reproduces the quantitative behavior of water as well as or better than conventional 3, 4 or 5 point charge models. Simulation rates have been reported that are 180 times faster than the least expensive 3 point charge model (specifically SPC/E) while the quantitative agreement of the melting temperature, enthalpy of melting, liquid–vacuum surface tension and liquid density as a function of temperature have been shown to be superior to the SPC, SPC/E, TIP3P, TIP4P and TIP5P models [20]. The orders-of-magnitude speedup relative to SPC/E has been attributed to: (a) a three-fold reduction in number of atoms, (b) the elimination of expensive *k*-space solvers and (c) the enabling of longer timesteps due to the lack of internal bonds. In particular, the mW model has been shown to facilitate the observation of spontaneous freezing in water [21] with much fewer timesteps relative to well-known traditional point-charge potentials [22,23] while still reproducing many quantitative water properties of interest. This makes molecular dynamics a more attractive tool to probe phenomena that span many orders of magnitude of space and/or time, such as our particular area of interest, which is the study of ice formation in the presence of surfaces.

Here, we evaluate performance of the mW model with acceleration compared to both the standard CPU implementation for Stillinger–Weber in LAMMPS and simulation with the SPC/E water model. Our benchmark simulations include periodic water boxes and production simulations that are used to study the microscopic mechanism of droplet freezing on a surface. For the latter, simulation sizes of one million water molecules are used in order to probe the types of complex crystallization behaviors [24] we have observed experimentally for water droplets freezing on surfaces [25,26].

2. Methods

2.1. LAMMPS

Our implementation for 3-body potentials has been performed within the LAMMPS molecular dynamics package [27]. LAMMPS is parallelized via MPI, using spatial-decomposition techniques that partition the 3D simulation domain into a grid of smaller 3D subdomains, one per MPI process. The algorithms we have previously developed for pairwise potentials and long-range electrostatics on accelerators/coprocessors supporting CUDA or OpenCL in LAMMPS have been published in detail [18,19,10]. LAMMPS supports acceleration for short-range force calculation [18] with optional acceleration for neighbor list builds and/or (P^3M) long-range electrostatics [19]. Neighbor list builds are performed on the accelerator by first constructing a cell list that is utilized to build a Verlet list using a radix sort to assert deterministic results. The van der Waals and short-range electrostatic forces are computed in a separate kernel. For each particle, the force-accumulation is performed by one or multiple threads. A default number of threads is chosen based on the hardware and the potential model being used for calculation. The short-range calculation can be performed in single, mixed, or double precision. For mixed precision, all accumulation is performed in double precision and forces, torques, energies, and virials are stored in double precision. For long-range electrostatics, acceleration for P^3M is supported for charge assignment to the mesh and force interpolation. The parallel FFT is performed on the host (see below). The P^3M calculation can be performed in single or double precision.