LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

UCRL-JRNL-224275

De Novo Ultrascale Atomistic Simulations On High-End Parallel Supercomputers

A. Nakano, R. K. Kalia, K. Nomura, A. Sharma, P. Vashishta, F. Shimojo, A. C. T. van Duin, W. A. Goddard, III, R. Biswas, D. Srivastava, L. H. Yang

September 8, 2006

The International Journal of High Performance Computing Applications

DE NOVO ULTRASCALE ATOMISTIC SIMULATIONS ON HIGH-END PARALLEL SUPERCOMPUTERS

Aiichiro Nakano,[a] Rajiv K. Kalia,[a] Ken-ichi Nomura,[a] Ashish Sharma,[a,b] Priya Vashishta,[a] Fuyuki Shimojo[a,c]
[a]Collaboratory for Advanced Computing and Simulations,
University of Southern California, Los Angeles, CA 90089-0242, USA
[b]Department of Biomedical Informatics, Ohio State University, Columbus, OH 43210, USA
[c]Department of Physics, Kumamoto University, Kumamoto 860-8555, Japan
(anakano, rkalia, knomura, sharmaa, priyav)@usc.edu, shimojo@kumamoto-u.ac.jp

Adri C. T. van Duin, William A. Goddard, III
Materials and Process Simulation Center, California Institute of Technology, Pasadena, CA 91125, USA
(duin, wag)@wag.caltech.edu

Rupak Biswas, Deepak Srivastava
NASA Advanced Supercomputing (NAS) Division, NASA Ames Research Center,
Moffett Field, CA 94035, USA
(rbiswas, dsrivastava)@mail.arc.nasa.gov

Lin H. Yang
Physics/H Division, Lawrence Livermore National Laboratory,
Livermore, CA 94551, USA
lyang@llnl.gov

Abstract

We present a de novo hierarchical simulation framework for first-principles based predictive simulations of materials and their validation on high-end parallel supercomputers and geographically distributed clusters. In this framework, high-end chemically reactive and non-reactive molecular dynamics (MD) simulations explore a wide solution space to discover microscopic mechanisms that govern macroscopic material properties, into which highly accurate quantum mechanical (QM) simulations are embedded to validate the discovered mechanisms and quantify the uncertainty of the solution. The framework includes an embedded divide-and-conquer (EDC) algorithmic framework for the design of linear-scaling simulation algorithms with minimal bandwidth complexity and tight error control. The EDC framework also enables adaptive hierarchical simulation with automated model transitioning assisted by graph-based event tracking. A tunable hierarchical cellular decomposition parallelization framework then maps the $O(N)$ EDC algorithms onto Petaflops computers, while achieving performance tunability through a hierarchy of parameterized cell data/computation structures, as well as its implementation using hybrid Grid remote procedure call + message passing + threads programming. High-end computing platforms such as IBM BlueGene/L, SGI Altix 3000 and the NSF TeraGrid provide an excellent test grounds for the framework. On these platforms, we have achieved unprecedented scales of quantum-mechanically accurate and well validated, chemically reactive atomistic simulations—1.06 billion-atom fast reactive force-field MD and 11.8 million-atom (1.04 trillion grid points) quantum-mechanical MD in the framework of the EDC density functional theory on adaptive multigrids—in addition to 134 billion-atom non-reactive space-time multiresolution MD, with the parallel efficiency as high as 0.998 on 65,536 dual-processor BlueGene/L nodes. We have also achieved an automated execution of hierarchical QM/MD simulation on a Grid consisting of 6 supercomputer centers in the US and Japan (in total of 150 thousand processor-hours), in which the number of processors change dynamically on demand and resources are allocated and migrated dynamically in response to faults. Furthermore, performance portability has been demonstrated on a wide range of platforms such as BlueGene/L, Altix 3000, and AMD Opteron-based Linux clusters.

Keywords: Hierarchical simulation, molecular dynamics, reactive force field, quantum mechanics, density functional theory, parallel computing, Grid computing

1 Introduction

Petaflops computers[1] to be built in near future and Grids[2-4] on geographically distributed parallel supercomputers will offer tremendous opportunities for high-end computational sciences. Their computing power will enable unprecedented scales of first-principles based predictive simulations to quantitatively study system-level behavior of complex dynamic systems.[5] An example is the understanding of microscopic mechanisms that govern macroscopic materials behavior, thereby enabling rational design of material compositions and microstructures to produce desired material properties.

Multitude of length and time scales and the associated wide solution space have thus far precluded such first-principles approaches. A promising approach is hierarchical simulation,[6-8] in which atomistic molecular dynamics (MD) simulations[9-12] of varying accuracy and computational costs (from classical non-reactive MD to chemically reactive MD based on semi-classical approaches) explore a wide solution space to discover new mechanisms, in which highly accurate quantum mechanical (QM) simulations[13-17] are embedded to validate the discovered mechanisms and quantify the uncertainty of the solution.

A simple estimate indicates that a 100 billion-atom MD simulation for one nanosecond (or 500 thousand time steps), which embeds 200 million-atom reactive MD and 1 million-atom QM simulations, will require 45 days of computing on a Petaflops platform. To enable such ultrascale simulations in near future, however, nontrivial developments in algorithmic and computing techniques, as well as thorough scalability tests, are required today.

We are developing a ***de novo hierarchical simulation framework*** to enable first-principles based hierarchical simulations of materials and their validation on Petaflops computers and Grids. The framework includes:

- An ***embedded divide-and-conquer (EDC) algorithmic framework*** for: 1) the design of linear-scaling algorithms for approximate solutions of hard simulation problems with minimal bandwidth complexity and codified tight error control; and 2) adaptive embedding of QM simulations in MD simulation so as to guarantee the quality of the overall solution, where ***graph-based event tracking*** (i.e., shortest-path circuit analysis of the topology of chemical bond networks) automates the embedding upon the violation of error tolerance.
- A ***tunable hierarchical cellular decomposition (HCD) parallelization framework*** for: 1) mapping the linear-scaling EDC algorithms onto Petaflops computers, while achieving performance tunability through a hierarchy of parameterized cell data/computation structures; and 2) enabling tightly coupled computations of considerable scale and duration on distributed clusters, based on ***hybrid Grid remote procedure call + message passing + threads programming*** to combine flexibility, fault tolerance, and scalability.

High-end parallel supercomputers such as IBM BlueGene/L and SGI Altix 3000, as well as Grid test-beds such as the NSF TeraGrid, are excellent test grounds for such scalable de novo hierarchical simulation technologies. This paper describes scalability tests of our hierarchical simulation framework on these platforms as well as its portability to other platforms such as AMD Opteron-based Linux clusters. In the next section, we describe the EDC algorithmic framework. Section 3 discusses the tunable HCD parallelization framework. Results of benchmark tests are given in Sec. 4, and Sec. 5 contains conclusions.

2 Embedded Divide-and-Conquer Algorithms for De Novo Hierarchical Simulations

Prerequisite to successful hierarchical simulations and validation at the Petaflops scale are simulation algorithms that are scalable beyond 10^5 processors. We use a unified embedded divide-and-conquer (EDC) algorithmic framework based on data locality principles to design linear-scaling algorithms for broad scientific applications with tight error control.[18, 19] In EDC algorithms, spatially localized sub-problems are solved in a global embedding field, which is efficiently computed with tree-based algorithms (Fig. 1). Examples of the embedding field are: 1) the electrostatic field in molecular dynamics (MD) simulation;[11] 2) the self-consistent Kohn-Sham potential in the density functional theory (DFT) in quantum mechanical (QM) simulation;[19] and 3) a coarser but less compute-intensive simulation method in hierarchical simulation.[7]

2.1 Linear-Scaling Molecular-Dynamics and Quantum-Mechanical Simulation Algorithms

In the past several years, we have used the EDC framework to develop a suite of linear-scaling MD algorithms, in which interatomic forces are computed with varying accuracy and complexity: 1) classical MD involving the formally $O(N^2)$ N-body problem; 2) reactive force-field (ReaxFF) MD involving the $O(N^3)$ variable N-charge problem; 3) quantum mechanical (QM) calculation based on the DFT to provide approximate solutions to the exponentially complex quantum N-body problem; and 4) adaptive hierarchical QM/MD simulations that embed highly accurate QM simulations in MD simulation only when and where high fidelity is required. The Appendix describes the three EDC simulation algorithms that are used in our adaptive hierarchical simulations:

- Algorithm 1—MRMD: space-time multiresolution molecular dynamics.
- Algorithm 2—F-ReaxFF: fast reactive force-filed molecular dynamics.
- Algorithm 3—EDC-DFT: embedded divide-and-conquer density functional theory on multigrids for quantum-mechanical molecular dynamics.

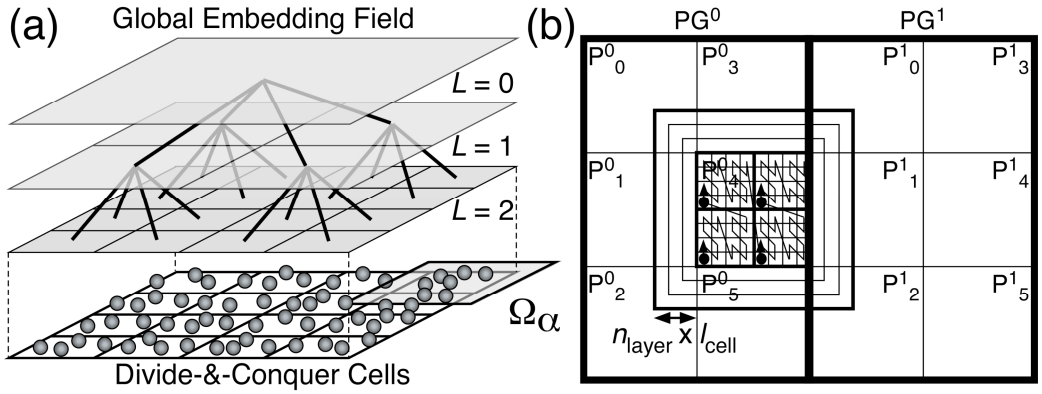

Fig. 1 (a) Schematic of embedded divide-and-conquer (EDC) algorithms. The physical space is subdivided into spatially localized cells, with local atoms constituting sub-problems (bottom), which are embedded in a global field (shaded) solved with a tree-based algorithm. (b) In tunable hierarchical cellular decomposition (HCD), the physical volume is subdivided into process groups, PG^γ, each of which is spatially decomposed into processes, P^γ_π. Each process consists of a number of computational cells (e.g., linked-list cells in MD or domains in EDC-DFT) of size l_{cell}, which are traversed concurrently by threads (denoted by dots with arrows) to compute blocks of cells. P^γ_π is augmented with n_{layer} layers of cached cells from neighbor processes.

2.2 Controlled Errors of Embedded Divide-and-Conquer Simulation Algorithms

A major advantage of the EDC simulation algorithms for automated hierarchical simulations and validation is the ease of codifying (i.e., turning into a coded representation, in terms of programs, which is mechanically executable by other program components) error management. The EDC algorithms have a well-defined set of localization parameters, with which the computational cost and the accuracy are controlled. Figures 2a and 2b show the rapid convergence of the EDC-DFT energy as a function of its localization parameters (the size of a domain and the length of buffer layers to augment each domain for avoiding artificial boundary effects). The EDC-DFT MD algorithm has also overcome the energy drift problem, which plagues most $O(N)$ DFT-based MD algorithms, especially with large basis sets ($> 10^4$ unknowns per electron, necessary for the transferability of accuracy) (Fig. 2c).[19]

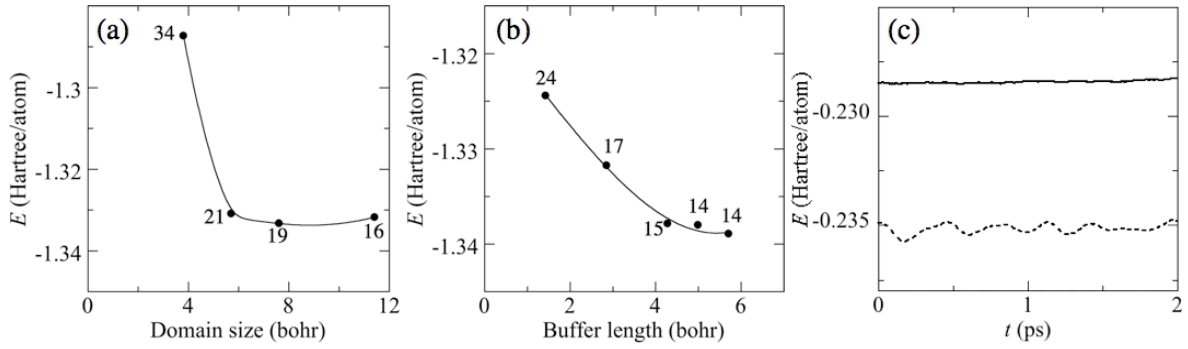

Fig. 2 Controlled convergence of the potential energy of amorphous CdSe by localization parameters: (a) domain size (with the buffer size fixed as 2.854 a.u.); (b) buffer length (with the domain size fixed as 11.416 a.u.). Numerals are the number of self-consistent iterations required for the convergence of the electron density within 10^{-4} of the bulk density. (c) Energy conservation in EDC-DFT based MD simulation of liquid Rb at 1,400 K. The domain and buffer sizes are 16.424 and 8.212 a.u., respectively.

2.3 Adaptive Hierarchical Simulation Framework

Adaptive hierarchical simulation combines a hierarchy of MD algorithms (e.g., MRMD, F-ReaxFF, and EDC-DFT described above) to enable atomistic simulations that are otherwise too large to solve, while retaining QM accuracy.[6-8, 20] The EDC framework achieves this by using less compute-intensive coarse simulation as an embedding field. We have developed an adaptive EDC hierarchical simulation framework, which embeds accurate but compute-intensive simulations in coarse simulation only when and where high fidelity is required. The hierarchical simulation framework consists of: 1) hierarchical division of the physical system into subsystems of decreasing sizes and increasing quality-of-solution (QoSn) requirements, $S_0 \supset S_1 \supset \ldots \supset S_n$; and 2) a suite of simulation services M_α ($\alpha = 0, 1, \ldots, n$) of ascending order of accuracy (e.g., MRMD p F-ReaxFF p EDC-DFT). In the additive hybridization scheme, an accurate estimate of the energy of the entire system is obtained from the recurrence relation,[8, 21]

$$E_\alpha(S_i) = E_{\alpha-1}(S_i) + E_\alpha(S_{i+1}) - E_{\alpha-1}(S_{i+1}) = E_{\alpha-1}(S_i) + \delta E_{\alpha/\alpha-1}(S_{i+1}). \quad (1)$$

This modular additive hybridization scheme not only allows the reuse of existing simulation codes but also minimizes the interdependence and communication between simulation modules. To further expose the data locality of hybrid QM/MD simulation, the EDC framework embeds EDC-DFT simulations of a number of domains within MD simulation of the total system:

$$E(\text{total}) = E_{\text{MD}}(\text{total}) + \sum_{\text{domain}} \left[E_{\text{QM}}(\text{domain}) - E_{\text{MD}}(\text{domain}) \right] = E_{\text{QM}}(\text{total}) + \sum_{\text{domain}} \delta E_{\text{QM/MD}}(\text{domain}). \quad (2)$$

Traditionally, termination atoms are added to the QM and MD domains to minimize artificial boundary effects. However the solution is sensitive to the choice of the termination atoms, and thus the domains need to be determined manually before the simulation.[8] The buffer-layer approach of the EDC-DFT algorithm considerably reduces this sensitivity, and accordingly we are among the first to automate the adaptive domain redefinition during simulations.[22]

2.4 Graph-based Event Tracking for Adaptive Hierarchical Simulations and Validation

An MD simulation method usually has a validation database that compares MD values with corresponding higher-accuracy QM values for a set of physical properties of selected atomic configurations (i.e., the training set). For example, the ReaxFF potential of RDX (1,3,5-trinitro-1,3,5-triazine) has an extensive training database of DFT calculations not only for 1,600 equilibrated molecular fragments but also for 40 key reaction pathways (i.e., chains of intermediate structures that interpolate the structures of reactant and product molecules).[23] Our QM/MD simulation ensures the overall accuracy by using MD simulation only within its range of validity. Our current adaptive hierarchical simulation manages the error based on a simple heuristic, i.e., the deviation of bond lengths from their equilibrium values, for which the MD interatomic potential has been trained.[22] For tighter error management, we extend this approach by encoding the deviation of local topological structures of atoms from those in the training set, based on an abstraction of the structures as a graph and its shortest-path circuit analysis. Though the modeling error is readily quantified by $|\delta E_{\alpha/\alpha-1}(S_{i+1})|$ in Eq. (1) and can be reduced by incrementally enlarging the size of the high-accuracy subsystem once it is defined, the challenge here is to speculate the error in advance so as to minimize subsequent readjustments (if the subsystem is defined too small) or costly speculative high-accuracy simulations (if it is too large).

We abstract the structure of a material as a graph $G = (V, E)$, in which atoms constitute the set of vertices V, and the edge set E consists of chemical bonds. Bonds are defined between a pair of atoms for which Pauling's bond order is larger than a threshold value or the pair distance is less than a cutoff radius. Each vertex is labeled with its three-dimensional position and auxiliary annotations such as atomic species, and each bond with attributes such as bond length and chemical bond order. Given a vertex x and two of its neighbors w and y of G, a shortest-path circuit generated by triplet (w, x, y) is any closed path that contains edges (w, x) and (x, y) and has a shortest path (w, y) in graph $G–x$.[24] The shortest-path circuit analysis has been used successfully to characterize topological order of amorphous materials and to identify and track topological defects such as dislocations (Fig. 3).[25-28] Efficient algorithms with near linear scaling are essential for the graph analysis to be embedded as part of simulation in real time. We have developed a scalable parallel shortest-path circuit analysis algorithm with small memory usage, based on dual-tree expansion and spatial hash-function tagging (SHAFT).[29] SHAFT utilizes the vertex-position label to design a compact, collision-free hash table, thereby avoiding the degradation of cache utilization for large graphs.

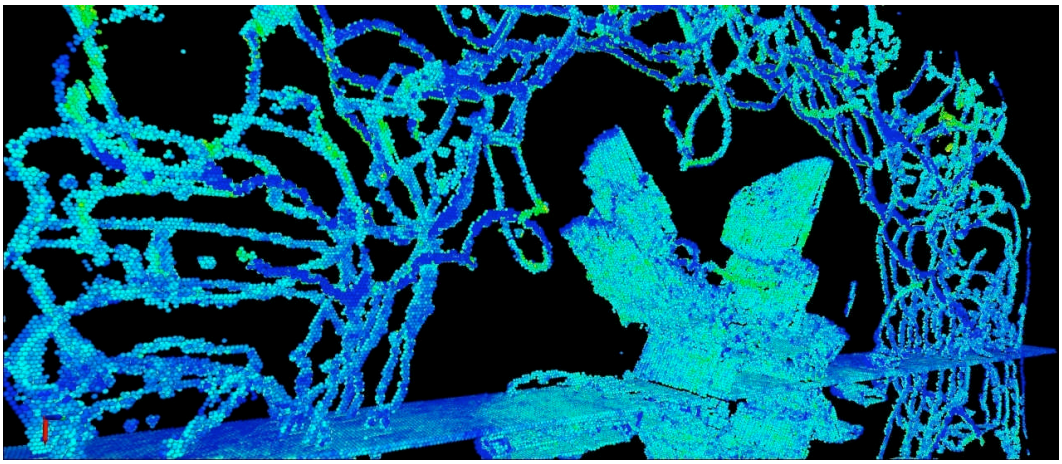

Fig. 3 Network of topological defects (or dislocations) during hypervelocity impact of aluminum nitride ceramic.[28] Only atoms that participate in non-6-membered circuits are visualized, where the color represents the pressure value. (A perfect crystal has only 6-membered circuits.)

The graph abstraction is also used to track discrete events to automate the discovery of mechanisms, i.e., cause-effect relations on a sequence of well-delineated microscopic events that govern system-level macroscopic behavior. An example is a damage mechanism recently discovered by our MD simulation, in which the intersection of an elastic shock-wave front (detected as a discrete jump in pressure) and a high-pressure structural transformation front (the boundary between a sub-graph of vertex degree 4 and that of vertex degree 6) nucleates topological defects and eventually causes the fracture of ceramic under impact.[28] Large classical MD simulations involving multibillion atoms are performed to search for events within a wide solution space, and only when distinct events are detected by the graph analysis, QM simulations are invoked only where the local topological anomaly has been detected.

3 Tunable Hierarchical Cellular Decomposition Parallelization Framework

Data locality principles are key to developing a scalable parallel computing framework as well. We have developed a tunable hierarchical cellular decomposition (HCD) framework for mapping the $O(N)$ EDC algorithms onto massively parallel computers with deep memory hierarchies. The HCD maximally exposes data locality and exploits parallelism at multiple decomposition levels, while providing performance tunability[30] through a hierarchy of parameterized cell data/computation structures (Fig. 1b). At the finest level, the EDC algorithms consist of computational cells—linked-list cells (which are identical to the octree leaf cells in the fast multipole method[31]) in MRMD and F-ReaxFF,[11] or domains in EDC-DFT.[19] In the HCD framework, each compute node (often comprising multiple processors with shared memory) of a parallel computer is identified as a subsystem (P^{γ}_{π} in Fig. 1b) in spatial decomposition, which contains a large number of computational cells. Our EDC algorithms are implemented as hybrid message passing interface (MPI)[32] + shared memory (OpenMP)[33] programs, in which inter-node communication for caching and migrating atoms between P^{γ}_{π}'s is handled with messages, whereas loops over cells within each P^{γ}_{π} (or MPI process) are parallelized with threads (denoted as dots with arrows in Fig. 1b). To avoid performance-degrading critical sections, the threads are ordered by blocking cells, so that the atomic n-tuples being processed by the threads share no common atom. On top of computational cells, cell-blocks, and spatial-decomposition subsystems, the HCD framework introduces a coarser level of decomposition by defining process groups ($PG^{\gamma} = \cup_{\pi} P^{\gamma}_{\pi}$ in Fig. 1b) as MPI Communicators (within a tightly coupled parallel computer) or Grid remote procedure calls[34] (on a Grid of clusters).

Our programs are designed to minimize global operations across PG^{γ}'s and to overlap computations with inter-group communications.[4] For example, the potential energy is computed locally within each group and the global sum is computed only when it needs to be reported to the user. Also our spatial decomposition scheme splits the computations on each processor into those involving only interior linked-list cells and those involving boundary cells. The interior computation is then fully overlapped with the communication of the boundary data. The effect of these latency-hiding techniques on performance is most noticeable on Grid environments, since the communication overhead is already very small on each parallel supercomputer as shown in Sec. 4.

The cellular data structure offers an effective abstraction mechanism for performance optimization. We optimize both data layouts (atoms are sorted according to their cell indices and the linked lists) and computation layouts (force computations are re-ordered by traversing the cells according to a spacefilling curve, a mapping from the 3D space to a 1D list). Cells are traversed along either a Morton curve (Fig. 1b) or a Hilbert curve.[35] In a multi-threading case, the Morton curve ensures maximal separation between the threads and thus eliminates critical sections. Furthermore, the cell size is made tunable to optimize the performance. There is also a trade-off between spatial-decomposition/message-passing and threads parallelisms in the hybrid MPI+OpenMP programs.[36-38] While spatial decomposition involves extra computation on cached cells from neighbor subsystems, its disjoint memory subspaces are free from shared-memory protocol overhead. The computational cells are also used in our multilayer cellular decomposition scheme for inter-node caching of atomic n-tuple (n = 2-6) information (Fig. 1b), where n changes dynamically in the MTS or MPCG algorithm (see Appendix). The Morton curve also facilitates a data compression algorithm based on data locality to reduce the I/O. The algorithm uses octree indexing and sorts atoms accordingly on the resulting Morton curve.[39] By storing differences between successive atomic coordinates, the I/O requirement for a given error tolerance level reduces from $O(N\log N)$ to $O(N)$. An adaptive, variable-length encoding scheme is used to make the scheme tolerant to outliers and optimized dynamically. An order-of-magnitude improvement in the I/O performance was achieved for MD data with user-controlled error bound. The HCD framework includes a topology-preserving computational spatial decomposition scheme to minimize latency through structured message passing and load-imbalance/communication costs through a wavelet-based load-balancing scheme.[40, 41]

High-end hierarchical simulations often run on thousands of processors for months. Grids of geographically distributed parallel supercomputers could satisfy the need of such 'sustained' supercomputing. In collaboration with scientists at the National Institute for Advanced Industrial Science and Technology (AIST) and Nagoya Institute of Technology in Japan, we have recently proposed a sustainable Grid supercomputing paradigm, in which supercomputers that constitute the Grid change dynamically according to a reservation schedule as well as to faults.[22] We use a hybrid Grid remote procedure call (GridRPC) + MPI programming to combine flexibility and scalability. GridRPC enables asynchronous, coarse-grained parallel tasking and hides the dynamic, insecure and unstable aspects of the Grid from programmers (we have used the Ninf-G GridRPC system, http://ninf.apgrid.org), while MPI supports efficient parallel execution on clusters.

4 Performance Tests

Scalability tests of the three parallel simulation algorithms, MRMD, F-ReaxFF and EDC-DFT, have been performed on a wide range of platforms, including the 10,240-processor SGI Altix 3000 at the NASA Ames Research Center, the 131,072-processor IBM BlueGene/L at the Lawrence Livermore National Laboratory (LLNL), and the 2,048-processor AMD Opteron-based Linux cluster at the University of Southern California (USC). We have also tested our sustainable Grid supercomputing framework for hierarchical QM/MD simulations on a Grid of 6 supercomputer centers in the US and Japan. The codes have been ported without any modifications to all the platforms, except that only the pure MPI implementations have been run on BlueGene/L since it does not support OpenMP.

4.1 Experimental Platforms

The SGI Altix 3000 system, named Columbia, at NASA Ames uses the NUMAflex global shared-memory architecture, which packages processors, memory, I/O, interconnect, graphics, and storage into modular components called bricks (detailed information is found at http://www.nas.nasa.gov/Resources/Systems/columbia.html). The computational building block of Altix is the C-Brick, which consists of four Intel Itanium2 processors (in two nodes), local memory, and a two-controller application-specific integrated circuit called the Scalable Hub (SHUB). Each SHUB interfaces to the two CPUs within one node, along with memory, I/O devices, and other SHUBs. The Altix cache-coherency protocol implemented in the SHUB integrates the snooping operations of the Itanium2 and the directory-based scheme used across the NUMAflex interconnection fabric. A load/store cache miss causes the data to be communicated via the SHUB at the cache-line granularity and automatically replicated in the local cache.

The 64-bit Itanium2 processor operates at 1.5GHz and is capable of issuing two multiply-add operations per cycle for a peak performance of 6Gflops. The memory hierarchy consists of 128 floating-point registers and three on-chip data caches (32KB L1, 256KB L2, and 6MB L3). The Itanium2 cannot store floating-point data in L1, making register loads and spills a potential source of bottlenecks; however, a relatively large register set helps mitigate this issue. The superscalar processor implements the Explicitly Parallel Instruction set Computing (EPIC) technology, where instructions are organized into 128-bit VLIW bundles. The Altix platform uses the NUMAlink3 interconnect, a high-performance custom network in a fat-tree topology, in which the bisection bandwidth scales

linearly with the number of processors. Columbia runs 64-bit Linux version 2.4.21. Our experiments use a 6.4TB parallel XFS file system with a 35-fiber optical channel connection to the CPUs.

Columbia is configured as a cluster of 20 Altix boxes, each with 512 processors and 1TB of global shared-access memory. Of these 20 boxes, 12 are model 3700 and the remaining eight are BX2—a double-density version of the 3700. Four of the BX2 boxes are linked with NUMAlink4 technology to allow the global shared-memory constructs to significantly reduce inter-processor communication latency. This 2,048-processor subsystem within Columbia provides a 13Tflops peak capability platform, and was the basis of the computations reported here.

The BlueGene/L (http://www.llnl.gov/ASC/platforms/bluegenel) has been developed by IBM and LLNL, and it uses a large number of low power processors coupled with powerful interconnects and communication schemes. The BlueGene/L comprises of 65,536 compute nodes (CN), each with two IBM PowerPC 440 processors (at 700MHz clock speeds) and 512 MB of shared memory. The theoretical peak performance is 5.6Gflops per CN, or 367Tflops for the full machine. Each processor has a 32KB instruction and data cache, a 2KB L2 cache and a 4MB L3 cache, which is shared with the other processor on the CN. Each CN has two floating-point units that can perform fused multiply-add operations. In its default mode (co-processor mode), one of the processors in the CN manages the computation, while the other processor manages the communication. In an alternative mode of operation (virtual mode), both processors can be used for computation. It uses a highly optimized lightweight Linux distribution and does not allow access to individual nodes.

The nodes are interconnected through multiple complementary high-speed low-latency networks, including a 3D torus network and a tree network. The CNs are connected as a $64 \times 32 \times 32$ 3D torus, which is used for common inter-processor communications. The tree network is optimized for collective operations such as broadcast and gather. The point-to-point bandwidth of the 3D torus network is 175MB/s per link and 350MB/s for the tree network.

The Opteron cluster used for the scalability test consists of 512 nodes, each with two dual-core AMD Opteron processors (at 2GHz clock speeds) and 4GB of memory (in total of 2,048 cores), which is part of a 5,384-processor Linux cluster at USC (http://www.usc.edu/hpcc/systems/l-overview.php). Each core has a 64KB instruction cache, a 64KB data cache, and a 1MB L2 cache. A front side bus operating at 2GHz provides a maximum I/O bandwidth of 24GB/s. The floating-point part of the processor contains three units: a Floating Store unit that stores results to the Load/Store Queue Unit and Floating Add and Multiply units that can work in superscalar mode, resulting in two floating-point results per clock cycle. The 512 nodes are interconnected with Myrinet, which provides over 200MB/s in a ping-pong experiment. Besides the high bandwidth, an advantage of Myrinet is that it entirely operates in the user space, thus avoiding operating systems interference and associated delays. This reduces the latency for small messages to 15μs.

4.2 Scalability Test Results

We have first tested the trade-off between MPI and OpenMP parallelisms on various shared-memory architectures. For example, we have compared different combinations of the number of OpenMP threads per MPI process, n_{td}, and that of MPI processes, n_p, while keeping $P = n_{td} \times n_p$ constant, on $P = 8$ processors in an 8-way 1.5GHz Power4 node. The optimal combination of (n_{td}, n_p) with the minimum execution time is (1, 8) for the MRMD algorithm for an 8,232,000-atom silica material and is (4, 2) for the F-ReaxFF algorithm for a 290,304-atom RDX crystal. Since BlueGene/L does not support OpenMP, we will use $n_{td} = 1$ in the following performance comparisons.

Figure 4a shows the execution time of the MRMD algorithm for silica material as a function of the number of processors P. We scale the problem size linearly with the number of processors, so that the number of atoms $N = 1{,}029{,}000P$ ($P = 1, ..., 1{,}920$). In the MRMD algorithm, the interatomic potential energy is split into the long-range and short-range contributions, and the long-range contribution is computed every 10 MD time steps. The execution time increases only slightly as a function of P, and this signifies an excellent parallel efficiency. We define the speed of an MD program as a product of the total number of atoms and time steps executed per second. The constant-granularity speedup is the ratio between the speed of P processors and that of one processor. The parallel efficiency is the speedup divided by P. On 1,920 processors, the isogranular parallel efficiency of the MRMD algorithm is 0.87. A better measure of the inter-box scaling efficiency based on NUMAlink4 is the speedup from 480 processors in 1 box to 1,920 processors in 4 boxes, divided by the number of boxes. On 1,920 processors, the inter-box scaling efficiency is 0.977. Also the algorithm involves very small communication time, see Fig. 4a.

Figure 4b shows the execution time of the F-ReaxFF MD algorithm for RDX material as a function of P, where the number of atoms is $N = 36{,}288P$. The computation time includes 3 conjugate gradient (CG) iterations to solve the electronegativity equalization problem for determining atomic charges at each MD time step. On 1,920 processors, the isogranular parallel efficiency of the F-ReaxFF algorithm is 0.953 and the inter-box scaling

efficiency is 0.995.

Figure 4c shows the performance of the EDC-DFT based MD algorithm for 720*P* atom alumina systems. In the EDC-DFT calculations, each domain of size 6.66×5.76×6.06Å^3 contains 40 electronic wave functions, where each wave function is represented on 28^3 = 21,952 grid points. The execution time includes 3 self-consistent (SC) iterations to determine the electronic wave functions and the Kohn-Sham potential, with 3 CG iterations per SC cycle to refine each wave function iteratively. The largest calculation on 1,920 processors involves 1,382,400 atoms, for which the isogranular parallel efficiency is 0.907 and the inter-box scaling efficiency is 0.966.

The largest benchmark tests on Columbia include 18,925,056,000-atom MRMD, 557,383,680-atom F-ReaxFF, and 1,382,400-atom (121,385,779,200 electronic degrees-of-freedom) EDC-DFT calculations. We have observed perfect linear scaling for all the three algorithms, with prefactors spanning five orders-of-magnitude. The only exception is the F-ReaxFF algorithm below 100 million atoms, where the execution time scales even sub-linearly. This is due to the decreasing communication overhead, which scales as $O((N/P)^{-1/3})$.

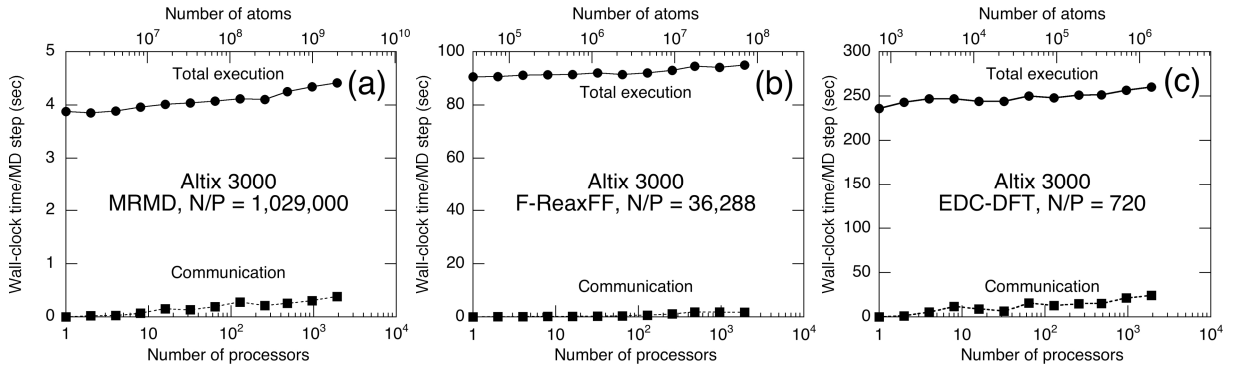

Fig. 4 Total execution (circles) and communication (squares) times per MD time step as a function of the number of processors *P* (= 1, ..., 1,920) of Columbia, for three MD simulation algorithms: (a) MRMD for 1,029,000 *P* atom silica systems; (b) F-ReaxFF MD for 36,288*P* atom RDX systems; and (c) EDC-DFT MD for 720*P* atom alumina systems.

The EDC simulation algorithms are portable and have been run on various high-end computers including IBM BlueGene/L and dual-core AMD Opteron. Since the EDC framework exposes maximal locality, the algorithms scale well consistently on all platforms. Figure 5 compares the isogranular parallel efficiency of the F-ReaxFF MD algorithm for RDX material on Altix 3000, BlueGene/L, and Opteron. In Fig. 5a for Altix 3000, the granularity is the same as that in Fig. 4b. Figure 5b shows the parallel efficiency as a function of *P* on BlueGene/L, where the number of atoms is $N = 36{,}288P$. On 65,536 BlueGene/L nodes (the computation uses only one processor per node in the co-processor mode), the isogranular parallel efficiency of the F-ReaxFF algorithm is over 0.998. Figure 5c shows the parallel efficiency of F-ReaxFF on Opteron, where the number of atoms is $N = 107{,}520P$. The measurements on Opteron have been curried out on one core per CPU for $P = 1$ and two cores per CPU for the other cases. The inter-core communication overhead (between $P = 1$ and 2) is negligible. The intra-node bandwidth ($P = 4$) and network speed ($P \geq 8$) affect the total execution time by 4~12%. The sharp drop of efficiency in Fig. 5c above 1,000 cores may be attributed to interference with other jobs on the Linux cluster, which was in general use during the non-dedicated scalability test. The parallel efficiency is high for all three platforms, where the higher efficiency is achieved for a platform with the higher communication-bandwidth/processor-speed ratio (in descending order for BlueGene/L > Altix 3000 > Opteron/Myrinet).

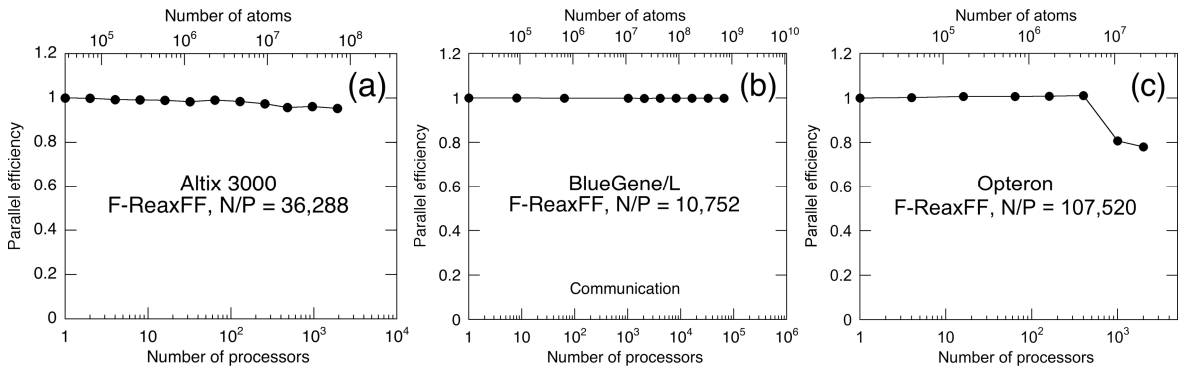

Fig. 5 Isogranular parallel efficiency of F-ReaxFF as a function of the number of processors *P* for RDX material: (a) the number of atoms per processor, *N*/*P* = 36,288 on Altix 3000 (*P* = 1, ...,1,920); (b) *N*/*P* = 10,752 on BlueGene/L (*P* = 1, ..., 65,536); and (c) *N*/*P* = 107,520 on an AMD Opteron cluster (*P* = 1, ..., 2,000).

Major design parameters for reactive and nonreactive MD simulations of materials include the number of atoms in the simulated material and the method to compute interatomic forces (classically in MRMD, semi-empirically in F-ReaxFF MD, or quantum-mechanically in EDC-DFT MD). Figure 6 shows a design-space diagram for MD simulations on BlueGene/L, Altix 3000, and Opteron. The largest benchmark tests in this study include 133,982,846,976-atom MRMD, 1,056,964,608-atom F-ReaxFF, and 11,796,480-atom (1,035,825,315,840 electronic degrees-of-freedom) EDC-DFT calculations on 65,536 dual-processor BlueGene/L nodes.

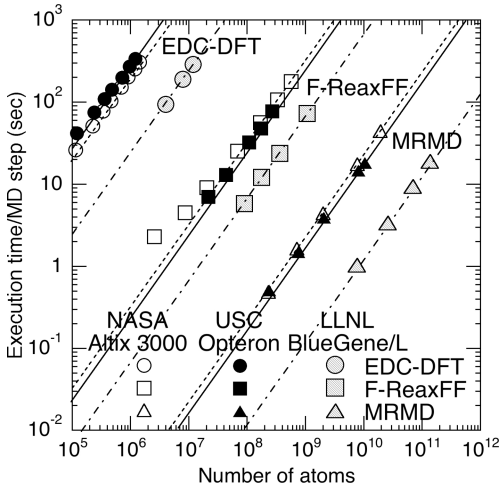

Fig. 6 Benchmark tests of reactive and nonreactive MD simulations on 1,920 Itanium2 processors of the Altix 3000 at NASA (open symbols), 2,000 Opteron processors at USC (solid symbols), and 65,536 dual-processor BlueGene/L nodes at LLNL (shaded symbols). The execution time per MD step is shown as a function of the number of atoms for: quantum-mechanical MD based on the embedded divide-and-conquer density functional theory (EDC-DFT, circles); fast reactive force-field MD (F-ReaxFF, squares); and nonreactive space-time multiresolution MD (MRMD, triangles). Lines show *O*(*N*) scaling.

Different characteristics of the MRMD, F-ReaxFF and EDC-DFT algorithms are reflected in their floating-point performances. The interatomic potential in MRMD is precomputed and tabulated as a function of the interatomic distance. The MRMD computation is thus predominantly table look-ups for atomic pairs and triplets. The F-ReaxFF algorithm, on the contrary, performs a large number of floating-point operations, but it involves more complex list management for atomic n-tuples (n = 2-6). In contrast to these particle-based algorithms, the EDC-DFT algorithm deals with wave functions on regular mesh points. In all the three $O(N)$ algorithms, however, the data layout and computations are highly irregular compared with their higher-complexity counterparts. The floating-point performances of the MRMD (N/P = 1,029,000), F-ReaxFF (N/P = 36,288) and EDC-DFT (N/P = 720) algorithms on

1,920 Itanium2 processors are 1.31, 1.07, and 1.49 Tflops, respectively, whereas the theoretical peak performance is 11.5 Tflops.

4.3 Grid Test Results

Using our sustainable Grid supercomputing framework, we have achieved an automated execution of hierarchical QM/MD simulation on a Grid consisting of 6 supercomputer centers in the US (USC and two NSF TeraGrid nodes at the Pittsburgh Supercomputing Center and the National Center for Supercomputing Applications) and Japan (AIST, University of Tokyo, and Tokyo Institute of Technology).[22] The simulation was sustained autonomously on ~700 processors for 2 weeks, involving in total of 150,000 CPU-hours, where the number of processors changed dynamically on demand and resources were allocated and migrated dynamically according to both reservations and unexpected faults.

5 Conclusions

We have motivated and supported the need for Petaflops computing for advanced matarials research, and have demonstrated that judicious use of divide-and-conquer algorithms and hierarchical parallelization frameworks should make these applications highly scalable on Petaflops platforms. We have illustrated the value of this work with real-world experiments involving quantum-mechanical and molecular-dynamics simulations on high-end parallel supercomputers such as SGI Altix 3000, IBM BlueGene/L and AMD Opteron, as well as on a Grid of globally distributed parallel supercomputers.

We are currently applying the de novo hierarchical simulation framework to study deformation and damage mechanisms of nanophase ceramics and nanoenergetic materials in harsh environments, thereby assisting the design of superhard, tough and damage-tolerant nanomaterials as well as nanoenergetic materials with high specific impact and reduced sensitivity.

One application is hypervelocity impact damage of advanced ceramics (aluminum nitride, silicon carbide, and alumina),[28] for which we have recently performed 500 million-atom MD simulations. The simulation has revealed atomistic mechanisms of fracture accompanying structural phase transformation in AlN under hypervelocity impact at 15 km/s. We are extending these classical MD simulations to those involving surface chemical reactions under high temperatures and flow velocities relevant to micrometeorite impact damages to the thermal and radiation protection layers of aerospace vehicles, understanding of which is essential for safer space flights.

Another application is the combustion of nanoenergetic materials. We have performed 1.3 million-atom F-ReaxFF MD simulations to study shock-initiated detonation of RDX (1,3,5-trinitro-1,3,5-triazine, $C_3N_6O_6H_6$) matrix embedded with aluminum nanoparticles (n-Al) on 1,024 dual-core Opteron processors at the Collaboratory for Advanced Computing and Simulations of USC (Fig. 7). In the simulation, a 320 × 210 × 204 $Å^3$ RDX/n-Al composite is impacted by a plate at a velocity of 5 km/s. The simulation has revealed atomistic processes of shock compression and subsequent explosive reaction. Strong attractive forces between oxygen and aluminum atoms break N-O and N-N bonds in the RDX and, subsequently, the dissociated oxygen atoms and NO molecules oxidize Al, which has also been observed in our DFT-based MD simulation.[42]

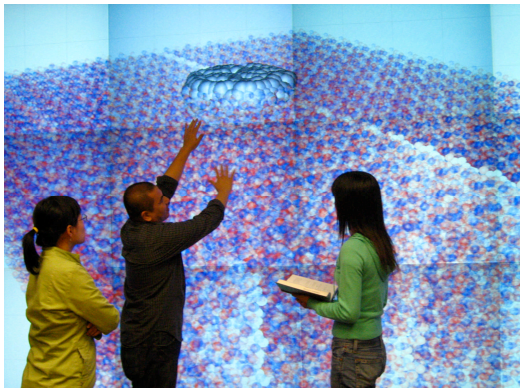

Fig. 7 F-ReaxFF MD simulation of n-Al/RDX simulation shown on a tiled display at the Collaboratory for Advanced Computing and Simulations of USC. Visualization software, with embedded graph analysis algorithms,[29] has been developed by Sharma, et al.[43]

De novo hierarchical simulations also have broad applications in nanoelectronics.[20] The hybrid QM/MD simulation on the US-Japan Grid described in Sec. 4 has studied the SIMOX (separation by implantation by oxygen) technique for fabricating high speed and low power-consumption semiconductor devices. The simulation of the implantation of oxygen atoms toward a Si substrate has revealed a strong dependence of the oxygen penetration depth on the incident beam position, which should be taken into consideration in extending the SIMOX technique to lower incident energies.

These applications on high-end computing platforms today are paving the way for predictive, first-principles simulation based sciences in coming years.[5]

Acknowledgements

The work at the University of Southern California (USC) was partially supported by AFOSR-DURINT, ARO-MURI, DOE, NSF, and Chevron-CiSoft. The work of LHY was supported under the auspices of the U.S. Department of Energy by the University of California Lawrence Livermore National Laboratory (LLNL) under contract No. W-7405-ENG-48. Benchmark tests were performed using the Columbia supercomputer at the NASA Ames Research Center, the BlueGene/L at LLNL, the NSF TeraGrid, the 5,384-processor (13.8Tflops) Linux cluster at USC, the 11Tflops Itanium/Opteron cluster at the National Institute for Advanced Industrial Science and Technology (AIST), and Linux clusters at the University of Tokyo and Tokyo Institute of Technology. Programs have been developed using the 2,048-processor (4Tflops) Opteron/Xeon/Apple G5 cluster at the Collaboratory for Advanced Computing and Simulations of USC. The authors thank Davin Chan, Johnny Chang, Bob Ciotti, Edward Hook, Art Lazanoff, Bron Nelson, Charles Niggley, and William Thigpen for technical discussions on Columbia, Shuji Ogata, Satoshi Sekiguchi, Hiroshi Takemiya, and Yoshio Tanaka for their collaboration on the adaptive QM/MD simulations on the US-Japan Grid, and Bhupesh Bansal, Paulo S. Branicio, and Cheng Zhang for their contribution on graph-based data mining.

Appendix: Embedded Divide-and-Conquer Simulation Algorithms

This Appendix describes computational characteristics of the three embedded divide-and-conquer simulation algorithms that are used in our adaptive hierarchical simulations: 1) MRMD: space-time multiresolution molecular dynamics; 2) F-ReaxFF: fast reactive force-field molecular dynamics; and 3) EDC-DFT: embedded divide-and-conquer density functional theory on adaptive multigrids for quantum-mechanical molecular dynamics.

Algorithm 1—MRMD: Space-Time Multiresolution Molecular Dynamics

MRMD is used as a template for developing broad particle and continuum simulation algorithms. The MD approach follows the time evolution of the positions, $\mathbf{r}^N = \{\mathbf{r}_i | i = 1,...,N\}$, of N atoms by solving coupled ordinary differential equations.[11] Atomic force law is mathematically encoded in the interatomic potential energy $E_{\mathrm{MD}}(\mathbf{r}^N)$, which is often an analytic function $E_{\mathrm{MD}}(\{\mathbf{r}_{ij}\},\{\mathbf{r}_{ijk}\})$ of atomic pair, $\mathbf{r}_{ij}$, and triplet, $\mathbf{r}_{ijk}$, positions. For the long-range electrostatic interaction, we use the fast multipole method (FMM) to reduce the $O(N^2)$ computational complexity of the N-body problem to $O(N)$.[31] In the FMM, the physical system is recursively divided into subsystems to form an octree data structure, and the electrostatic field is computed recursively on the octree with $O(N)$ operations, while maintaining spatial locality at each recursion level. Our scalable parallel implementation of the FMM has a unique feature to compute atomistic stress tensor components based on a complex charge method.[44] MRMD also utilizes temporal locality through multiple time stepping, which uses different force-update schedules for different force components.[11, 45, 46] Specifically, forces from neighbor atoms are computed at every MD step, whereas forces from farther atoms are updated less frequently. For parallelization, we use spatial decomposition. The total volume is divided into P subsystems of equal volume, and each subsystem is assigned to a node in an array of P compute nodes. To calculate the force on an atom in a subsystem, the coordinates of the atoms in the boundaries of neighbor subsystems are 'cached' from the corresponding nodes. After updating the atom positions due to time stepping, some atoms may have moved out of its subsystem. These atoms are 'migrated' to the proper neighbor nodes. With spatial decomposition, the computation scales as N/P, while communication scales as $(N/P)^{2/3}$. The FMM incurs an $O(\log P)$ overhead, which is negligible for coarse-grained ($N/P >> P$) applications.

Algorithm 2—F-ReaxFF: Fast Reactive Force-Field Molecular Dynamics

In the past 5 years, we have developed a first principles-based reactive force-field (ReaxFF) approach to significantly reduce the computational cost of simulating chemical reactions.[23, 47] However, its parallelization has seen only limited success, with the previously largest ReaxFF MD involving $N < 10^4$ atoms. We have developed F-ReaxFF to enable ReaxFF MD involving 10^9 atoms.[18, 48] The variable N-charge problem in ReaxFF amounts to solving a dense linear system of equations to determine atomic charges $\{q_i | i = 1,...,N\}$ at every MD step.[49, 50] F-ReaxFF reduces its $O(N^3)$ complexity to $O(N)$ by combining the FMM based on spatial locality and iterative minimization to utilize the temporal locality of the solution. To accelerate the convergence, we use a multilevel preconditioned conjugate-gradient (MPCG) method that splits the Coulomb-interaction matrix into short- and long-range parts and uses the sparse short-range matrix as a preconditioner.[51] The extensive use of the sparse preconditioner enhances the data locality and thereby improves the parallel efficiency. The chemical bond order B_{ij} is an attribute of atomic pair (i, j) and changes dynamically adapting to the local environment. In ReaxFF, the potential energy $E_{\mathrm{ReaxFF}}(\{\mathbf{r}_{ij}\},\{\mathbf{r}_{ijk}\},\{\mathbf{r}_{ijkl}\},\{q_i\},\{B_{ij}\})$ between atomic pairs $\mathbf{r}_{ij}$, triplets $\mathbf{r}_{ijk}$, and quadruplets $\mathbf{r}_{ijkl}$ depends on the bond orders of all constituent atomic pairs. Force calculations in ReaxFF thus involve up to atomic 6-tuples due to chain-rule differentiations through B_{ij}. To efficiently handle the multiple interaction ranges, the parallel F-ReaxFF algorithm employs a multilayer cellular decomposition scheme for caching atomic n-tuples (n = 2-6).[18]

Algorithm 3—EDC-DFT: Embedded Divide-and-Conquer Density Functional Theory on Adaptive Multigrids for Quantum-Mechanical Molecular Dynamics

EDC-DFT describes chemical reactions with a higher quantum-mechanical accuracy than ReaxFF. The DFT problem is formulated as a minimization of the energy functional $E_{\mathrm{QM}}(\mathbf{r}^N, \psi^{N\mathrm{el}})$ with respect to electronic wave functions (or Kohn-Sham orbitals) $\psi^{N\mathrm{el}}(\mathbf{r}) = \{\psi_n(\mathbf{r}) | n = 1,...,N_{\mathrm{el}}\}$, subject to orthonormality constraints (N_{el} is the number of wave functions on the order of N).[52] The data locality principle called quantum nearsightedness[53] in DFT is best implemented with a divide-and-conquer algorithm,[54, 55] which naturally leads to $O(N)$ DFT calculations.[56] However, it is only in the past several years that $O(N)$ DFT algorithms, especially with large basis sets ($> 10^4$ unknowns per electron, necessary for the transferability of accuracy), have attained controlled error bounds, robust

convergence properties, and energy conservation during MD simulations, to make large DFT-based MD simulations practical.[19, 57] We have designed an embedded divide-and-conquer density functional theory (EDC-DFT) algorithm, in which a hierarchical grid technique combines multigrid preconditioning and adaptive fine mesh generation.[19] The EDC-DFT algorithm represents the physical system as a union of overlapping spatial domains, $\Omega = \cup_\alpha \Omega_\alpha$, and physical properties are computed as linear combinations of domain properties. For example, the electronic density is expressed as $\rho(\mathbf{r}) = \Sigma_\alpha p^\alpha(\mathbf{r}) \Sigma_n f_n^\alpha |\psi_n^\alpha(\mathbf{r})|^2$, where $p^\alpha(\mathbf{r})$ is a support function that vanishes outside the α-th domain Ω_α, and f_n^α and $\psi_n^\alpha(\mathbf{r})$ are the occupation number and the wave function of the n-th Kohn-Sham orbital in Ω_α. The domains are embedded in a global Kohn-Sham potential, which is a functional of $\rho(\mathbf{r})$ and is determined self-consistently with $\{f_n^\alpha, \psi_n^\alpha(\mathbf{r})\}$. We use the multigrid method to compute the global potential in $O(N)$ time. The DFT calculation in each domain is performed using a real-space approach,[58] in which electronic wave functions are represented on grid points. The real-space grid is augmented with coarser multigrids to accelerate the iterative solution. Furthermore, a finer grid is adaptively generated near every atom, in order to accurately operate ionic pseudopotentials for calculating electron-ion interactions. The EDC-DFT algorithm on the hierarchical real-space grids is implemented on parallel computers based on spatial decomposition. Each compute node contains one or more domains of the EDC algorithm. Then only the global density but not individual wave functions needs to be communicated. The resulting large computation/communication ratio makes this approach highly scalable.

References

1. Dongarra, J. & Walker, D. The quest for Petascale computing. Computing in Science and Engineering 3(3), 22 (2001).
2. Foster, I. & Kesselman, C. The Grid 2: Blueprint for a New Computing Infrastructure (Morgan Kaufmann, 2003).
3. Allen, G. et al. Supporting efficient execution in heterogeneous distributed computing environments with Cactus and Globus. Proceedings of Supercomputing 2001 (ACM, 2001).
4. Kikuchi, H. et al. Collaborative simulation Grid: multiscale quantum-mechanical/classical atomistic simulations on distributed PC clusters in the US and Japan. Proceedings of Supercomputing 2002 (IEEE, 2002).
5. Emmott, S. & Rison, S. Towards 2020 Science (Microsoft Research, Cambridge, UK, 2006).
6. Broughton, J. Q., Abraham, F. F., Bernstein, N. & Kaxiras, E. Concurrent coupling of length scales: Methodology and application. Physical Review B 60, 2391-2403 (1999).
7. Nakano, A. et al. Multiscale simulation of nanosystems. Computing in Science & Engineering 3(4), 56-66 (2001).
8. Ogata, S. et al. Hybrid finite-element/molecular-dynamics/electronic-density-functional approach to materials simulations on parallel computers. Computer Physics Communications 138, 143-154 (2001).
9. Abraham, F. F. et al. Simulating materials failure by using up to one billion atoms and the world's fastest computer: Brittle fracture. Proceedings of the National Academy of Sciences of the United States of America 99, 5777-5782 (2002).
10. Kadau, K., Germann, T. C., Lomdahl, P. S. & Holian, B. L. Microscopic view of structural phase transitions induced by shock waves. Science 296, 1681-1684 (2002).
11. Nakano, A. et al. Scalable atomistic simulation algorithms for materials research. Scientific Programming 10, 263 (2002).
12. Kale, L. et al. NAMD2: Greater scalability for parallel molecular dynamics. Journal of Computational Physics 151, 283-312 (1999).
13. Car, R. & Parrinello, M. Unified approach for molecular dynamics and density functional theory. Physical Review Letters 55, 2471 (1985).
14. Truhlar, D. G. & McKoy, V. Computational chemistry. Computing in Science & Engineering 2, 19-21 (2000).
15. Kendall, R. A. et al. High performance computational chemistry: An overview of NWChem a distributed parallel application. Computer Physics Communications 128, 260-283 (2000).
16. Gygi, F. et al. Large-scale first-principles molecular dynamics simulations on the BlueGene/L platform using the Qbox code. Proceedings of Supercomputing 2005 (ACM, 2005).
17. Ikegami, T. et al. Full electron calculation beyond 20,000 atoms: ground electronic state of photosynthetic proteins. Proceedings of Supercomputing 2005 (ACM, 2005).
18. Nakano, A. et al. A divide-and-conquer/cellular-decomposition framework for million-to-billion atom simulations of chemical reactions. Computational Materials Science (2006).
19. Shimojo, F., Kalia, R. K., Nakano, A. & Vashishta, P. Embedded divide-and-conquer algorithm on hierarchical real-space grids: parallel molecular dynamics simulation based on linear-scaling density functional theory. Computer Physics Communications 167, 151-164 (2005).
20. Ogata, S., Shimojo, F., Kalia, R. K., Nakano, A. & Vashishta, P. Environmental effects of H_2O on fracture initiation in silicon: a hybrid electronic-density-functional/molecular-dynamics study. Journal of Applied Physics 95, 5316-5323 (2004).
21. Dapprich, S., Komáromi, I., Byun, K. S., Morokuma, K. & Frisch, M. J. A new ONIOM implementation in Gaussian 98. I. The calculation of energies, gradients, vibrational frequencies, and electric field derivatives. J. Mol. Struct. (Theochem) 461-462, 1 (1999).
22. Takemiya, H. et al. Sustainable adaptive Grid supercomputing: multiscale simulation of semiconductor processing across the Pacific. Proceedings of Supercomputing 2006 (IEEE/ACM, 2006).
23. Strachan, A., van Duin, A. C. T., Chakraborty, D., Dasgupta, S. & Goddard, W. A. Shock waves in high-energy materials: the initial chemical events in nitramine RDX. Physical Review Letters 91, 098301 (2003).
24. Franzblau, D. S. Computation of ring statistics for network models of solids. Physical Review B 44, 4925-4930 (1991).

25. Rino, J. P., Ebbsjo, I., Kalia, R. K., Nakano, A. & Vashishta, P. Structure of Rings in Vitreous SiO2. Physical Review B 47, 3053-3062 (1993).
26. Nakano, A., Kalia, R. K. & Vashishta, P. Scalable molecular-dynamics, visualization, and data-management algorithms for materials simulations. Computing in Science & Engineering 1, 39-47 (1999).
27. Szlufarska, I., Nakano, A. & Vashishta, P. A crossover in the mechanical response of nanocrystalline ceramics. Science 309, 911-914 (2005).
28. Branicio, P. S., Kalia, R. K., Nakano, A. & Vashishta, P. Shock-induced structural phase transition, plasticity, and brittle cracks in aluminum nitride ceramic. Physical Review Letters 96, 065502 (2006).
29. Zhang, C. et al. Collision-free spatial hash functions for structural analysis of billion-vertex chemical bond networks. Computer Physics Communications 175, 339-347 (2006).
30. Whaley, R. C., Petitet, A. & Dongarra, J. J. Automated empirical optimizations of software and the ATLAS project. Parallel Computing 27, 3-35 (2001).
31. Greengard, L. & Rokhlin, V. A fast algorithm for particle simulations. Journal of Computational Physics 73, 325-348 (1987).
32. Gropp, W., Lusk, E. & Skjellum, A. Using MPI, 2nd Ed. (MIT Press, Cambridge, 1999).
33. Chandra, R. & McDonald, R. M. L. D. D. K. D. M. J. Parallel Programming in OpenMP (Morgan Kaufmann, San Francisco, 2000).
34. Tanaka, Y., Nakada, H., Sekiguchi, S., Suzumura, T. & Matsuoka, S. Ninf-G: a reference implementation of RPC-based programming middleware for Grid computing. Journal of Grid Computing 1, 41-51 (2003).
35. Moon, B., Jagadish, H. V., Faloutsos, C. & Saltz, J. H. Analysis of the clustering properties of the Hilbert space-filling curve. IEEE Transactions on Knowledge and Data Engineering 13, 124-141 (2001).
36. Henty, D. S. Performance of hybrid message-passing and shared-memory parallelism for discrete element modeling. Proceedings of Supercomputing 2000 (IEEE, 2000).
37. Shan, H. Z., Singh, J. P., Oliker, L. & Biswas, R. A comparison of three programming models for adaptive applications on the Origin2000. Journal of Parallel and Distributed Computing 62, 241-266 (2002).
38. Shan, H. Z., Singh, J. P., Oliker, L. & Biswas, R. Message passing and shared address space parallelism on an SMP cluster. Parallel Computing 29, 167-186 (2003).
39. Omeltchenko, A. et al. Scalable I/O of large-scale molecular dynamics simulations: A data-compression algorithm. Computer Physics Communications 131, 78-85 (2000).
40. Nakano, A. & Campbell, T. J. An adaptive curvilinear-coordinate approach to dynamic load balancing of parallel multiresolution molecular dynamics. Parallel Computing 23, 1461-1478 (1997).
41. Nakano, A. Multiresolution load balancing in curved space: the wavelet representation. Concurrency: Practice and Experience 11, 343-353 (1999).
42. Umezawa, N., Kalia, R. K., Nakano, A., Vashishta, P. & Shimojo, F. RDX (1,3,5-trinitro-1,3,5-triazine) decomposition and chemisorption on Al(111) surface: first-principles molecular dynamics study. Journal of Chemical Physics (2006).
43. Sharma, A. et al. Immersive and interactive exploration of billion-atom systems. Presence-Teleoperators and Virtual Environments 12, 85-95 (2003).
44. Ogata, S. et al. Scalable and portable implementation of the fast multipole method on parallel computers. Computer Physics Communications 153, 445-461 (2003).
45. Martyna, G. J., Tuckerman, M. E., Tobias, D. J. & Klein, M. L. Explicit reversible integrators for extended systems dynamics. J. Chem. Phys. 101, 4177 (1994).
46. Schlick, T. et al. Algorithmic challenges in computational molecular biophysics. Journal of Computational Physics 151, 9-48 (1999).
47. van Duin, A. C. T., Dasgupta, S., Lorant, F. & Goddard, W. A. ReaxFF: A reactive force field for hydrocarbons. Journal of Physical Chemistry A 105, 9396-9409 (2001).
48. Vashishta, P., Kalia, R. K. & Nakano, A. Multimillion atom simulations of dynamics of oxidation of an aluminum nanoparticle and nanoindentation on ceramics. Journal of Physical Chemistry B 110, 3727-3733 (2006).
49. Rappe, A. K. & Goddard, W. A. Charge Equilibration for Molecular-Dynamics Simulations. Journal of Physical Chemistry 95, 3358-3363 (1991).
50. Campbell, T. J. et al. Dynamics of oxidation of aluminum nanoclusters using variable charge molecular-dynamics simulations on parallel computers. Physical Review Letters 82, 4866-4869 (1999).
51. Nakano, A. Parallel multilevel preconditioned conjugate-gradient approach to variable-charge molecular dynamics. Computer Physics Communications 104, 59-69 (1997).
52. Hohenberg, P. & Kohn, W. Inhomogeneous electron gas. Physical Review 136, B864-B871 (1964).

53. Kohn, W. Density functional and density matrix method scaling linearly with the number of atoms. Physical Review Letters 76, 3168-3171 (1996).
54. Yang, W. Direct calculation of electron density in density-functional theory. Physical Review Letters 66, 1438-1441 (1991).
55. Yang, W. & Lee, T.-S. A density-matrix divide-and-conquer approach for electronic structure calculations of large molecules. Journal of Chemical Physics 103, 5674-5678 (1995).
56. Goedecker, S. Linear scaling electronic structure methods. Reviews of Modern Physics 71, 1085-1123 (1999).
57. Fattebert, J.-L. & Gygi, F. Linear scaling first-principles molecular dynamics with controlled accuracy. Computer Physics Communications 162, 24-36 (2004).
58. Chelikowsky, J. R., Saad, Y., Ögüt, S., Vasiliev, I. & Stathopoulos, A. Electronic structure methods for predicting the properties of materials: grids in space. Physica Status Solidi (b) 217, 173 (2000).