Million-to-Billion Atom Simulation of Chemical Reactions: Embedded Divide-and-Conquer and Hierarchical Cellular Decomposition Frameworks for Scalable Scientific Computing

Aiichiro Nakano,[a] Rajiv K. Kalia,[a] Ken-ichi Nomura,[a] Ashish Sharma,[a] Priya Vashishta,[a]
Fuyuki Shimojo[a,b]
[a]Collaboratory for Advanced Computing and Simulations, Department of Computer Science,
Department of Physics & Astronomy, Department of Materials Science & Engineering,
University of Southern California
[b]Department of Physics, Kumamoto University, 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, Division of Chemistry and Chemical Engineering,
California Institute of Technology
(duin, wag)@wag.caltech.edu

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

Abstract

Simulating chemical reactions involving billions of atoms has been a dream of scientists, with broad societal impacts. This paper realizes the dream through novel simulation methods, algorithms, and parallel computing and visualization techniques. We have designed $O(N)$ embedded divide-and-conquer (EDC) algorithms for 1) first principles-based parallel reactive force-field (P-ReaxFF) molecular dynamics (MD), and 2) density functional theory (DFT) on adaptive multigrids for quantum mechanical MD, based on a space-time multiresolution MD (MRMD) algorithm. To map these $O(N)$ algorithms onto parallel computers, we have developed a hierarchical cellular decomposition (HCD) framework, including 1) wavelet-based computational-space decomposition for adaptive load balancing, and 2) octree-based probabilistic visibility culling for interactive visualization of billion-atom datasets. Preliminary tests on 1,920 Itanium2 processors of the NASA Columbia supercomputer have achieved unprecedented scales of reactive atomistic simulations: 0.56 billion-atom P-ReaxFF and 1.4 million-atom (0.12 trillion grid points) EDC-DFT simulations, in addition to 18.9 billion-atom MRMD simulation.

Keywords: Linear-scaling algorithm, embedded divide-and-conquer, parallel computing, hierarchical cellular decomposition, molecular dynamics, reactive force field, quantum mechanics, density functional theory, load balancing, massive data visualization

1 Introduction

There is growing interest in large-scale molecular dynamics (MD) simulations [1,14,27,32] involving million-to-billion atoms, in which interatomic forces are computed quantum mechanically [5] to accurately describe chemical reactions. Such large reactive MD simulations would for the first time provide requisite coupling of chemical reactions, atomistic processes, and macroscopic materials phenomena, to solve a wide spectrum of problems of great societal impact. Examples of technological significance include: stress corrosion cracking,* where chemical

* Corrosion-related direct costs make up 3.1% of the gross domestic product in the US.

reactions at the crack tip are inseparable from long-range stress fields [30]; energetic nanomaterials to boost the impulse of rocket fuels, in which chemical reactions sustain shock waves (see Fig. 1) [39]; and micro-meteorite impact damages to the thermal and radiation protection layers of aerospace vehicles, understanding of which is essential for safer space flights. Emerging Petaflops computers could potentially extend the realm of quantum mechanical simulation to the macroscopic scales, but only if scalable simulation technologies were developed. We have assembled a multidisciplinary team of physicists, chemists, materials scientists and computer scientists, at USC, Caltech and NASA, to solve this challenging problem. The team has developed a scalable parallel computing framework for reactive atomistic simulations, based on data locality principles.

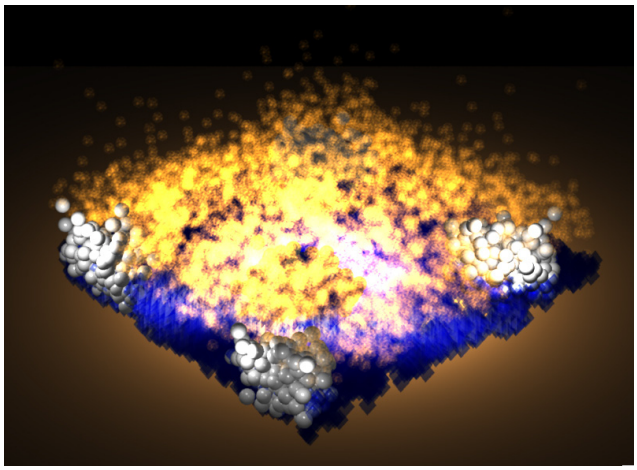

Figure 1: Reactive force-field molecular dynamics simulation of shock-initiated combustion of an energetic nanocomposite material (nitramine matrix embedded with aluminum nanoparticles).

The density functional theory (DFT) has reduced the exponentially complex quantum mechanical (QM) N-body problem to $O(N^3)$, by solving N one-electron problems self-consistently instead of an N-electron problem [13,18].† Unfortunately, DFT-based MD simulations [5] are rarely performed over $N \sim 10^2$ atoms because of the $O(N^3)$ computational complexity, which severely limits their scalability. In the past few years, two promising approaches have emerged toward achieving million-to-billion atom simulations of chemical reactions.

One computational approach toward QM-based million-atom MD simulations is to perform a number of small DFT calculations "on the fly" to compute interatomic forces quantum mechanically during an MD simulation. This concurrent DFT-based MD approach is best implemented with a divide-and-conquer algorithm [43]. This algorithm is based on a data locality principle called quantum nearsightedness [17], which naturally leads to $O(N)$ DFT calculations [9,10,37,43]. However, it is only in the past one year that $O(N)$ DFT algorithms, especially with large basis sets ($> 10^4$ unknowns per electronic wave function), have attained controlled error bounds, robust convergence properties, and energy conservation, to make large DFT-based MD simulations practical [9,38]. For example, we have recently designed an embedded divide-and-conquer DFT algorithm, in which a hierarchical grid technique combines multigrid preconditioning and adaptive fine mesh generation [38]. Here we present the first million-atom DFT-based MD calculation, where electronic wave functions are represented on 10^{11} grid points.

Alternative to this concurrent DFT-MD approach is a sequential DFT-informed MD approach, which employs environment-dependent interatomic potentials based on variable atomic charges to describe charge transfers and reactive bond orders to describe chemical bond formation and breakage. In our first principles-based reactive force-field (ReaxFF) approach, the parameters in the interatomic potentials are "trained" to best fit thousands of DFT calculations on small ($N \sim 10$) clusters of various atomic-species combinations [7,39]. Because of its $O(N^3)$ complexity associated with the variable N-charge problem and the multitude of atomic n-tuple information (n = 2-6) required to compute interatomic forces, however, parallelization of the ReaxFF algorithm has only seen limited success, and the largest ReaxFF-MD simulations to date have involved $N < 10^4$ atoms. This paper presents a new $O(N)$ parallel ReaxFF algorithm, which for the first time enables 10^9-atom MD simulations of chemical reactions.

A major technical contribution of this paper is a unified embedded divide-and-conquer (EDC) algorithmic framework for designing linear-scaling algorithms for broad scientific and engineering problems, with specific

† Walter Kohn received the 1998 Nobel prize in chemistry for the development of the density functional theory.

applications to the ReaxFF and DFT based MD simulations. Mapping these $O(N)$ algorithms onto multi-Teraflops to Petaflops parallel computers, however, poses a number of challenges, e.g., achieving high scalability for irregularly distributed billion atoms, and visualizing the resulting billion-atom datasets. To overcome these challenges, the second contribution of this paper is a hierarchical cellular decomposition (HCD) framework, which is aware of deep memory hierarchies, by maximally exposing data locality and exploiting parallelism at each decomposition level. The framework features topology-preserving computational space decomposition and wavelet-based adaptive load balancing. We also apply the framework to achieve the first interactive visualization of a billion-atom dataset, based on multilevel probabilistic visibility culling algorithms. The major accomplishment of this paper is the unprecedented scales of chemically reactive atomistic simulations on the NASA Columbia supercomputer. Preliminary tests on 1,920 Itanium2 processors have achieved 0.56 billion-atom ReaxFF and 1.4 million-atom (0.12 trillion grid points) DFT simulations, in addition to 18.9 billion-atom MD simulation. Currently, benchmark tests that are five-times larger are under way on Columbia, and will be reported in the final paper.

This paper is organized as follows. In the next section, we describe the EDC algorithmic framework. Section 3 discusses the HCD parallel computing and visualization framework. Results of benchmark tests are given in Sec. 4, and Sec. 5 contains conclusions.

2 Linear-scaling embedded divide-and-conquer simulation algorithms

We have developed a unified algorithmic framework to design linear-scaling algorithms for broad scientific and engineering problems, based on data locality principles. In the ***embedded divide-and-conquer (EDC) algorithms***, spatially localized subproblems are solved in a global embedding field, which is efficiently computed with tree-based algorithms (see Fig. 2). Examples of the embedding field are the electrostatic field in molecular dynamics (MD) simulations and the self-consistent Kohn-Sham potential in the density functional theory (DFT).

Specifically, we have developed a suite of linear-scaling MD simulation algorithms for materials simulations, in which interatomic forces are computed with increasing accuracy and complexity. The linear-scaling algorithms encompass a wide spectrum of physical reality: 1) classical MD based on a many-body interatomic potential model, which involves the formally $O(N^2)$ N-body problem; 2) environment-dependent, reactive force-field (ReaxFF) MD, which involves the $O(N^3)$ variable N-charge problem; and 3) quantum mechanical (QM) calculation based on the DFT, to provide approximate solutions to the exponentially complex quantum N-body problem.

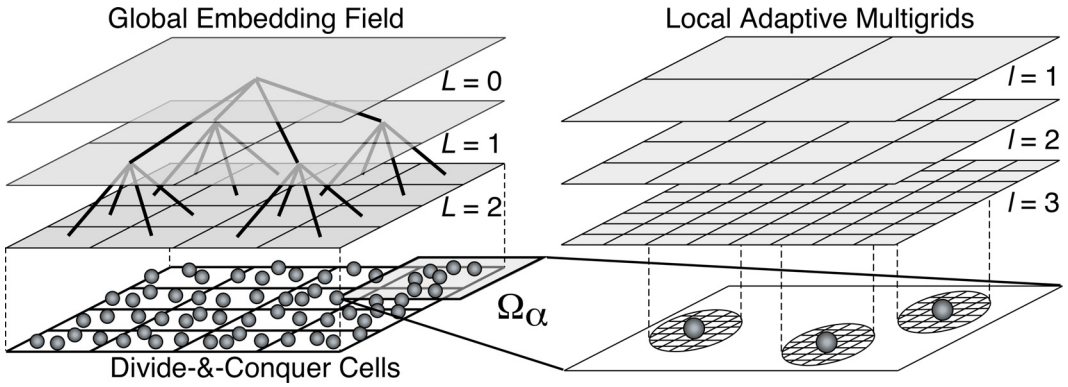

Figure 2: Schematic of an embedded divide-and-conquer (EDC) algorithm. (Left) The physical space is subdivided into spatially localized cells, with local atoms constituting subproblems (bottom), which are embedded in a global field (shaded) solved with a tree-based algorithm. (Right) To solve the subproblem in domain Ω_α in the EDC-DFT algorithm, coarse multigrids (gray) are used to accelerate iterative solutions on the original real-space grid (corresponding to the grid refinement level, $l = 3$). The bottom panel shows that fine grids are adaptively generated near the atoms (spheres) to accurately operate the ionic pseudopotentials on the electronic wave functions.

2.1 Space-time multiresolution molecular dynamics algorithm

We have developed chemically reactive $O(N)$ MD simulation algorithms, on the basis of our ***space-time multiresolution molecular dynamics (MRMD)*** algorithm [27]. In the MD approach, one obtains the phase-space trajectories of the system (positions and velocities of all atoms at all time). Atomic force laws for describing how atoms interact with each other is mathematically encoded in the interatomic potential energy, $E_{\mathrm{MD}}(\mathbf{r}^N)$, which is a

function of the positions of all N atoms, $\mathbf{r}^N = \{\mathbf{r}_1, \mathbf{r}_2, ..., \mathbf{r}_N\}$, in the system. In our many-body interatomic potential scheme, $E_{MD}(\mathbf{r}^N)$ is expressed as an analytic function that depends on relative positions of atomic pairs and triplets. Time evolution of $\mathbf{r}^N$ is governed by a set of coupled ordinary differential equations. For interatomic potentials with finite ranges, the computational cost is made $O(N)$ using a linked-list cell approach [27]. 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 $O(N)$ [11,26,28]. 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 the 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 novel complex charge method [28]. The MRMD algorithm also utilizes temporal locality through multiple time stepping (MTS), which uses different force-update schedules for different force components [21,23,26,41]. Specifically, forces from the nearest-neighbor atoms are computed at every MD step, whereas forces from farther atoms are updated less frequently.

For parallelization of MD simulations, we use spatial decomposition [27]. The total volume of the system 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 atomic positions due to a time-stepping procedure, some atoms may have moved out of its subsystem. These atoms are "migrated" to the proper neighbor nodes. With the spatial decomposition, the computation scales as N/P, while communication scales in proportion to $(N/P)^{2/3}$ for an N-atom system. Tree-based algorithms such as the FMM incur an $O(\log P)$ overhead, which is negligible for coarse-grained ($N/P > 10^4$) applications.

2.2 Parallel reactive force-filed molecular dynamics

Physical realism of MD simulations is greatly enhanced by incorporating variable atomic charges and reactive bond orders, which dynamically adapt to the local environment [7,39]. However, the increased realism of this reactive force-field (ReaxFF) MD is accompanied by increased computational complexity, $O(N^3)$, for solving a dense linear system of equations to determine atomic charges at every MD step, i.e., the variable N-charge problem. We have developed a scalable ***parallel reactive force-field (P-ReaxFF) MD algorithm***, which reduces the complexity to $O(N)$ by combining the FMM based on spatial locality and an iterative minimization approach to utilize the temporal locality of the solutions. To further accelerate the convergence, we use a ***multilevel preconditioned conjugate-gradient (MPCG)*** method [4,20], by splitting the Coulomb-interaction matrix into short- and long-range components and using the sparse short-range matrix as a preconditioner. The extensive use of the sparse preconditioner enhances the data locality, and thereby improves the parallel efficiency [20].

The chemical bond order, B_{ij}, is an attribute of an atomic pair, (i, j), and changes dynamically depending on the local environment. In ReaxFF, the interatomic potential energies between atomic pairs, triplets, and quartets depend on the bond orders of all constituent atomic pairs. Force calculations in ReaxFF MD thus include up to atomic 4-tuples explicitly, and require information on 6-tuples implicitly through the bond orders [7]. To efficiently handle the resulting multiple interaction ranges, the P-ReaxFF employs a multilayer cellular decomposition (MCD) scheme for caching atomic n-tuple (n = 2-6) information (see Sec. 3).

2.3 Linear-scaling quantum-mechanical calculation based on the density functional theory

An atom consists of a nucleus and surrounding electrons, and quantum mechanics explicitly treats the electronic degrees-of-freedom. The density functional theory (DFT) reduces the exponentially complex quantum N-body problem to a self-consistent matrix eigenvalue problem, which can be solved with $O(M^3)$ operations (M is the number of independent electronic wave functions and is on the order of N) [13,18]. The DFT can be formulated as a minimization of the energy functional, $E_{QM}(\mathbf{r}^N, \psi^M)$, with respect to electronic wave functions, $\psi^M(\mathbf{r}) = \{\psi_1(\mathbf{r}), \psi_2(\mathbf{r}), ..., \psi_M(\mathbf{r})\}$, subject to orthonormality constraints.

For scalable quantum-mechanical calculations [15], linear-scaling DFT algorithms are essential [9,10,37,43]. At SC2001, we presented an $O(M)$ DFT algorithm based on unconstrained minimization of a modified energy functional and a localized-basis approximation [27]. Recently, we have designed a new $O(M)$ DFT algorithm with considerably more robust convergence properties, controlled error bounds, and the energy conservation during MD simulations [38]. The convergence of the new algorithm, as well as its energy conservation in MD simulations, has been verified for nontrivial problems such as amorphous CdSe and liquid Rb [38]. The ***embedded divide-and-conquer density functional theory (EDC-DFT) algorithm*** represents the physical system as a union of overlapping spatial domains, $\Omega = \cup_\alpha \Omega_\alpha$ (see Fig. 2), 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 electronic state (i.e., 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 [3,38].

The DFT calculation in each domain is performed using a real-space approach, in which electronic wave functions are numerically represented on grid points, see Fig. 2 [6]. The real-space grid is augmented with coarser multigrids to accelerate the convergence of iterative solutions [3,8]. Furthermore, a finer grid is adaptively generated near every atom, in order to accurately operate ionic pseudopotentials to describe electron-ion interactions [38]. We include electron-ion interactions using norm-conserving pseudopotentials [40] and the exchange-correlation energy in a generalized gradient approximation [33].

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. For each domain, its electronic structure is computed independently, with little information needed from other compute nodes (only the global density but not individual wave functions is communicated). The resulting large computation/communication ratio makes this approach highly scalable on parallel computers.

3 Hierarchical cellular decomposition parallelization framework

Data locality principles are key to developing a scalable parallel computing framework as well. We have developed a ***hierarchical cellular decomposition (HCD) framework*** to map the above $O(N)$ algorithms onto massively parallel computers with deep memory hierarchies. The HCD maximally exposes data locality and exploits parallelism at multiple decomposition levels (see Fig. 3).

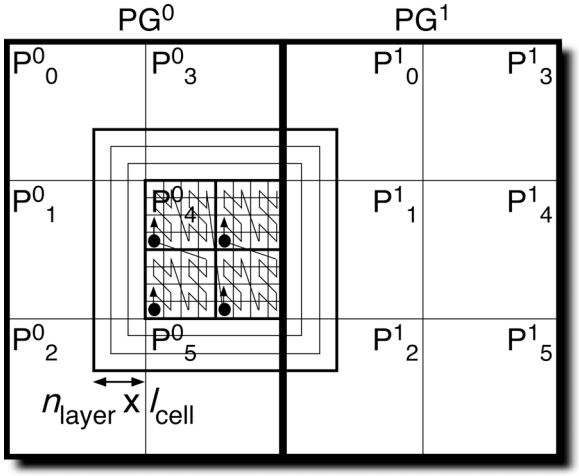

Figure 3: In hierarchical cellular decomposition (HCD), the physical volume is subdivided into process groups, PG^γ, which, in turn, are spatially decomposed into processes, P^γ_π. Each process consists of a number of computational cells (e.g., linked-list cells in MD or domains in EDC-DFT) of size l_{cell}, which are traversed by concurrent threads (denoted by dots with arrows) to compute interatomic forces in blocks of cells. P^γ_π is dynamically augmented with n_{layer} layers of cached cells from neighbor processes.

At the finest level, EDC algorithms consist of computational cells—linked-list cells (which are identical to the octree leaf cells in the FMM) [26,28] in MRMD and P-ReaxFF, or domains in EDC-DFT [38] (see Fig. 3). 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. 3) in spatial decomposition, which contains a large number of computational cells. Our EDC algorithms are implemented as hybrid message passing interface (MPI)-shared memory (OpenMP) programs, in which inter-node communication for caching and migrating atoms between P^γ_π's is handled with messages [27], whereas loops over cells within each P^γ_π (or MPI process) are parallelized with threads (denoted as dots with allows in Fig. 3). 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. The cellular data

structure also 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, see Fig. 3) [19]. Furthermore, the cell size is made tunable [42] to optimize the performance. For MRMD, we have observed ~10% performance gain by the data and computation re-ordering and cell-size optimization. The computational cells are also used in the multilayer cellular decomposition (MCD) scheme for inter-node caching of atomic n-tuple (n = 2-6) information (see Fig. 3), where n changes dynamically in the MTS or MPCG algorithm (see Sec. 2).

On top of the computational cells, cell blocks, and spatial-decomposition subsystems, the HCD framework introduces a coarser level of decomposition by defining process groups as MPI Communicators ($PG^{\prime}$ in Fig. 3). This provides a mechanism to optimize EDC applications distributed over a loosely coupled collection of parallel computers, e.g., a Grid of globally distributed parallel computers [2,16]. Our programs are designed to minimize global operations across $PG^{\prime}$'s, and to overlap computations with inter-group communications [16].

3.1 Wavelet-based computational-space decomposition for adaptive load balancing

The HCD framework includes a topology-preserving computational spatial decomposition scheme to minimize latency through structured message passing [24] and load-imbalance/communication costs through a novel wavelet-based load-balancing scheme [22].

The load-balancing problem can be stated as an optimization problem, i.e., one minimizes the load-imbalance cost as well as the size and the number of messages [25]. To minimize the number of messages, we preserve the 3D mesh topology, so that message passing is performed in a structured way in only 6 steps. To minimize the load imbalance cost as well as the message size, we have developed a computational-space decomposition scheme [24]. The main idea of this scheme is that the computational space shrinks where the workload density is high, so that the workload is uniformly distributed in the computational space. To implement the curved computational space, we introduce a curvilinear coordinate transformation. The sum of load imbalance and communication costs is then minimized as a functional of the coordinate transformation, using simulated annealing. We have found that wavelet representation leads to compact representation of curved partition boundaries, and accordingly speeds up the convergence of the minimization procedure [22].

3.2 Probabilistic octree algorithms for massive data visualization

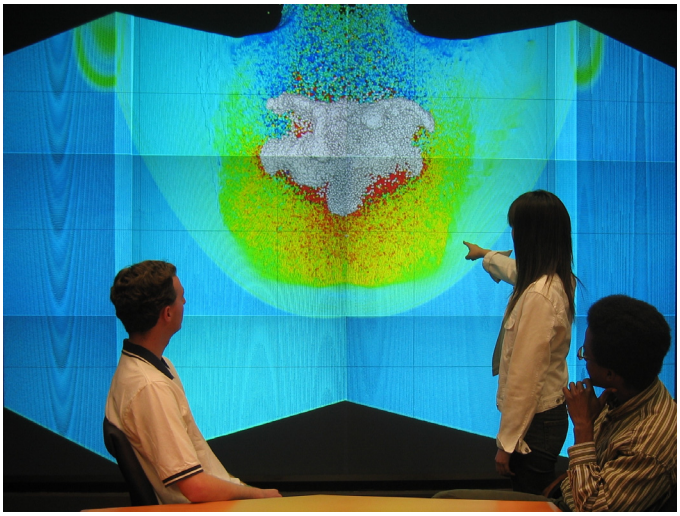

Figure 4: Rendering of an MD simulation on a tiled display, showing hypervelocity impact damage of a ceramic plate with impact velocity 15 km/s, where one quarter of the system is cut to show the internal pressure distribution (the projectile is shown in white). Such simulations help design thermal and radiation protection systems of aerospace vehicles, which are tolerant to micro-meteorite impacts (where impact speeds are as high as 40 km/s).

Data locality also plays a critical role in designing scalable data visualization and management techniques in the HCD framework. Interactive exploration of large-scale atomistic simulations is important for identifying and tracking atomic features that are responsible for macroscopic phenomena. We have developed a scalable

visualization system, Atomsviewer, to allow the viewer to walk through a billion atoms [36]. The system uses the octree data structure as an efficient abstraction mechanism to extract atoms within the field-of-view (view frustum culling). A novel probabilistic approach then removes far atoms that are hidden by other atoms (occlusion culling). The system off-loads these culling tasks to a Linux cluster, so that the graphics server is dedicated to the rendering of reduced data subsets. Furthermore, we use a machine-learning approach to predict the user's next movement and prefetch data from the Linux cluster to the graphics server. Finally, multiresolution rendering is used to further speed up the rendering. The resulting system renders a billion-atom dataset at nearly interactive frame rates on a dual processor SGI Onyx2 with an InfiniteReality2 graphics pipeline, connected to a 4-processor Linux cluster [36]. At the Collaboratory for Advanced Computing and Simulations (CACS) at USC, the Atomsviewer is used on an 8'×14' tiled display wall (Fig. 4) driven by a 26-processor Linux cluster and an immersive and interactive virtual environment called ImmersaDesk.

A billion-atom MD simulation produces 100GB of data per MD time step, including atomic species, positions, velocities, and stress tensor components. For scalable input/output (I/O) of such large datasets, the HCD framework uses a data compression algorithm based on data locality. It uses octree indexing and sorts atoms accordingly on the resulting spacefilling curve [31]. 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 actual MD data with user-controlled error bound.

4 Performance tests

The three parallel algorithms—MRMD, P-ReaxFF, and EDC-DFT—are portable and have been run on various platforms, including Intel Itanium2, Intel Xeon, AMD Opteron and IBM Power4 based parallel computers. This section presents performance tests of the three algorithms on some of the architectures.

There is a trade-off between spatial-decomposition (MPI) and thread (OpenMP) parallelisms [12,34,35] in our hybrid MPI+OpenMP programs. While spatial decomposition involves extra computation on cached cells from neighbor subsystems, its disjoint memory subspaces are free from shared-memory protocol overhead. A marked contrast in this trade-off is observed between the MRMD and P-ReaxFF algorithms. The MRMD is characterized by a low caching overhead and a small number of floating-point operations per memory access, since it mostly consists of look-ups for pre-computed interatomic force tables. In contrast, P-ReaxFF has a large caching overhead for 6-tuple information and a large computation/memory-access ratio. Table 1 shows the execution time of the MRMD algorithm for an 8,232,000-atom silica system and that of the P-ReaxFF algorithm for a 290,304-atom RDX system on $P = 8$ processors in an 8-way 1.5GHz Power4 node. (The test was performed on the iceberg Power4 system at the Arctic Region Supercomputing Center.) We compare 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. The optimal combination of (n_{td}, n_p) with the minimum execution time is (1, 8) for the MRMD and is (4,2) for the P-ReaxFF.

Number of OpenMP threads, n_{td}	Number of MPI processes, n_p	Execution time/MD time step (sec)	
		MRMD	P-ReaxFF
1	8	**4.19**	62.5
2	4	5.75	58.9
4	2	8.60	**54.9**
8	1	12.5	120

Table 1: Execution time per MD time step on $P = n_{td} \times n_p = 8$ processors in an 8-way 1.5GHz Power4 node, with different combinations of the number of OpenMP threads per MPI process, n_{td}, and that of MPI processes, n_p, for: (1) the MRMD algorithm for an 8,232,000-atom silica system; and (2) the P-ReaxFF algorithm for a 290,304-atom RDX system. The minimum execution time for each algorithm is typed in boldface.

Scalability tests of the three parallel algorithms—MRMD, P-ReaxFF, and EDC-DFT—have been performed on the 10,240-processor Columbia supercomputer at the NASA Ames Research Center. The SGI Altix 3000 system uses the NUMAflex global shared-memory architecture, which packages processors, memory, I/O, interconnect, graphics, and storage into modular components called bricks. 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 architecture operates at 1.5GHz and is capable of issuing two multiply-add operations per cycle for a peak performance of 6Gflop/s. 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 approximately 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 13Tflop/s peak capability platform, and was the basis of the computations reported here. The final paper will report results from the full 10,240-processor system.

Figure 5 shows the execution time of the MRMD algorithm for silica material as a function of the number of processors, P. In this and following figures, we set n_{td} to 1. We scale the system 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, where 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 parallel efficiency of the MRMD algorithm is 0.87. Also the algorithm involves very small communication time, see Fig. 5.

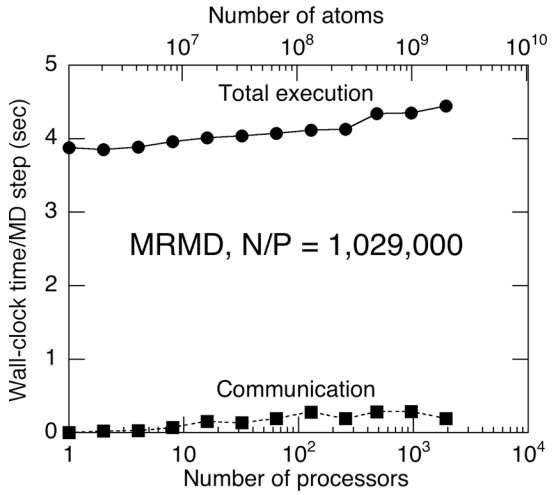

Figure 5: Total execution (circles) and communication (squares) times per MD time step as a function of the number of processors for the MRMD algorithm with scaled workloads—1,029,000 P atom silica systems on P processors (P = 1, ..., 1,920) of Columbia.

Figure 6 shows the total execution (circles) and communication (squares) times per MD time step of the P-ReaxFF MD algorithm with scaled workloads—36,288P-atom RDX (1,3,5-trinitro-1,3,5-triazine) systems on P processors (P = 1, ..., 1,920). The computation time includes 3 conjugate gradient (CG) iterations to solve the electronegativity equalization problem for determining atomic charges at each MD time step. The execution time increases only slightly as a function of P, and the parallel efficiency is 0.91 on 1,920 processors.

Figure 7 shows the performance of the EDC-DFT based MD algorithm with scaled workloads—720P-atom alumina systems on P processors (P = 1, ..., 1,920). In the EDC-DFT calculations, each domain of size

6.66×5.76×6.06Å^3 contains 40 electronic wave functions (i.e., Kohn-Sham orbitals), where each wave function is represented on 28^3 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 and 5,529,600 electronic wave functions on 121,385,779,200 grid points, for which the parallel efficiency is 0.76.

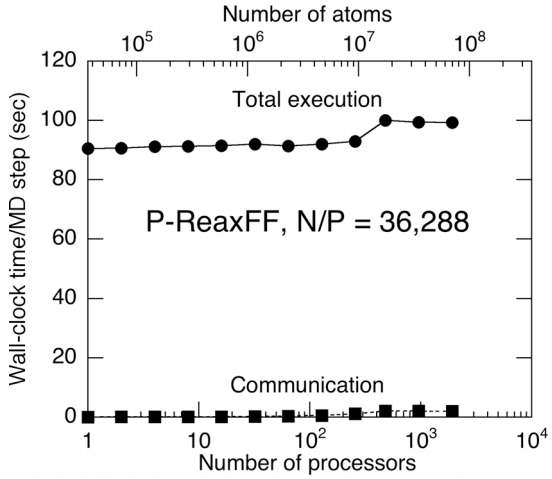

Figure 6: Total execution (circles) and communication (squares) times per MD time step as a function of the number of processors for the P-ReaxFF MD algorithm with scaled workloads—36,288 *P* atom RDX systems on *P* processors (*P* = 1, ..., 1,920) of Columbia.

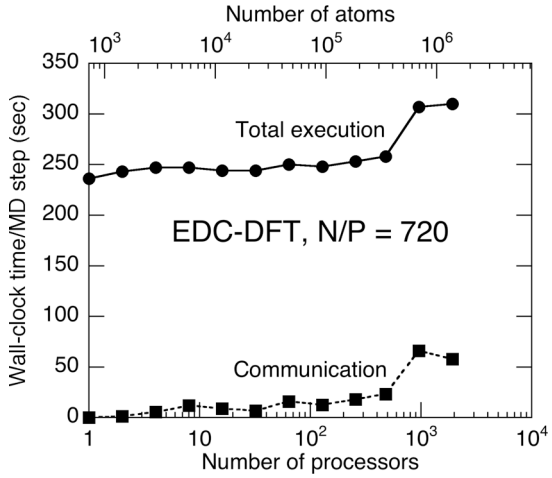

Figure 7: Total execution (circles) and communication (squares) times per MD time step as a function of the number of processors for the EDC-DFT MD algorithm with scaled workloads—720*P* atom alumina systems on *P* processors (*P* = 1, ..., 1,920) of Columbia.

Major design parameters for MD simulations of materials include the number of atoms in the simulated system and the methods to compute interatomic forces (classically in MRMD, semi-empirically in P-ReaxFF MD, or quantum-mechanically in EDC-DFT MD). Figure 8 shows a design-space diagram for classical and quantum-mechanical MD simulations on 1,920 Itanium2 processors of Columbia. The largest benchmark tests in this study include 18,925,056,000-atom MRMD, 557,383,680-atom P-ReaxFF, and 1,382,400-atom (121,385,779,200 electronic degrees-of-freedom) EDC-DFT calculations. The figure demonstrates perfect linear scaling for all the three algorithms, with prefactors spanning five orders-of-magnitude. Only exception is the P-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 $O((N/P)^{-1/3})$.

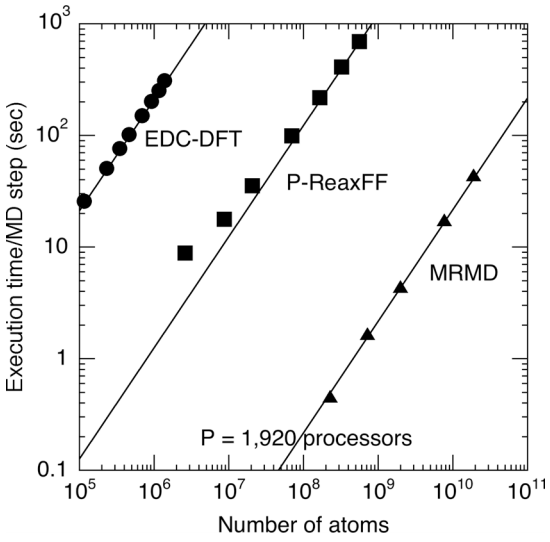

Figure 8: Design-space diagram for MD simulations on 1,920 Itanium2 processors of Columbia. The figure shows the total execution per MD step as a function of the number of atoms for three linear-scaling algorithms: Quantum-mechanical MD based on the embedded divide-and-conquer density functional theory (EDC-DFT, circles); parallel reactive force-field MD (P-ReaxFF, squares); space-time multiresolution MD (MRMD, triangles). Lines show ideal $O(N)$ scaling.

5 Conclusions

Since we demonstrated a hundred thousand-atom reactive molecular dynamics simulation based on the density functional theory at SC2001 [27], significant progresses have been made in simulation methods (e.g., first principles-based reactive force-field molecular dynamics), linear-scaling algorithms (e.g., embedded divide-and-conquer density functional theory algorithm on adaptive multigrids), and scalable parallel computing technologies (e.g., hierarchical cellular decomposition with wavelet-based adaptive load balancing). Based on these innovations, we are currently performing larger performance tests on NASA Columbia, including 2.8 billion-atom reactive force-field and 6.9 million-atom (0.61 trillion grid points) density functional theory based molecular dynamics simulations of chemical reactions, in addition to 95 billion-atom molecular dynamics simulation. These results show considerable promise for atomistic simulations of chemical reactions with unprecedented scales and accuracy on emerging Petaflops computer architectures.

The hierarchy of molecular dynamics simulation algorithms developed in this paper can be integrated seamlessly into a hierarchical simulation framework, which embeds accurate but compute-intensive simulations in coarse simulations only when and where high fidelity is required [29,30]. Our hierarchical simulation framework consists of: 1) hierarchical division of the physical system into subsystems of decreasing sizes and increasing quality-of-solution (QoSn) requirements, $S_0 \supset S_1 \supset ... \supset S_n$; and 2) a suite of simulation services, M_α ($\alpha = 0, 1, ..., n$), of ascending order of accuracy (e.g., MRMD $\prec$ P-ReaxFF $\prec$ EDC-DFT). In our additive hybridization scheme, an accurate estimate of the energy of the entire system is obtained from the recurrence relation, $E_\alpha(S_i) = E_{\alpha-1}(S_i) + E_\alpha(S_{i+1}) - E_{\alpha-1}(S_{i+1})$. The scalable parallel simulation techniques presented in this paper, with such a dynamically extensible hierarchical simulation framework, should open up enormous opportunities for scientific computing on high-end computers. Together with the massive data visualization techniques in this paper, such ultrascale simulations promise to bring in fundamental advances in science.

Acknowledgements

This work was partially supported by AFOSR-DURINT, ARO-MURI, DARPA-PROM, DOE, and NSF. Benchmark tests were performed using the NASA Columbia supercomputers at the NASA Ames Research Center, and at Department of Defense's Major Shared Resource Centers under a DoD Challenge Project. Programs have been developed using the 600-processor (2.0Tflops) Intel Xeon-AMD Opteron-Apple G5 clusters at the

Collaboratory for Advanced Computing and Simulations and the 2,000-processor (7.3Tflops) Xeon cluster at the High Performance Computing Center at the University of Southern California. Visualization was performed on the 8'×14' tiled display and the immersive and interactive virtual environment, ImmersaDesk, at the Visualization Collaboratorium at the USC. The authors thank Walter Kohn and Emil Prodan for discussions on $O(N)$ density functional theory algorithms, Bhupesh Bansal for visualization data processing, and Davin Chan, Johnny Chang, Bob Ciotti, Edward Hook, Art Lazanoff, Bron Nelson, Charles Niggley, and William Thigpen for technical discussions on Columbia.

References

[1] F. F. Abraham, R. Walkup, H. J. Gao, M. Duchaineau, T. D. de la Rubia, and M. Seager. Simulating materials failure by using up to one billion atoms and the world's fastest computer: brittle fracture. *Proceedings of the National Academy of Science*, 99:5777-5782 (2002).

[2] G. Allen, T. Dramlitsch, I. Foster, N. T. Karonis, M. Ripeanu, E. Seidel, and B. Toonen. Supporting efficient execution in heterogeneous distributed computing environments with Cactus and Globus. In *Proceedings of SC2001*. ACM, 2001.

[3] T. L. Beck. Real-space mesh techniques in density-functional theory. *Reviews of Modern Physics*, 72:1041-1080 (2000).

[4] T. J. Campbell, R. K. Kalia, A. Nakano, P. Vashishta, S. Ogata, and S. Rodgers. Dynamics of oxidation of aluminum nanoclusters using variable charge molecular-dynamics simulations on parallel computers. *Physical Review Letters*, 82:4866-4869 (1999).

[5] R. Car and M. Parrinello. Unified approach for molecular dynamics and density functional theory. *Physical Review Letters*, 55:2471-2474 (1985).

[6] J. R. Chelikowsky, Y. Saad, S. Ögüt, I. Vasiliev, and A. Stathopoulos. Electronic structure methods for predicting the properties of materials: Grids in space. *Phyica Status Solidi (b)*, 217:173-195 (2000).

[7] A. C. T. van Duin, S. Dasgupta, F. Lorant, and W. A. Goddard, III. ReaxFF: a reactive force field for hydrocarbons. *Journal of Physical Chemistry A*, 105:9396-9409 (2001).

[8] J.-L. Fattebert and J. Bernholc. Towards grid-based $O(N)$ density-functional theory methods: optimized nonorthogonal orbitals and multigrid acceleration. *Physical Review B*, 62:1713-1722 (2000).

[9] J.-L. Fattebert and F. Gygi. Linear scaling first-principles molecular dynamics with controlled accuracy. *Computer Physics Communications*, 162:24-36 (2004).

[10] S. Goedecker. Linear scaling electronic structure methods. *Reviews of Modern Physics*, 71:1085-1123 (1999).

[11] L. Greengard and V. Rokhlin. A fast algorithm for particle simulations. *Journal of Computational Physics*, 73:325-348 (1987).

[12] D. S. Henty. Performance of hybrid message-passing and shared-memory parallelism for discrete element modeling. In *Proceedings of SC2000*. IEEE, 2000.

[13] P. Hohenberg and W. Kohn. Inhomogeneous electron gas. *Physical Review*, 136:B864-B871 (1964).

[14] K. Kadau, T. C. Germann, P. S. Lomdahl, and B. Lee Holian. Microscopic view of structural phase transitions induced by shock waves. *Science*, 296:1681-1684 (2002).

[15] R. A. Kendall, E. Apra, D. E. Bernholdt, E. J. Bylaska, M. Dupuis, G. I. Fann, R. J. Harrison, J. Ju, J. A. Nichols, J. Nieplocha, T. P. Straatsma, T. L. Windus, and A. T. Wong. High performance computational chemistry: an overview of NWChem a distributed parallel application. *Computer Physics Communications*, 128:260-283 (2000).

[16] H. Kikuchi, R. K. Kalia, A. Nakano, P. Vashishta, F. Shimojo, and S. Saini. Collaborative simulation Grid: multiscale quantum-mechanical/classical atomistic simulations on distributed PC clusters in the US and Japan. In *Proceedings of SC2002*. IEEE, 2002.

[17] W. Kohn. Density functional and density matrix method scaling linearly with the number of atoms. *Physical Review Letters*, 76:3168-3171 (1996); E. Prodan and W. Kohn, private communication.

[18] W. Kohn and P. Vashishta. General density functional theory. In *Inhomogeneous Electron Gas*, eds. N. H. March and S. Lundqvist, pages 79-184. Plenum, New York, 1983.

[19] J. Mellor-Crummey, D. Whalley, and K. Kennedy. Improving memory hierarchy performance for irregular applications using data and computation reorderings. *International Journal of Parallel Programming*, 29:217-247 (2001).

[20] A. Nakano. Parallel multilevel preconditioned conjugate-gradient approach to variable-charge molecular dynamics. *Computer Physics Communications*, 104:59-69 (1997).

[21] A. Nakano. Fuzzy clustering approach to hierarchical molecular dynamics simulation of multiscale materials phenomena. *Computer Physics Communications*, 105:139-150 (1997).
[22] A. Nakano. Multiresolution load balancing in curved space: the wavelet representation. *Concurrency: Practice and Experience*, 11:343-353 (1999).
[23] A. Nakano. A rigid-body based multiple time-scale molecular dynamics simulation of nanophase materials. *The International Journal of High Performance Computing Applications*, 13:154-162 (1999).
[24] A. Nakano and T. J. Campbell. An adaptive curvilinear-coordinate approach to dynamic load balancing of parallel multi-resolution molecular dynamics, *Parallel Computing*, 23:1461-1478 (1997).
[25] A. Nakano, R. K. Kalia, A. Sharma, and P. Vashishta. Virtualization-aware application framework for hierarchical multiscale simulations on a Grid. *Annual Review of Scalable Computing*, 6 (2005) in press.
[26] A. Nakano, R. K. Kalia, and P. Vashishta. Multiresolution molecular dynamics algorithm for realistic materials modeling on parallel computers. *Computer Physics Communications*, 83:197-214 (1994).
[27] A. Nakano, R. K. Kalia, P. Vashishta, T. J. Campbell, S. Ogata, F. Shimojo, and S. Saini. Scalable atomistic simulation algorithms for materials research. *Scientific Programming*, 10:263-270 (2002); **The Best Paper Award at *IEEE/ACM SC2001***.
[28] S. Ogata, T. J. Campbell, R. K. Kalia, A. Nakano, P. Vashishta, and S. Vemparala. Scalable and portable implementation of the fast multipole method on parallel computers. *Computer Physics Communications*, 153:445-461 (2003).
[29] S. Ogata, E. Lidorikis, F. Shimojo, A. Nakano, P. Vashishta, and R. K. Kalia. Hybrid finite-element/molecular-dynamics/electronic-density-functional approach to materials simulations on parallel computers. *Computer Physics Communications*, 138:143-154 (2001).
[30] S. Ogata, F. Shimojo, R. K. Kalia, A. Nakano, and P. Vashishta. 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).
[31] A. Omeltchenko, T. J. Campbell, R. K. Kalia, X. Liu, A. Nakano, and P. Vashishta. Scalable I/O of large-scale molecular-dynamics simulations: a data-compression algorithm. *Computer Physics Communications*, 131:78-85 (2000).
[32] J. Phillips, G. Zheng, S. Kumar, and L. V. Kalé. NAMD: biomolecular simulation on thousands of processors. In *Proceedings of SC02*. IEEE, 2002.
[33] J. P. Perdew, K. Burke, and M. Ernzerhof. Generalized gradient approximation made simple. *Physical Review Letters*, 77:3865-3868 (1996).
[34] H. Shan, J. P. Singh, L. Oliker, and R. Biswas. A comparison of three programming models for adaptive applications on the Origin2000. *Journal of Parallel and Distributed Computing*, 62:241-266 (2002); **The Best Student Paper Award at *IEEE/ACM SC2000***.
[35] H. Shan, J. P. Singh, L. Oliker, and R. Biswas. Message passing and shared address space parallelism on an SMP cluster. *Parallel Computing*, 29:167-186 (2003).
[36] A. Sharma, A. Nakano, R. K. Kalia, P. Vashishta, S. Kodiyalam, P. Miller, W. Zhao, X. Liu, T. J. Campbell, and A. Haas. Immersive and interactive exploration of billion-atom systems. *Presence: Teleoperators and Virtual Environments*, 12:85-95 (2003); **Best Paper of *IEEE Virtual Reality 2002***.
[37] F. Shimojo, R. K. Kalia, A. Nakano, and P. Vashishta. Linear-scaling density-functional-theory calculations of electronic structure based on real-space grids: design, analysis, and scalability test of parallel algorithms. *Computer Physics Communications*, 140:303-314 (2001).
[38] F. Shimojo, R. K. Kalia, A. Nakano, and P. Vashishta. 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).
[39] A. Strachan, A. C. T. van Duin, D. Chakraborty, S. Dasgupta, and W. A. Goddard, III. Shock waves in high-energy materials: initial chemical events in nitramine RDX. *Physical Review Letters*, 91:098301:1-4 (2003).
[40] N. Troullier and J. L. Martins. Efficient pseudopotentials for plane-wave calculations. *Physical Review B*, 43:1993-2006 (1991).
[41] M. E. Tuckerman, D. A. Yarne, S. O. Samuelson, A. L. Hughes, and G. J. Martyna. Exploiting multiple levels of parallelism in molecular dynamics based calculations via modern techniques and software paradigms on distributed memory computers. *Computer Physics Communications*, 128:333-376 (2000).
[42] R. C. Whaley, A. Petitet, and J. J. Dongarra. Automated empirical optimizations of software and the ATLAS project. *Parallel Computing*, 27:3-35 (2001).
[43] W. Yang. Direct calculation of electron density in density-functional theory. *Physical Review Letters*, 66:1438-1441 (1991).