A virtualization-aware application framework is developed, based on data-locality principles, to perform hierarchical multiscale simulations of materials on a Grid of distributed computing and visualization platforms. The framework combines: linear-scaling algorithms based on space-time multiresolution techniques; topology-preserving computational-space decomposition with wavelet-based adaptive load balancing; and immersive and interactive visualization of billion-atom datasets. Multiscale simulations are performed on a Grid to seamlessly integrate quantum mechanical calculation based on the density functional theory and atomistic simulation based on the molecular dynamics method.

1. Introduction

Metacomputing on a Grid of geographically distributed Teraflop-to-Petaflop computers and immersive virtual reality environments connected via high-speed networks will revolutionize science and engineering, by enabling hybrid simulations that integrate multiple expertise distributed globally. Such multidisciplinary applications are
emerging at the forefront of computational science and engineering. For example, hierarchical multiscale simulations embed accurate calculations—e.g., quantum mechanical (QM) calculations to handle chemical reactions—within coarse models—e.g., molecular dynamics (MD) simulation to describe large-scale atomistic processes and finite element (FE) calculation for continuum mechanics—only where/when high-fidelity modeling is required.\textsuperscript{3-5}

On a Grid, applications will likely be virtualized, i.e., the user will not know on which computers the code is running. However, virtualization of high-end applications (such as billion-atom MD and thousand-atom QM simulations) is an unsolved challenge. To support virtualization, applications need to be: 1) scalable from a single processor to thousands of processors; 2) portable from one architecture to another in terms of performance; and 3) adaptive to dynamically changing computing resources. In the past years, we have been developing a virtualization aware application framework for hierarchical multiscale materials simulations, based on data locality principles.

Our framework includes \(O(N)\) algorithms (\(N\) is the problem size) for broad applications such as: 1) space-time multiresolution molecular dynamics (MRMD) algorithm for large-scale atomistic simulations; and 2) linear-scaling density functional theory (LSDFT) algorithm to reduce the exponential complexity of the QM problem. To implement these algorithms on massively parallel and distributed computing platforms, our framework also includes: 1) topology-preserving computational-space decomposition for structured message passing to minimize the latency; and 2) wavelet-based curvilinear-coordinate load balancing to minimize the load-imbalance and communication costs. Finally, the framework enables interactive visualization of high-dimensional datasets from extreme-scale (e.g., billion atoms) simulations in immersive virtual environments, using multilevel, probabilistic and parallel/distributed techniques.

In Sec. 2, we describe the \(O(N)\) MRMD and LSDFT algorithms, and Sec. 3 deals with the scalable parallel computing framework. Multiscale simulations and immersive/interactive visualization techniques are presented in Secs. 4 and 5, respectively. Finally, Sec. 6 contains conclusions.
2. Linear-Scaling Atomistic Simulation Algorithms

Spatiotemporal localities inherent in scientific/engineering problems can be utilized to design algorithms with low computational complexities. This section shows two such examples in MD and QM simulations.

2.1. Multiresolution Molecular Dynamics

In the MD approach, one obtains the phase-space trajectories of the system (positions and velocities of all atoms at all time) by numerically integrating Newton’s equations of motion,

\[ M_i \frac{d^2 r_i}{dt^2} = -\frac{\partial}{\partial r_i} E_{MD}(r^N) \quad (i = 1, \ldots, N) \]  

(1)

where \( M_i \) and \( r_i \) are the mass and position of the \( i \)-th atom, respectively. In Eq. (1), atomic force laws for describing how atoms interact with each other is encoded in the interatomic potential energy, \( E_{MD}(r^N) \), which is a function of the positions of all \( N \) atoms, \( r^N = \{ r_1, r_2, \ldots, r_N \} \). In our many-body interatomic potential scheme, \( E_{MD}(r^N) \) is expressed as an analytic function that depends on relative positions of atomic pairs and triplets.

The hardest computation in MD simulations is the evaluation of long-range electrostatic forces, which requires \( O(N^2) \) operations. The multiresolution molecular dynamics (MRMD) algorithm\(^6,7,9\) reduces this \( O(N^2) \) complexity to \( O(N) \), by making use of spatial localities based on the fast multipole method (FMM)\(^10,11\).

The first essential idea of the FMM is clustering, \( i.e., \) instead of computing interactions between all atomic pairs, atoms are clustered and cluster-cluster interactions are computed (see Fig. 1). At the source of interaction, cluster information is encapsulated in terms of the multipoles of the charge distribution with a well-defined error bound. At the destination, the electrostatic potential is expanded in terms of local terms, which is similar to the Taylor expansion. The second essential idea is to use larger clusters for longer distances, in order to reduce the computation and keep the error constant. This is achieved by recursively subdividing the simulation box into smaller cells to form an octree data structure. The \( O(N) \) algorithm traverses this tree twice. In the upward pass, multipoles are computed for all cells at all levels. First the multipoles of the leaf cells are computed using atomic charges and coordinates, and the multipoles of these children cells are shifted and combined to obtain the multipoles of the parent cells. This procedure is
repeated until the root of the octree is reached. In the downward pass, these multipoles are translated to local terms for all cells at all levels, starting from the root. For a given cell at each level, only the multipoles of a constant number of interactive cells contribute to the local terms. Contributions from farther cells have already been computed at the previous coarse level, and they are inherited from the parent cell. On the other hand, the contributions from the nearest-neighbor cells will be computed at the next fine level. This procedure is repeated until we reach the leaf level. Finally, the nearest-neighbor-cell contributions at the leaf level are computed by direct summation over atoms. Since constant computation is performed at each of the $O(N)$ octree nodes, the complexity of the FMM algorithm is $O(N)$. Our scalable and portable parallel FMM algorithm, which has the unique capability to compute stress tensors with a novel complex charge method, is freely available from an open source public library.\textsuperscript{11}

Fig. 1. Schematic of the FMM. (Left) Atoms (dots) are clustered (squares), and cluster-cluster interactions are computed. (Center) Two-dimensional example of hierarchical clustering, in which the entire system at level $l=0$ is recursively divided into subsystems. (Right) For a given cell at an octree level (shown in black in the bottom panel), only the multipoles from a constant number of neighbor cells (shown in gray in the bottom panel) are translated to the local terms. Contributions from farther cells (hatched) have already been computed at the previous coarse level, and they are inherited from the parent cell (black in the top panel). On the other hand, translation of the multipoles from the nearest-neighbor cells (white in the bottom panel) will be delegated to the children cells at the next fine level.

The MRMD algorithm also uses multiple time stepping (MTS)\textsuperscript{9,12,13} to take advantage of temporal localities. The MTS uses different force-update schedules for different force components, i.e., forces from the nearest-neighbor atoms are computed at every MD step, and forces from
farther atoms are computed with less frequency. This not only reduces the computational cost but also enhances the data locality, and accordingly the parallel efficiency is increased. These different force components are combined using a reversible symplectic integrator and the resulting algorithm consists of nested loops to use forces from different spatial regions. It has been proven that the phase-space volume occupied by atoms is a simulation-loop invariant in this algorithm, and this loop invariant results in excellent long-time stability of the solutions.

2.2. Linear-Scaling Density Functional Theory on Hierarchical Grids

Breakage and formation of bonds in chemically reactive regions need to be described by a QM method, which explicitly treats the electronic degrees-of-freedom. Since each electron’s wave function is a linear combination of many states, the combinatorial solution space for the many-electron problem is exponentially large. The density functional theory (DFT) avoids the exhaustive enumeration of many-electron correlations by solving $M$ single-electron problems in a common average environment ($M$ is the number of independent wave functions and is on the order of $N$). As a result, the problem is reduced to a self-consistent matrix eigenvalue problem, which can be solved with $O(M^3)$ operations. The DFT problem can also be formulated as a minimization of the energy, $E_{QM}(\mathbf{r}^M, \psi^M)$, with respect to electron wave functions, $\psi^M(\mathbf{r}) = \{\psi_1(\mathbf{r}), \psi_2(\mathbf{r}), ..., \psi_M(\mathbf{r})\}$, subject to orthonormalization constraints between the wave functions. We include electron-ion interaction using norm-conserving pseudopotentials and the exchange-correlation energy associated with electron-electron interaction in a generalized gradient approximation.

For efficient parallel implementation of DFT, we have developed a hierarchical real-space grid method based on higher-order finite differencing and multigrid acceleration. In the hierarchical grid method, these real-space multigrids are adaptively refined near each atom to accurately describe pseudopotentials (see Fig. 2).

For larger systems ($M > 1,000$), however, the $O(M^3)$ orthonormalization becomes the bottleneck, and hence, linear-scaling DFT algorithms become essential. We have implemented an $O(M)$ algorithm based on the data locality principle called “quantum nearsightedness”—the observation that, for most materials at most temperatures, the off-diagonal elements of the density matrix,
\[ \rho(\mathbf{r}, \mathbf{r}') = \sum_{n=1}^{M} \psi_n^*(\mathbf{r})\psi_n(\mathbf{r}'), \] (2)

decay exponentially, i.e., \( \rho(\mathbf{r}, \mathbf{r}') \propto \exp(-C|\mathbf{r} - \mathbf{r}'|) \) for \( |\mathbf{r} - \mathbf{r}'| \to \infty \) (\( C \) is a constant). Such a diagonally dominant matrix can be represented by maximally localizing each wave function, \( \psi_n(\mathbf{r}) \), by a unitary transformation and then truncating it with a finite cut-off radius. In addition, a Lagrange-multiplier-like technique is used to perform unconstrained minimization, avoiding the \( O(M^3) \) orthonormalization procedure.

Fig. 2. Schematic of hierarchical real-space grids. Coarse multigrids (gray) are used to accelerate iterative solutions of the DFT problem. Fine grids (meshes in the bottom panel) are adaptively generated near the atoms (spheres in the bottom panel) to accurately operate pseudopotentials on the wave functions.

3. Scalable Parallel Computing Framework

Data locality principles are key to developing a scalable parallel computing framework as well. For parallelization of MRMD and LSDFT algorithms, we have developed a topology-preserving computational spatial decomposition scheme\(^6\) to minimize latency through structured message passing\(^9\) and load-imbalance/communication costs through a novel wavelet-based load-balancing scheme\(^{29}\). In spatial decomposition, the total volume of the system is divided into \( P \) subsystems of equal volume, and each subsystem is assigned to a processor in an array of \( P \) processors. 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 processors. 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 processors. 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.

Many MD simulations are characterized by irregular atomic distribution and associated load imbalance. For irregular data structures, the number of atoms assigned to each processor varies significantly, and this load imbalance degrades the parallel efficiency. 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:

$$T = t_{\text{comp}} \left( \max_p \left\{ i | r_i \in p \right\} \right) + t_{\text{comm}} \left( \max_p \left\{ i | \| r_i - \partial p \| < r_c \right\} \right) + t_{\text{latency}} \left( \max_p \left[ N_{\text{message}}(p) \right] \right),$$  \hspace{1cm} (3)

where the three terms are the load-imbalance cost, the size of messages, and the number of messages, respectively. In Eq. (3), $\partial p$ and $N_{\text{message}}(p)$ denote the boundary surface of the physical volume assigned to processor $p$, and the number of messages per MD step for $p$, respectively. The expression, $\max_p f(p)$, denotes the maximum value of function $f(p)$ over all the processors, and $r_c$ is the range of the interatomic potential. The prefactors, $t_{\text{comp}}, t_{\text{comm}}$ and $t_{\text{latency}}$, are constants related to the processor speed, communication bandwidth and latency, respectively, and they are determined experimentally by test runs on the parallel computer under consideration.

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.\(^{30}\) The main idea of this scheme is that the computational space shrinks where the workload density is high and expands where the density is low, so that the workload is uniformly distributed in the computational space. To implement the curved computational space, we introduce a curvilinear coordinate transformation,

$$\xi = \mathbf{x} + \mathbf{u}(\mathbf{x}), \hspace{1cm} (4)$$

where $\mathbf{x}$ is a position in the physical Euclidean space and $\mathbf{u}(\mathbf{x})$ is a deformation field. We then use regular 3D mesh topology in the computational space, $\xi$, to map atom $i$ to processor $p$ in an array of $P_x \times P_y \times P_z$ processors:
$$\begin{align*}
&\left\{ p(\xi_i) = \sum_{\alpha=x,y,z} \left( p^\alpha(\xi_{i\alpha}) P^\alpha_\alpha P_\alpha \right) + p(\xi_{iy}) P_z + p(\xi_{iz}) \right\}, \\
&\left\{ p^\alpha_{\alpha}(\xi_{i\alpha}) = \left[ \xi_{i\alpha} P^\alpha_\alpha / L_\alpha \right] \ (\alpha = x,y,z) \right\}
\end{align*}$$

where $\xi_i = (\xi_{ix}, \xi_{iy}, \xi_{iz})$ is the coordinate of atom $i$ and $L_\alpha$ is the simulation box size in the $\alpha$ direction in the computational space. This regular 3D mesh partition in the computational space results in curved partition boundaries in the physical space, $x$. The load-imbalance and communication costs are minimized as a functional of the coordinate transformation, $\tilde{\xi}(x)$, using simulated annealing. We have found that wavelet representation leads to compact representation of curved partition boundaries, and accordingly to fast convergence of the minimization procedure.\(^{29}\)

Another critical issue in high-end parallel/distributed computing is the handling of large datasets. For example, a 1.5-billion-atom MD simulation we are currently performing produces 150 GB of data per time step (or every few seconds), including atomic species, positions, velocities, and stresses. For scalable input/output (I/O) of such large datasets, we have designed a data compression algorithm based on data localities.\(^{31}\) It uses octree indexing and sorts atoms accordingly on the resulting spacefilling curve, which is a mapping from the three-dimensional space to a one-dimensional list, while preserving the spatial proximity of successive list elements. By storing differences between successive atomic coordinates, the I/O requirement for a given error tolerance reduces from $O(N\log N)$ to $O(N)$. An adaptive, variable-length encoding scheme is used to make the scheme tolerant to outliers. An order-of-magnitude improvement in the I/O performance was achieved for actual MD data with user-controlled error bound.

The $O(N)$ algorithms in Sec. 2 and the parallel computing framework in this section have been combined to achieve scalability up to 6.4 billion-atom MRMD and 0.44 million-electron LSDFT simulations on multi-Teraflop architectures.\(^{6}\)

### 4. Hierarchical Multiscale Simulations on a Grid

Our hierarchical multiscale simulation framework\(^{4,5}\) consists of: i) hierarchical division of the physical system into subsystems of decreasing sizes and increasing quality-of-solution (QoS) requirements (e.g., needs for describing nonlinear atomistic processes or chemical reactions), $S_0 \supset S_1 \supset \ldots \supset S_n$; and ii) a suite of simulation services, $M_\alpha (\alpha$
= 0, 1, ..., n), of ascending order of accuracy (e.g., FE ≲ MD ≲ DFT), see Fig. 3. In this scheme, an accurate estimate of the energy of the entire system is obtained from the recurrence relation,

\[ E_\alpha(S_i) = E_{\alpha-1}(S_i) + E_\alpha(S_{i+1}) - E_{\alpha-1}(S_{i+1}), \tag{6} \]

where \( E_\alpha(S_i) \) is the energy of system \( S_i \) calculated with method \( M_\alpha \). This modular, additive hybridization scheme has minimal interdependency and communication between simulation modules. Other physical quantities such as interatomic forces are obtained by derivatives of Eq. (6), and accordingly are additive as well.

![Diagram](image)

Fig. 3. Extrapolation in the two-dimensional meta-model space in a hierarchical multiscale simulation. Recursive applications of Eq. (6) accurately describe a large system (denoted by star), using less compute-intensive calculations (circles).

We have used the additive hybridization framework to perform: 1) MD/DFT simulations of crack initiation in Si in the presence of water molecules;\(^{33}\) 2) FE/MD simulations of stress distributions at Si/amorphous Si\(_3\)N\(_4\) interfaces;\(^{34}\) and 3) multiscale FE/MD/DFT simulations of oxidation in Si.\(^{4}\)

We have also performed a multidisciplinary, collaborative MD/DFT simulation on a Grid of geographically distributed Linux clusters in the US and Japan, based on the modular, additive hybridization scheme (see Fig. 4).\(^{2}\) The multiscale MD/QM simulation code has been Grid-enabled based on a divide-and-conquer scheme, in which the QM region is a union of multiple QM clusters.
Fig. 4. Multiscale MD/DFT simulation of the reaction of water at a crack tip in silicon (top), on a Grid of distributed Linux clusters in the US and Japan (bottom). In this figure, five QM calculations (circles) around five water molecules are embedded in an MD simulation.

Since the energy is a sum of the QM energy corrections for the clusters in the additive divide-and-conquer hybridization scheme,\[ E = E_{\text{MD}}(\text{system}) + \sum_{\text{cluster}} [E_{\text{DFT}}(\text{cluster}) - E_{\text{MD}}(\text{cluster})]. \] (7)
each QM-cluster calculation does not access the atomic coordinates in the other clusters, and accordingly its parallel implementation involves no inter-QM-cluster communication. Furthermore, the multiple-QM-cluster scheme is computationally more efficient than the single-QM-cluster scheme because of the $O(N^3)$ scaling. (The large prefactor of $O(N)$ DFT algorithms makes conventional $O(N^3)$ algorithms faster below a few hundred atoms.)\(^{21}\)

The hybrid MD/DFT simulation algorithm has been implemented on parallel computers, by first dividing processors into the MD and DFT calculations (task decomposition) and then using spatial decomposition in each task. The additive hybridization scheme makes the MD and DFT subtasks entirely independent except for the exchange of cluster-atom coordinates and calculated forces. The MD processors compute the
energy and forces of the entire system and send the atomic coordinates of the QM clusters to each of the QM processor groups. Subsequently, the MD and QM processors independently perform the MD and QM computations on the atomic clusters. The QM energy and forces are then returned to the MD processors, where the total energy and corresponding forces are calculated and the equations of motion are integrated to update the atomic positions and velocities. The communications between the MD and QM processors are minimal, since the MD processors only need to send several hundred atomic coordinates to each QM cluster, which in return sends back the calculated several hundred force components.

We have implemented the multiscale MD/DFT simulation algorithm as a single MPI (Message Passing Interface) program. The Globus middleware and the Grid-enabled MPI implementation, MPICH-G2, have been used to implement the MPI-based multiscale MD/DFT simulation code in a Grid environment.

In the initial implementation, processors on multiple PC clusters are statically allocated using a host file. The user specifies the number of processors for each QM-cluster calculation in a configuration file. In more recent MD/DFT simulations, a simple local error indicator based on atomic bond lengths has been used to automatically change the size of QM calculations in run-time.

The Gridified MD/QM simulation code has been used to study environmental effects of water molecules on fracture in silicon. A preliminary run of the code has achieved a parallel efficiency of 94% on 25 PCs distributed over 3 PC clusters in the US and Japan, see Fig. 4.

5. Immersive and Interactive Visualization of Massive Datasets

Data locality principles also play a critical role in designing scalable visualization techniques. Interactive visualization/mining of high-dimensional datasets is essential for understanding hybrid multiscale material simulations. An immersive virtual environment is ideal for interactively exploring complex material processes, e.g., in nanoceramics and nanocomposites. We have an immersive and interactive visualization platform called ImmersaDesk, which is used to render billion-atom datasets at a near interactive speed.

To achieve this capability, we have developed a visualization system based on a parallel and distributed approach with a Linux cluster to efficiently select a data subset within the field-of-view (view-frustum culling) using the octree data structure (see Fig. 5).
We have also developed a novel probabilistic approach to efficiently remove hidden atoms (occlusion culling) far from the viewer.\textsuperscript{38, 39} This approach is based on a recurrence relation,

\begin{equation}
  v_c = (1 - D_c) v_{c-1},
\end{equation}

where \( v_c \) is the visibility (i.e., the fraction of atoms that are probably seen by the user) of the \( c \)-th octree cell and \( D_c \) is the density of atoms (normalized in the range between 0 and 1) of the \( c \)-th octree cell. In Eq. (8), the leaf octree cells are ordered in an ascending order of distance from the user, based on a line-drawing algorithm. When the viewer is moving, the probabilistic occlusion culling is activated to decimate atoms with probability \( 1 - v_c \), with typically a few percent of pixel loss.

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.\textsuperscript{40}

Using this visualization system, we have demonstrated interactive visualization of a billion-atom dataset in an immersive virtual reality environment.\textsuperscript{38}

6. Conclusion

In summary, we are developing a virtualization aware framework for hierarchical multiscale simulations based on data locality principles.
Such a framework will become increasingly more important in the coming era of “cyber-infrastructures for science and engineering,” when globally distributed multidisciplinary teams will collaborate on a Grid.

The scope of hierarchical multiscale simulations is rapidly expanding. For example, we have recently performed multiscale FE/MD/DFT simulations to study the oxidation of silicon surface. Such hierarchical multiscale simulations will enhance the scalability on Grid architectures by performing more accurate but less scalable computations only when and where they are needed, and they are expected to play a significant role in scientific research and engineering developments in the cyber-infrastructure era.

Acknowledgments

This work was partially supported by AFRL, ARL, DOE, NSF, and USC-Berkeley-Princeton DURINT. Benchmark tests were performed at Department of Defense (DoD)’s Major Shared Resource Center under a DoD Challenge Project. Parallel simulations were also performed on the 1,512-processor HPC cluster at the Research Computing Facility and 400+ processor Linux clusters at the Collaboratory for Advanced Computing and Simulations at the University of Southern California.
References