Collaborative Simulation Grid:
Multiscale Quantum-Mechanical/Classical Atomistic Simulations
on Distributed PC Clusters in the US and Japan

Hideaki Kikuchi,[*] Rajiv K. Kalia,[*,†] Aiichiro Nakano,[*,†] Priya Vashishta[*,†]
†Computer Science Department, Department of Materials Science,
Department of Physics & Astronomy, Department of Biomedical Engineering,
University of Southern California, USA
*Concurrent Computing Laboratory for Materials Simulations, Department of Computer Science,
Department of Physics & Astronomy, Louisiana State University, USA
hkikuchi@phys.lsu.edu, (anakano, kalia, priyav)@usc.edu

Hiroshi Iyetomi
Department of Physics, Niigata University, Japan
hiyetomi@sc.niigata-u.ac.jp

Shuji Ogata, Takahisa Kouno
Department of Applied Sciences, Yamaguchi University, Japan
ogata@yamaguchi-u.ac.jp

Fuyuki Shimojo
Faculty of Integrated Arts and Sciences, Hiroshima University, Japan
shimojo@minerva.ias.hiroshima-u.ac.jp

Kenji Tsuruta
Department of Electrical and Electronic Engineering, Okayama University, Japan
tsuruta@elec.okayama-u.ac.jp

Subhash Saini
IT Modeling and Simulation, NASA Ames Research Center, USA
saini@nas.nasa.gov

Abstract

A multidisciplinary, collaborative simulation has been performed on a Grid of geographically distributed PC clusters. The multiscale simulation approach seamlessly combines i) atomistic simulation based on the molecular dynamics (MD) method and ii) quantum mechanical (QM) calculation based on the density functional theory (DFT), so that accurate but less scalable computations are performed only where they are needed. The multiscale MD/QM simulation code has been Grid-enabled using i) a modular, additive hybridization scheme, ii) multiple QM clustering, and iii) computation/communication overlapping. 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, and a larger test involving 154 processors on 5 distributed PC clusters is in progress.

Keywords: Grid application, multiscale simulation, molecular dynamics, quantum mechanics, density functional theory

1 Introduction

Metacomputing on a Grid of geographically distributed computing platforms connected via high-speed networks could revolutionize computational research, by enabling collaborative, hybrid computations that integrate multiple expertise distributed over wide geographical locations [1]. The availability of inexpensive PC clusters at the research-group level suggests a new collaborative mode for computational research, in which multiple research groups of diverse expertise participate in a metacomputing project by providing both expert-maintained application programs and computational resources to run them. Such a multidisciplinary application is emerging at the forefront of computational sciences and engineering. The multiscale simulation embeds accurate quantum mechanical (QM) calculations to handle chemical reactions (below a length scale of 10^{-8} m) within a molecular dynamics (MD) simulation to describe large-scale atomistic processes (up to a length scale of 10^{-6} m), see Fig. 1 [2-5]. Modern design of high-performance materials and devices focuses on controlling structures at diverse length scales from atomic to macroscopic [6], and such multiscale MD/QM simulations are expected to play an important role in scaling down engineering concepts to nanometer scales.
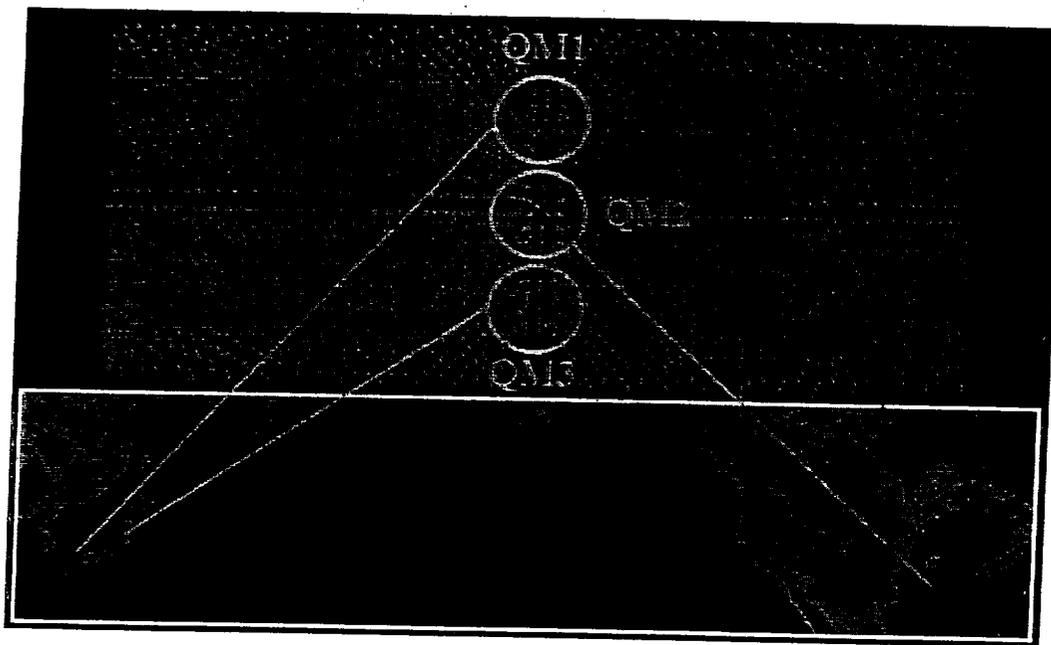

Figure 1: Multiscale MD/QM simulation of the reaction of water molecules at a crack tip in silicon (top), performed on geographically distributed PC clusters in the US and Japan (bottom). In this example, three QM calculations are embedded in an MD simulation, where green spheres represent QM silicon atoms; blue, handshake silicon atoms; red, QM oxygen atoms; yellow, QM hydrogen atoms; gray, MD silicon atoms.

Unfortunately, only a limited class of scientific and engineering applications has been shown to scale on such Grid computing platforms [7]. This paper describes our efforts to enable MD/QM simulations on a Grid by designing scalable multiscale simulation algorithms. In the next section, we describe MD and QM algorithms as well as their Grid-enabled hybridization schemes. Section 3 discusses Grid implementation of the resulting MD/QM simulation algorithm. Results of benchmark tests are given in Sec. 4, and Sec. 5 contains conclusions.

2 Multiscale simulation algorithms

We have developed a multiscale simulation algorithm that combines a classical MD simulation and a self-consistent QM calculation based on the density functional theory (DFT).

2.1 Space-time multiresolution molecular dynamics algorithm

The MD approach obtains the phase-space trajectories (positions and velocities of all atoms at all time) by numerically integrating coupled ordinary differential equations. The dynamics is encoded in the interatomic potential energy, $E_{MD}(\mathbf{r}^N)$, 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 an analytic function that depends on relative positions of atomic pairs and triples [3].

We have developed highly efficient, multiresolution algorithms to carry out large-scale MD simulations on parallel computers [8,9]. The most compute-intensive problem in an MD simulation is the $O(N^2)$ computation of the electrostatic energy for N charged atoms. We use the Fast Multipole Method (FMM) [10] to reduce the complexity to $O(N)$ by computing the electrostatic field recursively on an octree. In our multiresolution molecular dynamics (MRMD) algorithm, the FMM is combined with a symplectic, multiple time-scale (MTS) method [11] that applies different force-updating schedules for different force components while keeping the phase-space volume occupied by the atoms as a simulation loop invariant.

Our MRMD program also features wavelet-based computational-space decomposition for adaptive load balancing [12, 13] and spacefilling-curve-based adaptive data compression for scalable I/O [14].

2.2 Real-space quantum-mechanical calculation based on the density functional theory

Empirical interatomic potentials used in MD simulations fail to describe chemical processes. Instead, interatomic interaction in reactive regions needs to be calculated by a QM method that can describe breaking and formation of bonds. An atom consists of a nucleus and surrounding electrons, and quantum mechanics explicitly treats the electronic degrees-of-freedom. Since each electron's wave function is a linear combination of multiple states, the combinatorial solution space for a many-electron problem is exponentially large. Walter Kohn received a Nobel prize in 1998 for the development of a heuristic called the density functional theory (DFT) [15-17]. The DFT avoids the exhaustive enumeration of many-electron correlations by solving N_{wf} single-electron problems in a common average environment, which is determined self-consistently (N_{wf} is the number of independent wave functions or electronic bands). As a result, the problem is reduced to a self-consistent matrix eigenvalue problem, which can be solved with $O(N_{wf}^3)$ operations. The DFT problem can be formulated as the minimization of an energy functional, $E_{QM}(\psi^{N_{wf}})$, with respect to electron wave functions, $\psi^{N_{wf}}(\mathbf{r}) = \{\psi_1(\mathbf{r}), \psi_2(\mathbf{r}),..., \psi_{N_{wf}}(\mathbf{r})\}$, subject to orthonormalization constraints on the wave functions.

For efficient parallel implementation of the DFT, we employ real-space approaches based on higher-order finite differencing [18] and multigrid acceleration [19, 20]. Our parallel DFT code [9, 21, 22] includes electron-ion interactions using norm-conserving pseudopotentials [23] and the exchange-correlation energy in a generalized gradient approximation [24].

2.3 Multiscale molecular-dynamics/quantum-mechanical simulation algorithm

We have developed a multiscale MD/QM simulation scheme, in which atomic clusters described by QM calculation are embedded in an atomistic region (see Fig. 1) [3-5]. The motion of atoms is described with a real-space multigrid-based DFT in the QM clusters and with the MD approach in the surrounding region. To make the MD/QM simulation scalable on a Grid, we have used the following techniques.

Additive hybridization. To minimize the modification of existing MD and QM codes, we employ an additive hybridization approach based on a linear combination of MD and QM energies [25]. The total energy, E, is written as

$$E = E_{MD}^{system} + E_{QM}^{cluster}(\{\mathbf{r}_{QM}\},\{\mathbf{r}_{HS}\}) - E_{MD}^{cluster}(\{\mathbf{r}_{QM}\},\{\mathbf{r}_{HS}\}) , \tag{1}$$

where E_{MD}^{system} is the classical MD energy for the entire system and the last two terms represent the QM energy correction for the QM cluster: $E_{QM}^{cluster}$ is the QM energy for the cluster (its dangling bonds are terminated by hydrogen atoms to provide appropriate boundary conditions), and $E_{MD}^{cluster}$ is the MD potential energy of the cluster (terminated by appropriate MD atoms). In Eq. (1), $\{\mathbf{r}_{QM}\}$ is the set of the positions of the QM atoms, and $\{\mathbf{r}_{HS}\}$ are the positions of handshake (HS) atoms, *i.e.*, classical atoms bonded to the QM atoms (see Fig. 1). In QM calculation, each bond connecting a QM atom and a HS atom is terminated by a hydrogen atom. Positions of the termination hydrogen atoms are determined dynamically with the scaled-position link atom method as a function of

$\{r_{QM}\}$ and $\{r_{HS}\}$. Other physical quantities, such as interatomic forces, are derived from Eq. (1) as a linear combination as well. Because of this additivity, our cut-and-paste type hybridization scheme is highly modular. Separate research groups can develop the MD and QM modules independently on their own PC clusters, and then these modules can easily be combined to a single metacomputing application on the Grid of their PC clusters. (This is not trivial for most hybridization schemes, in which the computation on the QM cluster involves the atomic coordinates in the outer region.)

Equation (1) shows that, far from the MD/QM boundary, the MD and QM systems are characterized by the MD and QM energies, respectively. The additive MD/QM scheme linearly interpolates the two physical models at the MD/QM boundary, and as such, its validity must be tested by the smoothness of the interpolation. For oxidation of a Si surface, we have calculated the time variation of the kinetic energy distribution to show that the dissociation energy of an oxygen molecule is dissipated from the QM region to the MD region without artificial reflection at the MD/QM boundary [4].

Multiple QM clustering. For scalable MD/QM simulations on a Grid, we have implemented a divide-and-conquer scheme, in which the QM region is a union of multiple QM clusters (see Fig. 1) [5]. Since the energy is a sum of the QM energy corrections for the clusters in the additive hybridization scheme,

$$E = E_{MD}^{system} + \sum_{cluster} [E_{QM}^{cluster}(\{r_{QM}\},\{r_{HS}\}) - E_{MD}^{cluster}(\{r_{QM}\},\{r_{HS}\})] , \tag{2}$$

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 [9, 21, 22].)
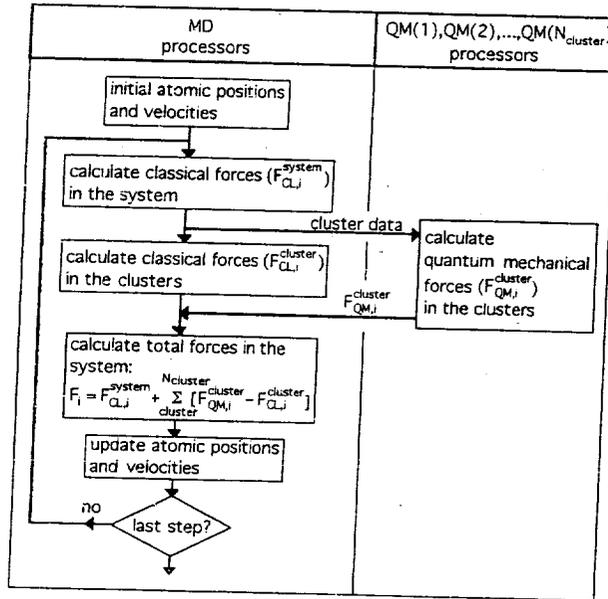

Figure 2: A flowchart of parallel computations in the hybrid MD/QM simulation algorithm.

The hybrid MD/QM simulation algorithm has been implemented on parallel computers, by first dividing processors into the MD and QM calculations (task decomposition) and then using spatial decomposition in each task. The additive hybridization scheme makes the MD and QM subtasks entirely independent except for the exchange of cluster-atom coordinates and calculated forces, as shown in the flowchart in Fig. 2. 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. (This is in contrast to non-additive hybridization schemes, in which the QM tasks continuously access the atomic information in the outer region during their computations.)

Computation/communication overlap. To hide the latency, the communications between the MD and QM processors have been overlapped with the computations. Our spatial decomposition scheme splits the computation on each processor into the interior and boundary computations. The interior computation is then fully overlapped with the communication of the boundary data.

3 Grid implementation

The above hybridization scheme is amenable to meta-computing on a Grid. We have implemented the multiscale MD/QM simulation algorithm as a single MPI (Message Passing Interface) program. The Globus middleware (www.globus.org) and the Grid-enabled MPI implementation, MPICH-G2 (www3.niu.edu/mpi), have been used to implement the MPI-based multiscale MD/QM simulation code in a Grid environment.

In the parallel MD/QM program, all the tasks constitute a single MPI communicator, MPI_COMM_WORLD, and processors are grouped into MD and QM groups by defining multiple MPI communicators. (The MD calculation, as well as each of the QM-cluster calculations, is assigned a dedicated communicator.) The code is written in a single-program multiple-data (SPMD) style, which uses selection statements for the MD and QM processors to execute only the corresponding code segments. Dynamic memory allocation/deallocation operations in Fortran 90 are used to reduce the memory size.

In the current 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. (On heterogeneous platforms, load balancing needs to be manually achieved by assigning more processors on slower platforms.) The additive hybridization and multiple QM clustering schemes minimize the inter-task communications, and thus the maximal parallel efficiency is achieved by assigning each QM cluster to a single PC cluster. Each PC cluster, on the other hand, can have multiple QM clusters without much communication overhead.

The QM calculation involves significant I/O for saving wave functions of all the electrons. For example, the QM calculation of 76 atoms (36 Si and 40 H atoms) produces 24 MBytes of data. (The wave functions in the previous 2 MD steps are saved to obtain a good initial guess for the iterative solution by extrapolation.) To avoid the I/O bottleneck, these wave functions are saved locally within each PC cluster.

4 Performance tests

We have implemented the hybrid MD/QM simulation code on geographically distributed PC clusters in the US and Japan. The Linux-based PC clusters used for this project include:
- A 65-processor Intel Pentium IV 2.0GHz cluster at Yamaguchi University in Japan;
- Three 24-processor Intel Pentium IV 1.8GHz clusters at Hiroshima, Okayama, and Niigata Universities in Japan;
- A 17-processor AMD Athlon XP 1900+ cluster at the Louisiana State University (LSU) in the US.

These PCs are connected using Gigabit Ethernet switches inside each cluster and uplink to campus backbone. Non-dedicated networks used for the Grid include the SINET/SuperSINET (OC192), APAN/TransPAC (OC12), Abilene (OC48), LaNet (OC3), and campus networks of the participating institutions. However, the measured bandwidth between LSU and the other institutions in Japan is low (ranging from 50 to 200 kBytes/s) due to the bottleneck of the low-bandwidth campus networks and bandwidth sharing.

A preliminary benchmark test of the hybrid MD/QM simulation code has been performed on the Grid of PC clusters described above. (A larger benchmark test involving 154 processors is being performed on 5 PC clusters in the US and Japan.) MD/QM simulations have been performed to study environmental effects of H_2O molecules on fracture of strained Si, in which atoms near the crack tip are treated quantum-mechanically in the framework of the DFT. Significant effects of the stress intensity factor on the reaction of the H_2O molecules at the crack tip have been thereby observed [5]. The simulated system is a cracked Si with (110) crack surfaces under uniaxial tension, containing 91,256 atoms. The total number of QM atoms is $76N_{cluster}$ (36 Si and 40 termination-H atoms per cluster)

where $N_{cluster}$ is the total number of QM clusters. At least several hundred QM atoms are needed to accurately model the crack tip region.

Figure 3 shows the execution time of the MD/QM simulation code as a function of $N_{cluster}$, where each QM-cluster calculation is performed on an 8-processor PC cluster. The total number of processors is thus $P = 1 + 8N_{cluster}$ (the MD simulation is performed on 1 processor). In the run on 25 processors, the PC cluster at LSU performs the MD task as well as 1 QM-cluster calculation on 9 (= 1 + 8) processors, whereas each of the PC clusters at Niigata and Yamaguchi uses 8 processors to perform 1 QM-cluster calculation.

Since the system size in the QM calculations scales linearly with the number of processors, the constant execution time per simulation step signifies perfect speedup. (As shown in Fig. 3, the execution time for the MD task is nearly constant, since the number of MD atoms is kept constant.) We first define the speed of the MD/QM program as a product of the total number of QM atoms and time steps executed in a second. The scaled speedup is given by the ratio between the speed of $N_{cluster}$ clusters and that of one cluster. The scaled efficiency is the scaled speedup divided by $N_{cluster}$. On 25 processors (3 PC clusters) in the US and Japan, the measured execution time per MD step is 224.8 seconds. This is only a slight increase from 211.2 seconds on 1 PC cluster in a single location, and the resulting efficiency is 0.94.
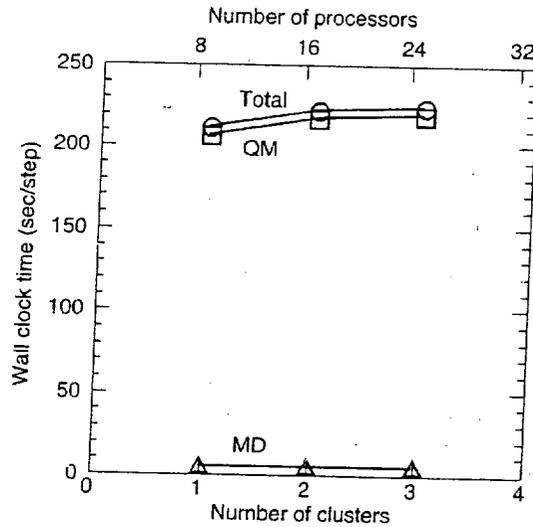

Figure 3: Wall clock time per simulation step for the MD/QM simulation code as a function of the number of QM clusters, $N_{cluster}$. (The number of simulated QM atoms is $76N_{cluster}$.) Each QM-cluster calculation is performed on an 8-processor PC cluster, and the MD simulation is performed on 1 processor. The number of processors in the US and Japan is thus $1 + 8N_{cluster}$. The total execution time and the partial execution times for the QM and MD tasks are denoted by circles, squares, and triangles, respectively.

As is evident in Fig. 3, the dominant computation in the MD/QM simulation is the QM task. Table I shows the averaged wall clock and communication times per MD step for three different stages of the QM task: i) Initial setup; ii) iterative solution of the eigenvalue problem; and iii) force calculation. Not only is the second stage computationally dominant, but it also involves considerable communication.

TABLE I: Partial wall clock and communication times per MD step for the QM task.

	Initialization	Eigenvalue problem	Force calculation
Wall clock time (sec)	16.5	192.7	1.5
Communication (sec)	1.0	67.9	1.3

The computation of the QM task scales as $O(N^3/P)$ for N QM atoms on P processors, whereas the communication scales as $O(N(N/P)^{2/3} + N^2\log P)$ [22]. The communication overhead becomes less for large grain sizes, N/P, but could become a bottleneck if a QM cluster is assigned on multiple PC clusters with weak communication links.

5 Conclusions

We have demonstrated the scalability of multiscale MD/QM simulations on a Grid of geographically distributed PC clusters in the US and Japan. The resulting high efficiency (despite the weak communication link) suggests that this is an efficient collaboration mode in computational research. Such a multiscale simulation collaboratory will consist of geographically distributed application specialists who not only maintain their computational modules with most updated simulation algorithms but also provide compute servers that are best suited to their own application codes. We have shown that the additive multiple-clustering scheme is effective in Gridifying such multiscale simulation codes. Furthermore, the additive hybridization scheme allows multiple layers of nested hybridization including, e.g., MD, DFT, and higher accuracy QM calculations such as configuration interaction. These methods are currently used by separate groups of scientists to solve similar problems at different levels of accuracy and problem sizes. The multiscale simulation collaboratory on a Grid will allow these scientists to jointly solve challenging scientific/engineering problems. The Gridification approach in this paper should be applicable to other modular multiscale simulations as well. An example is the black-box hybridization scheme based on the multilevel Newton method, in which the residual equation is formulated in a coupled model space combining, e.g., electrostatic and mechanical problems [26].

Acknowledgements

References

[1] I. Foster and C. Kesselman. *The Grid: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann, San Francisco, 1999.

[2] F. F. Abraham, J. Q. Broughton, N. Bernstein, and E. Kaxiras. Spanning the length scales in dynamic simulation. *Computational Physics*, 12:538-546 (1998).

[3] A. Nakano, M. E. Bachlechner, R. K. Kalia, E. Lidorikis, P. Vashishta, G. Z. Voyiadjis, T. J. Campbell, S. Ogata, and F. Shimojo. Multiscale simulation of nanosystems. *Computing in Science and Engineering*, 3(4):56-66 (2001).

[4] 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).

[5] S. Ogata, F. Shimojo, A. Nakano, P. Vashishta, and R. K. Kalia. Environment effects of H_2O on fracture initiation in Si: a hybrid electronic-density-functional/molecular-dynamics study. *Journal of Applied Physics*, submitted (2002).

[6] A. Pechenik, R. K. Kalia, and P. Vashishta. *Computer-Aided Design of High-Temperature Materials*. Oxford Univ. Press, Oxford, 1999.

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

[8] 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).

[9] A. Nakano, R. K. Kalia, P. Vashishta, T. J. Campbell, S. Ogata, F. Shimojo, and S. Saini. Scalable atomistic simulation algorithms for materials research. In *Proceedings of Supercomputing 2001*, ACM, New York, 2001.

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

[11] 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).

[12] A. Nakano. Multiresolution load balancing in curved space: The wavelet representation. *Concurrency: Practice and Experience*, 11:343-353 (1999).

[13] 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).

[14] 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).

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

[16] 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.

[17] M. C. Payne, M. P. Teter, D. C. Allan, T. A. Arias, and J. D. Joannopoulos. Iterative minimization techniques for *ab initio* total energy calculations: molecular dynamics and conjugate gradients. *Reviews of Modern Physics*, 64:1045-1097 (1992).

[18] J. R. Chelikowsky, Y. Saad, S. Öğüt, 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).

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

[20] 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).

[21] F. Shimojo, T. J. Campbell, R. K. Kalia, A. Nakano, P. Vashishta, S. Ogata, and K. Tsuruta. A scalable molecular-dynamics-algorithm suite for materials simulations: Design-space diagram on 1,024 Cray T3E processors. *Future Generation Computer Systems*, 17:279-291 (2000).

[22] 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).

[23] N. Troullier and J. L. Martins. Efficient pseudopotentials for plane-wave calculations. *Physical Review B*, 43:1993-2006 (1991).

[24] J. P. Perdew, K. Burke, and M. Ernzerhof. Generalized gradient approximation made simple. *Physical Review Letters*, 77:3865-3868 (1996).

[25] S. Dapprich, I. Kom romi, K. S. Byun, K. Morokuma, and M. J. Frisch. A new ONIOM implementation in Gaussian 98. I. The calculation of energies, gradients, vibrational frequencies, and electric field derivatives. *Journal of Molecular Structure (Theochem)* 461-462:1-21 (1999).

[26] N. R. Aluru and J. White. A multilevel Newton method for mixed-energy domain simulation of MEMS. *Journal of Microelectromechanical Systems*, 8:299-308 (1999).